BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC THĂNG LONG**



**BÀI TẬP LỚN**

**Môn: Internet Of Things**

**Đề tài : Xây dựng ngôi nhà thông minh**

**Giáo viên hướng dẫn: Thành viên nhóm:**

NGUYỄN NGỌC TÂNA39003 PHẠM ĐỨC ANH

A36237 NGUYỄN LONG VŨ

A39173 PHẠM MINH HIẾU

A39006 VŨ VĂN TRUNG

A41252 NGUYỄN HẢI PHONG

**HÀ NỘI – 2023**

**MỤC LỤC**

[CHƯƠNG 1. Tổng quan về đề tài 8](#_Toc139479990)

[1.1. Tổng quan về ngôi nhà thông minh 8](#_Toc139479991)

[1.2. Các mô hình nhà thông minh đang được áp dụng hiện nay 10](#_Toc139479992)

[1.2.1. Các giải pháp nhà thông minh trên thế giới 10](#_Toc139479993)

[1.2.2. Các giải pháp nhà thông minh ở Việt Nam 12](#_Toc139479994)

[1.3. Lựa chọn hướng thiết kế 13](#_Toc139479995)

[CHƯƠNG 2. Thiết kế hệ thống VÀ chức năng 16](#_Toc139479996)

[2.1. Giới thiệu các thiết bị sử dụng trong đề tài 16](#_Toc139479997)

[2.1.1. Cảm biến khí gas 16](#_Toc139479998)

[2.1.2. Cảm biến nhiệt độ 17](#_Toc139479999)

[2.1.3. Relay 18](#_Toc139480000)

[2.1.4. Động cơ Servo 19](#_Toc139480001)

[2.1.5. Mạch hạ áp 5V 20](#_Toc139480002)

[2.1.6. Còi chíp 5V 21](#_Toc139480003)

[2.1.7. Cảm biến hồng ngoại 22](#_Toc139480004)

[2.1.8. Cảm biến mưa 23](#_Toc139480005)

[2.1.9. Motor DC 5V 25](#_Toc139480006)

[2.1.10. Cảm biến chuyển động 26](#_Toc139480007)

[2.1.11. NodeMCU ESP8266 27](#_Toc139480008)

[2.1.12. Đèn led 28](#_Toc139480009)

[2.2. Sơ đồ kết cấu ngôi nhà và chức năng 29](#_Toc139480010)

[2.2.1. Sơ đồ kết cấu 29](#_Toc139480011)

[2.2.2. Chức năng 29](#_Toc139480012)

[CHƯƠNG 3. Blynk và MQTT 32](#_Toc139480013)

[3.1. Blynk và MQTT 32](#_Toc139480014)

[3.2. Giao diện Blynk 33](#_Toc139480015)

[3.2.1. Trên web 33](#_Toc139480016)

[*3.3.* Trên app Blynk 35](#_Toc139480017)

[CHƯƠNG 4. Code dự án 38](#_Toc139480018)

[4.1. IOT Gateway 38](#_Toc139480019)

[4.2. IOT Node: PHòng Bếp 49](#_Toc139480020)

[4.3. IOT Node: Phòng Khách 55](#_Toc139480021)

[4.4. IOT Node: Phòng Ngủ 63](#_Toc139480022)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 Mô hình ngôi nhà thông minh 9](#_Toc139466582)

[Hình 1.2 Biểu đồ tăng trưởng thị trường Smarthome thế giới 9](#_Toc139466583)

[Hình 1.3 Biểu đồ tăng trưởng thị trường Smarthome chỉ tính riêng thị trường Bắc Mỹ 10](#_Toc139466584)

[Hình 1.4 Mô hình Smart home của công ty Compro Technology. 11](#_Toc139466585)

[Hình 1.5 Mô hình Smart home của công ty IEI Integration 11](#_Toc139466586)

[Hình 1.6 Mô hình Smart home Eco-Future-World 12](#_Toc139466587)

[Hình 1.7 Mô hình Smart home của BKAV 12](#_Toc139466588)

[Hình 1.8 Mô hình Smart home của Lumi 13](#_Toc139466589)

[Hình 1.9 Xu hướng phát triển của smarthome 14](#_Toc139466590)

[Hình 2.1 Cảm biến khí gas 16](#_Toc139466591)

[Hình 2.2 Nguyên lý hoạt động của cảm biến khí gas 17](#_Toc139466592)

[Hình 2.3 Cảm biến nhiệt độ(DHT11) 17](#_Toc139466593)

[Hình 2.4 Relay 18](#_Toc139466594)

[Hình 2.5 Động cơ servo 19](#_Toc139466595)

[Hình 2.6 Mạch hạ áp 5V 20](#_Toc139466596)

[Hình 2.7 Còi chip 5V 21](#_Toc139466597)

[Hình 2.8 Cảm biến hồng ngoại 22](#_Toc139466598)

[Hình 2.9 Sơ đồ nguyên lý cảm biến hồng ngoại 23](#_Toc139466599)

[Hình 2.10 Cảm biến mưa 23](#_Toc139466600)

[Hình 2.11 Motor 5V 25](#_Toc139466601)

[Hình 2.12 Cảm biến chuyển động 26](#_Toc139466602)

[Hình 2.13 Cấu hình sơ đồ bảng phát triển NodeMCU 27](#_Toc139466603)

[Hình 2.14 Đèn led 28](#_Toc139466604)

[Hình 2.15 . Sơ đồ mạch cảm biến hồng ngoại bật đèn, cảm biệt nhiệt độ bật quạt, cảm biến mưa bật mái 30](#_Toc139466605)

[Hình 2.16 Sơ đồ mạch cảm biến khí gas bật còi và quạt hút khí 30](#_Toc139466606)

[Hình 2.17 Sơ đồ mạch cảm biến kéo rèm tự động khi ánh sáng quá cao và cửa tự động mở khi phát hiện người chuyển động ở cửa nhà 31](#_Toc139466607)

[Hình 3.1 Nguyên lý hoạt động của MQTT tích hợp với Blynk 32](#_Toc139466608)

[Hình 3.2 Datastream hiển thị cảm biến 33](#_Toc139466609)

[Hình 3.3 Datastream điều khiển 33](#_Toc139466610)

[Hình 3.4 Giao diện Dashboard 34](#_Toc139466611)

[Hình 3.5 Giao diện điều khiển 34](#_Toc139466612)

[Hình 3.6 Giao diện event 35](#_Toc139466613)

[Hình 3.7 Giao diện các chỉ số 35](#_Toc139466614)

[Hình 3.8 Giao diện điều khiển 36](#_Toc139466615)

[Hình 3.9 Giao diện cảnh báo 37](#_Toc139466616)

**Bảng chi tiết công việc nhóm 7**

|  |  |  |
| --- | --- | --- |
| Mã sinh viên | Họ và Tên | Nhiệm vụ |
| A39006 | Vũ Văn Trung | Code Ardunio, MQTT,BLYNK, Làm Word, Làm Powpoitn, xây dựng dự án và hỗ trợ |
| A39003 | Phạm Đức Anh | Code Ardunio, làm mô hình Làm Word, Powpoint, xây dựng dự án và hỗ trợ |
| A39173 | Phạm Minh Hiếu | Làm mô hình, Làm Word, Powpoint, xây dựng dự án và hỗ trợ |
| A36237 | Nguyễn Long Vũ | Làm Word, Powpoint xây dựng và hỗ trợ dự án |
| A41252 | Nguyễn Hải Phong | Hỗ trợ powpoint |

**LỜI NÓI ĐẦU**

IoT (Internet of Things) nghĩa là Internet vạn vật. Một hệ thống các thiết bị tính toán, máy móc cơ khí và kỹ thuật số hoặc con người có liên quan với nhau và khả năng truyền dữ liệu qua mạng mà không yêu cầu sự tương tác giữa con người với máy tính.Từ vài năm nay, cụm từ IoT (Internet of Things) được xuất hiện khá phổ biến ở Việt Nam cũng như trên thế giới và IoT đang trở thành làn sóng mạnh mẽ của cuộc cách mạng công nghiệp 4.0. IoT không chỉ mang đến cho mọi người một cái nhìn lớn hơn, đầy đủ hơn về những công nghệ, ứng dụng của tương lai, mà còn đem đến tiềm năng ứng dụng thực sự đáng kinh ngạc. Hiện tại, trên thế giới, chưa có định nghĩa thống nhất về IoT, nhưng có thể hiểu theo một cách khái quát là mọi vật đều có thể kết nối với nhau thông qua Internet. Với IoT người dùng có thể kiểm soát đồ vật của mình qua một thiết bị thông minh như máy tính xách tay, điện thoại di động hay máy tính bảng. Và nó là một phần vô cùng hữu ích cho cuộc sống sau này và sự phát triển của đất nước để tiến thêm một bước sánh với các cường quốc.

Xã hội thế kỷ 21 chứng kiến sự phát triển vượt bậc của công nghệ và đánh dấu sự mở đầu của những thiết bị thông minh. Smart phone, Smart Tivi đều là những thiết bị ngày càng phổ biến, thông dụng trong đời sống hằng ngày của con người. Đúng như tên gọi, những thiết bị này không những có khả năng đáp ứng những yêu cầu cơ bản của con người, mà còn hơn thế, các thiết bị smart ra đời đã thay thế con người trong việc kiểm soát và điều khiển các chức năng khác 1 cách chuyên nghiệp, dễ dàng và hiệu quả.

Tiếp nối thành công của những thiết bị thông minh ấy, Smart home ra đời như một sự khởi đầu táo bạo về tư duy làm chủ công nghệ ngay trong cuộc sống của con người. Một ngôi nhà thông minh với khả năng thấu hiểu tư duy điều khiển của con người nhanh chóng trở thành đề tài công nghệ có sức hấp dẫn.

Nhà thông minh hay smart home, home automation là kiểu nhà được lắp đặt các thiết bị điện, điện tử có tác dụng tự động hóa hoàn toàn hoặc bán tự động, thay thế con người trong thực hiện một hoặc một số thao tác quản lý, điều khiển. Hệ thống điện tử này giáo tiếp với chủ nhân nhà thông qua bẳng điện tử được đặt sẵn trong nhà, phần mềm điện thoại di động, máy tính bảng hoặc một giao diện web. Đề tài là một sản phẩm có tính thực tế cao dựa trên nhu cầu công nghệ hiện nay, được nghiên cứu, chế tạo dựa trên những kiến thức chúng em đã học, kế thừa và phát huy những kết quả của các công trình nghiên cứu trước đây.

# Tổng quan về đề tài

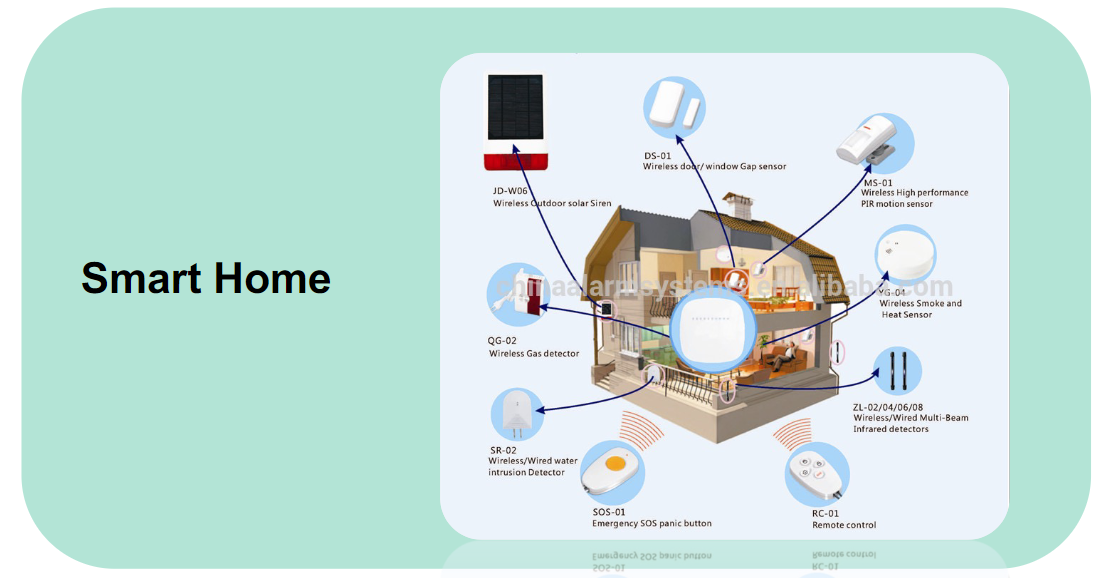
## Tổng quan về ngôi nhà thông minh

Ngày nay, khi đời sống ngày càng được nâng cao, những nhu cầu của con người đòi hỏi những sự tiện nghi và hỗ trợ tốt nhất. Cùng với đó là sự mở rộng không ngừng của mạng lưới internet trên khắp các vùng quốc gia và lãnh thổ làm cho việc giám sát và điều khiển hệ thống qua mạng internet trở thành tất yếu. Từ những yêu cầu và điều kiện thực tế đó , ý tưởng về ngôi nhà thông minh được hình thành, nơi mà mọi hoạt động của con người đều được hỗ trợ và giúp đỡ một cách linh hoạt, ngoài ra ngôi nhà còn có thể tự động quản lí một cách thông minh nhất.

