**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

**BỘ MÔN VIỄN THÔNG**

---------------o0o---------------

****

**ĐỒ ÁN MÔN HỌC**

**“GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ**

**TRONG NHÀ QUA APP BLYNK”.**

**GVHD: Ths. ĐINH QUỐC HÙNG**

**SVTH: VÕ THÀNH ĐẠT 1710064**

**LÊ TIẾN ĐẠT 1710948**

**TP. HỒ CHÍ MINH, THÁNG 7 NĂM 2020**

***LỜI CẢM ƠN***

Để thực hiện đồ án này, chúng em xin gửi lời cảm ơn sâu sắc đến thầy Đinh Quốc Hùng đã tận tình giúp đỡ tạo điều kiện, góp ý và chia sẻ nhiều kinh nghiệm quý báu cho chúng em thực hiện tốt đề tài.

Chúng em xin gửi lời cảm ơn chân thành các thầy cô bộ môn Viễn Thông nói riêng và trường Đại học Bách Khoa TP.HCM nói chung và Điện - Điện tử nói chung đã tận tình dạy dỗ, chỉ bảo, cung cấp những kiến thức nền chuyên môn làm cơ sở cũng như tạo điều kiện tốt nhất cho chúng em hoàn thành đề tài. Do thời gian và kiến thức còn nhiều hạn chế, đề tài của em sẽ không tránh khỏi những sai sót, em mong thầy cô góp ý, chỉnh sửa để có thể hoàn thiện hơn.

Cảm ơn gia đình, bạn bè đã động viên và luôn luôn bên cạnh trong những lúc khó khăn nhất.

Chúng em cũng gửi lời cảm ơn các bạn sinh viên khoa Điện-Điện tử đã chia sẻ, trao đổi kiến thức cũng như những kinh nghiệm quý báu trong thời gian thực hiện đề tài.

Xin chân thành cảm ơn!

*Tp. Hồ Chí Minh, tháng 07 năm 2020.*

**Nhóm sinh viên thực hiện**

**TÓM TẮT ĐỒ ÁN**

Ngày nay cùng với sự phát triển mạnh mẽ 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 một 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.

Với ý tưởng trên nhóm đưa ra đề tài “GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ TRONG NHÀ QUA APP BLYNK”. Hệ thống có khả năng hỗ trợ giúp người dùng giám sát và điều khiển các thiết bị trong nhà từ xa thông qua điện thoại kết nối wifi. 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.

Với đề tài này, nhóm hy vọng sẽ nghiên cứu cho các nhóm sau có thể mở rộng, phát triển hơn nữa.

MỤC LỤC

[1. GIỚI THIỆU 6](#_Toc45806529)

[1.1 Đặt vấn đề 6](#_Toc45806530)

[1.2 Mục tiêu 6](#_Toc45806531)

[1.3 Giới hạn 7](#_Toc45806532)

[1.4 Nội dung nghiên cứu 7](#_Toc45806533)

[1.5 Phân chia công việc trong nhóm 8](#_Toc45806534)

[2. LÝ THUYẾT 9](#_Toc45806535)

[**2.1** **Nguyên lý truyền nhận dữ liệu:** 9](#_Toc45806536)

[**2.2** **Giao tiếp với người dùng qua App Blynk:** 12](#_Toc45806537)

[**2.3.** **Giới thiệu module Arduino** 14](#_Toc45806538)

[3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG 16](#_Toc45806539)

[**3.1.** **Lựa chọn phần cứng** 16](#_Toc45806540)

[**3.1.1** **Khối xử lý trung tâm** 16](#_Toc45806541)

[**3.1.2** **Cảm biến khí gas MQ2** 21](#_Toc45806542)

[**3.1.3.** **Relay** 24](#_Toc45806543)

[**3.1.4.** **Nguồn** 25](#_Toc45806544)

[**3.1.5.** **Module Buzzer:** 26](#_Toc45806545)

[**3.1.6** **Cảm biến nhiệt độ, độ ẩm DHT11** 27](#_Toc45806546)

[**3.2** **THI CÔNG PHẦN CỨNG** 28](#_Toc45806547)

[**3.2.1** **Yêu cầu hệ thống cụ thể** 28](#_Toc45806548)

[**3.2.2** **Sơ đồ khối hệ thống** 29](#_Toc45806549)

[**4.** **THIẾT KẾ VÀ THI CÔNG PHẦN MỀM** 32](#_Toc45806550)

[**4.1.** **Lưu đồ giải thuật** 32](#_Toc45806551)

