**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**VÀ TRUYỀN THÔNG VIỆT – HÀN**

****

**ĐỒ ÁN CƠ SỞ 3**

**ĐIỀU KHIỂN THIẾT BỊ BẰNG GIỌNG NÓI**

Sinh viên thực hiện : **Lê Đức Nhật**

**Lê Thiện Nhân**

Giảng viên hướng dẫn: **ThS. Trần Thị Trà Vinh**

Lớp : **19CE**

*Đà Nẵng, tháng 5 năm 2021*

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**VÀ TRUYỀN THÔNG VIỆT – HÀN**

**ĐỒ ÁN CƠ SỞ 3**

**ĐIỀU KHIỂN THIẾT BỊ BẰNG GIỌNG NÓI**

*Đà nẵng, tháng 4 năm 202*

# **LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian làm đồ án cơ sở 3, chúng em đã tìm hiểu, học hỏi và nhận được rất nhiều sự hỗ trợ từ các thầy cô và các anh chị khóa trước. Chúng em cũng được học được nghiêm kinh nghiệm thông qua sử chỉ dẫn tận tình, kèm theo những bài học có giá trị mà anh chị thầy cô đi trước truyền đạt dạy bảo.

Với lòng biết ơn sâu sắc, em xin gửi lời cảm ơn đến quý thầy cô Khoa Kĩ Thuật Máy Tính và Điện Tử - Trường học Công Nghệ Thông Tin Việt – Hàn đã truyền đạt vốn kiến thức quý báu cho em cũng như các bạn cùng khóa. Qua đây chúng em cũng xin cảm ơn các anh khóa trước đã góp ý kiến giúp chúng em để hoàn thành đồ án cơ sở này.

Đặt biệt hơn, chúng em xin chân thành cảm ơn cô ThS.Trần Thị Trà Vinh đã tận tâm hướng dẫn chúng em qua từng buổi hướng dẫn để hoàn thành đồ án cơ sở. Nếu không có những lời hướng dẫn, dạy bảo của cô thì đồ án cơ sở lần này khó có thể hoàn thiện được. Mặc dù đã rất cố gắng, nhưng trong một khoảng thời gian cho phép, cũng như hạn chế về mặt kiến thức của chúng em, cuốn đồ án này không thể tránh khỏi nhiều thiếu sót... Chính vì vậy, chúng em rất mong nhận được sự góp ý của các thầy giáo, cô giáo cũng như của bạn bè để hoàn thành đề tài này tốt hơn. Một lần nữa chúng em xin chân thành cảm ơn.

# **NHẬN XÉT**

**(Của giảng viên hướng dẫn)**

Chữ kí của giảng viên

# **DANH MỤC HÌNH**

[Hình 3. 1 Sơ đồ khối của hệ thống 13](#_Toc72418433)

[Hình 3. 2 Giao diện màn hình ứng dụng 14](#_Toc72418434)

[Hình 3. 3 Lưu đồ giải thuật 15](#_Toc72418435)

[Hình 3. 4 Mô-đun wifi ESP8266 ESP-12E 16](#_Toc72418436)

[Hình 3. 5 Sơ đồ chân và sơ đồ khối của ESP8266-12E 19](#_Toc72418437)

[Hình 3. 6 Module relay 4 kênh 22](#_Toc72418438)

[Hình 4. 1 Cửa sổ làm việc của Arduino IDE 24](#_Toc72418444)

[Hình 4. 2 Thêm thư viện cho Arduino IDE 24](#_Toc72418445)

[Hình 4. 3 Cửa sổ Library Manager 25](#_Toc72418446)

[Hình 4. 4 Lưu đồ một chương trình trong Arduino IDE 25](#_Toc72418447)

[Hình 4. 5 Mã truy cập trong firebase 26](#_Toc72418448)

[Hình 4. 6 Lưu đồ module wifi 28](#_Toc72418449)

[Hình 4. 7 Khai báo biến 30](#_Toc72418450)

[Hình 4. 8 Lấy dữ liệu từ firebase về 30](#_Toc72418451)

[Hình 4. 9 Xử lí dữ liệu 31](#_Toc72418452)

[Hình 4. 10 xử lí dữ liệu nhận từ voice 31](#_Toc72418453)

[Hình 5. 1 Giao diện màn hình ứng dụng 36](#_Toc73135809)

[Hình 5. 2 relay lúc bật 36](#_Toc73135810)

[Hình 5. 3 relay lúc tắt 37](#_Toc73135811)

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc72419016)

[NHẬN XÉT ii](#_Toc72419017)

[DANH MỤC HÌNH iii](#_Toc72419018)

[MỤC LỤC iv](#_Toc72419019)

[PHẦN 1 TỔNG QUAN 1](#_Toc72419020)

[1.1 Đặt vấn đề 1](#_Toc72419021)

[1.2 Mục tiêu 1](#_Toc72419022)

[1.3 Nội dung thực hiện 2](#_Toc72419023)

[1.4 Giới hạn 2](#_Toc72419024)

[1.5 Cấu trúc 2](#_Toc72419025)

[PHẦN 2 CƠ SỞ LÝ THUYẾT 3](#_Toc72419026)

[2.1 IoT và ứng dụng trong đời sống 3](#_Toc72419027)

[2.2 Nguyên lý và hoạt động của mô hình 8](#_Toc72419028)

[2.3 Lịch sử hệ điều hành android 8](#_Toc72419029)

[2.3.1 Các phiên bản android 8](#_Toc72419030)

[2.4 Khái niệm WIFI 10](#_Toc72419031)

[2.5 Cơ sở dữ liệu Firebase: 11](#_Toc72419032)

[2.6 Cách thức hoạt động Firebase: 11](#_Toc72419033)

[PHẦN 3 THIẾT KẾ HỆ THỐNG 13](#_Toc72419034)

[3.1 Bài toán đặt ra 13](#_Toc72419035)

[3.2 Thiết kế hệ thống 13](#_Toc72419036)

[PHẦN 4 XÂY DỰNG MÔ HÌNH VÀ KẾT QUẢ 24](#_Toc72419037)

[4.1 Thi công mạch điều khiển 24](#_Toc72419038)

[4.2 Thiết kế ứng dụng 31](#_Toc72419039)

[PHẦN 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 35](#_Toc72419040)

[5.1 Kết luận về đề tài: 35](#_Toc72419041)

[5.2 Hướng phát triển: 36](#_Toc72419042)

[TÀI LIỆU THAM KHẢO 37](#_Toc72419043)

**MỞ ĐẦU**

Ngày nay, xã hội càng hiện đại, khoa học kỹ thuật ngày càng phát triển thì cuộc sống của con người càng có nhu cầu sử dụng đầy đủ các thiết bị thông minh để phục vụ cho sinh hoạt và công việc của mình. Một thực tế rất gần với con người là trong chính căn nhà của mình, mong muốn được sử dụng công nghệ tự động hóa càng được rộng rãi, tất cả đồ dùng từ trong nhà từ phòng ngủ, phòng khách đến toliet đều gắn các bộ điều khiển điện tử có thể kết nối internet và điện thoại di động, cho phép chủ nhân điều khiển vật dụng từ xa hoặc lập trình cho thiết bị ở nhà hoạt động theo lịch thời gian đúng mong muốn.

Về nhu cầu con người về kiểm soát hệ thống thiết bị điện và điều khiển thiết bị thông minh ngày càng phổ biến như kiểm tra trạng thái của đèn, quạt, máy lạnh, các thiết bị khác có thể mở hay tắt và các thiết điện trong nhà từ xa bằng thiết bị điện thoại di động, thiết bị máy tính thông qua mạng internet. Đối với hiện nay với nền khoa học phát triển với các trợ lý ảo thông minh hỗ trợ AI như: Google Assistant (google), Alexa (Amazon ), Siri (Apple), Cortana (Microsoft),.. là một nền công nghệ cao IoT là sắp trở thành một lĩnh vực mà mọi con người chú ý đến.

Ngoài ra, chúng ta còn thấy được lợi ích cụ thể khi bắt đầu và cùng sử dụng các thiết bị thông minh chỉ thông qua một chiếc smart phone bé nhỏ và có thể điều khiển dễ dàng thông qua việc kết nối wifi và giao diện thân thiện của app thật dễ dùng. Vậy nên ngại gì mà chúng ta không nên thiết kế và đưa những thiết bị thông minh này vào cuộc sống thực tế

Đặt biệt hơn nữa, các thiết bị trong ngôi nhà thông minh ngoài điều khiển qua trợ lý ảo cũng dần được phát triển và ngày càng chiếm ưu thế mạnh trong đời sống hiện nay, mang lại sự dễ dàng cho người sử dụng và dễ quản lí. Vì vậy, chúng em đã tìm hiểu, nghiên cứu, chọn đề tài: “Điều khiển thiết bị bằng giọng nói qua Google Assistant và điều khiển thông các hiệu ứng nút button bật/tắt” để làm đề tài đồ án cơ sở 3.

# **PHẦN 1 TỔNG QUAN**

1. **Đặt vấn đề**

Mạng lưới vạn vật kết nối Internet hoặc là mạng lưới thiết bị kết nối Internet viết tắt là IoT (Internet of Things) là một kịch bản của thế giới, khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, người với máy tính. IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và Internet. Nó đơn giản là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet, và với thế giới bên ngoài để thực hiện một công việc nào đó.

Giọng nói của con người đang ngày càng có quyền lực hơn khi đã có thể điều khiển mọi thiết bị công nghệ hiện đại ngày nay. Nếu trở về thời điểm của 30 năm trước thì có lẽ ý tưởng "điều khiển bằng giọng nói" sẽ được xem như một ý tưởng "viển vông", thì tính tới nay, việc điều khiển thiết bị bằng giọng nói thông qua ứng dụng đã trở nên rất phổ biến. Ngày càng nhiều ứng dụng ra đời nhằm vào điều khiển các thiết bị bằng giọng nói dễ dàng và tiện lợi hơn. Có thể kể đên những cái tên lớn như Apple Siri, Google Assistant, Amazon Alexa, Microsoft Cortana hay Samsung Bixby, và các công nghệ này tồn tại trên hầu hết mọi Smart Phone và Smart TV hiện đại trên thế giới.

Ngoài những ưu việt ở trên, thì việc áp dụng các thiết bị vào đời sống hay thiết kế một thiết bị thông minh đem lại sự tiện lợi trong việc quản lí và tránh được những lúc hay quên mà chúng ta ở xa không tắt hoặc bật được các thiết bị ở nhà. Chưa kể đến việc thiết kế ra sản phẩm hay một thiết bị IoT đem lại thu nhập cũng khá cao và cũng được mọi người khá ưa chuộng trong xã hội hiện nay.

Chính vì một số lý do trên nên nhóm chúng em quyết định làm một sản phẩm giúp nhận biết giọng nói của con người để bật tắt các thiết bị đèn chiếu sáng khi không sử dụng, hay tự động bật tắt các thiết bị qua ứng dụng, giúp tiết kiệm năng lượng. Nhằm đem lại sử quản lí dễ dàng cho mọi người khi đang ở trong nhà hoặc ở mọi nơi mà vẫn quản lí tốt các thiết bị hoạt động trong chính ngôi nhà của mình.

1. **Mục tiêu**

Các vấn đề cần giải quyết:

* Tìm hiểu về mạch và nguyên lý hoạt động của mạch.
* Xây dựng ứng dụng.
* Tìm hiểu về cơ sở dữ liệu thời gian thực.
* Tìm hiểu về Arduino.
* Tìm hiểu về hệ thống điều khiển thông minh
* Kiểm tra đánh giá ứng dụng về đề tài

1. **Nội dung thực hiện**
   * Tìm hiểu nguyên lý hoạt động và thiết kế mạch điều khiển.
   * Thiết kế mô hình
   * Xây dựng cơ sở dữ liệu.
   * Thiết kế lưu đồ giải thuật.
   * Viết chương trình, thiết kế giao diện để điều khiển thiết bị.
2. **Giới hạn**
   * Mô hình điều khiển các thiết bị trong nhà có công suất dưới 1000W.
   * Chỉ điều khiển và giám sát thiết bị.
3. **Cấu trúc**

Kết cấu quyển đồ án được trình bày theo 05 phần chính:

- Mở đầu

- Lời cảm ơn

- Mục lục

* Phần 1 Giới thiệu.
* Phần 2 Cơ sở lý thuyết.
* Phần 3 Thiết kế hệ thống
* Phần 4 Xây dựng mô hình, kết quả
* Phần 5 Kết luận, kết quả và hướng phát triển.

- Danh mục tài liệu tham khảo

# **PHẦN 2 CƠ SỞ LÝ THUYẾT**

1. **IoT và ứng dụng trong đời sống**

**2.1.1 IoT là gì?**

IoT (Internet of Things) được nghiên cứu ra đời vào năm 1999 khi con người nhận thấy tiềm năng phát triển rất nhiều nghành, nghề của xu hướng này. Song song với việc Internet vẫn còn một số rào cản ngăn cách sự phát triển thì đối ngành khoa học công nghệ đã và đang dần được khai phá.