Vậy, như thế nào là nhà thông minh ?

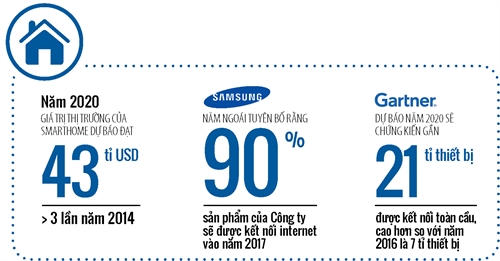
Sự thông minh của một ngôi nhà được thể hiện trên 4 phương diện như sau:

* Thứ nhất, là khả năng tự động hóa. Căn nhà được trang bị hệ thống các cảm biến như: cảm biến nhiệt độ, cảm biến độ ẩm, cảm biến khí gas, cảm biến báo cháy, cảm biến vật cản, cảm biến ánh sáng… với khả năng tự động hoạt động theo điều kiện môi trường. Nhà thông minh giúp chúng ta giám sát được mức tiêu thụ điện, nước tốt hơn so với thông thường.
* Thứ hai, là khả năng thỏa mãn nhu cầu của người sử dụng. Chủ nhân ngôi nhà có thể điều khiển theo ý muốn hoặc theo những kịch bản được lập trình sẵn.
* Thứ ba rò rỉ khí gas sẽ tự động báo trạng thái của ngôi nhà qua mạng internet và còi thông báo.
* Thứ tư, là khả năng điều khiển, cảnh báo từ xa thông qua kết nối internet thông qua wifi,3g… Các thiết bị như: bóng đèn, quạt,….. cũng đều được kết nối tới mạng internet. Người sử dụng chỉ cần có một thiết bị kết nối internet là có thể theo dõi dữ liệu từ các cảm biến và điều khiển các thiết bị trong nhà theo ý muốn của bản thân.

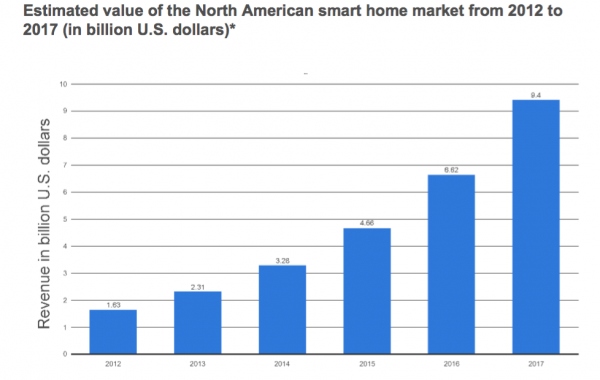


Hình 1.1 Mô hình ngôi nhà thông minh

Hiện nay, nhà thông minh đã và đang là một thị trường tiềm năng với thị trường toàn cầu lên đến con số tỉ đô. Không những vậy, chỉ riêng thị trường Bắc Mỹ, theo các con số thống kê, hoàn toàn là có cơ sở để nhận định rằng đây chính là tương lai của một ngôi nhà mà chúng ta cần phải có.



Hình 1.2 Biểu đồ tăng trưởng thị trường Smarthome thế giới



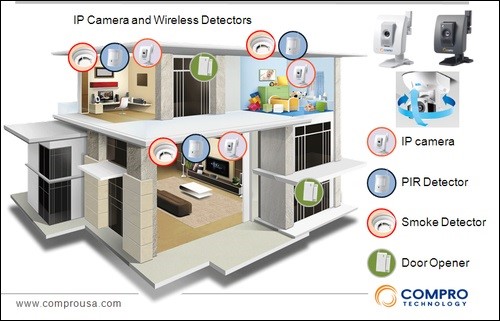
Hình 1.3 Biểu đồ tăng trưởng thị trường Smarthome chỉ tính riêng thị trường Bắc Mỹ

## Các mô hình nhà thông minh đang được áp dụng hiện nay

### Các giải pháp nhà thông minh trên thế giới

Hiện nay là thị trường Smarthome lớn nhất thế giới là Bắc Mỹ. Với quy mô cũng như tính tiện nghi dành cho một ngôi nhà với 4 người, sẽ có thiết kế cơ bản với khả năng như: Cảnh báo đột nhập, cảnh báo khí gas, hệ thống cửa tự động, hệ thống camera an ninh, hệ thống giải trí…

Dưới đây là ví dụ về một ngôi nhà thông minh của một số nhà sản xuất tại Mỹ và châu Âu, với tiêu chuẩn từ cơ bản đến cao cấp dành cho một gia đình:



Hình 1.4 Mô hình Smart home của công ty Compro Technology.



Hình 1.5 Mô hình Smart home của công ty IEI Integration

******

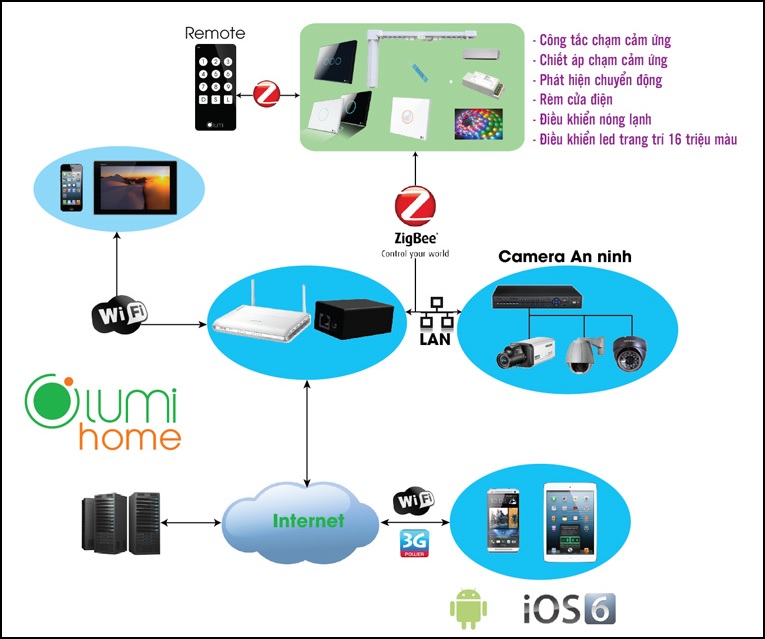
Hình 1.6 Mô hình Smart home Eco-Future-World

### Các giải pháp nhà thông minh ở Việt Nam

Tại Việt Nam, không đứng ngoài dòng chảy công nghệ về nhà thông minh, đã có rất nhiều nhà sản xuất cũ và mới tham gia thị trường đầy tiềm năng này, dẫn đầu là BKAV và Lumi Smarthome. Với đầy đủ các chức năng như các nhà sản xuất nước ngoài, lại thêm yếu tố phù hợp với riêng thị trường Việt Nam, hiện nay họ đang có một lợi thế không nhỏ so với các nhà sản xuất nước ngoài tại Việt Nam.



Hình 1.7 Mô hình Smart home của BKAV



Hình 1.8 Mô hình Smart home của Lumi

## Lựa chọn hướng thiết kế

Nhà thông minh là một đề tài rộng và có nhiều vấn đề đặt ra. Tùy theo mục đích sử dụng của chủ nhân để thiết kế, một phần quan trọng trong hệ thống nhà thông minh là hệ thống điều khiển và giám sát.

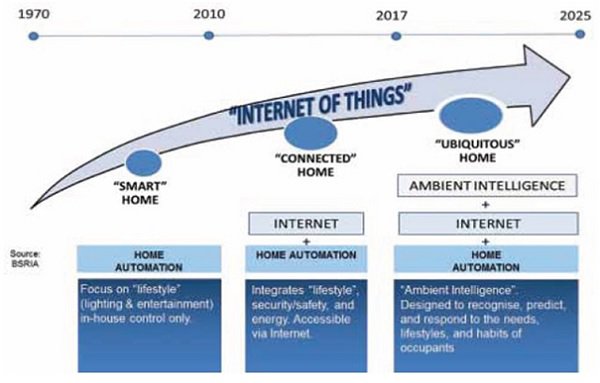
Trước đây, nhà thông minh chỉ hoàn toàn nằm trong trí tưởng tượng cũng như trên phim ảnh. Nhờ sự phát triển không ngừng của khoa học công nghệ, các giải pháp nhà thông minh ngày càng phong phú và thuận tiện hơn cho người sử dụng.

Từ ban đầu, nhà thông minh chỉ có các thiết bị điều khiển từ xa ở trong phạm vi ngôi nhà phục vụ cho một số nhu cầu của con người. Tiếp theo là sự tự động hóa các thiết bị trong ngôi nhà với khả năng tự động điều chỉnh theo môi trường cũng như người sử dụng.

Sau đó, với sự phát triển và lan rộng của mạng internet, người ta đưa ra giải pháp kết nối và điều khiển các thiết bị trong nhà thông qua mạng internet và thêm vào các tiện ích như hệ thống đảm bảo an toàn, tính toán năng lượng sử dụng,… giúp chủ nhân có thể điều khiển thiết bị ở khoảng cách xa chứ không bó hẹp trong khuôn viên ngôi nhà nữa.

Khả năng bảo mật an ninh cũng được đặt lên hàng đầu, vì đi cùng với kết nối internet là khả năng bị hack vào hệ thống nhằm chiếm quyền điều khiển cũng vì thế mà gia tăng. Chủ nhân có thể sử dụng mật khẩu riêng để đăng nhập vào hệ thống cũng như ngôi nhà qua các hình thức như Passcode, bảo mật vân tay, bảo mật mống mắt… Đi kèm đó là khả năng cảnh báo đột nhập giúp gia chủ có thể phát hiện ra tại bất kì nơi đâu với kết nối Wifi/GPRS.

Và gần đây, xu thế điều khiển thiết bị bằng giọng nói cũng được thêm vào giải pháp xây dựng nhà thông minh, giúp cho việc sử dụng trở nên dễ dàng hơn mọi người trong nhà. Trong tương lai, nhờ các thiết bị công nghệ mới kết hợp trí tuệ nhân tạo, ngôi nhà có thể phân biệt giọng nói từng thành viên và ghi nhớ thói quen của mỗi người trong gia đình.



Hình 1.9 Xu hướng phát triển của smarthome

Hiện nay, ở Việt Nam, giải pháp xây dựng nhà thông minh với hệ thống điều khiển và giám sát thông qua internet vẫn phổ biến và phát triển hơn cả vì nó phù hợp với khả năng công nghệ và điều kiện kinh tế hiện có.

Nhóm em cảm thấy đề tài này là phù hợp với IOT, vậy nên ở đề tài này, nhóm em xin chọn thiết kế ngôi nhà thông minh theo giải pháp sử dụng hệ thống điều khiển và giám sát các thiết bị trong ngôi nhà thông qua mạng internet, mà cụ thể là mạng wifi trên một mô hình nhà thông minh với các chức năng cơ bản như: cửa đóng/mở tự động, giám sát và cảnh báo cháy, cảnh báo khí gas rò rỉ, rèm cửa tự động theo ánh sáng, đèn và quạt bật tự động theo người sử dụng và nhiệt độ môi trường , hệ thống …

# Thiết kế hệ thống VÀ chức năng

## Giới thiệu các thiết bị sử dụng trong đề tài

### Cảm biến khí gas

A picture containing silver

Description automatically generated

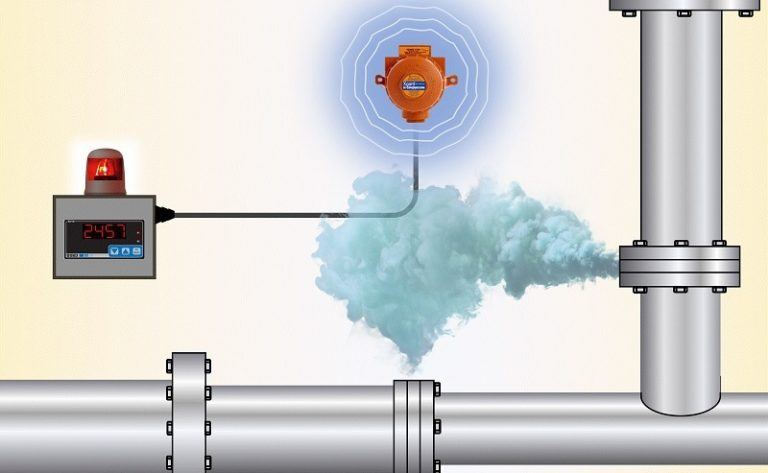
Hình 2.1 Cảm biến khí gas

Thông số kĩ thuật:

* Điện áp hoạt động: 3.3V-5V
* Kích thước PCB: 3cm \* 1.6cm
* Led đỏ báo nguồn vào, Led xanh báo gas
* IC so sánh : LM393
* VCC: 3.3V-5V
* GND: 0V
* DO: Đầu ra tín hiệu số (0 và 1)
* AO: Đầu ra Analog (Tín hiệu tương tự)
* Cấu tạo từ chất bản dẫn Sno2

Nguyên lý hoạt động:

* Thiết bị này hoạt động dựa trên nguyên lý các loại khí khác nhau sẽ tương tác với các chất khác nhau và làm làm thay đổi điện trở của vật liệu dẫn điện. Mà vật liệu dẫn điện ở đây cụ thể là chất bán dẫn hoặc oxide kim loại.