[**4.2.** **Các bước tạo project app Blynk** 34](#_Toc45806552)

[5. KẾT QUẢ, KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 41](#_Toc45806553)

[**5.1.** **Đo đạc và thực nghiệm hệ thống:** 41](#_Toc45806554)

[**5.1.1. Mô hình thực tế** 41](#_Toc45806555)

[**5.1.2** **. Kết quả giám sát một số thiết bị** 41](#_Toc45806556)

[**5.2.** **Đánh giá kết quả làm việc nhóm.** 43](#_Toc45806557)

[**5.3** **Kết luận** 43](#_Toc45806558)

[**5.4** **Hướng phát triển** 44](#_Toc45806559)

[6. TÀI LIỆU THAM KHẢO 45](#_Toc45806560)

[7. PHỤ LỤC 47](#_Toc45806561)

**DANH SÁCH HÌNH MINH HỌA**

Hình 2.1a: Giao tiếp UART 9

Hình 2.1b: Giao tiếp dữ liệu song song UART 10

Hình 2.2a: Biểu tượng của App Blynk 12

Hình 2.2b: Cách thức Blynk và phần cứng hoạt động 13

Hình 2.3: Hình ảnh Arduino Uno R3 15

Hình 3.1.1a: Sơ đồ chân NodeMCU Esp8266 17

Hình 3.1.1b: Hình ảnh Arduino Nano 18

Hình 3.1.2a: Hình ảnh Module cảm biến khí Gas MQ-2 21

Hình 3.1.2b*.* Cấu tạo module cảm biến khí gas MQ2 23

Hình 3.1.3: Hình ảnh Relay 5V 1 kênh 24

Hình 3.1.4: Hình ảnh pin sạc Xiaomi 25

Hình 3.1.5:Hình ảnh Module Buzzer 26

Hình 3.1.6: Sơ đồ nối chân DHT11 28

Hình 3.2.2: Sơ đồ khối hệ thống 29

Hình 3.2.3: Sơ đồ chi tiết 31

Hình 4.1a: Lưu đồ giải thuật phía Server 32

Hình 4.1b: Lưu đồ giải thuật phía Client 33

Hình 5.1.1. Mô hình ngôi nhà thực tế 41

Hình 5.1.2a. Cảnh báo trên điện thoại 42

Hình 5.1.2b Biểu đồ nhiệt độ, độ ẩm theo thời gian 42

# GIỚI THIỆU

## Đặt vấn đề

Cuộc sống ngày càng phát triển nhanh chóng và hiện đại hơn, những công nghệ mới ngày cảng được phát minh và phát triển để đưa vào phục vụ cuộc sống hàng ngày của con người. Những ứng dụng của IOT được sử dụng ngày càng rộng rãi và trong rất nhiều lĩnh vực từ đời sống đến sản xuất. Với một đất nước đang phát triển như nước ta, việc ứng dụng IOT vào ngôi nhà của mỗi người để tăng tính tiện ích và đáp ứng nhu cầu cơ sở vật chất ngày càng tăng là một nhu cầu thiết yếu và đầy tiềm năng.

Điều khiển và giám sát thông minh, là tích hợp các hệ thống như điều khiển và giám sát đèn, quạt, máy lạnh,… và các thiết bị cá nhân khác thành một hệ thống thống nhất. Mỗi chức năng của điều khiển và giám sát thông minh đều có khả năng tự vận hành hoặc dưới sự điều khiển của người dùng. Người dùng có thể truy cập từ xa để kiểm tra, điều khiển hệ thống hoạt động theo yêu cầu của mình.

Điều khiển và giám sát thông minh là một trong những đề tài công nghệ ứng dụng được ứng dụng trong rất nhiều dự án. Không chỉ hạn chế với những tính năng nêu trên, ngày càng có nhiều nghiên cứu đề xuất phát triển hệ thống điều khiển và giám sát để theo kịp sự phát triển của công nghệ, tối ưu hóa hiệu năng sử dụng cũng như giá cả hợp lý.

Cùng với sự ngày càng phổ bến của mạng không dây, việc kết nối thông tin ngày càng trở nên tiện lợi và nhanh chóng với giá thành ngày càng thấp. Do đó, nhóm quyết định chọn đề tài: “GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ TRONG NHÀ QUA APP BLYNK”.

## Mục tiêu

Thiết kế một hệ thống có khả năng điều khiển thiết bị và cảnh báo cho người dùng khi có nguy cơ cháy nổ xảy ra. Hệ thống ứng dụng công nghệ IoT, giúp cho người dùng ở khu vực có kết nối internet có thể dễ dàng điều khiển, giám sát hệ thống của mình.