Công nghệ IoT được hiểu là sự kết nối, kiểm soát và theo dõi trạng thái của mọi thiết bị điện tử thông minh bởi internet. Công tắc thông minh, đèn chiếu sáng, chuông cửa, máy bơm thông minh được điều khiển bằng điện thoại smartphone từ xa, thì thiết bị đó sẽ được gọi là một thiết bị IoT.

Chính vì sự ra đời và phát triển của IoT khiến cho các chuyên gia đầu ngành trong lĩnh vực khoa học công nghệ đã nghiên cứu và áp dụng vào trong cuộc sống cực kì hiệu quả và đem lại rất nhiều thành tựu về các lĩnh vực như: Quản lý hạ tầng cơ sở, ứng dụng trong nhà thông minh, giao thông thông minh,..

**2.1.2 Ứng dụng của internet of things trong nhà thông minh**

Với IoT đây có lẽ là ứng dụng phổ biến nhất, đem lại hiệu quả thiết thực nhất. Được nghiên cứu thiết kế nhằm cung cấp cho người sử dụng sự an toàn tiện ích tối ưu. Những ngôi nhà thông minh giúp bạn khi về chính ngôi nhà của mình giống như bước vào một thiên đường mới trong lành, hiện đại, tinh tế đến từng chi tiết.

Nhà thông minh là một hệ thống cho phép giám sát, theo dõi quản lý và điều khiển từ xa thông qua điện thoại hoặc máy tính bảng, để người sử dụng có thể điều khiển các trạng théo theo ý muốn ở bất cư nơi nào và với tất cả các thiết bị có kết nối thông minh như bóng đèn chiếu sáng, cửa cuốn, máy bơm phun tưới, rèm cửa,…

Hệ thống nhà thông minh sẽ giúp bạn và gia đình cảm thấy luôn thoải mái, an toàn, tiện lợi hơn bao giờ hết. Hãy tưởng tượng điều khiển ánh sáng của ngôi nhà của bạn từ xa để tạo ấn tượng như bạn đang thực sự đi nghỉ ở một khu resort 5 sao.Nếu điều hòa của bạn được bật chính xác 20 phút trước khi bạn về đến nhà để có nhiệt độ phòng hoàn hảo một không khi tươi mới, giống như có một người đang âm thầm chăm chút tỉ mỉ cho cuộc sống của bạn. Thật quá thuận tiện nhất phải không?

Ở tại Việt Nam đang rất phát công nghệ IoT trong lĩnh vực nhà thông minh, thiết bị điện thông minh đang rất phát triển, với nhiều công dụng điển hình cho điều khiển hệ thống ánh sáng, chuông cửa, quạt, điều hòa, bình nóng lạnh, cửa cuốn, máy bơm nước và các thiết bị khác.



*Hình 1.1 Ứng dụng Internet Of Things trong nhà thông minh*

**2.1.3 Ứng dụng của Internet of things trong quản lý hạ tầng**

Internet of things và những ứng dụng quan trọng nhất của chúng trong quản lý cơ sở hạ tầng, IoT hoàn toàn có thể giám sát và kiểm soát chặt chẽ tất cả các hoạt động cơ sở hạ tầng ở đô thị và nông thôn như trang trại, tàu hòa, đường đi, cầu cảng…

Mục đính chính của IoT là để sử dụng theo dõi bất kỳ thay đổi hoặc những sự kiện trong trường hợp mà cơ cấu có thể ảnh hưởng đến mức độ an toàn và sự nguy hiểm đến hạ tầng đang thi công điều này đã đem lại thành tựu vô cùng hiệu quả

Đồng thời ứng dụng IoT được sử dụng lập kế hoạch giúp cho hoạt động sửa chữa và bảo trì một cách hiệu quả.



*Hình 1.2 Ứng dụng Internet Of Things trong quản lí hạ tầng*

**2.1.4 Ứng dụng của Internet of things trong giao thông**

Các sản phẩm của IoT dùng để hỗ trợ trong trường hợp tích hợp các thông tin liên lạc, kiểm soát chính xác kết hợp luôn xử lý thông tin với nhiều hệ thống giao thông vận tải. Ứng dụng này còn được mở rộng đến hầu hết các khía cạnh về giao thông chủ yếu là đường bộ, ví dụ như là xe, cơ sở vật chất và hạ tầng, cùng với người lái xe sử dụng.

Kết nối giữa các thành phần của một hệ thống giao thông vận tải cho phép con người có thể điều khiển giao thông thông minh, không cần phải đến tận nơi điều khiển như phương pháp truyền thống, đồng thời hình thành nên bãi, nhà để xe thông minh, hàng loạt hệ thống thu phí điện tử được lắp đặt, quản lý tất cả đội xe, với mục tiêu an toàn và hỗ trợ cho ngành đường bộ.

Mặt khác IoT còn có khả năng làm giảm thiểu phần nào tai nạn giao thông, những va chạm trên các tuyến đường một cách hiệu quả. Nhờ đó mà chúng ta đã ứng dụng vào đến các trụ đèn phát sáng hai bên đường để thiết bị tự động cảm biến, tự nhận dạng ngày và đêm để thiết lập các chương trình hoạt động hợp lý nhất.



*Hình 1.3 Ứng dụng Internet Of Things trong giao thông*

**2.1.5 Ứng dụng của Internet of things trong công nghiệp**

Thiết bị internet of thing hoàn toàn có thể sử dụng để theo dõi sức khỏe từ xa và hệ thống thông báo khẩn cấp khi có những thay đổi về tình trạng sức khỏe cơ thể.

Những thiết bị theo dõi sức khỏe này có thể thực hiện đo huyết áp và nhịp tim bất cứ lúc nào với các thiết bị tiên tiến cao cấp hơn thì chúng còn có khả năng giám sát cấy ghép đặc biệt, ví dụ như máy điều hòa nhịp tim hoặc trợ thính.

Bộ cảm biến đặc biệt còn được trang bị trong không gian sinh hoạt của người già giúp theo dõi thông báo tình trạng sức khỏe thường xuyên tránh được rất nhiều nguy cơ đột tử

Thành tự đắt giá mà công nghệ IoT đem lại cho ngành y tế phải kể đến đó là những thiết bị thông minh theo dõi sức khỏe con người như đo nhịp tim, theo dõi đường huyết, kiểm tra đường huyết,..



*Hình 1.4 Ứng dụng Internet Of Things trong y tế*

**2.1.6 Ứng dụng của Internet of things trong công nghiệp**

Thiết bị IoT có thể hỗ trợ người nông dân giám sát thông số về nhiệt đô, độ ẩm không khí, tốc độ gió… giúp người nông dân luôn chủ động đối với thời tiết thay đổi thất thường. Giảm khả năng sâu bệnh, tăng hiệu quả năng suất chất lượng của sản phẩm.

Hệ thông tưới tiêu tự động cũng là một trong những ứng dụng hàng đầu giúp nông nghiệp ngày một phát triển, tiết kiệm thời gian và công sức cho người nông dân.



*Hình 1.5 Ứng dụng Internet Of Things trong công nghiệp*

1. **Nguyên lý và hoạt động của mô hình**
2. **Lịch sử hệ điều hành android**

Android là hệ điều hành được thiết kế dành cho các thiết bị di động màn hình cảm ứng với mã nguồn mở và dựa trên nền tảng Linux.

Cha đẻ của Android là một kỹ sư công nghệ yêu thích robot tên Andy Rubin sáng chế ra. Trong những ngày đầu mới hình thành, hệ điều hành android chỉ được phát triển bởi một số ít các kỹ sư chuyên ngành. Họ nhanh chóng nhận ra rằng hệ điều hành này đã không còn phù hợp với thị trường máy ảnh kỹ thuật số nữa và đã chuyển sang điện thoại di động.

Rubin đã đem dự án này giới thiệu với công ty Danger và từ những kinh nghiệm trong nhiều năm làm việc, nghiên cứu ông đã tạo ra một chiếc điện thoại mới với hệ thống bàn phím trượt cổ điển với tên gọi là T-Mobile Sidekick. Nhưng chỉ sau khoảng một năm rưỡi hoạt động thì dự án Android này phải dừng hoạt động vì cạn kiệt chi phí. Rubin đã đề nghị với công ty SamSung về việc bán lại dự án nhưng lại bị từ chối. Nhưng vận may của ông vẫn chưa hết, minh chứng đó là sự kết hợp hoàn hảo của Android và Google vào năm 2005.

Với sự trợ giúp của Google, Android đã từng bước phát triển và đạt được những thành tựu như ngày hôm nay. Mặc dù Google không phải là nơi sinh ra hệ điều hành Android, nhưng nó là nơi khiến Android được tái sinh một lần nữa.

**2.3.1 Các phiên bản android**

Đến năm 2008, sau khi trải qua nhiều lần cập nhật và dần cải thiện, bổ sung thêm các tính năng mới, đồng thời sửa chữa các lỗi trong các phiên bản trước. Hiện nay, các phiên bản của Android gồm:

*+ Android 1.0 đến 1.1:*

– Hệ điều hành di động ban đầu của Google cung cấp khả năng cơ bản với các ứng dụng tích hợp như Gmail, Maps, Lịch và YouTube.

*+ Android 1.5 hoặc Cupcake :*

– Được phát hành vào đầu năm 2009, đây là phiên bản được đặt tên đầu tiên của hệ điều hành này. Nó bao gồm một bàn phím trên màn hình và giới thiệu khuôn khổ cho các ứng dụng của bên thứ ba có thể chạy trên thiết bị di động.

*+ Android 2.0-2.1 hoặc Eclair:*

– Phiên bản 2.0 đã thêm thông tin giao thông thời gian thực, điều hướng có hướng dẫn bằng giọng nói và khả năng chụm để thu phóng vào hệ điều hành.

*+ Android 2.3 hoặc Gingerbread :*

– Bản phát hành năm 2010 của hệ điều hành này tập trung vào giao diện màu đen và xanh lá cây khi Android bắt đầu phát triển giao diện đặc biệt.

*+ Android 3.0-3.2 hoặc Honeycomb*

– Được phát hành vào năm 2011, phiên bản hệ điều hành này dành riêng cho các thiết bị máy tính bảng và được giới thiệu các nút trên màn hình.

*+ Android 4.0 hay Ice Cream Sandwich*

– Đây là hệ điều hành thống nhất dành cho điện thoại và máy tính bảng, tất cả đều được phát hành vào năm 2011, có giao diện ba chiều và sử dụng rộng rãi thao tác vuốt khi sử dụng hệ thống.

*+ Android 4.4 hoặc Kit-Kat*

– Cuối năm 2013 đã thấy bản cập nhật này làm sáng màu giao diện và giới thiệu với thế giới hỗ trợ “OK, Google”.

*+ Android 5.0-5.1 hoặc Lollipop*

– Những thay đổi lớn đã được thực hiện với bản nâng cấp của năm 2014. Google đã giới thiệu tiêu chuẩn Thiết kế Vật liệu dựa trên thẻ được sử dụng để thống nhất giao diện của các mục được Hệ điều hành hiển thị.

*+ Android 6.0 hoặc Marshmallow*

– Bản cập nhật tương đối nhỏ này vào năm 2015 là khởi đầu cho các mô hình chuyển tiếp phiên bản mới, được đánh số của Google hàng năm.

*+ Android 7.0-7.1 hoặc Nougat :*

– Mục nhập của năm 2016 trong từ điển hệ điều hành Android đã thêm chế độ chia màn hình gốc và khởi chạy Trợ lý Google.

*+ Android 9 hoặc Pie :*

– Google đã phát hành phiên bản Android được đặt tên cuối cùng này vào tháng 8 năm 2018. Các cập nhật dễ thấy nhất của phiên bản này đối với người dùng là nút Home lớn và nút Quay lại nhỏ cho giao diện người dùng và các tính năng bảo mật mới .

+ *Android 10 :*

**-** Android 10 được phát hành vào ngày 3 tháng 9 năm 2019. Với đầy đủ các tính năng năng mới mẽ như trả lời thông minh, bộ khuếch đại âm thanh, điều hướng bằng cữ chỉ,...

Hiện nay có nhiều phiên bản nỗi bật như Android 11 và 12 cũng như phiên meta Android 13 sắp ra mắt với các tính năng vợt bậc so với tất cả các phiên bản.

**2.4 Khái niệm WIFI**

Wifi là chữ viết tắt của từ Wireless Fidelity, là mạng kết nối Internet không dây, có khả năng sử dụng sóng vô tuyến để truyền tín hiệu. Loại sóng này tương tự như sóng điện thoại, sóng truyền hình hay sóng radio và hầu hết các thiết bị điện tử thông minh hiện nay đều có thể kết nối được Wifi.

Wifi chủ yếu hoạt động trên băng tần 54 Mbps, dựa trên chuẩn kết nối IEEE 802.11 và có thể đạt tín hiệu mạnh nhất trong khoảng cách gần 31 mét theo lý thuyết. Còn trong thực tế thì do có nhiều vật cản trên đường truyền sóng Wifi nên khoảng cách đạt tín hiệu mạnh sẽ bị thu hẹp lại.