Hình 2.2 Nguyên lý hoạt động của cảm biến khí gas

* Khi khí gas tương tác với vật liệu dẫn điện, nó sẽ làm thay đổi điện trở của vật liệu này. Cảm biến sử dụng một mạch điện để đo lường điện trở này và phát hiện sự thay đổi khi có sự hiện diện của khí gas.

### Cảm biến nhiệt độ



Hình 2.3 Cảm biến nhiệt độ(DHT11)

Thông số kĩ thuật:

* Nguồn: 3 → 5 VDC
* Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
* Khoảng đo độ ẩm: 20%-90% RH (sai số 5%RH)
* Khoảng đo nhiệt độ: 0-50°C (sai số 2°C)
* Tần số lấy mẫu tối đa: 1Hz (1 giây / lần)
* Kích thước 15mm x 12mm x 5.5mm.

Nguyên lý hoạt động

* Để có thể giao tiếp với DHT11 theo chuẩn 1 chân vi xử lý thực hiện theo 2 bước:
* Gửi tin hiệu muốn đo (Start) tới DHT11, sau đó DHT11 xác nhận lại.
* Khi đã giao tiếp được với DHT11, Cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được

### Relay



Hình 2.4 Relay

Thông số:

* Điện áp cuộn hút: 12VDC
* Dòng đóng cắt: 10A/250VAC – 10A/30VDC
* Kích thước: 19.2 x 15.5 x 15.2mm
* Trọng lượng: 8g

Nguyên lý hoạt động:

* Khi dòng điện công suất nhỏ chạy qua mạch điện thứ nhất sẽ kích hoạt nam châm điện, tạo ra từ trường, tín hiệu. Từ trường này sẽ thu hút 1 tiếp điểm để kích hoạt mạch điện thứ 2, cho phép thiết bị kết nối sử dụng dòng có cường độ lớn hơn rất nhiều.
* Khi dòng điện bị ngắt, nam châm ngừng hoạt động, không tạo ra thị trường. Lúc này, tiếp điểm sẽ bị lực kéo của lò xo ban đầu kéo về vị trí cũ, tương ứng với mạch điện thứ 2 bị ngắt.

### Động cơ Servo



Hình 2.5 Động cơ servo

Thông số:

* Chủng loại: Analog RC Servo.
* Điện áp hoạt động: 4.8~6.6VDC
* Lực kéo:
* 3.5 kg-cm (180.5 ozin) at 4.8V-1.5A
* 5.5 kg-cm (208.3 ozin) at 6V-1.5A
* Tốc độ quay:
* 0.17sec / 60 degrees (4.8V no load)
* 0.13sec / 60 degrees (6.0V no load)
* Kích thước: 40mm x 20mm x 43mm
* Trọng lượng: 55g

Nguyên lý hoạt động:

* Nguyên lý hoạt động của Servo DC dựa trên cấu tạo của bốn thành phần chính. Đó là động cơ DC, thiết bị cảm biến vị trí, cụm bánh răng và mạch điều khiển.
* Tốc độ của động cơ Servo DC dựa trên điện áp được sử dụng.
* Để điều khiển tốc độ động cơ, thường sử dụng 1 chiết áp để tạo ra điện áp tương ứng.
* Trong một số mạch thì  xung điều khiển được sử dụng để tạo ra điện áp tham chiếu DC tương ứng với một vị trí hoặc tốc độ mong muốn của động cơ. Và nó sẽ được áp dụng cho bộ chuyển đổi điện áp độ rộng xung.
* Độ dài của xung quyết định đến điện áp đặt tại bộ khuếch đại. Điều đó để tạo ra một điện áp tương đương với tốc độ hoặc vị trí mong muốn.
* Đối với điều khiển kỹ thuật số, PLC hoặc là bộ điều khiển chuyển động khác được sử dụng để tạo xung theo chu kỳ nhiệm vụ nhằm mục đích xây dựng nên những quy trình điều khiển chính xác hơn.

### Mạch hạ áp 5V



Hình 2.6 Mạch hạ áp 5V

Thông số:

* Điện áp đầu vào: Từ 3V đến 30V.
* Điện áp đầu ra: Điều chỉnh được trong khoảng 1.5V đến 30V.
* Dòng đáp ứng tối đa là 3A.
* Hiệu suất: 92%
* Công suất: 15W
* Kích thước: 45 (dài) \* 20 (rộng) \* 14 (cao) mm

Nguyên lý hoạt động:

* Điện áp sẽ đi từ các cuộn lọc nhiễu qua đi ốt chỉnh lưu. Lúc đầu là dòng điện áp xoay chiều có U từ 80 đến 220V. Sau đó, chúng giảm còn 130 – 300V. Tiếp đó, tụ lọc nguồn điện sơ cấp tích năng lượng điện một chiều để biến áp xung có thể hoạt động.
* Hiện nay, bạn sẽ thường thấy một số loại tụ lọc sơ cấp gồm 4,7uF – 400V, 10uF-400V, 220uF-400V và 10uF-200V.
* Tiếp đến, ở cuộn sơ cấp được cấp điện thông qua một số linh kiện như Transistor, mosfet hoặc IGBT… Bộ tạo xung hoặc mạch dao động điện tử sẽ tạo ra các xung điện. Một số loại mạch phổ biến trên thị trường gồm Viper22, Viper12, hx202, Tl494, Sg3525.
* Sau cùng, cuộn dây thứ cấp nhờ mạch chỉnh lưu để tạo ra điện một chiều cấp điện cho tải tiêu thụ. Chúng duy trì ở một điện áp nhất định và cụ thể như 3.3V; 5V… 24V. Tiếp đó, mạch hồi tiếp sẽ lấy tín hiệu điện để tạo ra xung dao động.

### Còi chíp 5V

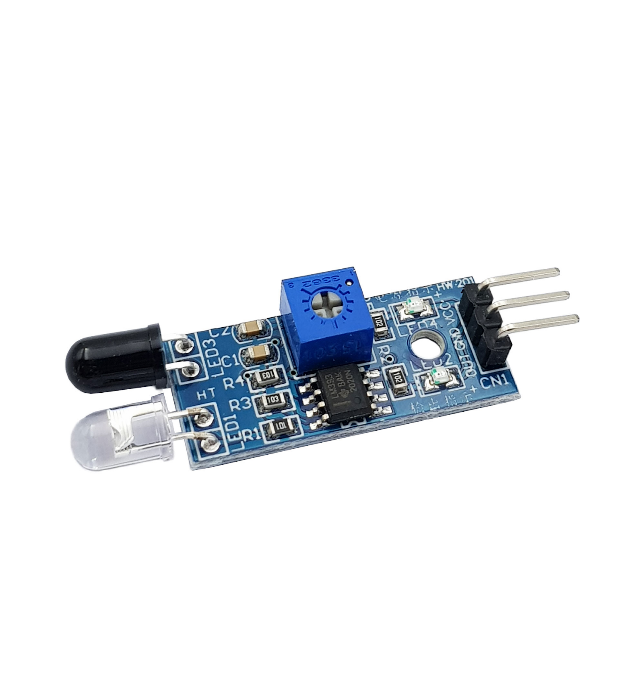


Hình 2.7 Còi chip 5V

Thông số:

* Điện áp sử dụng: Từ 3V đến 12V.
* Dòng điện tiêu thụ: <25mA
* Tàn số âm thanh: 2300hz ± 500
* Biên độ âm thanh: >80dB
* Âm thanh đầu ra: bíp bíp
* 2 cực: cực âm, cực dương

### Cảm biến hồng ngoại



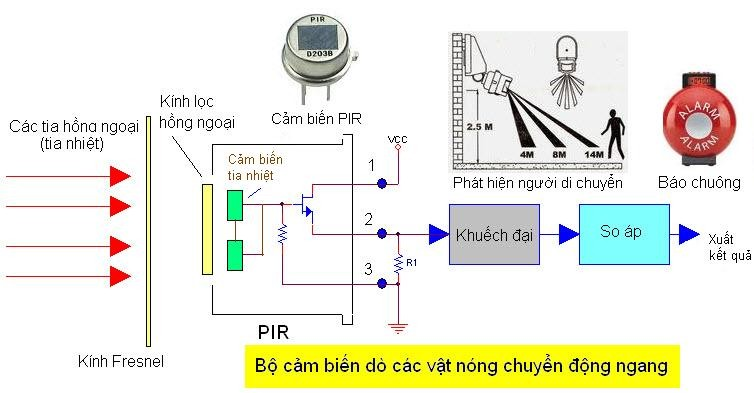
Hình 2.8 Cảm biến hồng ngoại

Thông số kĩ thuật:

* Bộ so sánh sử dụng LM393, làm việc ổn định
* Điện áp làm việc: 3.3V – 5V DC.
* Khi bật nguồn, đèn báo nguồn màu đỏ sáng.
* Lỗ vít 3 mm, dễ dàng cố định, lắp đặt.
* Kích thước: 3.2cm \* 1.4cm
* Các mô-đun đã được so sánh điện áp ngưỡng thông qua chiết áp, nếu sử dụng ở chế độ thông thường, xin vui lòng không tự ý điều chỉnh chiết áp.

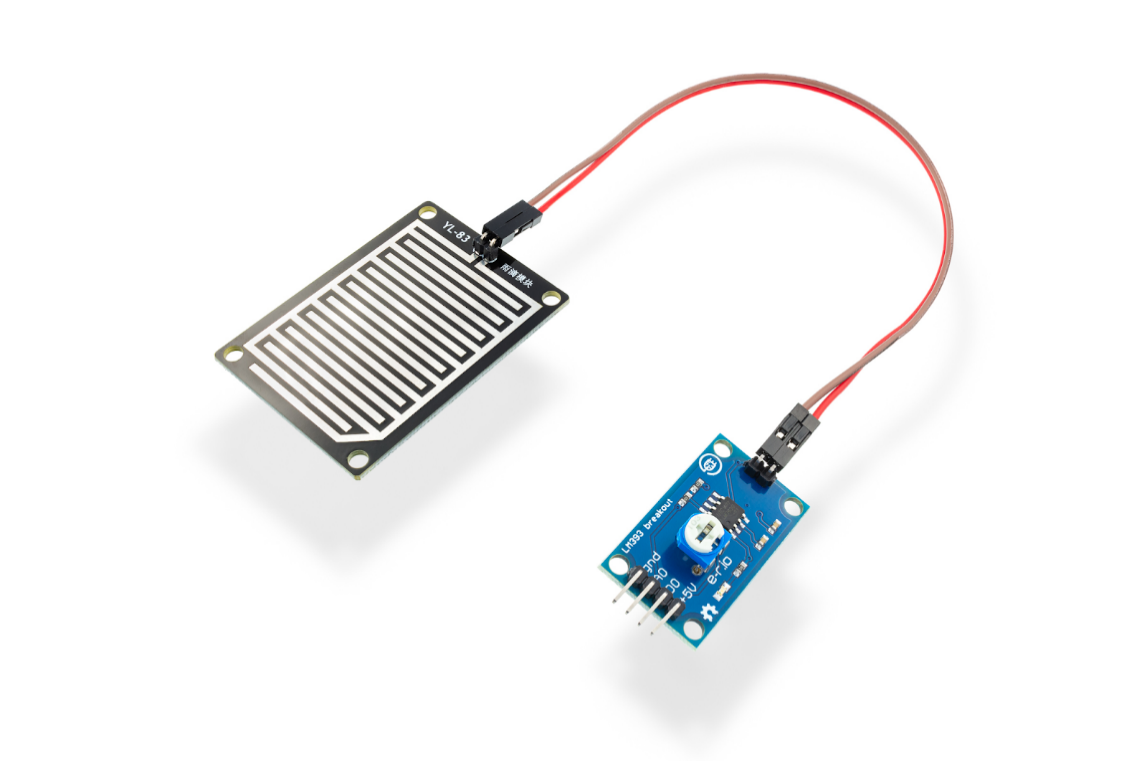
Nguyên lý hoạt động:

* cảm biến hồng ngoại sử dụng một cảm biến ánh sáng cụ thể để phát hiện bước sóng ánh sáng chọn trong giới hạn phổ hồng ngoại (IR). Thông qua việc sử dụng đèn LED để tạo ra ánh sáng có cùng bước sóng với cảm biến đang tìm kiếm, người thực hiện có thể xem cường độ của ánh sáng nhận được. Khi một vật ở gần với cảm biến, ánh sáng từ đèn LED sẽ bật ra khỏi vật thể và đi vào cảm biến ánh sáng tạo nên một bước nhảy lớn về cường độ.



Hình 2.9 Sơ đồ nguyên lý cảm biến hồng ngoại

### Cảm biến mưa



Hình 2.10 Cảm biến mưa

Thông số:

* Điện áp sử dụng: 5VDC
* Kích thước tấm cảm biến mưa: 54 x 40mm
* Kích thước board PCB: 30 x 16mm
* Tín hiệu đầu ra: Digital TTL (0VDC / 5VDC) và đầu ra Analog A0 trả giá trị điện áp tuyến tính theo lượng nước tiếp xúc với cảm biến.
* Lỗ cố định bu lông dễ dàng để cài đặt
* Có đèn báo hiệu nguồn và đầu ra
* Độ nhạy có thể được điều chỉnh thông qua chiết áp
* LED sáng lên khi không có mưa đầu ra cao, có mưa, đầu ra thấp LED tắt.