## Giới hạn

* Hệ thống chỉ dừng lại ở công việc thu thập, gửi lên internet thông số khí gas, khói khi vượt quá mức cho phép.
* Hệ thống không quá to với quy mô đồ án
* Sai số chấp nhận được
* Do giới hạn về lập trình nên hệ thống vẫn còn một độ trễ nhất định.

## Nội dung nghiên cứu

* Nội dung 1: Tìm hiểu và lựa chọn các giải pháp thiết kế.
* Nội dung 2: Thu nhập tài liệu về các linh kiện, phần mềm thiết kế sử dụng.
* Nội dung 3: Thiết kế phần cứng của hệ thống.
* Nội dung 4: Lập trình cho hệ thống, phần mềm giúp giao tiếp với người dùng qua internet.
* Nội dung 5: Thử nghiệm thực tế hệ thống, đánh giá kết quả thực hiện.
* Nội dung 6: Viết báo cáo và bảo vệ đồ án.

## Phân chia công việc trong nhóm

|  |  |  |
| --- | --- | --- |
| **BẢNG PHÂN CÔNG CÔNG VIỆC** | | |
| **Thành viên** | | |
| Lê Tiến Đạt |
| Võ Thành Đạt |  | |
| **Công việc** | | **Thành viên phụ trách** |
| * Tìm hiểu, nghiên cứu thực tế | | Cả nhóm |
| * Xây dựng sơ đồ khối | | Tiến Đạt |
| * Tìm hiểu NodeMCU esp8266 | | Cả nhóm |
| * Tìm hiểu các cảm biến | | Thành Đạt |
| * Xây dựng giải thuật | | Tiến Đạt |
| * Lập trình cho hệ thống | | Tiến Đạt |
| * Xây dựng app blynk | | Thành Đạt |
| * Thi công phần cứng | | Thành Đạt |
| * Xây dựng mô hình | | Cả nhóm |
| * Đánh giá kết quả, viết báo cáo | | Cả nhóm |
| **Họp nhóm** | | Thứ 6 tại phòng Lab 110B3 |
| **Quy định** | | * Hoàn thành công việc được giao đúng thời gian * Tôn trọng ý kiến các thành viên trong nhóm |

Bảng 1: Bảng phân chia công việc nhóm.

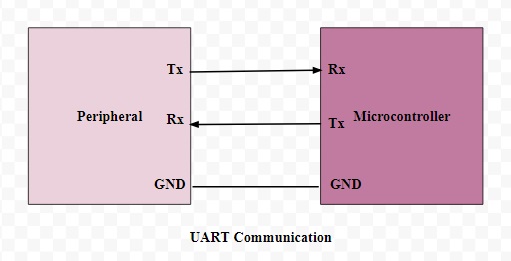
# LÝ THUYẾT

* 1. **Nguyên lý truyền nhận dữ liệu:**

**Chuẩn giao tiếp UART TTL:**

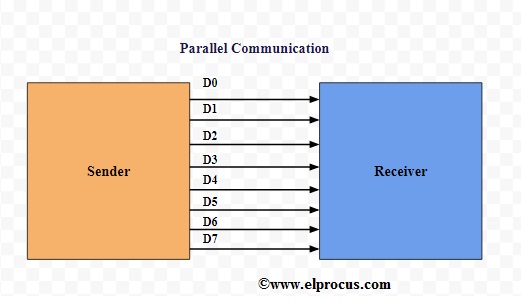
* **Khái niệm:**

Có tên đầy đủ UART là “Universal Asynchronous Receiver/Transmitter”, nó là một vi mạch sẵn có trong một vi điều khiển nhưng không giống như một giao thức truyền thông (I2C & SPI). Chức năng chính của UART là truyền dữ liệu nối tiếp. Trong UART, giao tiếp giữa hai thiết bị có thể được thực hiện theo hai cách là giao tiếp dữ liệu nối tiếp và giao tiếp dữ liệu song song.



*Hình 2.1a*

* **Truyền thông nối tiếp và song song:**
* Trong giao tiếp dữ liệu nối tiếp, dữ liệu có thể được truyền qua một cáp hoặc một đường dây ở dạng bit-bit và nó chỉ cần hai cáp. Truyền thông dữ liệu nối tiếp không đắt khi chúng ta so sánh với giao tiếp song song. Nó đòi hỏi rất ít mạch cũng như dây. Vì vậy, giao tiếp này rất hữu ích trong các mạch ghép so với giao tiếp song song.
* Trong giao tiếp dữ liệu song song, dữ liệu có thể được truyền qua nhiều cáp cùng một lúc. Truyền dữ liệu song song tốn kém nhưng tốc độ nhanh, vì nó đòi hỏi phần cứng và cáp bổ sung. Các ví dụ tốt nhất cho giao tiếp này là máy in cũ, PCI, RAM, ...