*+ Nguyên lí hoạt động của Wifi:*

Mạng Internet sẽ được các ISP (nhà cung cấp dịch vụ Internet - Internet Services Provider) truyền đến bộ giải mã tín hiệu số (Modem), thông qua bộ định tuyến (Router) hay chúng ta hay gọi là “bộ phát wifi” chuyển tín hiệu hữu tuyến thành kết nối vô tuyến và đưa đến các thiết bị di động không dây thông qua chuẩn kết nối WiFi.

Các thiết bị không dây tiếp nhận sóng WiFi thông qua một thiết bị chuyển đổi tín hiệu gọi là Adapter (card Wifi) được cài đặt trực tiếp trên các thiết bị. Tín hiệu vô tuyến sẽ được giải mã ngay trên thiết bị, từ đây người dùng có thể trực tiếp truy cập Internet như bình thường.

**2.5 Cơ sở dữ liệu Firebase:**

Firebase là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Cụ thể là những giao diện lập trình ứng dụng API đơn giản. Mục đích nhằm tăng số lượng người dùng và thu lại nhiều lợi nhuận hơn.

Đặc biệt, còn là dịch vụ đa năng và bảo mật cực tốt. Firebase hỗ trợ cả hai nền tảng Android và IOS. Không có gì khó hiểu khi nhiều lập trình viên chọn Firebase làm nền tảng đầu tiên để xây dựng ứng dụng cho hàng triệu người dùng trên toàn thế giới.

**2.6 Cách thức hoạt động Firebase:**

Firebase hiện nay bao gồm các hoạt động như:

*+ Firebase Realtime Database:*

Khi đăng ký một tài khoản trên Firebase để tạo ứng dụng, bạn đã có một cơ sở dữ liệu thời gian thực. Dữ liệu bạn nhận được dưới dạng JSON. Đồng thời nó cũng luôn được đồng bộ thời gian thực đến mọi kết nối client.

Đối với các ứng dụng đa nền tảng, tất cả các client đều sử dụng cùng một cơ sở dữ liệu. Nó được tự động cập nhật dữ liệu mới nhất bất cứ khi nào các lập trình viên phát triển ứng dụng. Cuối cùng, tất cả các dữ liệu này được truyền qua kết nối an toàn SSL có bảo mật với chứng nhận 2048 bit.

Trong trường hợp bị mất mạng, dữ liệu được lưu lại ở local. Vì thế khi có mọi sự thay đổi nào đều được tự động cập nhật lên Server của Firebase. Bên cạnh đó, đối với các dữ liệu ở local cũ hơn với Server thì cũng tự động cập nhật để được dữ liệu mới nhất.

*+ Freebase Authentication:*

Hoạt động nổi bật của Firebase là xây dựng các bước xác thực người dùng bằng Email, Facebook, Twitter, GitHub, Google. Đồng thời cũng xác thực nặc danh cho các ứng dụng. Hoạt động xác thực có thể giúp thông tin cá nhân của người sử dụng được an toàn và đảm bảo không bị đánh cắp tài khoản.

*+ Freebase Hosting:*

Cách thức hoạt động cuối cùng của Firebase được đề cập trong bài viết này là cung cấp các hosting. Hosting được phân phối qua tiêu chuẩn công nghệ bảo mật SSL từ mạng CDN.

# 

# **PHẦN 3 THIẾT KẾ HỆ THỐNG**

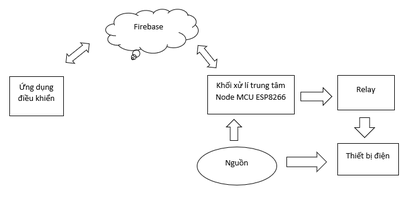
1. **Bài toán đặt ra**

Để đáp ứng nhu cầu có thể điều khiển các thiết bị thường được sử dụng như đèn, quạt, máy lạnh, TV thông qua ứng dụng di động. nhận thấy được nhu cầu ngày càng thiết thực trong cuộc sống hàng ngày nên đã quyết định thực hiện đề tài “bật thắt thiết bị trong nhà thông qua giọng nói” nhằm đáp ứng các nhu cầu trên.

Mạch điều khiển gồm những yêu cầu cần thiết sau:

* Điện áp vào: 5V
* Có kết nối chuẩn UART
* Có kết nối chuẩn SPI
* Có kết nối chuẩn I2C

1. **Thiết kế hệ thống**
   1. **Thiết kế sơ đồ khối hệ thống**

****

Hình 3. 1 Sơ đồ khối của hệ thống

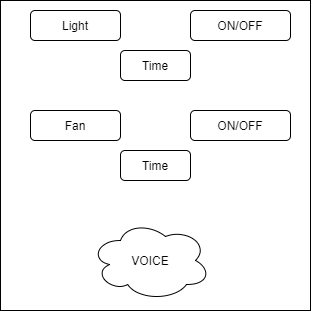
Chức năng của từng khối:

* **Khối ứng dụng điều khiển:** ứng dụng dùng để điều khiển thiết bị
* **Khối Firebase database:** thêm, điều chỉnh và đọc dữ liệu để tương tác
* **Khối xử lí trung tâm** : tạo ra lệnh điều khiển, điều khiển hoạt động của hệ thống. nhận dữ liệu từ firebase thông qua kết nối wifi, xử lí và gửi tín hiệu cho khối khác.
* **Khối nguồn**: đây là khối để cung cấp cho toàn bộ hệ thống điện, gồm 2 nguồn:

Nguồn cho mạch điều khiển: tạo ra dòng điện và điện thế ổn định cung cấp an toàn cho toàn mạch.

Nguồn cho các thiết bị điện: dùng điện 220V.

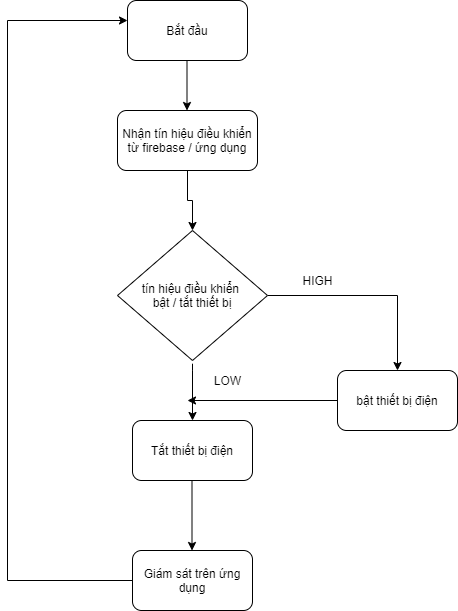
* **Khối relay**: dùng để đóng ngắt các thiết bị trong nhà. ở mô hình này, nhóm sử dụng module relay 4 kênh có thể đóng ngắt được 4 thiết bị trong nhà.
* **Khối thiết bị điện**: bao gồm các thiết bị điện trong nhà được kết nối để điều khiển được bằng giọng nói, các thiết bị điện có thể là: đèn, quạt, máy bơm, máy lạnh,…
  1. **Ứng dụng điều khiển**



Hình 3. 2 Giao diện màn hình ứng dụng

Giao diện gồm 2 thiết bị: đèn, quạt và micro, Các nút nhấn ON/OFF tương ứng với đèn và quạt. khi nhấn nút micro sẽ hiển thị giao diện nhận diện giọng nói của google, người dùng nói đúng thông tin bật tắt thì dữ liệu sẽ gửi lên firebase và xử lí.

* 1. **Lưu đồ thuật toán**

****

Hình 3. 3 Lưu đồ giải thuật

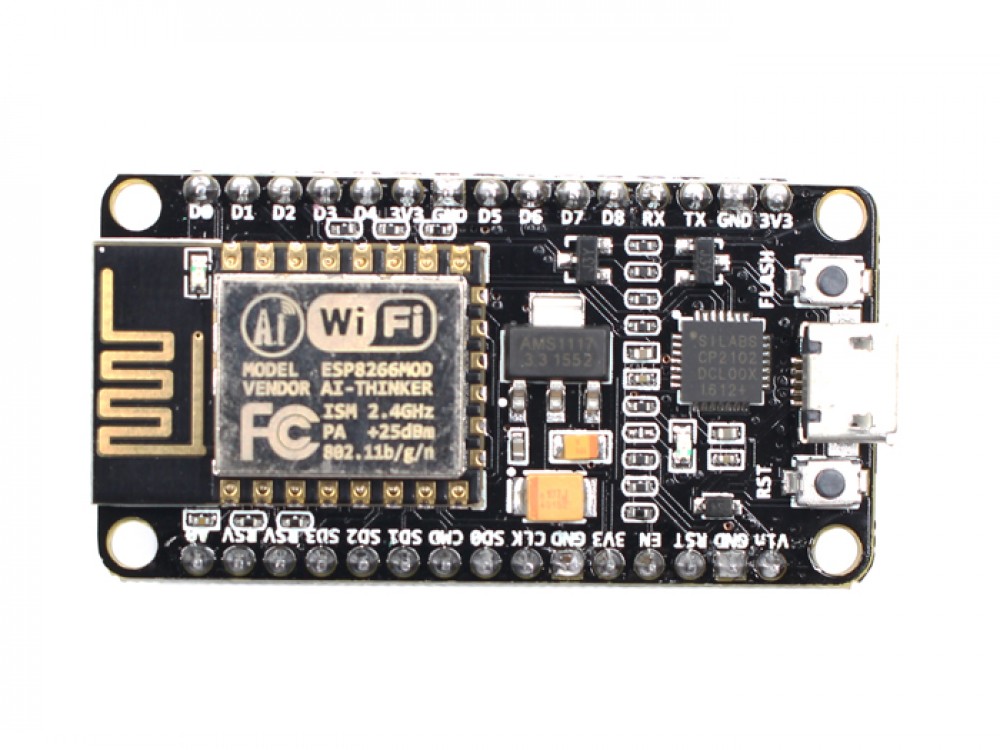
Nguyên lý hoạt động của hệ thống:

**Điều khiển bằng giọng nói**: khi ta nói một câu lệnh trên micro thì google Assistant sẽ nhận lệnh đó và sẽ chuyển đổi ra text, ứng dụng nhận kết quả và gửi lên firebase database, NodeMCU ESP8266 sẽ đọc giá trị trên firebase gửi tới và xử lí dữ liệu để kích hoạt relay tương ứng với các mức logic để bật tắt thiết bị theo câu lệnh đã cài đặt trên ứng dụng.

**Điều khiển bằng nút trên ứng dụng**: khi nhấn nút ứng dụng sẽ gửi giá trị lên firebase và NodeMCU ESP8266 sẽ đọc và xử lý dữ liệu nhận được để kích các relay tương ứng để điều khiển thiết bị điện.

* 1. **Mạch điều khiển**
     1. **Khối xử lí trung tâm NodeMCU ESP8266**

Chip ESP8266 được phát triển bởi Espressif để cung cấp giải pháp giao tiếp wifi cho các thiết bị IOT. Điểm đặc biệt của dòng ESP8266 là nó được tích hợp các mạch RF như balun, atenna switches, TX power amplifier và RX filter ngay bên trong chip với kích thước rất nhỏ chỉ 5x5mm nên các board sử dụng ESP8266 không cần kích thước board lớn cũng như không cần nhiều linh kiện xung quanh.

****

Hình 3. 4 Mô-đun wifi ESP8266 ESP-12E

Module thu phát wifi ESP8266 NodeMCU là kit phát triển dựa trên nền chip wifi SoC ESP8266 với thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 dễ dàng hơn.

Module ESP-12E chứa chip ESP8266 có bộ vi xử lí RISC 32 bit LX106 RISC hoạt động ở xung nhịp 80 – 160 MHz và hỗ trợ RTOS.

Chip ESP-12E

* Tensilica Xtensa® 32-bit LX106
* Đồng hồ Freq 80 đến 160 MHz.
* RAM nội bộ 128kB
* Đèn flash ngoài 4MB
* Bộ thu phát Wi-Fi 802.11b / g / n

Ngoài ra còn có 128KB RAM và 4 MB bộ nhớ Flash (để lưu trữ chương trình và dữ liệu) vừa đủ để đối phó với các chuỗi lớn tạo nên các trang web, dữ liệu JSON / XML và mọi thứ chúng ta hiện có trên các thiết bị IoT.

ESP8266 tích hợp bộ thu phát Wi-Fi 802.1140 / g / n HT40 , do đó, nó không chỉ có thể kết nối với mạng WiFi và tương tác với Internet mà còn có thể thiết lập một mạng riêng, cho phép các thiết bị khác kết nối trực tiếp với nó Điều này làm cho ESP8266 NodeMCU trở nên linh hoạt hơn.