Nguyên lý hoạt động:

* Cảm biến mưa hoạt động dựa theo cách so sánh hiệu điện thế của mạch cảm biến nằm ngoài trời với một giá trị định trước. Giá trị định trước này thay đổi được thông qua 1 biến trở màu xanh. Từ kết quả so sánh thì bộ xử lý sẽ phát ra tín hiệu đóng hoặc ngắt rơ le qua chân DO.
* Khi xuất hiện nước trên bề mặt cảm biến (trời mưa), độ dẫn điện của tấm bên ngoài sẽ tốt hơn khiến cho [điện trở](https://dientusangtaovn.com/dien-tro-la-gi/) giảm xuống. Chân DO sẽ được kéo xuống thấp (0V), đèn LED màu đỏ sẽ được bật sáng lên.
* Khi trời không mưa, độ dẫn điện của vật liệu sẽ kém khiến cho điện trở cao, chân DO của module cảm biến mưa được giữ ở mức cao (khoảng từ 5V-12V). Vì vậy, đầu ra của cảm biến mưa chủ yếu phụ thuộc vào điện trở. Bạn nên dùng các loại rơ le kích ở mức phù hợp với mục đích sử dụng để kích hoạt chức năng mong muốn.

### Motor DC 5V

A picture containing plug

Description automatically generated

Hình 2.11 Motor 5V

Thông số:

* Model: RS-385
* Công suất định mức: 3,6 (W)
* Loại sản phẩm: Động cơ DC chổi than
* Điện áp định mức: 12 (V)
* Dòng định mức: 0,15-0,75 (A)
* Tốc độ định mức: 9800 (vòng / phút)
* Mô-men xoắn định mức: 70 (g. Cm)
* Kích thước hình dạng: 27,7 \* 60 mm / 1,09 \* 2,36"
* Phạm vi tương thích: đa dạng

Nguyên lý hoạt động:

* Phần chính của motor điện (động cơ điện) gồm phần đứng yên (stator) và phần chuyển động (rotor) được quấn nhiều vòng dây dẫn hay có nam châm vĩnh cửu. Khi cuộn dây trên rotor và stator được nối với nguồn điện, xung quanh nó tồn tại các từ trường, sự tương tác từ trường của rotor và stator tạo ra chuyển động quay của rotor quanh trục hay 1 mômen.
* Phần lớn các motor điện (động cơ điện) hoạt động theo nguyên lý điện từ, những loại động cơ dựa trên nguyên lý khác như lực tĩnh điện và hiệu ứng điện áp cũng được sử dụng. Nguyên lý cơ bản mà các động cơ điện từ dựa vào là có một lực lực cơ học trên một cuộn dây có dòng điện chạy qua nằm trong một từ trường. Lực này theo mô tả của định luật lực Lorentz và vuông góc với cuộn dây và cả với từ trường.
* Phần lớn động cơ từ đều xoay nhưng cũng có động cơ tuyến tính. Trong động cơ xoay, phần chuyển động được gọi là rotor, và phần đứng yên gọi là stator.

### Cảm biến chuyển động



Hình 2.12 Cảm biến chuyển động

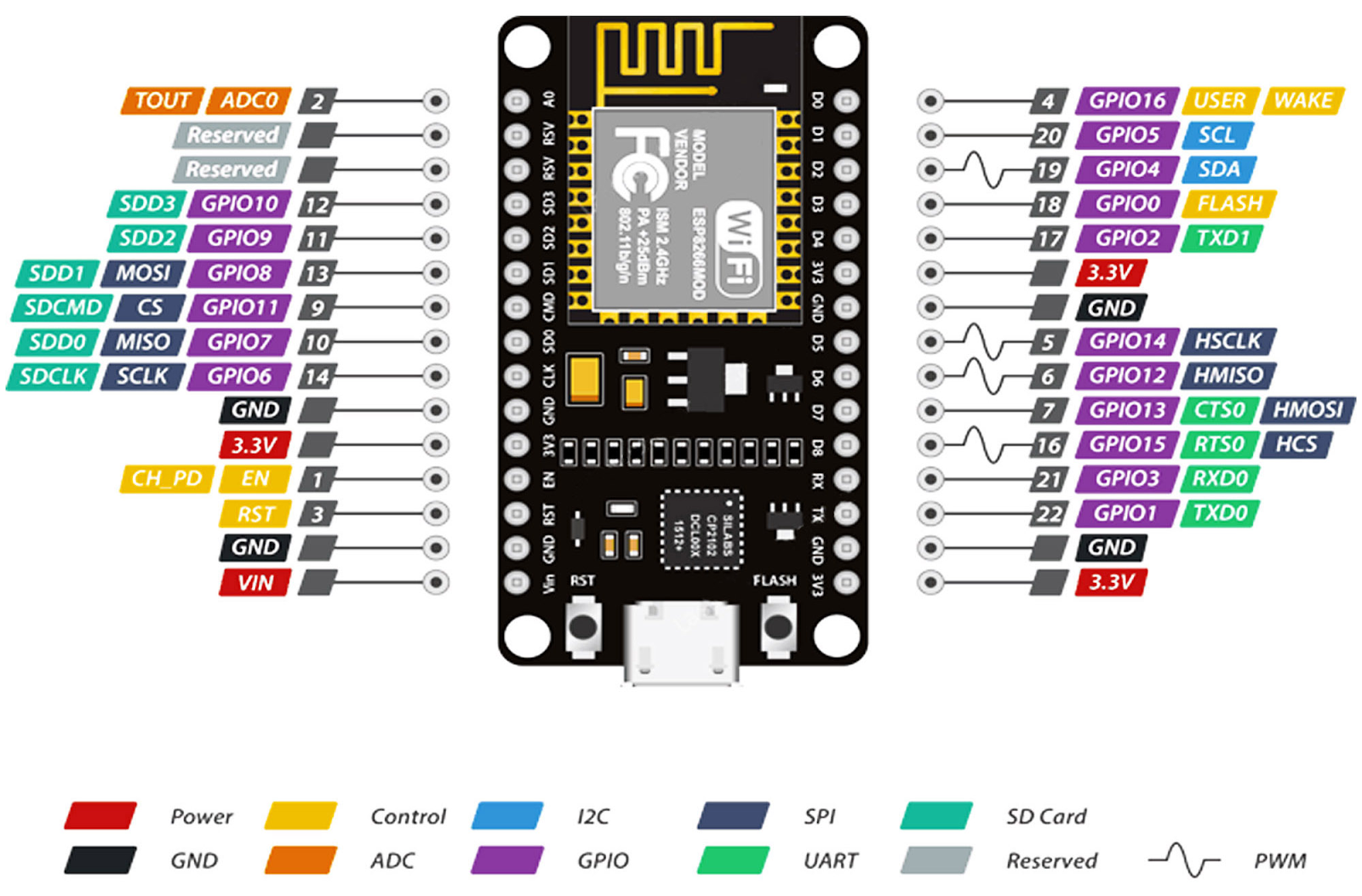
Thông số:

* Sử dụng điện áp: 5V
* Dòng tiêu thụ của cảm biến chuyển động SR501: 65mA
* Đầu ra: 0 – 3.3V
* Thời gian trễ: điều chỉnh trong khoảng 5 – 300 s
* Thời gian khóa: 0.2s
* 2 chế độ hoạt động:
* (L) không lặp lại kích hoạt
* (H) lặp lại kích hoạt.
* Phạm vi hoạt động:  <120 độ C, trong vòng 7 mét
* Nhiệt độ: -15 – 70 độ C
* Kích thước PCB:32mm x 24mm

Nguyên lý hoạt động:

* Nguyên lý hoạt động của thiết bị cảm biến chuyển động được hiểu đơn giản như sau: Khi có vật thể (con người, con vật, đồ vật,...) xuất hiện trong phạm vi không gian hoạt động của các loại cảm biến như: Tia hồng ngoại, vi sóng, sóng âm,... thì các tia/ sóng này ngay lập tức sẽ bị tán xạ khiến cho cảm biến bị ngắt và tín hiệu sẽ được gửi trực tiếp đến các trung tâm điều khiển được cài đặt sẵn từ trước như: Điện thoại thông minh, laptop,...

### NodeMCU ESP8266

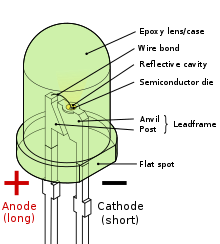


Hình 2.13 Cấu hình sơ đồ bảng phát triển NodeMCU

Thông số kĩ thuật:

* 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

### Đèn led



Hình 2.14 Đèn led

## Sơ đồ kết cấu ngôi nhà và chức năng

### Sơ đồ kết cấu

Từ một ngôi nhà thông thường, chúng em lựa chọn thiết kế ra một mô hình cơ bản dành cho 1 gia đình với 4 người ở, với thiết kế gồm:

* 1 phòng khách
* 1 phòng ngủ
* 1 khu bếp
* 1 nhà vệ sinh
* 1 ban công

### Chức năng

Với những tiêu chí về ngôi nhà thông minh tại Việt Nam, chúng em lựa chọn các chức năng tạo nên một ngôi nhà thông minh với các tiện ích như:

* Hệ thống rèm cửa tự động theo ánh sáng môi trường
* Hệ thống phơi quần áo tự động điều chỉnh theo thời tiết.
* Hệ thống báo rò rỉ khí Gas tự động.
* Hệ thống đèn, quạt tự động.
* Hệ thống điều khiển từ xa qua giao diện Web.

Phòng khách:

* Tự động bật quạt khi nhiệt độ cao
* Tự động bật đèn khi có người
* Tự động mở cửa khi có người ra ngoài

Phòng ngủ:

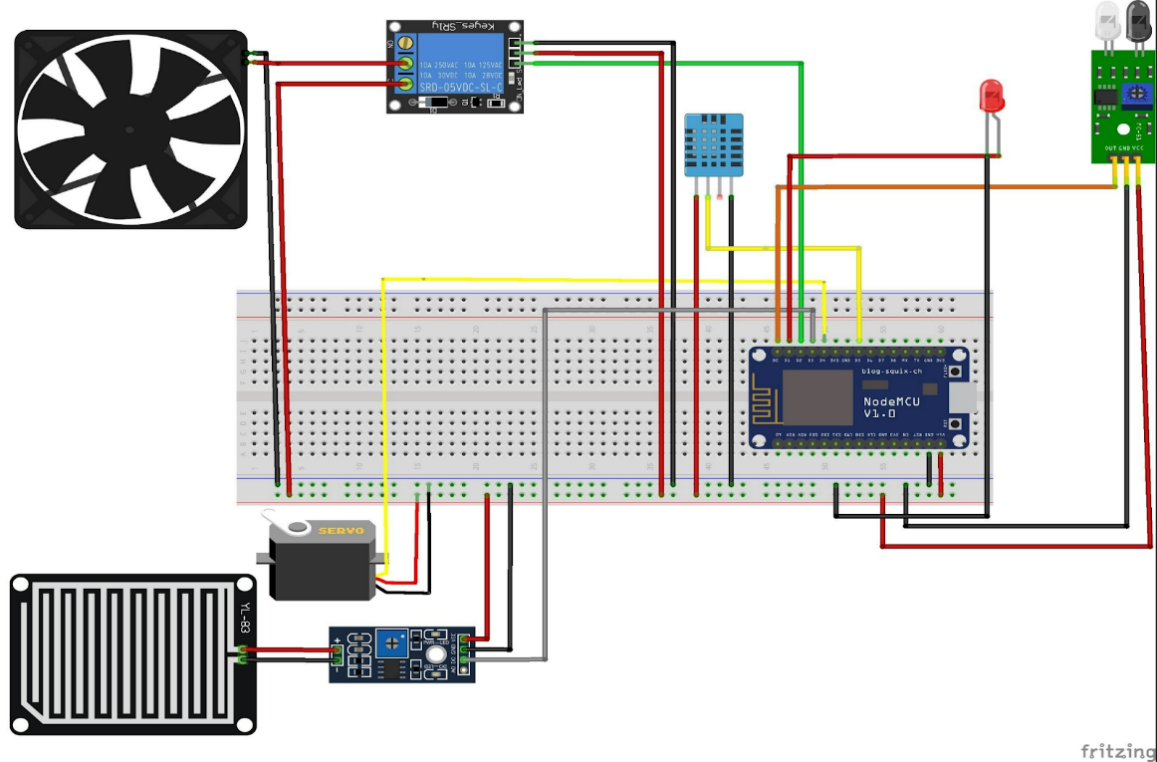
* Tự động đóng/mở rèm cửa theo ánh sáng ngoài trời, tự động đóng lại khi trời sáng và mở ra khi trời tối hoặc điều khiển trực tiếp qua giao diện web.