*Hình 2.1b*

* **Truyền thông UART:**
* Trong giao tiếp này, có hai loại UART có sẵn là truyền UART và nhận UART và giao tiếp giữa hai loại này có thể được thực hiện trực tiếp với nhau. Đối với điều này, chỉ cần hai cáp để giao tiếp giữa hai UART. Luồng dữ liệu sẽ từ cả hai chân truyền (Tx) và nhận (Rx) của UARTs. Trong UART, việc truyền dữ liệu từ Tx UART sang Rx UART có thể được thực hiện không đồng bộ (không có tín hiệu CLK để đồng bộ hóa các bit).
* Việc truyền dữ liệu của UART có thể được thực hiện bằng cách sử dụng bus dữ liệu ở dạng song song bởi các thiết bị khác như vi điều khiển, bộ nhớ, CPU,... Sau khi nhận được dữ liệu song song từ bus, nó tạo thành gói dữ liệu bằng cách thêm ba bit như bắt đầu, dừng lại và trung bình. Nó đọc từng bit gói dữ liệu và chuyển đổi dữ liệu nhận được thành dạng song song để loại bỏ ba bit của gói dữ liệu. Tóm lại, gói dữ liệu nhận được bởi UART chuyển song song về phía bus dữ liệu ở đầu nhận.
* **Start-bit:**

Start-bit còn được gọi là bit đồng bộ hóa được đặt trước dữ liệu thực tế. Nói chung, một đường truyền dữ liệu không hoạt động được điều khiển ở mức điện áp cao. Để bắt đầu truyền dữ liệu, truyền UART kéo đường dữ liệu từ mức điện áp cao (1) xuống mức điện áp thấp (0). UART thu được thông báo sự chuyển đổi này từ mức cao sang mức thấp qua đường dữ liệu cũng như bắt đầu hiểu dữ liệu thực. Nói chung, chỉ có một start-bit.

* **Bit dừng:**

Bit dừng được đặt ở phần cuối của gói dữ liệu. Thông thường, bit này dài 2 bit nhưng thường chỉ sử dụng 1 bit. Để dừng sóng, UART giữ đường dữ liệu ở mức điện áp cao.

* **Bit chẵn lẻ:**

Bit chẵn lẻ cho phép người nhận đảm bảo liệu dữ liệu được thu thập có đúng hay không. Đây là một hệ thống kiểm tra lỗi cấp thấp & bit chẵn lẻ có sẵn trong hai phạm vi như chẵn lẻ – chẵn lẻ cũng như chẵn lẻ – lẻ. Trên thực tế, bit này không được sử dụng rộng rãi nên không bắt buộc.

* **Dữ liệu bit hoặc khung dữ liệu:**

Các bit dữ liệu bao gồm dữ liệu thực được truyền từ người gửi đến người nhận. Độ dài khung dữ liệu có thể nằm trong khoảng 5-8. Nếu bit chẵn lẻ không được sử dụng thì chiều dài khung dữ liệu có thể dài 9 bit. Nói chung, LSB của dữ liệu được truyền trước tiên sau đó nó rất hữu ích cho việc truyền.

* **Tốc độ baud:**

Máy phát tốc độ baud giữa máy phát và máy thu tạo ra tốc độ dao động từ 110 bps đến 230400 bps. Thông thường, tốc độ truyền của vi điều khiển là 9600 đến 115200.

* **Ưu điểm và nhược điểm của UART:**
* Nó chỉ cần hai dây để truyền dữ liệu.
* Tín hiệu CLK là không cần thiết.
* Nó bao gồm một bit chẵn lẻ để cho phép kiểm tra lỗi.
* Sắp xếp gói dữ liệu có thể được sửa đổi vì cả hai mặt được sắp xếp.
* Kích thước khung dữ liệu tối đa là 9 bit.
* Nó không chứa một số hệ thống phụ.
* Tốc độ truyền của UART phải ở mức 10% của nhau.
  1. **Giao tiếp với người dùng qua App Blynk:**