Tóm lược thông số kĩ thuật:

* Vi điều khiển: CPU RISC 32-bit RISC 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
* UART: 1
* SPI: 1
* I2C: 1
* Bộ nhớ flash: 4 MB
* SRAM: 64 KB
* Tốc độ xung nhịp: 80 MHz
* USB-TTL dựa trên CP2102 được tích hợp trên bo mạch, kích hoạt Plug n Play
* Anten PCB
* Mô-đun cỡ nhỏ để phù hợp thông minh trong các dự án IoT của bạn

**Yêu cầu năng lượng:**

Vì dải điện áp hoạt động của ESP8266 là 3V đến 3,6V , bo mạch đi kèm với bộ điều chỉnh điện áp LDO để giữ điện áp ổn định ở mức 3,3V. Nó có thể cung cấp đáng tin cậy tới 600mA, mức này là quá đủ khi ESP8266 kéo tối đa 80mA trong khi truyền RF . Đầu ra của bộ điều chỉnh cũng được chia ra một trong các cạnh của bảng và được dán nhãn là 3V3. Pin này có thể được sử dụng để cung cấp năng lượng cho các thành phần bên ngoài.

Cấp nguồn cho ESP8266 NodeMCU được cung cấp thông qua đầu nối MicroB USB tích hợp . Ngoài ra, nếu bạn có nguồn điện áp 5V được điều chỉnh, chân VIN có thể được sử dụng để cung cấp trực tiếp cho ESP8266 và các thiết bị ngoại vi của nó.

*Chú ý*: ESP8266 yêu cầu nguồn điện 3,3V và mức logic 3,3V để liên lạc. Các chân GPIO không chịu được 5V! Nếu bạn muốn giao diện bảng với các thành phần 5V (hoặc cao hơn), bạn sẽ cần thực hiện một số thay đổi cấp độ.

**Thiết bị ngoại vi và I/O**

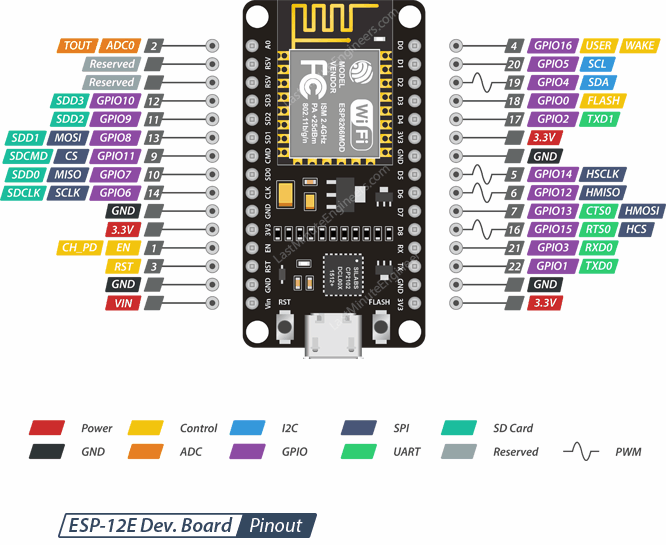
ESP8266 NodeMCU có tổng số 17 chân GPIO được chia ra cho các tiêu đề pin ở cả hai phía của bảng phát triển. Các chân này có thể được gán cho tất cả các loại nhiệm vụ ngoại vi, bao gồm:

* Kênh ADC - Kênh ADC 10 bit.
* Giao diện UART - Giao diện UART được sử dụng để tải mã ser seri.
* Đầu ra PWM - Chân PWM để làm mờ đèn LED hoặc điều khiển động cơ.
* Giao diện SPI, I2C & I2S - Giao diện SPI và I2C để kết nối tất cả các loại cảm biến và thiết bị ngoại vi.
* Giao diện I2S - Giao diện I2S nếu bạn muốn thêm âm thanh vào dự án của mình.

Nhờ tính năng ghép kênh pin của ESP8266 (Nhiều thiết bị ngoại vi được ghép kênh trên một pin GPIO duy nhất). Có nghĩa là một chân GPIO duy nhất có thể hoạt động như PWM / UART / SPI.

**Sơ đồ chân**

ESP8266 NodeMCU có tổng cộng 30 chân giao tiếp với thế giới bên ngoài. Các kết nối như sau:



Hình 3. 5 Sơ đồ chân và sơ đồ khối của ESP8266-12E

* **Power (chân nguồn)**: Có bốn chân điện viz. một chân VIN & ba chân 3,3V. Chân VIN có thể được sử dụng để cung cấp trực tiếp cho ESP8266 và các thiết bị ngoại vi của nó, nếu bạn có nguồn điện áp 5V quy định. Các chân 3,3V là đầu ra của một bộ ổn áp trên bo mạch. Các chân này có thể được sử dụng để cung cấp năng lượng cho các thành phần bên ngoài.
* **GND (chân đất)**: là một pin mặt đất của ban phát triển ESP8266 NodeMCU.
* **I2C**: được sử dụng để kết nối tất cả các loại cảm biến và thiết bị ngoại vi I2C trong dự án của bạn. Cả I2C Master và I2C Slave đều được hỗ trợ. Chức năng giao diện I2C có thể được hiện thực hóa bằng lập trình và tần số xung nhịp tối đa là 100 kHz. Cần lưu ý rằng tần số xung nhịp I2C phải cao hơn tần số xung nhịp chậm nhất của thiết bị phụ.
* **GPIO**: ESP8266 NodeMCU có 17 chân GPIO có thể được gán cho các chức năng khác nhau như I2C, I2S, UART, PWM, Điều khiển từ xa IR, Đèn LED và Nút theo chương trình. Mỗi GPIO được kích hoạt kỹ thuật số có thể được cấu hình để kéo lên hoặc kéo xuống bên trong hoặc đặt thành trở kháng cao. Khi được cấu hình làm đầu vào, nó cũng có thể được đặt thành kích hoạt cạnh hoặc kích hoạt mức để tạo ra các ngắt CPU.
* **ADC (kênh ADC)**: NodeMCU được nhúng với ADC ADC chính xác 10 bit. Hai chức năng có thể được thực hiện bằng cách sử dụng ADC viz. Kiểm tra điện áp nguồn của pin VDD3P3 và kiểm tra điện áp đầu vào của pin TOUT. Tuy nhiên, chúng không thể được thực hiện cùng một lúc.
* **UART**: ESP8266 NodeMCU có 2 giao diện UART, tức là UART0 và UART1, cung cấp giao tiếp không đồng bộ (RS232 và RS485) và có thể giao tiếp với tốc độ lên tới 4,5 Mb / giây. Các chân UART0 (chân TXD0, RXD0, RST0 & CTS0) có thể được sử dụng để liên lạc. Nó hỗ trợ kiểm soát chất lỏng. Tuy nhiên, UART1 (chân TXD1) chỉ có tính năng truyền tín hiệu dữ liệu, do đó, nó thường được sử dụng để in nhật ký.
* **SPI**: ESP8266 có hai SPI (SPI và HSPI) ở chế độ phụ và chủ. Các SPI này cũng hỗ trợ các tính năng SPI cho mục đích chung sau:
  + - * Bốn chế độ thời gian chuyển định dạng SPI
      * Lên đến 80 MHz và xung nhịp chia 80 MHz
      * Lên đến 64-byte
* **SDIO**: ESP8266 có Giao diện đầu vào / đầu ra kỹ thuật số an toàn (SDIO) được sử dụng để giao tiếp trực tiếp thẻ SD. Hỗ trợ 4 bit 25 MHz SDIO v1.1 và 4 bit 50 MHz SDIO v2.0.
* **Reserved - PWM**: Bảng mạch có 4 kênh điều chế độ rộng xung (PWM). Đầu ra PWM có thể được thực hiện theo chương trình và được sử dụng để điều khiển động cơ kỹ thuật số và đèn LED. Dải tần số PWM được điều chỉnh từ 1000 to đến 10000 s, tức là trong khoảng từ 100 Hz đến 1 kHz.
* **Control (chân điều khiển)**: được sử dụng để điều khiển ESP8266. Các chân này bao gồm chân Kích hoạt chip (EN), chân Đặt lại (RST) và chân WAKE.
  + - * Chân EN - Chip ESP8266 được bật khi chân EN được kéo CAO. Khi kéo THẤP, chip hoạt động ở công suất tối thiểu.
      * Chân RST - Chân RST được sử dụng để đặt lại chip ESP8266.
      * Pin WAKE - Pin Wake được sử dụng để đánh thức chip khỏi giấc ngủ sâu.
    1. **Module relay 4 kênh**

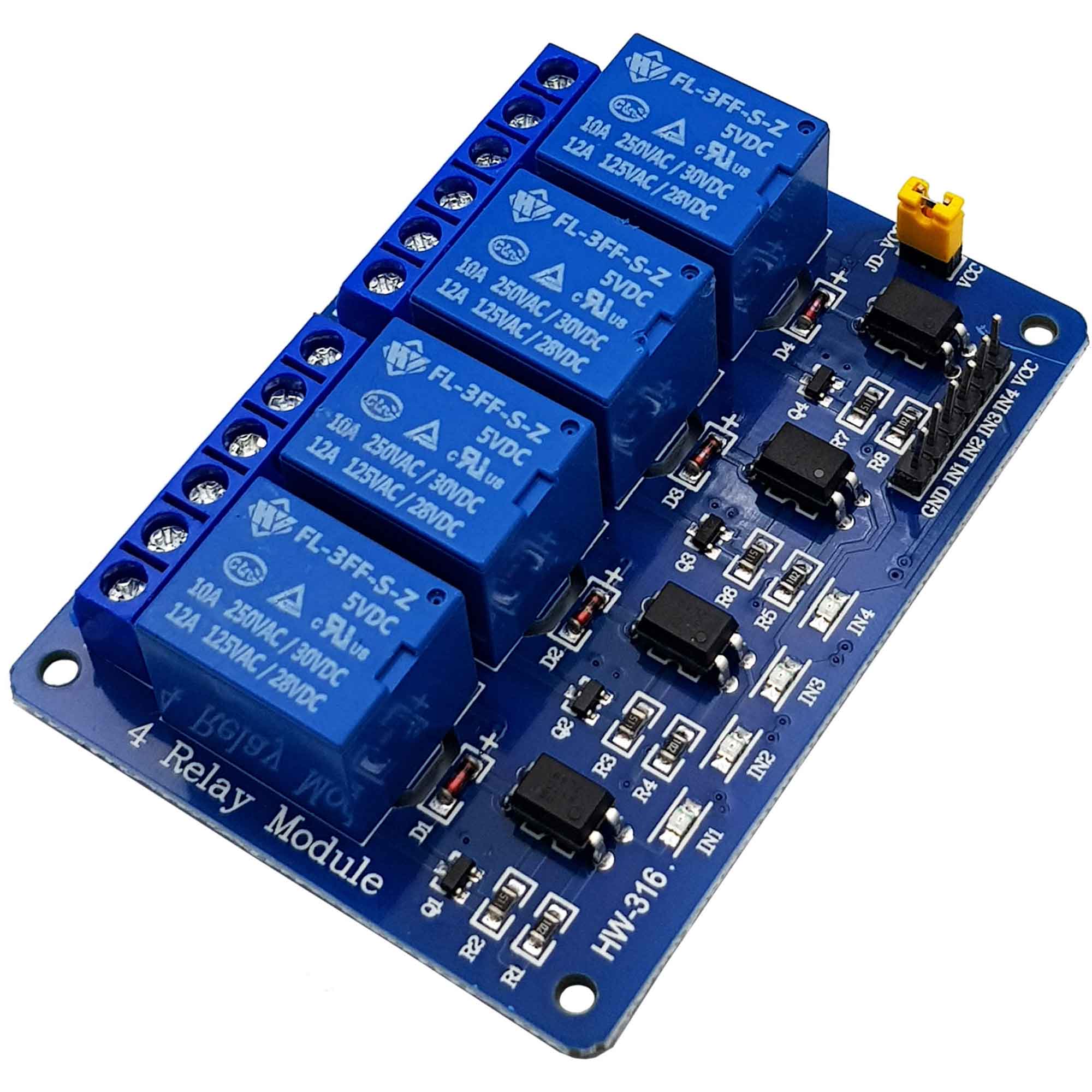
Relay là thiết bị đóng ngắt cơ bản, nó được sử dụng rất nhiều trong cuộc sống và trong các thiết bị điện tử.

Cấu tạo Relay gồm 2 phần:

* Cuộn hút: tạo ra năng lượng từ trường để hút tiếp điểm về phía mình. Tuỳ vào điện áp người ta chia relay ra:
  + DC: 5V, 12V, 24V.
  + AC: 110V, 220V
* Cặp tiếp điểm: khi không có từ trường (không cấp điện cho cuộn dây). Tiếp điểm 1 được tiếp xúc với 2 nhờ lực của lò xo. Tiếp điểm thường đóng, khi có năng lượng từ trường thì tiếp điểm 1 được hút chuyển sang 3. Trong relay có thể có 1 cặp tiếp điểm, 2 hoặc nhiều hơn.