Phòng bếp:

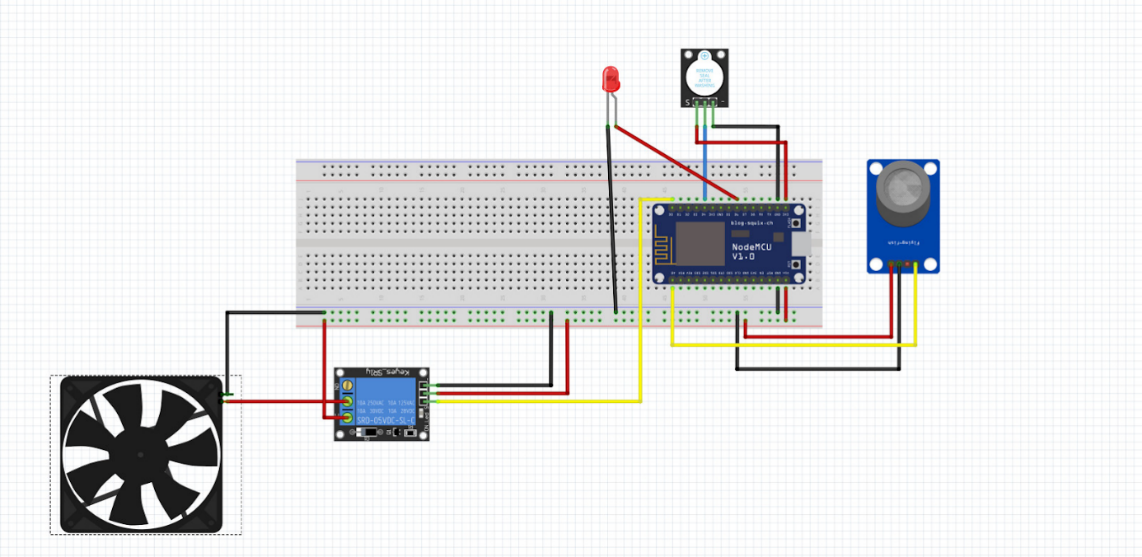
* Báo động còi khi nồng độ khí gas vượt quá mức quy định.
* Tự động bật quạt hút khí khi khí ga vượt quá quy định

Ban công

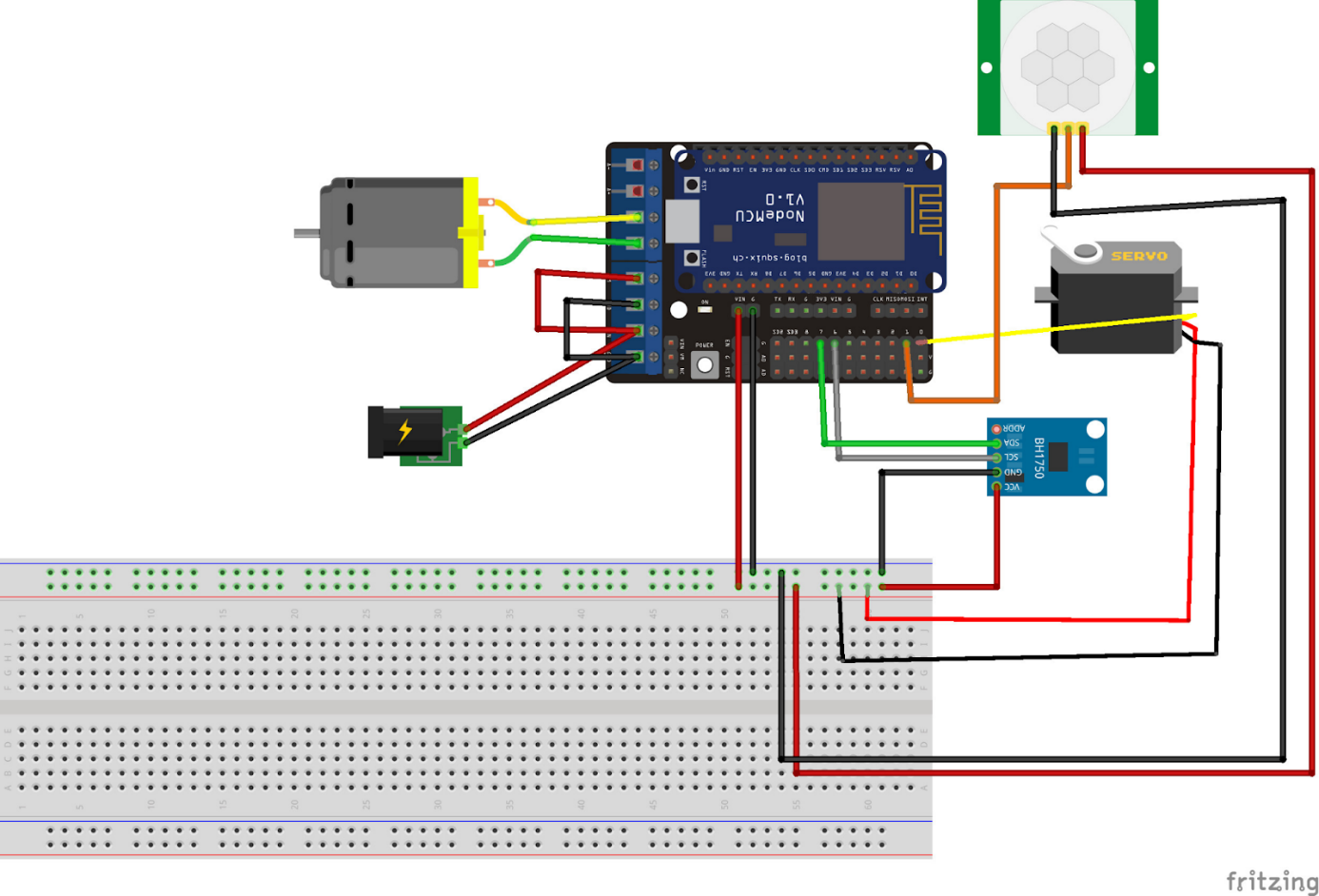
* Tự động che quần áo khi trời mưa



Hình 2.15 . Sơ đồ mạch cảm biến hồng ngoại bật đèn, cảm biệt nhiệt độ bật quạt, cảm biến mưa bật mái



Hình 2.16 Sơ đồ mạch cảm biến khí gas bật còi và quạt hút khí



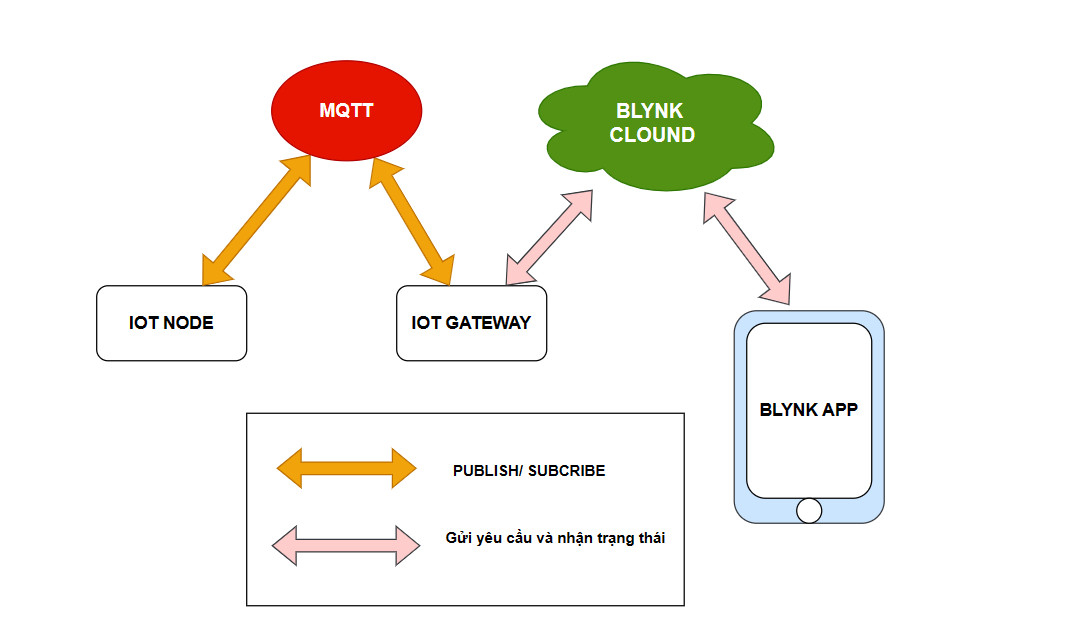
Hình 2.17 Sơ đồ mạch cảm biến kéo rèm tự động khi ánh sáng quá cao và cửa tự động mở khi phát hiện người chuyển động ở cửa nhà

# Blynk và MQTT

## Blynk và MQTT

Blynk là một nền tảng với các ứng dụng điện thoại thông minh cho phép bạn có thể dễ dàng tương tác với bộ vi điều khiển như: Arduino, Esp8266, Esp32 hoặc Raspberry qua Internet. Blynk App là một bảng điều khiển kỹ thuật số cho phép bạn có thể xây dựng giao diện đồ họa cho dự án của mình bằng cách kéo và thả các widget khác nhau mà nhà cung cấp thiết kế sẵn. Blynk không bị ràng buộc với một số bo hoặc shield cụ thể. Thay vào đó, nó hỗ trợ phần cứng mà bạn lựa chọn. Cho dù Arduino hoặc Raspberry Pi của bạn được liên kết với Internet qua Wi-Fi, Ethernet hoặc chip ESP8266, Blynk sẽ giúp bạn kết nối và sẵn sàng cho các dự án IoT.

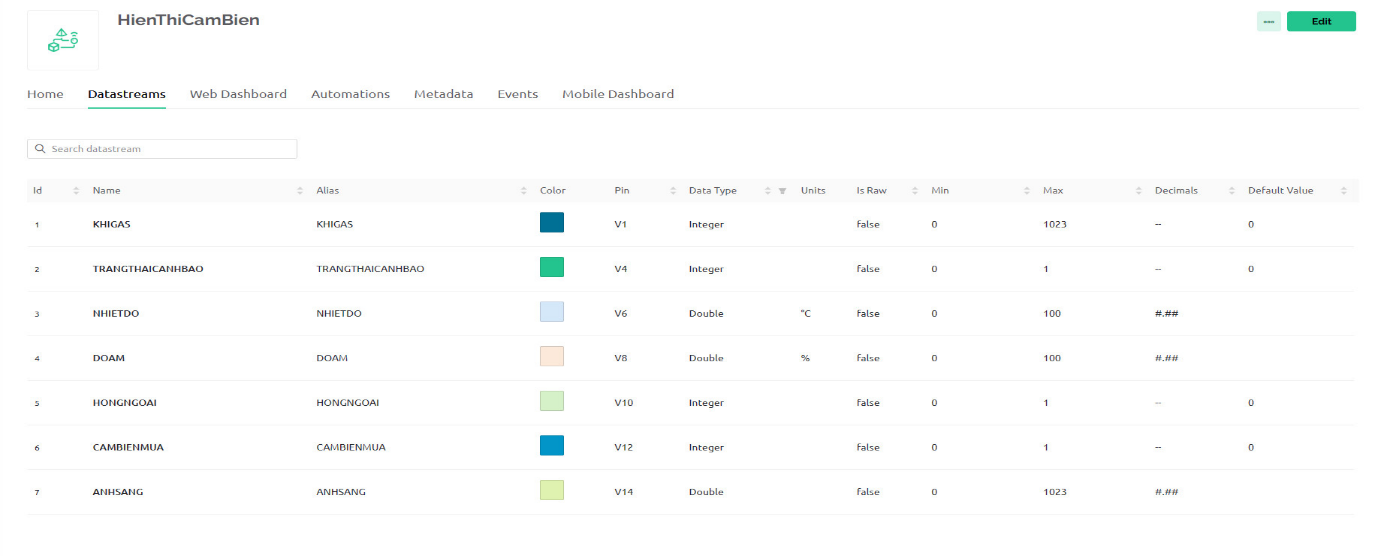
MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông nhẹ và đơn giản dùng cho việc giao tiếp giữa các thiết bị trong mạng IoT. Nó sử dụng mô hình Publish-Subscribe và giúp tiết kiệm băng thông. MQTT hỗ trợ chất lượng dịch vụ, bảo mật và được sử dụng rộng rãi trong các ứng dụng IoT như kiểm soát thiết bị thông minh và giám sát môi trường.