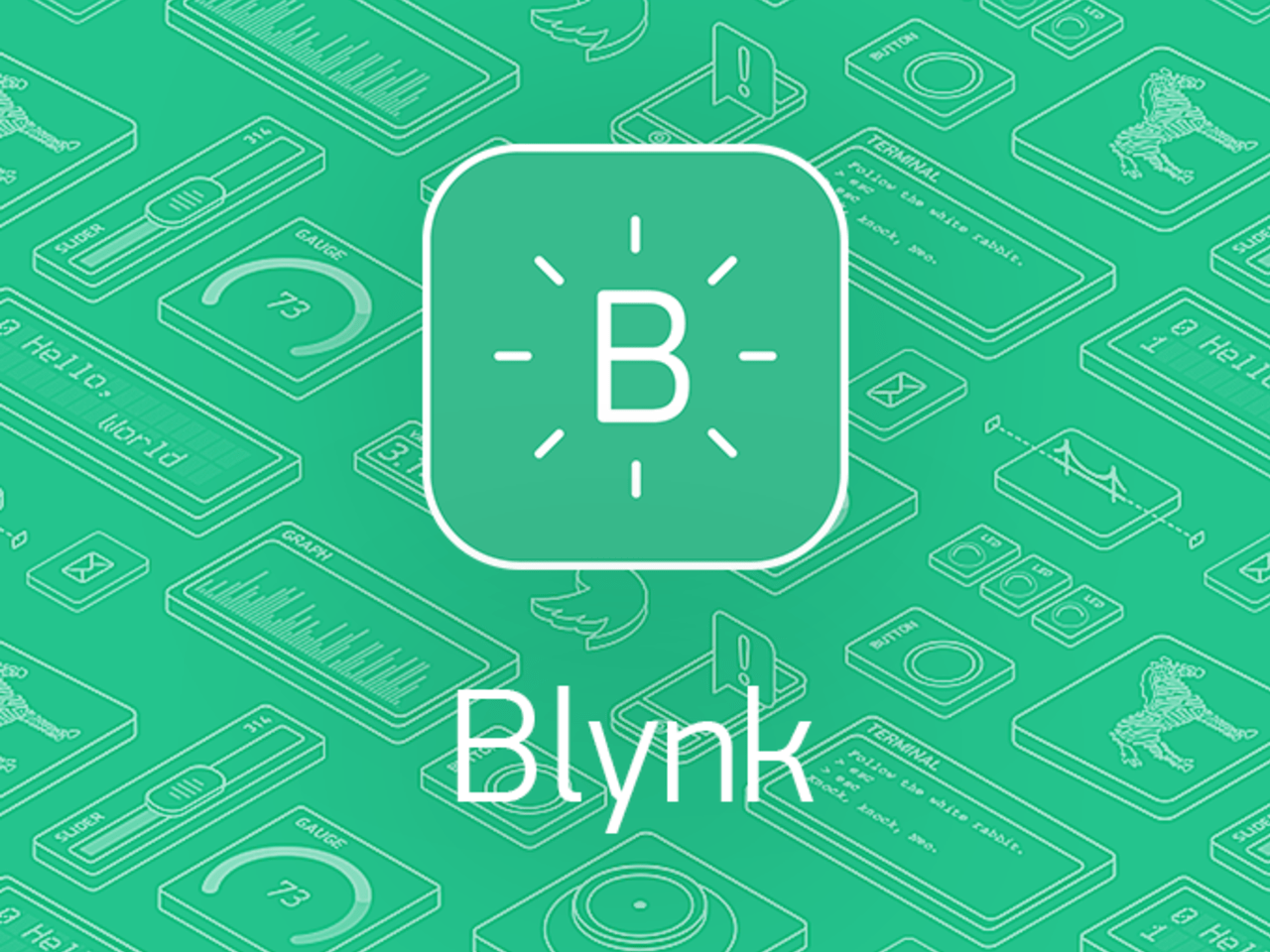
***Với những yêu cầu kỹ thuật đề ra:***

* Giúp người dùng giao tiếp với hệ thống từ xa, cụ thể là hiển thị thông số cảm biến đo được, trạng thái relay, gửi thông tin điều khiển relay từ người dùng đến khối xử lý.
* Kết nối qua internet.
* Làm việc theo thời gian thực, ổn định, đáng tin cậy.