Relay 4 kênh 5V gồm 4 rơ le hoạt động tại điện áp 5VDC. Chịu được hiệu điện thế lên đến 250VAC 10A. Relay 4 kênh 5V được thiết kế chắc chắn, khả năng cách điện tốt. trên module đã có sẵn mạch kích relay sử dụng transistor và IC cách ly quang giúp cách ly hoàn toàn mạch điều khiển với rơ le bảo đảm vi điều khiển hoạt động ổn định.

Relay 4 kênh sử dụng chân kích mức thấp (0V), khi có tín hiệu 0V vào chân IN thì relay sẽ nhảy qua trường hở. ứng dụng dùng với relat module khá nhiều bao gồm cả điện DC hay AC.



Hình 3. 6 Module relay 4 kênh

Thông số kỹ thuật:

* Sử dụng điện áp 5V DC.
* 4 relay đóng ngắt ở điện thế kích bằng 0V nên có thể sử dụng cho cả tín hiệu 5V hay 3V3. Mỗi relay tiêu thụ dòng khoảng 80mA.
* Điện thế đóng ngắt tối đa: AC250V-10A hay DC30V – 20A.
* Có đèn báo đóng ngắt trên mỗi relay.

# **PHẦN 4 XÂY DỰNG MÔ HÌNH VÀ KẾT QUẢ**

1. **Thi công mạch điều khiển**

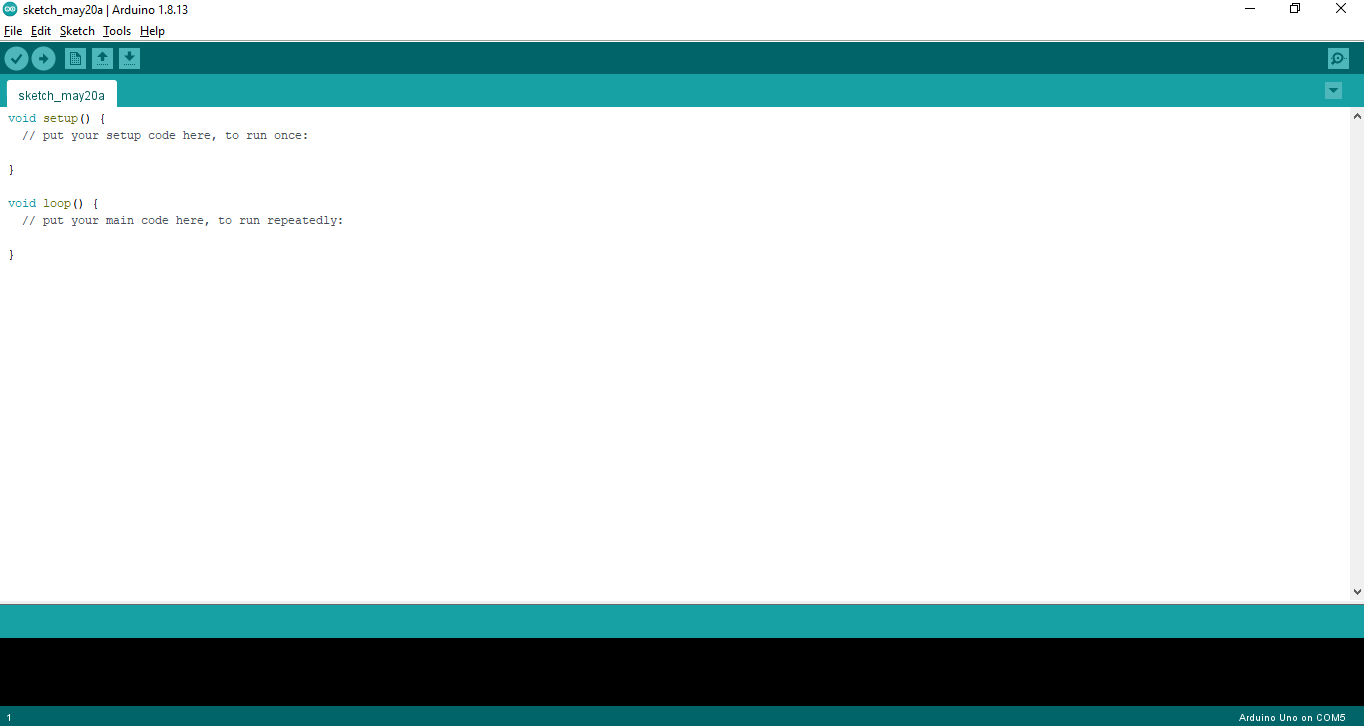
Chuẩn bị vật liệu: module ESP8266 NodeMCU, module relay 4 kênh, đèn 220V, quạt 12V, dây bus, pin.

**Phần mềm lập trình (ARDUINO IDE):**

Arduino IDE là môi trường phát triển tích hợp với mã nguồn mở của Arduino. Đây là một ứng dụng đa nền tảng và được viết trên nền tảng bằng ngôn ngữ Java, và từ IDE này chương trình sẽ được sử dụng cho ngôn ngữ lập trình nguồn mở khác. Chướng trình được thiết kế nhằm giúp cho những người dùng mới có thể làm quen dễ dàng với lĩnh vực phát triển phần mềm. Nó bao gồm đầy đủ các phần nhưu các phần mềm lập trình khác nhưng với mức độ dễ sử dụng hơn như: đánh dấu cú pháp, tự dộng canh lề, biên dịch và nạp chương trình lên board. Chương trình của Arduino được gọi là Sketch.

Các chương trình khi lập trình trên phầm mềm được viết bằng ngôn ngữ C hoặc C++. Trên Arduino IDE người dùng chỉ cần định nghĩa 2 hàm để tạo ra được một chương trình hoàn chỉnh để có thể chạy được gồm:

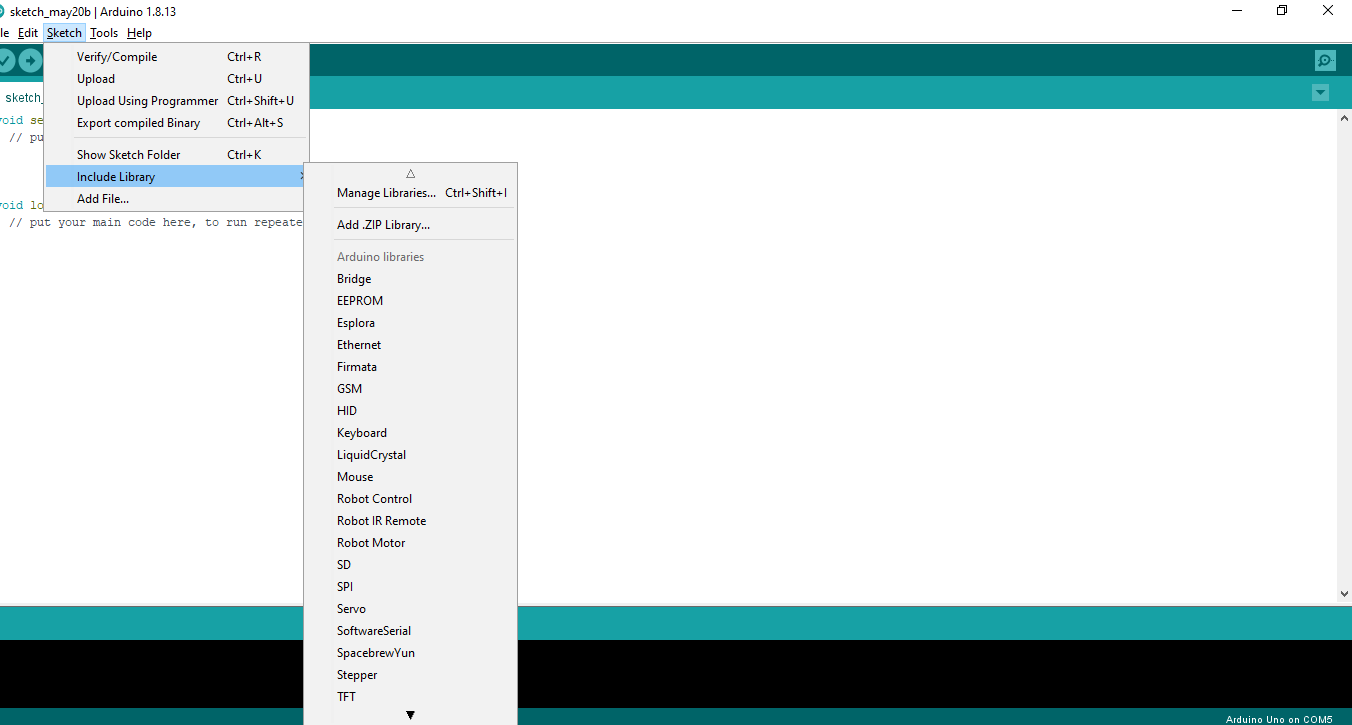
* Setup(): hàm này chạy mỗi khi khởi động chương trình, dùng để thiết đặt các thông số cài đặt lại từ đầu.
* Loop(): hàm này được hiểu là vòng lặp cho đến khi không sử dụng nưuax hay ngắt nguồn mạch điều khiển.



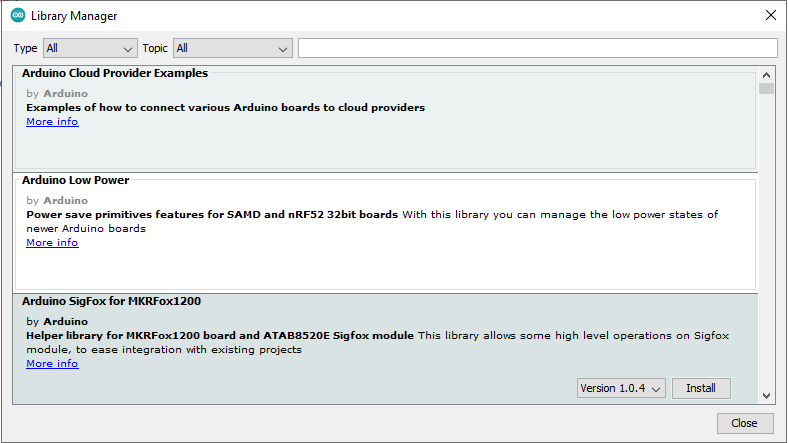
Hình 4. 1 Cửa sổ làm việc của Arduino IDE

Thêm thư viện cho Arduino IDE:

Vào Sketch 🡪 Include Library 🡪 Manage Libraries sau đó gõ vào thanh tìm kiếm thư viện cần cài đặt và nhấn Install ở góc phải

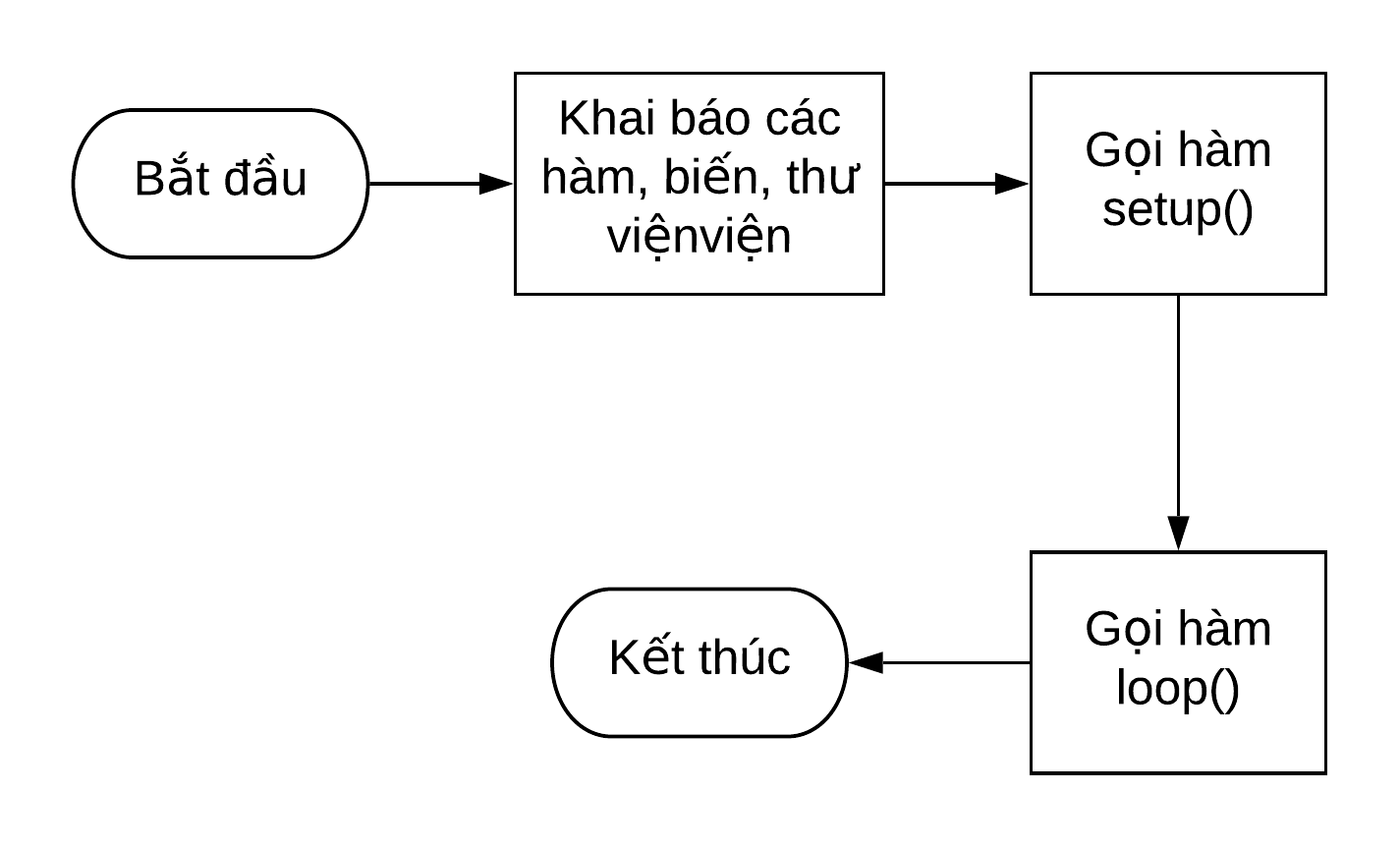


Hình 4. 2 Thêm thư viện cho Arduino IDE



Hình 4. 3 Cửa sổ Library Manager

Với yêu cầu điều khiển đã nêu như ở trên ta thiết lập được lưu đồ hệ thống sau:



Hình 4. 4 Lưu đồ một chương trình trong Arduino IDE

Sử dụng mô-đun wifi để đọc dữ liệu từ firebase thông qua mạng wifi nên cần phải khai báo hai thư viện liên quan:

**#include <ESP8266WiFi.h>**

**#include <FirebaseESP8266.h>**

Khai báo URL, mã truy cập của firebase, khai báo ID và Password của mạng wifi đang dùng.

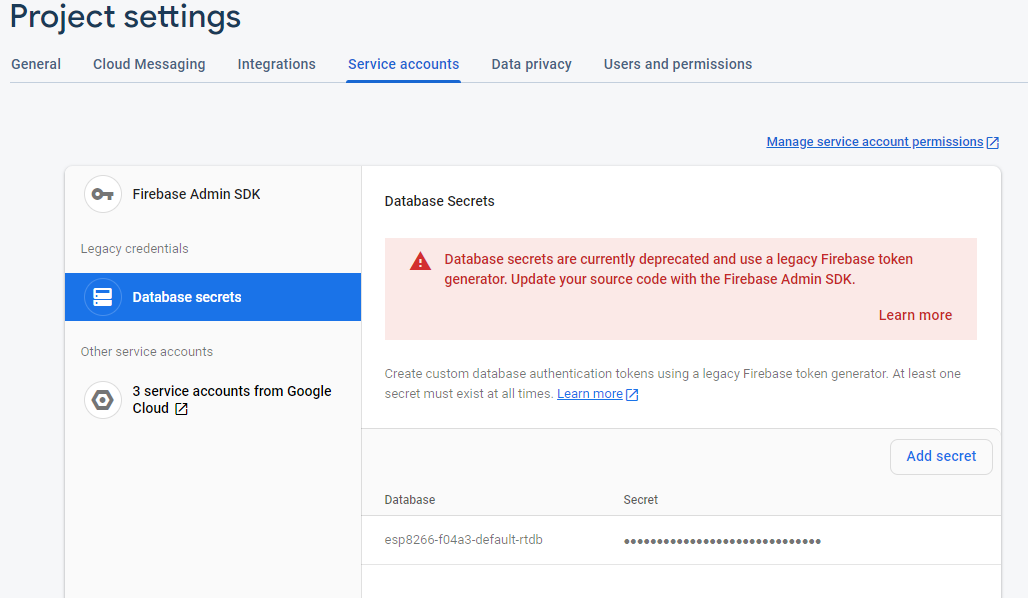
#define FIREBASE\_HOST ""

#define FIREBASE\_AUTH ""

#define WIFI\_SSID ""

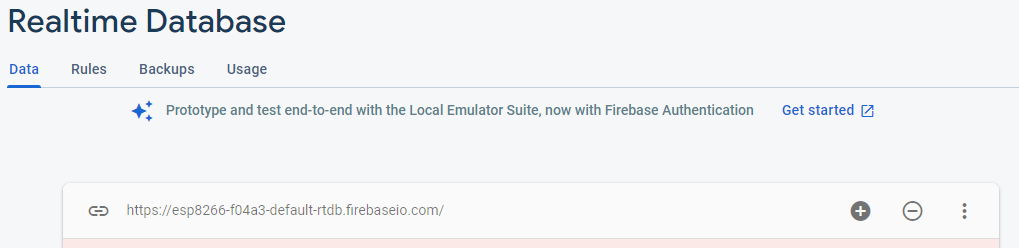
#define WIFI\_PASSWORD ""

Để lấy mã truy cập cho FIREBASE\_AUTH cho chương trình ESP8266, chúng ta vào Project Setting 🡪 Service Account 🡪 Database secrets.



Hình 4. 5 Mã truy cập trong firebase

Để lấy URL firebase chúng ta vào Database 🡪 Realtime database



Thiết lập kết nối wifi và kết nối firebase trong hàm setup():

Serial.begin(115200);

pinMode(ledPin,OUTPUT);

pinMode(fanPin, OUTPUT);

Serial.println("Serial communication started\n\n");

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //try to connect with wifi

Serial.print("Connecting to ");

Serial.print(WIFI\_SSID);

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

Serial.print(".");

delay(500);

}

Serial.println();

Serial.print("Connected to ");

Serial.println(WIFI\_SSID);

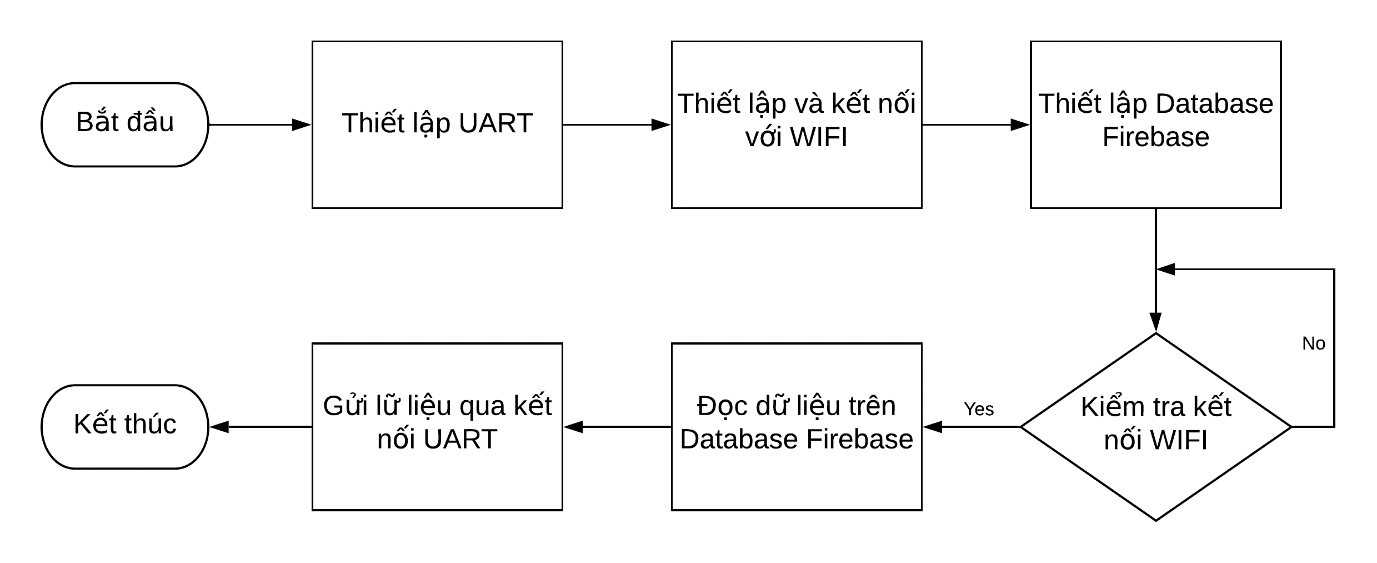
Serial.print("IP Address is : ");

Serial.println(WiFi.localIP()); //print local IP address

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH); // connect to firebase

Firebase.reconnectWiFi(true);

delay(1000);



Hình 4. 6 Lưu đồ module wifi

**Giải thích lưu đồ**:

* Bắt đầu hệ thống sẽ thiết lập UART của ESP8266, thiết lập và kết nối đến điểm chỉ định, thiết lập database firebase.
* Tiếp theo kiểm tra kết nối Wifi, nếu có kết nối thì sẽ đọc dữ liệu trên database Firebase và ngược lại thì sẽ thiết lập và kết nối lại với wifi.
* Sau khi đọc dữ liệu database Firebase, ESP8266 NodeMCU sẽ truyền dữ liệu sang cho vi điều khiển xử lý.

Xử lý dữ liệu thay đổi trong firebase ở hàm loop():

Void handleLight{

if (Firebase.getDouble(firebaseData, "/light\_time")) {

lightTime = firebaseData.doubleData();

if(lightTime == -1){

if (Firebase.getString(firebaseData, "/light\_state")){

if (firebaseData.dataType() == "string")

val = firebaseData.stringData();

Serial.println(val);

if(val == "turn\_on\_light"){

digitalWrite(ledPin, HIGH);

}else if(val == "turn\_off\_light"){

digitalWrite(ledPin, LOW);

}

}

}

}else{

if((lightTime - currentTime) < 0){

Serial.println("light time over");

Firebase.setDouble(firebaseData,"/light\_time", -1);

Firebase.setString(firebaseData,"/light\_state", "turn\_off\_light");

digitalWrite(ledPin, LOW);

}else{

digitalWrite(ledPin, HIGH);

}

}

}else {

Serial.println(firebaseData.errorReason());

}

}

Void handleFan(){

if(Firebase.getDouble(firebaseData, "/fan\_time")){

fanTime = firebaseData.doubleData();

if(fanTime == -1){

if(Firebase.getString(firebaseData, "/fan\_state")){

if(firebaseData.dataType() == "string"){

fanStatus = firebaseData.stringData();

Serial.println(fanStatus);

Serial.println("\n fan not time.");

if(fanStatus == "turn\_on\_fan"){

digitalWrite(fanPin, HIGH);

}else if(fanStatus == "turn\_off\_fan"){

digitalWrite(fanPin, LOW);

}

}

}

}else{

if((fanTime - currentTime) < 0){

Serial.println("fan time over");

Firebase.setDouble(firebaseData,"/fan\_time", -1);

Firebase.setString(firebaseData,"/fan\_state", "turn\_off\_fan");

digitalWrite(fanPin, LOW);

}else{

digitalWrite(fanPin, HIGH);

}

}

}

}

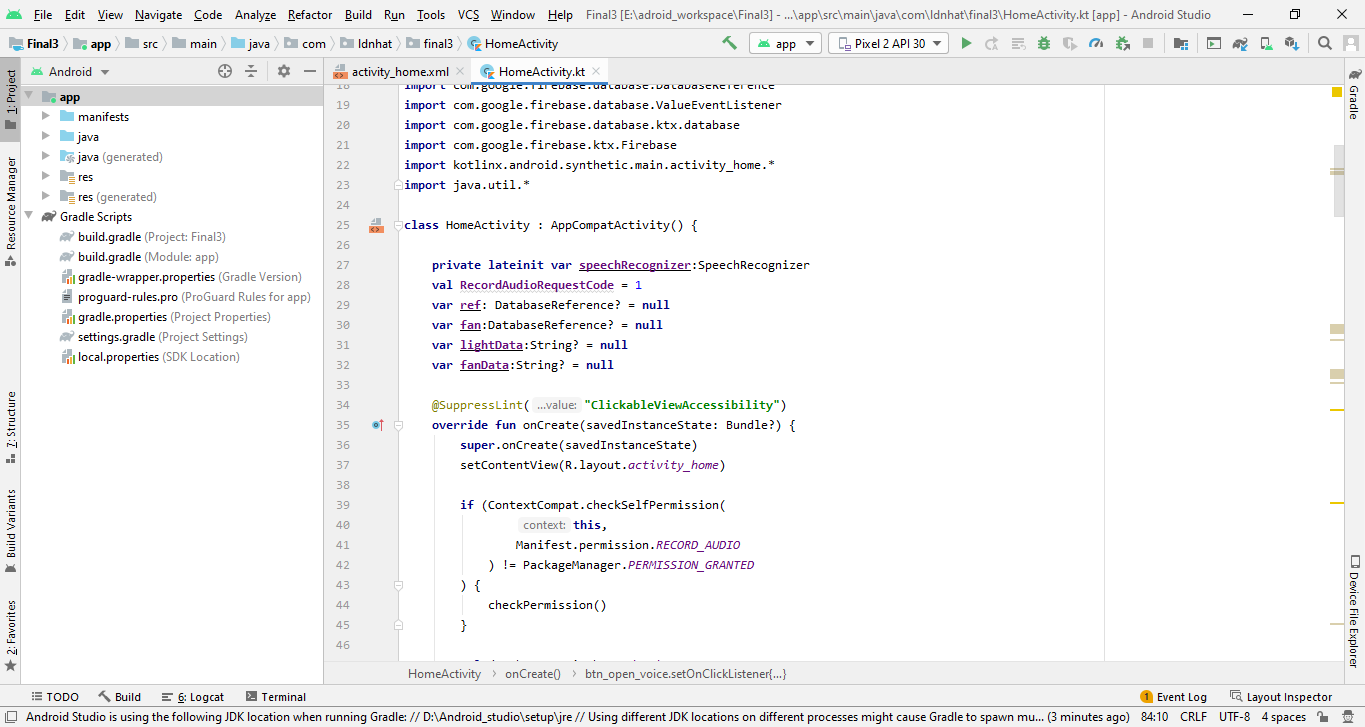
1. **Thiết kế ứng dụng**

**Phần mềm lập trình ứng dụng di động (Android Studio):**

Android Studio là một phầm mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