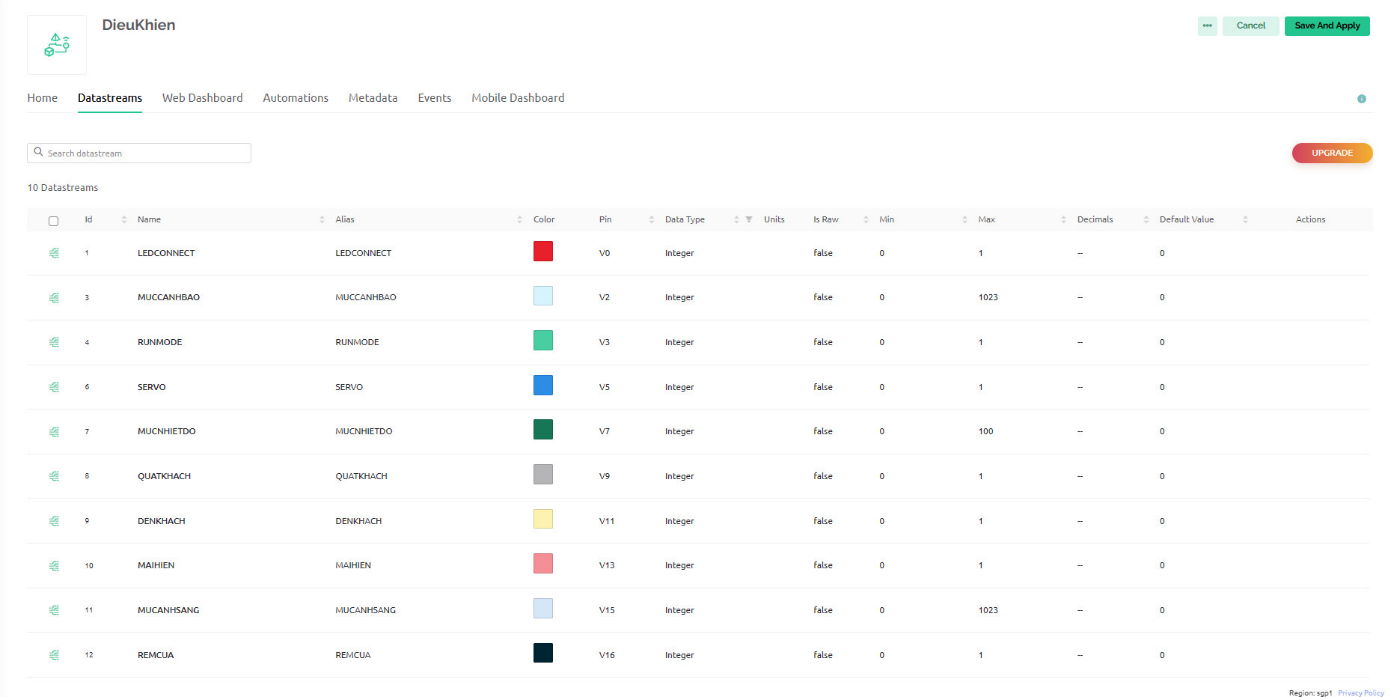
Hình 3.1 Nguyên lý hoạt động của MQTT tích hợp với Blynk

## Giao diện Blynk

### Trên web



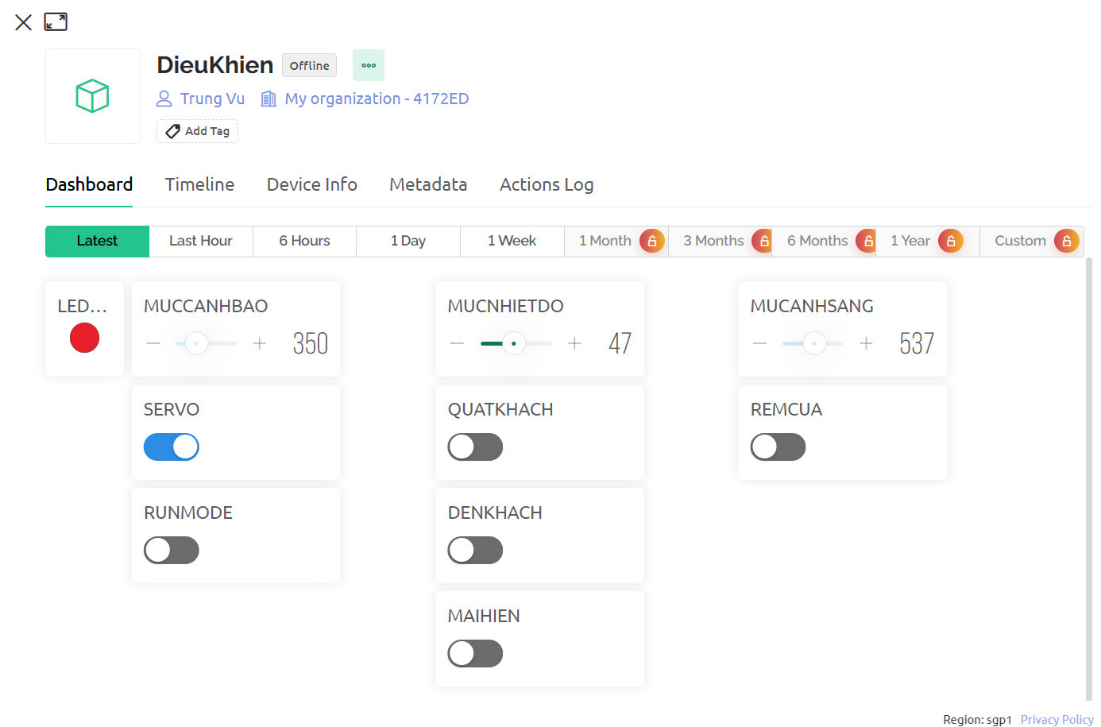
Hình 3.2 Datastream hiển thị cảm biến



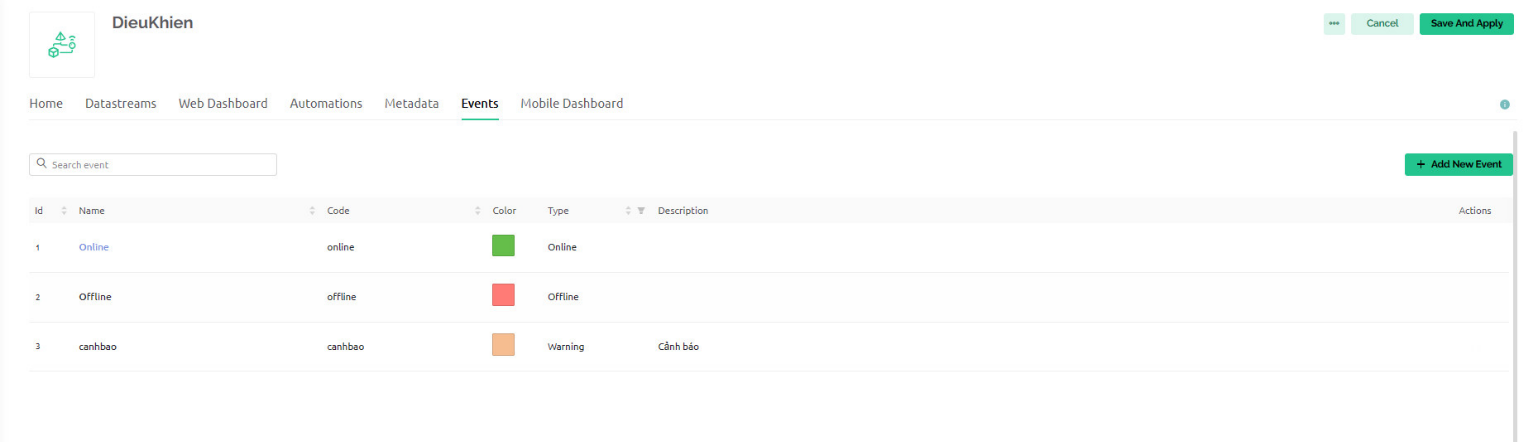
Hình 3.3 Datastream điều khiển



Hình 3.4 Giao diện Dashboard



Hình 3.5 Giao diện điều khiển

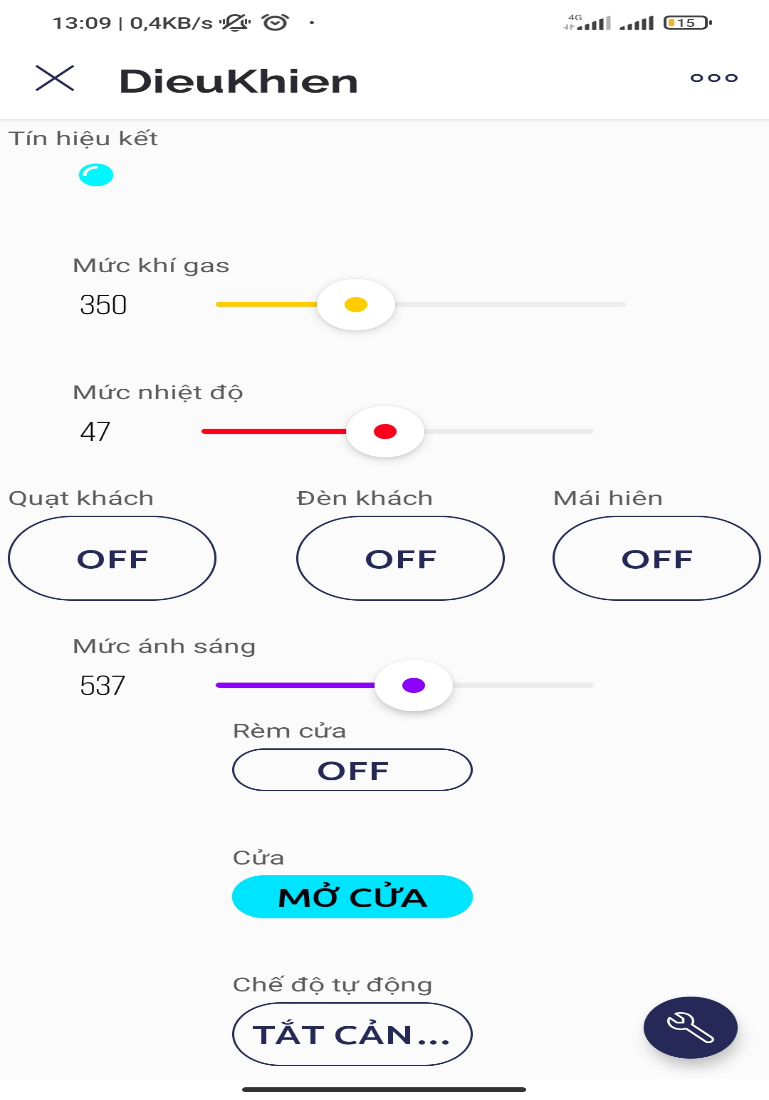


Hình 3.6 Giao diện event

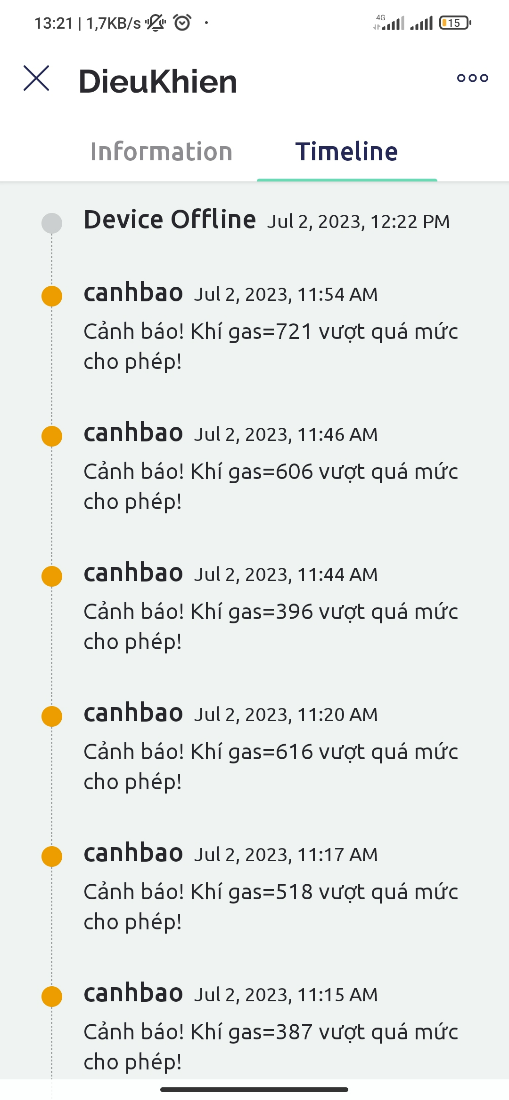
## Trên app Blynk



Hình 3.7 Giao diện các chỉ số



Hình 3.8 Giao diện điều khiển



Hình 3.9 Giao diện cảnh báo

# Code dự án

## IOT Gateway

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <BlynkSimpleEsp8266.h>

#include <string.h>

#define BLYNK\_TEMPLATE\_ID "TMPL63CBGctdQ"

#define BLYNK\_TEMPLATE\_NAME "DieuKhien"

#define BLYNK\_AUTH\_TOKEN "AoN7S0BQm9QYXQvOu\_jecZCBqX-VGj2z"

unsigned long times=millis();

int mucCanhbao=500; // Biến lưu mức cảnh báo gas

int mucNhietDo = 30; // Biến thiết lập nhiệt độ để tự động chạy quạt

int mucAnhSang = 200; // Biến thiết lập ánh sáng tự động kéo rèm

boolean runMode=1;//Bật/tắt chế độ cảnh báo

boolean cuaState=0; // Lưu trạng thái cửa

boolean quatState=0; // Lưu trạng thái quạt

boolean denState=0; // Lưu trạng thái đèn

boolean maiHienState=0; // Lưu trạng thái mái

boolean remCuaState=0; // Lưu trạng thái rèm

// KHAI BÁO BIẾN BLYNK ĐIỀU KHIỂN

WidgetLED led(V0); // Biến LED Widget của Blynk

#define MUCCANHBAO V2 // Nồng độ khi gas thiết lập

#define RUNMODE V3 // Bật chế độ tự động cho tòa nhà

#define SERVO V5 // Điều khiển cửa

// KHAI báo blynk nhiệt độ phòng khách

#define MUCNHIETDO V7 // Muc Nhiệt độ thiết lập chạy quạt

#define QUATKHACH V9 // Bật tăt quạt phòng khác

// KHái báo đèn

#define DENKHACH V11 // Bật tắt đèn phòng khách

// Khai báo mưa

#define MAIHIEN V13 // Bật tắt mái sân phơi đồ

// Khai báo biến phòng ngủ

#define MUCANHSANG V15 // Mức ánh sáng thiết lập kéo rèm

#define REMCUA V16 // Kèo rèm

BlynkTimer timer; // Thư viện hẹn giờ của Blynk

// Cấu hình Blynk

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssids[] = "Redmi 9T";

char pass[] = "11111111";