***Nhóm quyết định sử dụng Ứng dụng Blynk.***

***Blynk là gì?***

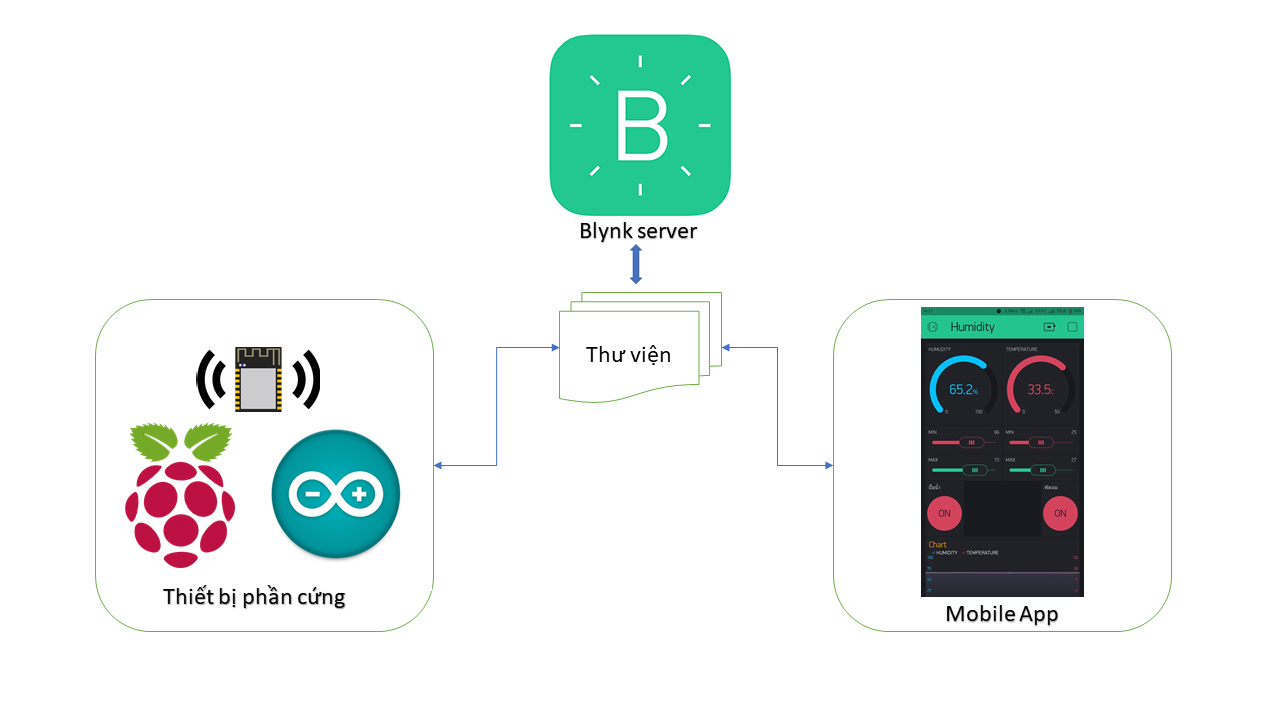
Blynk là một ứng dụng chạy trên nền tảng iOS và Android để điều khiển và giám sát thiết bị thông qua internet. Blynk không bị ràng buộc với những phần cứng cụ thể nào cả, thay vào đó, nó hỗ trợ phần cứng cho bạn lựa chọn như Arduino, Raspberry Pi, ESP8266 và nhiều module phần cứng phổ biến khác.

****

*Hình 2.2a*

***Hoạt động***

* Có ba thành phần chính trong nền tảng:
* Blynk App - cho phép tạo giao diện cho sản phẩm của người dùng bằng cách kéo thả các widget khác nhau mà nhà cung cấp đã thiết kế sẵn.
* Blynk Server - chịu trách nhiệm xử lý dữ liệu trung tâm giữa điện thoại, máy tính bảng và phần cứng. Chúng ta có thể sử dụng Blynk Cloud của Blynk cung cấp hoặc tự tạo máy chủ Blynk riêng của mình. Vì đây là mã nguồn mở, nên chúng ta có thể dễ dàng intergrate vào các thiết bị và thậm chí có thể sử dụng Raspberry Pi làm server của mình.
* Library Blynk – support cho hầu hết tất cả các nền tảng phần cứng phổ biến - cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và đi.