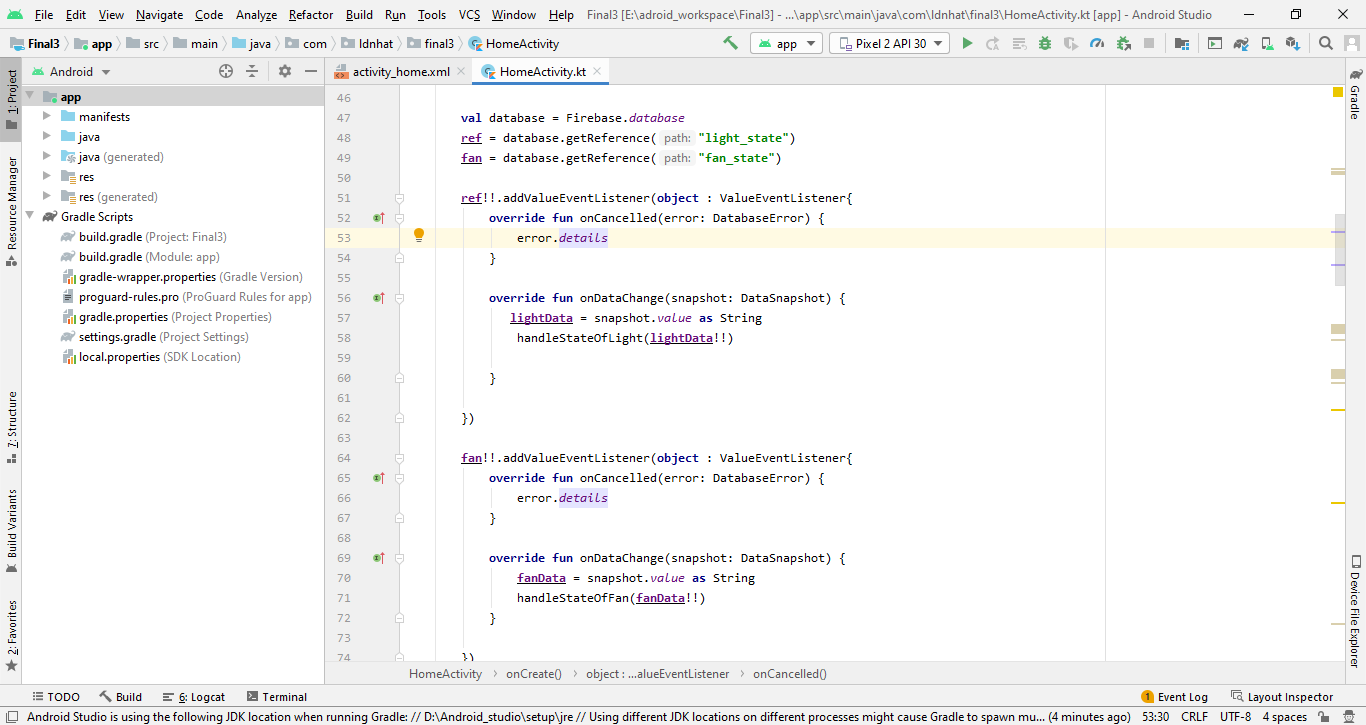
**Kết nối dữ liệu với firebase realtime database**:

Khai báo các biến:

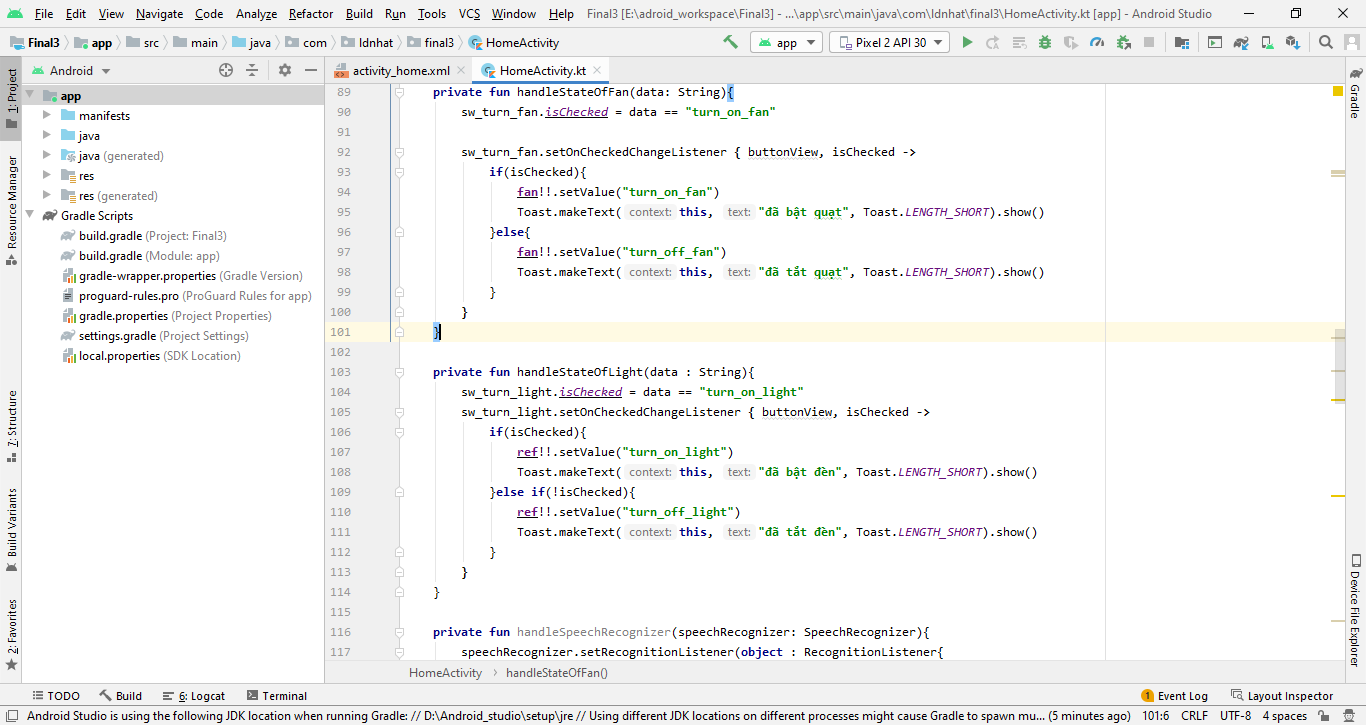


Hình 4. 7 Khai báo biến

Tham chiếu đến các node trong firebase database và xử lí dữ liệu trong hàm khởi tạo:

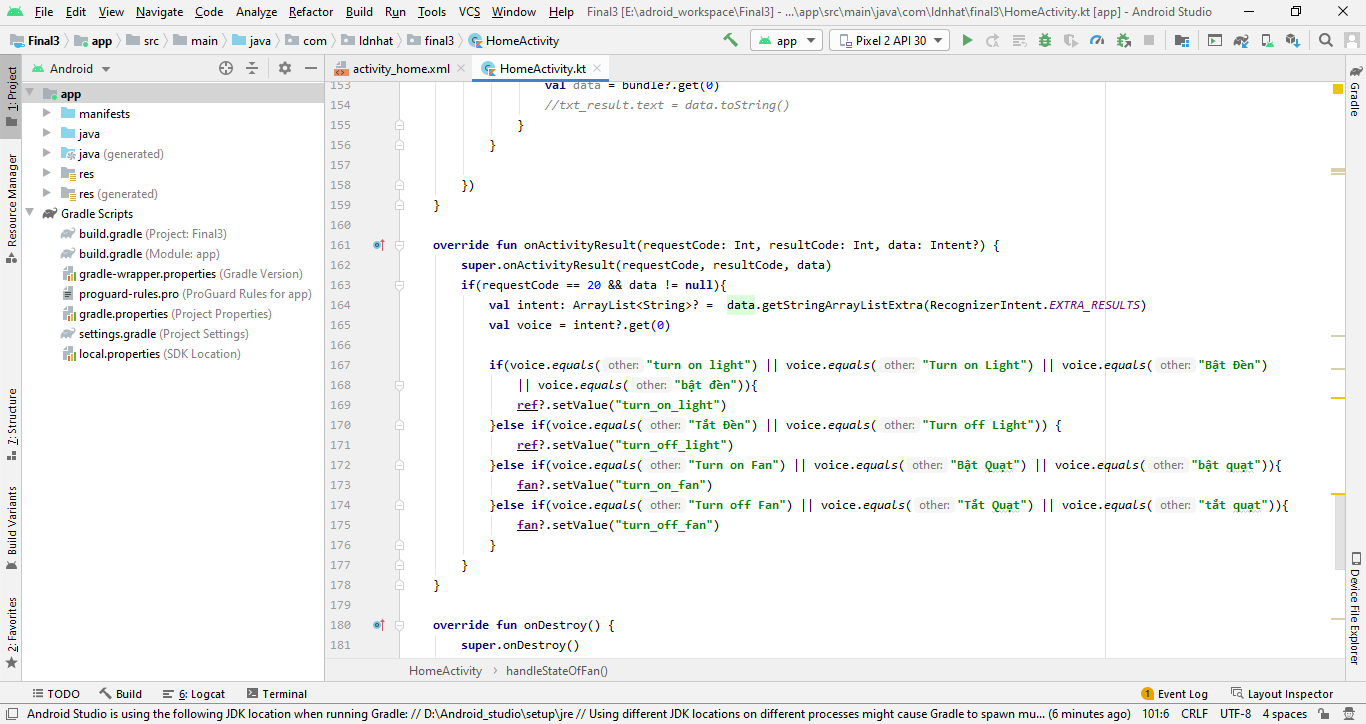


Hình 4. 8 Lấy dữ liệu từ firebase về



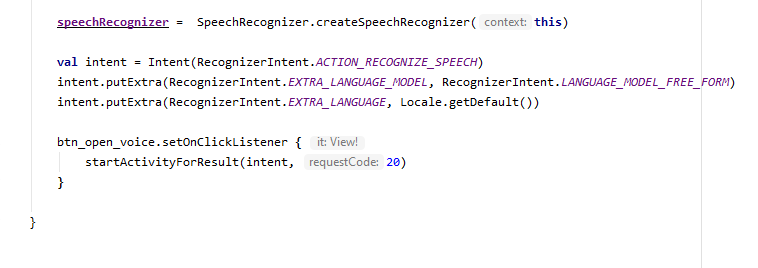
Hình 4. 9 Xử lí dữ liệu

Xử lý dữ liệu nhận được từ voice:



Hình 4. 10 xử lí dữ liệu nhận từ voice

Đoạn code này hiển thị màn hình thu âm của google:



# **PHẦN 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**5.1 Kết luận về đề tài:**

**5.1.1 Những mặt đã làm được:**

- Mạch điện với các module nhỏ trên mạch được thiết kế, thi công hoàn chỉnh và được thử nghiệm nhiều lần và đã hoạt động ổn định trong thực tế.