// Cấu hình MQTT

const char\* ssid = "Redmi 9T";

const char\* password = "11111111";

const char\* mqtt\_server = "test.mosquitto.org";

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

#define MSG\_BUFFER\_SIZE (50)

char msg[MSG\_BUFFER\_SIZE];

void setup\_wifi() {

delay(10);

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

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

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

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

Serial.print((char)payload[i]);

}

Serial.println();

// Lấy giá trị cảm biến gas từ topic sensingData/gas

if (strcmp(topic, "sensingData/gas") == 0) {

char payloadString[length + 1];

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

payloadString[i] = (char)payload[i];

}

payloadString[length] = '\0';

int gasValue = atoi(payloadString); // Chuyển gas về kiểu int đọc từ topic

Blynk.virtualWrite(MUCCANHBAO,mucCanhbao); // Đồng bộ mức cảnh báo

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",runMode);

// Kiểm tra trạng thái runMode bật hay tắt rồi publish lên topic

client.publish("controlData/led", msg);

// Kiểm tra nồng độ khí gas vượt quá mức thiết lập không

if(gasValue > mucCanhbao){

Serial.println("Khí Gas Cao");

// Trả về cảnh báo

Blynk.logEvent("canhbao", String("Cảnh báo! Khí gas=" + String(gasValue)+" vượt quá mức cho phép!"));

if(runMode==HIGH){

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",1);

}

client.publish("controlData/baodong", msg);

}else{

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",0);

client.publish("controlData/baodong", msg);

}

}

// Lấy tín hiệu trả về từ topic sensingData/temp

if (strcmp(topic, "sensingData/temp") == 0) {

char payloadString[length + 1]; // +1 để chứa ký tự null kết thúc chuỗi

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

payloadString[i] = (char)payload[i];

}

payloadString[length] = '\0'; // Ký tự null kết thúc chuỗi

float nhietdo = atoi(payloadString);

Serial.println(nhietdo);

// Nếu nhiệt độ cao hơn nhiệt độ thiết lập bật quạt

if(nhietdo > mucNhietDo){

Serial.println("Quạt Chạy");

if(runMode==HIGH){

Blynk.virtualWrite(QUATKHACH,1);

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",1);

client.publish("controlData/quatKhach", msg);

}

}else{

}

}

// Lấy tín hiệu trả về từ topic sensingData/hongNgoai

if (strcmp(topic, "sensingData/hongNgoai") == 0) {

char payloadString[length + 1]; // +1 để chứa ký tự null kết thúc chuỗi

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

payloadString[i] = (char)payload[i];

}

// Điều khiển đèn bật nếu tín hiệu trả về là 0 và ngược lại

if((char)payload[0] == '0'){

Serial.println("Đèn Chạy");

if(runMode==HIGH){

Blynk.virtualWrite(DENKHACH,1);

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",1);

client.publish("controlData/denKhach", msg);

}

}else{

if(runMode==HIGH){

Blynk.virtualWrite(DENKHACH,0);

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",0);

client.publish("controlData/denKhach", msg);

}

}

}

// Lấy tín hiệu trả về từ topic sensingData/rain

if (strcmp(topic, "sensingData/rain") == 0) {

char payloadString[length + 1]; // +1 để chứa ký tự null kết thúc chuỗi

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

payloadString[i] = (char)payload[i];

}

// Nếu tín hiệu mưa là 0 thi mái chạy và ngược lại

if((char)payload[0] == '0'){

if(runMode==HIGH){

Blynk.virtualWrite(MAIHIEN,1);

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",1);

client.publish("controlData/maiHien", msg);

}

}else{

if(runMode==HIGH){

Blynk.virtualWrite(MAIHIEN,0);

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",0);

client.publish("controlData/maiHien", msg);

}

}

}

// Lấy tín hiệu trả về từ topic sensingData/anhsang

if (strcmp(topic, "sensingData/anhsang") == 0) {

char payloadString[length + 1]; // +1 để chứa ký tự null kết thúc chuỗi

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

payloadString[i] = (char)payload[i];

}

payloadString[length] = '\0'; // Ký tự null kết thúc chuỗi

float anhsang = atoi(payloadString);

Serial.println(anhsang);

// Nếu mức ánh sáng lớn hơn mức ánh sáng thiết lập kéo rèm che và ngược lại

if(anhsang > mucAnhSang){

Serial.println("Rèm Chạy");

if(runMode==HIGH){

Blynk.virtualWrite(REMCUA,1);

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",1);

client.publish("controlData/remCua", msg);

}

}else{

if(runMode==HIGH){

Blynk.virtualWrite(REMCUA,0);

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",0);

client.publish("controlData/remCua", msg);

}

}

}

// Lấy tín hiệu trả về từ topic sensingData/camBienCD

if (strcmp(topic, "sensingData/camBienCD") == 0) {

char payloadString[length + 1]; // +1 để chứa ký tự null kết thúc chuỗi

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

payloadString[i] = (char)payload[i];

}

// Nếu tín hiệu bằng 1 thì mở cửa ngược lại đóng

if((char)payload[0] == '1'){

Serial.println("Mở cửa");

if(runMode==HIGH){

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",1);

client.publish("controlData/cuaTuDong", msg);

}

}else{

}

}

}

void reconnect() {

// Loop until we're reconnected

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

// Create a random client ID

String clientId = "ESP8266Client-";

clientId += String(random(0xffff), HEX);

// Attempt to connect

if (client.connect(clientId.c\_str())) {

// Publish các giá trị để điều khiển các IoTnode

Serial.println("connected");

client.publish("controlData/baodong", "hello IoT Node...");

client.publish("controlData/runmode", "%d",0);

client.publish("controlData/led", "%d",0);

client.publish("controlData/remCua","hello IoT Node...");

client.publish("controlData/servo","hello IoT Node...");

// ... and resubscribe

// Subcribe để lấy giá trị từ IoT để thực hiện logic

client.subscribe("sensingData/gas");

client.subscribe("sensingData/temp");

client.subscribe("sensingData/hongNgoai");

client.subscribe("sensingData/rain");

client.subscribe("sensingData/anhsang");

client.subscribe("sensingData/camBienCD");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

delay(5000);

}

}

}

// Kiểm tra kết nối

void blinkLedWidget(){

if (led.getValue()) {

led.off();

} else {

led.on();

}

}

void setup() {

Serial.begin(115200);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

client.setCallback(callback);

Blynk.begin(BLYNK\_AUTH\_TOKEN, ssids, pass);

delay(100);

timer.setInterval(1000L, blinkLedWidget); // Tạo hiệu ứng nhấp nháy cho LED Widget trong blynk mỗi 1s kiểm tra kết nốt

}

void loop() {

timer.run();

if (!client.connected()) {

reconnect();

}

client.loop();

}

// Đồng bộ tín hiệu với ứng dụng

BLYNK\_CONNECTED() {

Blynk.syncVirtual(MUCANHSANG,RUNMODE,MUCCANHBAO,SERVO,MUCNHIETDO,DENKHACH,MAIHIEN,REMCUA);

}

// Lấy giá trị từ Blynk để thiết lập mức cảnh báo

BLYNK\_WRITE(MUCCANHBAO) {

mucCanhbao = param.asInt();

Blynk.virtualWrite(MUCCANHBAO,mucCanhbao);

}

// Lấy giá trị từ Blynk để bật tắt đèn

BLYNK\_WRITE(RUNMODE) {

runMode = param.asInt();

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",runMode);

client.publish("controlData/led", msg);

}

// Lấy giá trị từ Blynk để điều khiển cửa

BLYNK\_WRITE(SERVO){

cuaState = param.asInt();

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",cuaState);

client.publish("controlData/servo", msg);

}

// Lấy giá trị từ Blynk để thiết lập mức nhiệt độ

BLYNK\_WRITE(MUCNHIETDO) {

mucNhietDo = param.asInt();

Blynk.virtualWrite(MUCNHIETDO,mucNhietDo);

}

// Lấy giá trị từ Blynk để điều khiển quạt

BLYNK\_WRITE(QUATKHACH){

quatState = param.asInt();

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",quatState);

client.publish("controlData/quatKhach", msg);

}

// Lấy giá trị từ Blynk để điều khiển đèn

BLYNK\_WRITE(DENKHACH){

denState = param.asInt();

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",denState);

client.publish("controlData/denKhach", msg);

}

// Lấy giá trị từ Blynk để điều khiển mái hiên

BLYNK\_WRITE(MAIHIEN){

maiHienState = param.asInt();

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",maiHienState);

client.publish("controlData/maiHien", msg);

}

// Lấy giá trị từ Blynk để thiết lập mức ánh sáng

BLYNK\_WRITE(MUCANHSANG) {

mucAnhSang = param.asInt();

Blynk.virtualWrite(MUCANHSANG,mucAnhSang);

}

// Lấy giá trị từ Blynk để điều khiển rèm cửa

BLYNK\_WRITE(REMCUA) {

remCuaState = param.asInt();

snprintf (msg, MSG\_BUFFER\_SIZE, "%d",remCuaState);

client.publish("controlData/remCua", msg);

}

## IOT Node: PHòng Bếp

#define BLYNK\_TEMPLATE\_ID "TMPL6ofIEFl\_Q"

#define BLYNK\_TEMPLATE\_NAME "HienThiCamBien"

#define BLYNK\_AUTH\_TOKEN "95UA4sFr1G0i-wXqMcOcWvy9bJYLX6Sj"

#define BLYNK\_PRINT Serial

unsigned long times=millis();

#include <BlynkSimpleEsp8266.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

BlynkTimer timer; // Thư viện hẹn giờ của Blynk

// Update these with values suitable for your network.

#define KHIGAS V1 // Biến hiển thị khí gas lên blynk

#define TRANGTHAICANHBAO V4 // Biến hiển thị trạng thái cảnh báo lên blynk

int mq2\_value;

// Cấu hình Blynk

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssids[] = "Redmi 9T";

char pass[] = "11111111";

// Cấu hình MQTT

const char\* ssid = "Redmi 9T";

const char\* password = "11111111";

const char\* mqtt\_server = "test.mosquitto.org";

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

#define MSG\_BUFFER\_SIZE (50)

char msg[MSG\_BUFFER\_SIZE];

int FAN = D0; // Kết nối tín hiệu quạt với D0

int buzzer = D4; // Kết nối còi với D4

int led = D6; // Kết nối led bật chế độ cảnh báo D6

int gas; // Biến lưu nồng độ khí gas

boolean runmode = 0; // Biến lưu giá trị khi chạy chế độ cảnh báo

void setup\_wifi() {

delay(10);

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

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

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

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

Serial.print((char)payload[i]);

Serial.println(length);

}

Serial.println();

// Kiểm tra topic mqtt controlData/buzzer để bật còi cảnh báo

if (strcmp(topic, "controlData/buzzer") == 0) {

if(payload[0] != '0'){

digitalWrite(buzzer, LOW);

}else{

digitalWrite(buzzer, HIGH);

}

}

// Kiểm tra topic mqtt controlData/led để bật led chế độ cảnh báo

if (strcmp(topic, "controlData/led") == 0) {

if(payload[0] != '0'){

runmode = 1;

digitalWrite(led, HIGH);

}else{

runmode = 0;

digitalWrite(led, LOW);

}

}

// Kiểm tra topic mqtt controlData/baodong để bật đèn, còi, quạt hút gas

if (strcmp(topic, "controlData/baodong") == 0 && runmode) {

Blynk.virtualWrite(TRANGTHAICANHBAO,0);

if(payload[0] != '0'){

digitalWrite(led, HIGH);

digitalWrite(buzzer, LOW);

Blynk.virtualWrite(TRANGTHAICANHBAO,1);

digitalWrite(FAN, HIGH);

}else{

}

}else{

digitalWrite(FAN, LOW);

digitalWrite(buzzer, HIGH);

}

}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

String clientId = "ESP8266Client-";

clientId += String(random(0xffff), HEX);

if (client.connect(clientId.c\_str())) {

Serial.println("connected");

// publish topic sensingData/gas đến mqtt để gửi tín hiệu đến gateway đăng ký

client.publish("sensingData/gas", "hello IoT Gateway...");

// publish topic subcribe controlData để nhận tín hiệu điều khiển từ gateway

client.subscribe("controlData/buzzer");

client.subscribe("controlData/led");

client.subscribe("controlData/baodong");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

delay(5000);