****

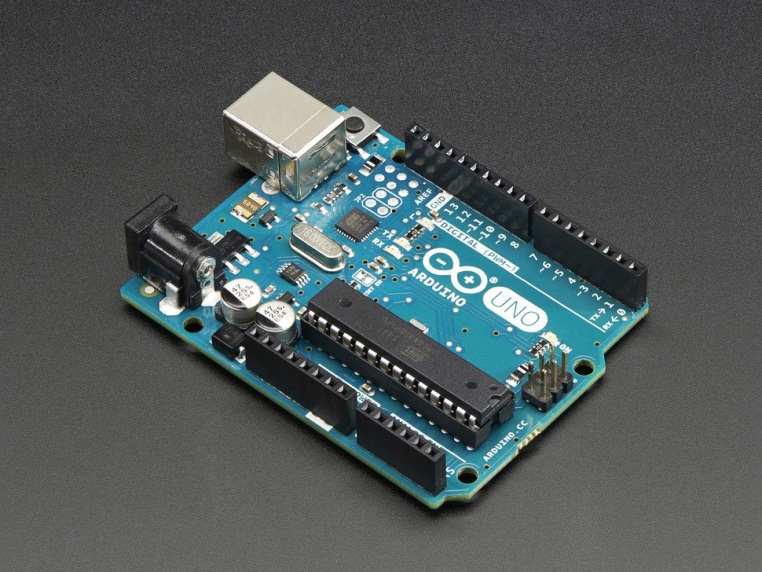
*Hình 2.2b*

***Tính năng, đặc điểm***

* Cung cấp API & giao diện người dùng tương tự cho tất cả các thiết bị và phần cứng được hỗ trợ
* Kết nối với server bằng cách sử dụng:
* Wifi
* Bluetooth và BLE
* Ethernet
* USB (Serial)
* GSM
* Các tiện ích trên giao diện được nhà cung cấp dễ sử dụng
* Thao tác kéo thả trực tiếp giao diện mà không cần viết mã
* Dễ dàng tích hợp và thêm chức năng mới bằng cách sử dụng các cổng kết nối ảo được tích hợp trên blynk app
* Theo dõi lịch sử dữ liệu
* Thông tin liên lạc từ thiết bị đến thiết bị bằng Widget
* Gửi email, tweet, thông báo realtime, v.v.
* Được cập nhật các tính năng liên tục.

***Những lý do sử dụng Blynk:***

* Dễ sử dụng: việc cài đặt ứng dụng và đăng ký tài khoản trên điện thoại rất đơn giản cho cả IOS và Android.
* Chức năng phong phú: Blynk hỗ trợ rất nhiều chức năng với giao diện đẹp và thân thiện, dễ sử dụng.
* Không phải lập trình ứng dụng, không cần kiến thức chuyên sâu về lập trình.
* Điều khiển, giám sát thiết bị ở bất kì đâu thông qua internet với khả năng đồng bộ hóa trạng thái và thiết bị.
  1. **Giới thiệu module Arduino**
* Arduino đã và đang được sử dụng rất rộng rãi trên thế giới, và ngày càng chứng tỏ được sức mạnh của chúng thông qua vô số ứng dụng độc đáo của người dùng trong cộng đồng nguồn mở (open-source). Tuy nhiên tại Việt Nam Arduino vẫn còn chưa được biết đến nhiều.
* Arduino cơ bản là một nền tảng tạo mẫu mở về điện tử (open-source electronics prototyping platform) được tạo thành từ phần cứng lẫn phần mềm. Về mặt kĩ thuật có thể coi Arduino là 1 bộ điều khiển logic có thể lập trình được. Đơn giản hơn, Arduino là một thiết bị có thể tương tác với ngoại cảnh thông qua các cảm biến và hành vi được lập trình sẵn. Với thiết bị này, việc lắp ráp và điều khiển các thiết bị điện tử sẽ dễ dàng hơn bao giờ hết. Một điều không hề dễ dàng cho những ai đam mê công nghệ và điều khiển học nhưng là người ngoại đạo và không có nhiều thời gian để tìm hiểu sâu hơn về về kĩ thuật lập trình và cơ điện tử.