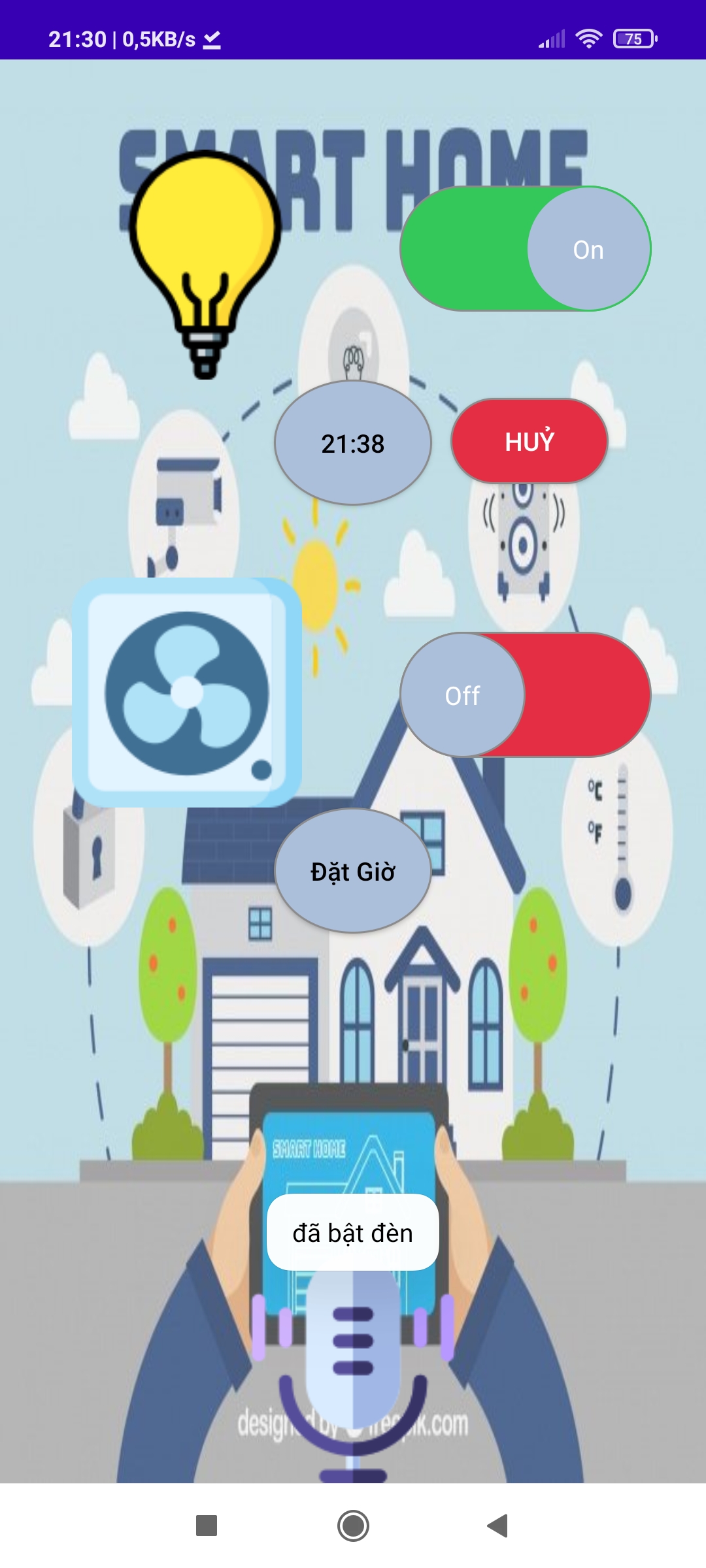
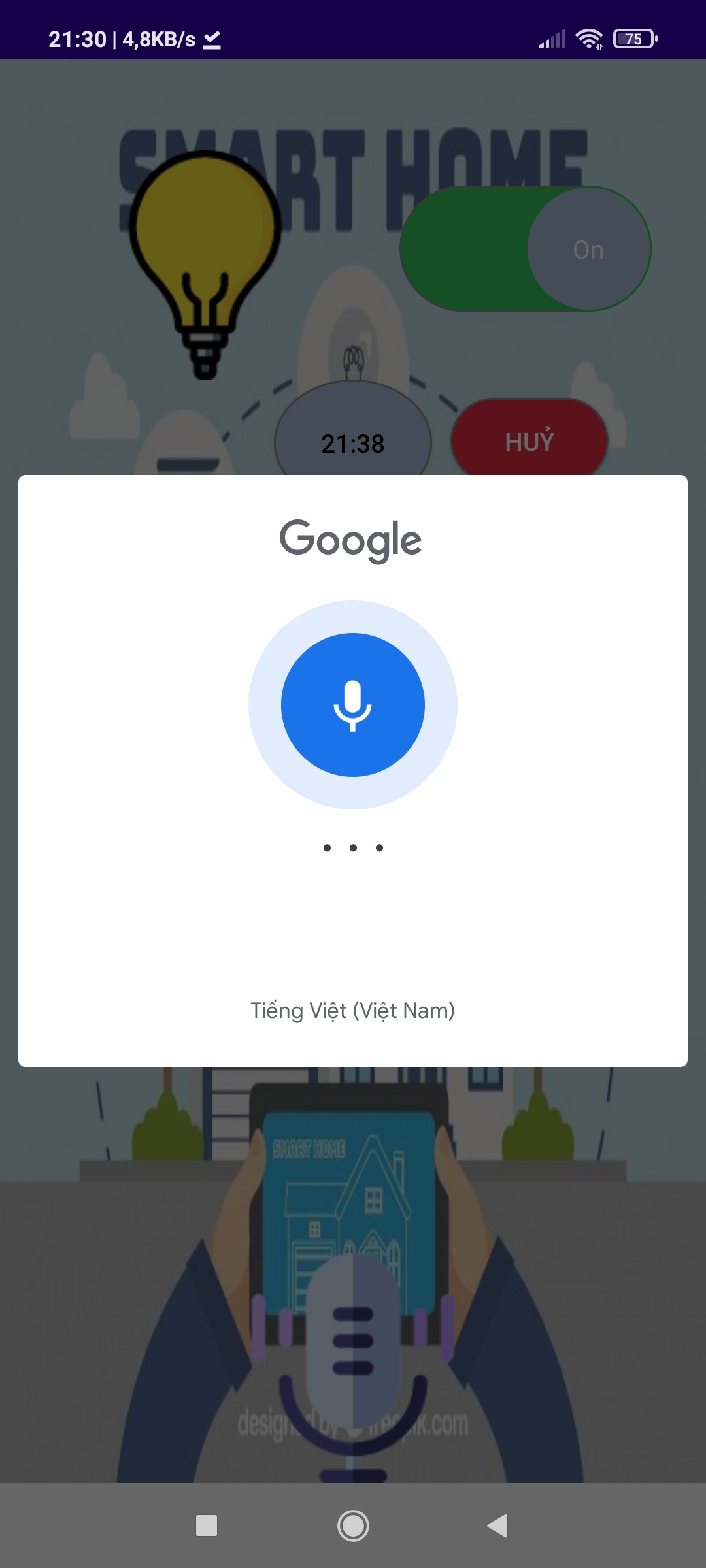
- Hệ thống điều khiển bằng giọng nói với google Assistant

- Hệ thống điều khiển thông qua các nút button được thiết lập trong hệ thống lập trình android

- Hệ thống có thể điều khiển các thiết bị điện tử từ xa thông qua wifi bằng cơ sở dữ liệu Firebase

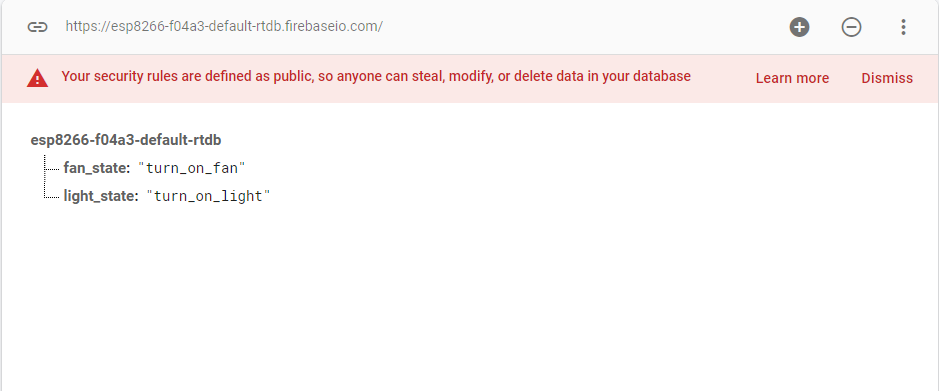
- Đã đồng bộ được tất cả các thiết bị bằng công tắt cảm ứng thông qua button thiết lập giúp chúng qua có thể quản lí và giám sát một cách tiện lợi thông qua giao diện app trên điện thoại di động.

**Giao diện của ứng dụng:**

** **

Hình 5. 1 Giao diện màn hình ứng dụng

**Giao diện của firebase database**



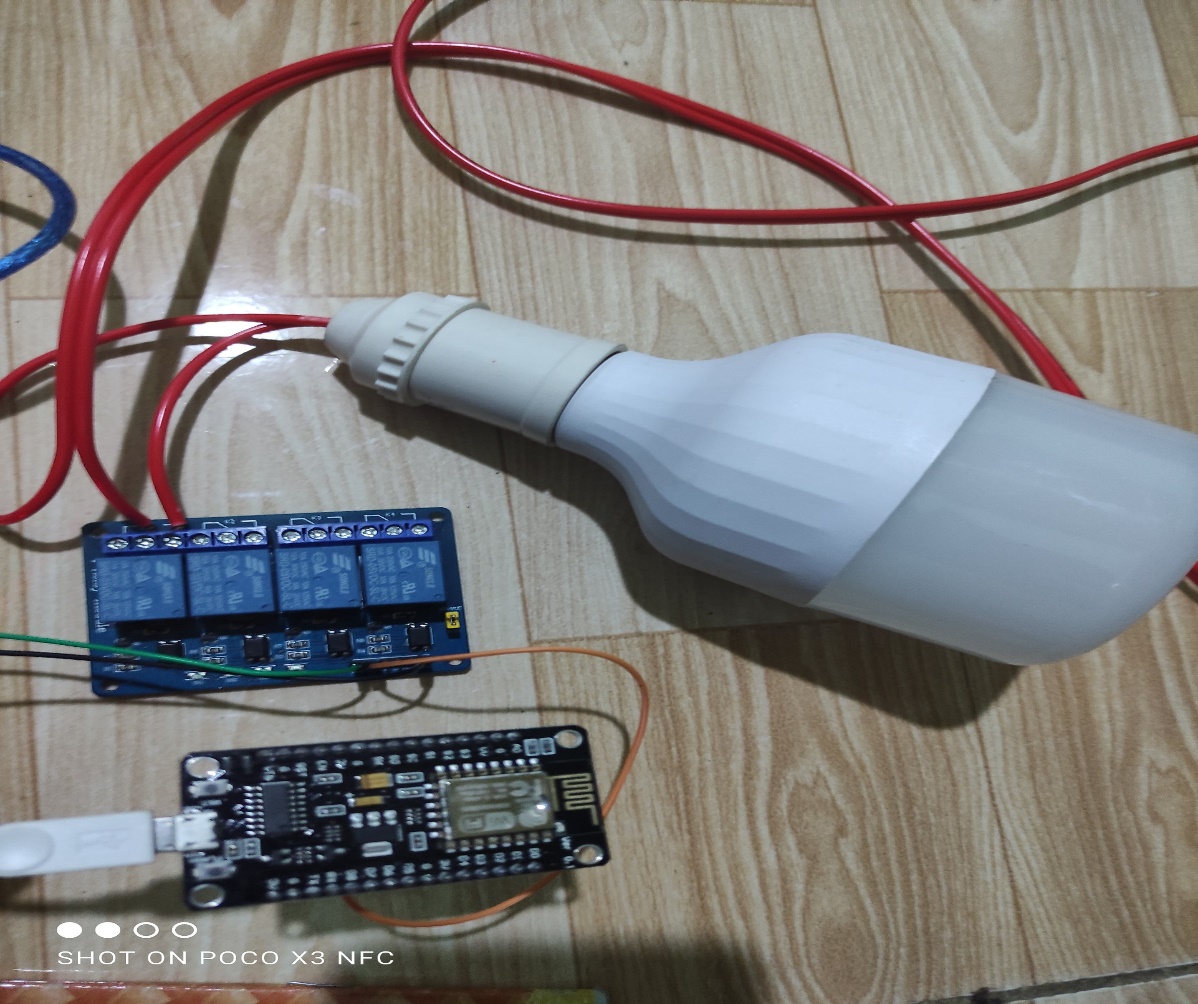
**Mạch điều khiển:**

Trạng thái của relay lúc bật



Hình 5. 2 relay lúc bật

Trạng thái của relay lúc tắt



Hình 5. 3 relay lúc tắt

## **5.2 Hướng phát triển:**

- Khắc phục những hạn chế, tồn tại của hệ thống

- Đưa ra giải pháp, phát triển mô hình lớn rộng hơn, đáp ứng được nhu cầu người sử dụng trong nước

- Phát triển thêm nhiều thiết bị mang tính chất tự động hóa cho thiết bị để dễ dàng quảng lí và thực thi cái lệnh thông qua điện thoại hay Wifi.

# **TÀI LIỆU THAM KHẢO**

[1]. Giáo trình *Arduino cho người mới bắt đầu*, Cộng động Việt Nam

[2]. Giáo trình *Internet Of things with* ESP8266

[3]. Giáo trình *Lập trình di động* Android Kotlin

[4]. Trang web: *Arduino.vn*