}

}

}

void setup() {

// Thiết lập giá trị khởi tạo còi, còi, quạt thoát khí

tone(buzzer, 1000);

digitalWrite(buzzer, HIGH);

pinMode(led, OUTPUT);

pinMode(FAN, OUTPUT);

digitalWrite(FAN, LOW);

pinMode(BUILTIN\_LED, OUTPUT); // Initialize the BUILTIN\_LED pin as an output

Serial.begin(9600);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

Blynk.begin(BLYNK\_AUTH\_TOKEN, ssids, pass);

client.setCallback(callback);

}

void loop() {

timer.run();

if (!client.connected()) {

reconnect();

}

client.loop();

unsigned long now = millis();

if (now - lastMsg > 2000) { // Gửi giá trị cảm biến mỗi 2s

lastMsg = now;

gas = analogRead(A0); // Đọc giá trị cảm biến khí gas

Blynk.virtualWrite(KHIGAS,gas); // Gửi giá trị lên Blynk

// publish giá trị lên topic

snprintf (msg, MSG\_BUFFER\_SIZE, "%d", gas);

Serial.print("Publish message: ");

Serial.println(msg);

client.publish("sensingData/gas", msg);

}

}

## IOT Node: Phòng Khách

#define BLYNK\_TEMPLATE\_ID "TMPL6ofIEFl\_Q"

#define BLYNK\_TEMPLATE\_NAME "HienThiCamBien"

#define BLYNK\_AUTH\_TOKEN "95UA4sFr1G0i-wXqMcOcWvy9bJYLX6Sj"

#define BLYNK\_PRINT Serial

#include <BlynkSimpleEsp8266.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

BlynkTimer timer;

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <DHT.h>

#include <Servo.h>

// Câu hình Blynk

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssids[] = "Redmi 9T";

char pass[] = "11111111";

// Cấu hình MQTT

const char\* ssid = "Redmi 9T";

const char\* password = "11111111";

const char\* mqtt\_server = "test.mosquitto.org";

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

#define MSG\_BUFFER\_SIZE (50)

char msg[MSG\_BUFFER\_SIZE];

//Khai báo chân của cảm biến nhiệt độ bật quạt(Phòng khách)

#define DHTPIN D5 //Chân kỹ thuật số được kết nối với cảm biến DHT

#define DHTTYPE DHT22 // DHT 22

#define FAN\_PIN D2 // FAN RELAY

//khai báo chân của cảm biến hồng ngoại bật đèn(Phòng khách)

#define SENSOR\_PIN D0 // Chân kết nối Sensor Module

#define LED\_PIN D1 // Chân kết nối đèn

//Khao báo chân cảm biến mưa thu quần áo (Sân nhà)

#define RAIN\_SENSOR\_PIN D3 // Chân kết nối cảm biến mưa

#define SERVO\_PIN D4 // Chân kết nối với servo là mái

// KHAI báo biến blynk

#define NHIETDO V6

#define DOAM V8

#define HONGNGOAI V10

#define CAMBIENMUA V12

float humDHT = 0; //Biến lưu giá trị độ ẩm

float tempDHT = 0; // Biến lưu giá trị nhiệt độ

DHT dht(DHTPIN, DHTTYPE);

Servo myServo;

int sensorValue; // Biến lưu giá trị hồng ngoại

int rainState = 1; // Biến lưu giá trị cảm biến mưa

void setup\_wifi() {

delay(10);

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

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

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

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

Serial.print((char)payload[i]);

Serial.println(length);

}

Serial.println();

// Kiểm tra tín hiệu gửi đến từ controlData/maiHien để điều khiển mái hiên

if (strcmp(topic, "controlData/maiHien") == 0) {

if(payload[0] != '0'){

myServo.write(180);

}else{

myServo.write(0);

}

}

// Kiểm tra tín hiệu gửi đến từ controlData/quatKhach để điều khiển quạt phòng khách

if (strcmp(topic, "controlData/quatKhach") == 0) {

if(payload[0] != '0'){

digitalWrite(FAN\_PIN , HIGH);

}else{

digitalWrite(FAN\_PIN, LOW);

}

}

// Kiểm tra tín hiệu gửi đến từ controlData/denKhach để điều khiển đèn phòng khách

if (strcmp(topic, "controlData/denKhach") == 0) {

if(payload[0] != '0'){

digitalWrite(LED\_PIN, HIGH);

}else{

digitalWrite(LED\_PIN, LOW);

}

}

}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

String clientId = "ESP8266Client-";

clientId += String(random(0xffff), HEX);

if (client.connect(clientId.c\_str())) {

Serial.println("connected");

// publish thông số nhiệt độ, độ ẩm, hồng ngoại phòng khách và thông số cảm biến mưa đến IoT gate way

client.publish("sensingData/humi", "hello IoT Gateway...");

client.publish("sensingData/temp", "hello IoT Gateway...");

client.publish("sensingData/hongNgoai", "hello IoT Gateway...");

client.publish("sensingData/rain", "hello IoT Gateway...");

// Nhận giá trị từ các topic để điều khiển cơ cấu chấp hành

client.subscribe("controlData/maiHien");

client.subscribe("controlData/quatKhach");

client.subscribe("controlData/denKhach");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

delay(5000);

}

}

}

void setup() {

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Phòng Khách\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//cảm biến nhiệt độ

pinMode(FAN\_PIN, OUTPUT);

digitalWrite(FAN\_PIN, LOW);

Serial.println(F("DHTxx test!"));

dht.begin();

//Cảm biến hồng ngoại

pinMode(SENSOR\_PIN, INPUT);

pinMode(LED\_PIN, OUTPUT);

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Sân nhà\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//cảm biến mưa

pinMode(RAIN\_SENSOR\_PIN, INPUT);

myServo.attach(SERVO\_PIN);

myServo.write(0);

pinMode(BUILTIN\_LED, OUTPUT); // Initialize the BUILTIN\_LED pin as an output

Serial.begin(9600);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

Blynk.begin(BLYNK\_AUTH\_TOKEN, ssids, pass);

client.setCallback(callback);

}

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

unsigned long now = millis();

if (now - lastMsg > 2000) {

lastMsg = now;

// Đọc giá trị cảm biến phòng khác

humDHT = dht.readHumidity();

Blynk.virtualWrite(DOAM,humDHT);

// Đọc nhiệt độ theo độ C (mặc định)

tempDHT = dht.readTemperature();

Blynk.virtualWrite(NHIETDO,tempDHT); // Gửi dữ liệu cảm biến lên Blynk

sensorValue = digitalRead(SENSOR\_PIN); //Đọc giá trị cảm biến hồng ngoại

Blynk.virtualWrite(HONGNGOAI,sensorValue);

// ----- Đọc giá trị cảm biến mưa----------

rainState = digitalRead(RAIN\_SENSOR\_PIN);

Blynk.virtualWrite(CAMBIENMUA,rainState); // Gửi dữ liệu cảm biến mưa lên Blynk

// Kiểm tra xem có lỗi đọc nào không và thoát sớm (để thử lại).

if (isnan(humDHT) || isnan(tempDHT))

{

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

return;

}

// publish cảm biến phòng khách nhiệt độ, độ ẩm

snprintf (msg, MSG\_BUFFER\_SIZE, "%.2f", humDHT);

Serial.print("Publish message: ");

Serial.println(msg);

client.publish("sensingData/humi", msg);

snprintf (msg, MSG\_BUFFER\_SIZE, "%.2f", tempDHT);

Serial.print("Publish message: ");

Serial.println(msg);

client.publish("sensingData/temp", msg);

// publish cảm biến mưa

snprintf (msg, MSG\_BUFFER\_SIZE, "%d", rainState);

Serial.print("Publish message: ");

Serial.println(msg);

client.publish("sensingData/rain", msg);

// publish cảm biến hồng ngoại

snprintf (msg, MSG\_BUFFER\_SIZE, "%d", sensorValue);

Serial.print("Publish message: ");

Serial.println(msg);

client.publish("sensingData/hongNgoai", msg);

}

}

## IOT Node: Phòng Ngủ

#define BLYNK\_TEMPLATE\_ID "TMPL6ofIEFl\_Q"

#define BLYNK\_TEMPLATE\_NAME "HienThiCamBien"

#define BLYNK\_AUTH\_TOKEN "95UA4sFr1G0i-wXqMcOcWvy9bJYLX6Sj"

#define BLYNK\_PRINT Serial

unsigned long times=millis();

#include <BlynkSimpleEsp8266.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

BlynkTimer timer;

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <BH1750.h>

#include <Wire.h>

#include <Servo.h>

Servo myservo;

BH1750 lightMeter; // Sử dụng thư viện của Cảm biến ánh sáng

// Cấu hình Blynk

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssids[] = "Redmi 9T";

char pass[] = "11111111";

// Cấu hình MQTT

const char\* ssid = "Redmi 9T";

const char\* password = "11111111";

const char\* mqtt\_server = "test.mosquitto.org";

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

#define MSG\_BUFFER\_SIZE (50)

char msg[MSG\_BUFFER\_SIZE];

// Khai Bao Bien CAMBIEN ANHSANG

#define ANHSANG V14 // Lưu giá trị cảm biến lên Blynk

#define LeftMotorSpeed 4 // Biến lưu tốc độ của motor rèm

#define LeftMotorDir 2 // Biến lưu hướng của motor rèm

int servo = D0; // Cổng D0 làm cổng tín hiệu servo(cửa)

int trangThaiRem=0; // Lưu trạng thái rèm hiện tại đóng hay mở

void setup\_wifi() {

delay(10);

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

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

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

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

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

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

Serial.print((char)payload[i]);

Serial.println(length);

}

Serial.println();

// Nhận tín hiệu từ topic controlData/remCua để điều khiển rèm cửa

if (strcmp(topic, "controlData/remCua" ) == 0 ) {

// Mở rèm cửa nếu trangthairem = 0 thì rèm chạy

if(payload[0] != '0' && trangThaiRem==0){

trangThaiRem = 1;

analogWrite(LeftMotorSpeed,1020);

digitalWrite(LeftMotorDir,HIGH);

delay(4000);

analogWrite(LeftMotorSpeed,0);

digitalWrite(LeftMotorDir,LOW);

}

// Đóng rèm cửa nếu trangthairem = 1 thì chạy

if(payload[0] == '0' && trangThaiRem==1){

trangThaiRem = 0;

analogWrite(LeftMotorSpeed,1020);

digitalWrite(LeftMotorDir,LOW);

delay(4000);

analogWrite(LeftMotorSpeed,0);

digitalWrite(LeftMotorDir,LOW);

}

}

// Nhận tín hiệu từ topic controlData/cuaTuDong để mở cửa, đóng cứa tự động sau 5s

if (strcmp(topic, "controlData/cuaTuDong" ) == 0 ) {

if(payload[0] != '0' ){

myservo.write(180);

delay(5000);

myservo.write(0);

}else{

}

}

// Nhận tín hiệu từ topic controlData/servo để mở cửa, đóng cứa thủ công

if (strcmp(topic, "controlData/servo") == 0) {

if(payload[0] != '0'){

myservo.write(180);

}else{

myservo.write(0);

}

}

}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

String clientId = "ESP8266Client-";

clientId += String(random(0xffff), HEX);

// Attempt to connect

if (client.connect(clientId.c\_str())) {

Serial.println("connected");

// publish chỉ số ánh sáng

client.publish("sensingData/anhsang", "hello IoT Gateway...");

// subcribe các topic để nhận tín hiệu điều khiển cơ cấu chấp hành

client.subscribe("controlData/remCua");

client.subscribe("controlData/cuaTuDong");

client.subscribe("controlData/servo");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

delay(5000);

}

}

}

void setup() {

pinMode(D1, INPUT); // Khởi tạo cổng D1 là cổng lưu giấ trị cảm biến chuyển động

pinMode(BUILTIN\_LED, OUTPUT); // Initialize the BUILTIN\_LED pin as an output

Serial.begin(9600);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

Blynk.begin(BLYNK\_AUTH\_TOKEN, ssids, pass);

client.setCallback(callback);

myservo.write(0);

myservo.attach(servo);

Wire.begin();

Wire.begin(D7, D6); // Cổng D7, D6 làm giá trị nhận cảm biến của cảm biến ánh sáng

lightMeter.begin();

Serial.println(F("BH1750 Test begin"));

// Cấu hình rèm

pinMode(LeftMotorSpeed, OUTPUT);

pinMode(LeftMotorDir, OUTPUT);

digitalWrite(LeftMotorSpeed, LOW);

digitalWrite(LeftMotorDir,HIGH);

}

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

unsigned long now = millis();

if (now - lastMsg > 2000) { // Gửi tín hiệu mỗi 2s theo thời gian thực

lastMsg = now;

float lux = lightMeter.readLightLevel(); // Đọc giá trị cảm biến ánh sáng

int camBienCD = digitalRead(D1); // Đọc giá trị cảm biến cd

Blynk.virtualWrite(ANHSANG,lux); // Gửi giá trị cảm biến lên Blynk

// Publish giá trị từ cảm biến ánh sáng lên topic

snprintf (msg, MSG\_BUFFER\_SIZE, "%.2f", lux);

Serial.print("Publish message: ");

Serial.println(msg);

client.publish("sensingData/anhsang", msg);

// Publish giá trị từ cảm biến chuyển động lên topic

snprintf (msg, MSG\_BUFFER\_SIZE, "%d", camBienCD);

Serial.print("Publish message: ");

Serial.println(msg);

client.publish("sensingData/camBienCD", msg);