Hình 2.3. Hình ảnh Arduino Uno R3

* Hiện tại có rất nhiều loại vi điều khiển và đa số được lập trình bằng ngôn ngữ C/C++ hoặc Assembly nên rất khó khăn cho những người có ít kiến thức sâu về điện tử và lập trình. Nó là trở ngại cho mọi người muốn tạo riêng cho mình một món đồ mang tính công nghệ. Do vậy đó là lí do Arduino được phát triển nhằm đơn giản hóa việc thiết kế, lắp ráp linh kiện điện tử cũng như lập trình trên vi xử lí và mọi người có thể tiếp cận dễ dàng hơn với thiết bị điện tử mà không cần nhiều về kiến thức điện tử và thời gian. Sau đây là nhưng thế mạnh của Arduino so với các nền tảng vi điều khiển khác:
* Chạy trên đa nền tảng: Việc lập trình Arduino có thể thể thực hiện trên các hệ điều hành khác nhau như Windows, Mac Os, Linux trên Desktop, Android trên di động.
* Ngôn ngữ lập trình đơn giản dễ hiểu.
* Nền tảng mở: Arduino được phát triển dựa trên nguồn mở nên phần mềm chạy trên Arduino được chia sẻ dễ dàng và tích hợp vào các nền tảng khác nhau.
* Mở rộng phần cứng: Arduino được thiết kế và sử dụng theo dạng module nên việc mở rộng phần cứng cũng dễ dàng hơn.
* Đơn giản và nhanh: Rất dễ dàng lắp ráp, lập trình và sử dụng thiết bị.
* Dễ dàng chia sẻ: Mọi người dễ dàng chia sẻ mã nguồn với nhau mà không lo lắng về ngôn ngữ hay hệ điều hành mình đang sử dụng.
* Arduino được chọn làm bộ não xử lý của rất nhiều thiết bị từ đơn giản đến phức tạp. Trong số đó có một vài ứng dụng thực sự chứng tỏ khả năng vượt trội của Arduino do chúng có khả năng thực hiện nhiều nhiệm vụ rất phức tạp. Sau đây là danh sách một số ứng dụng nổi bật của Arduino như trong công nghệ in 3D, robot dò đường theo hướng có nguồn nhiệt, tạo một thiết bị nhấp nháy theo âm thanh và đèn laser hay là một thiết bị báo cho khách hàng biết khi nào bánh mì ra lò.

# THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

* 1. **Lựa chọn phần cứng**
     1. **Khối xử lý trung tâm**

**Yêu cầu kỹ thuật:**

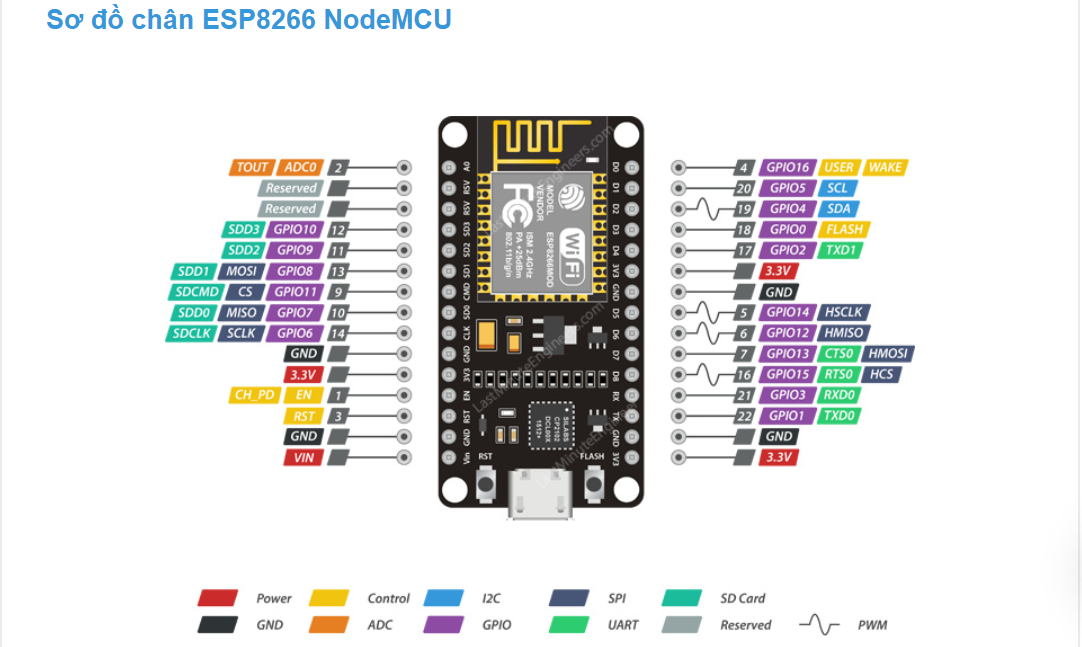
* Đọc thông số từ các cảm biến.
* Nhận tín hiệu điều khiển tại chỗ từ nút nhấn.
* Gửi thông tin và nhận lệnh điều khiển qua internet.
* Hiển thị thông số cảm biến tại chỗ qua Blynk cho người dùng ở xa.
* Chạy được chương trình đã được lập trình trước.

***Đề ra phương án:*** *Thiết kế phần cứng xoay quanh Arduino Nano và module esp8266.*

1. **NodeMCU ESP8266**

**Yêu cầu kỹ thuật:**

* Nhận tín hiệu điều khiển tại chỗ từ nút nhấn từ các khối bên dưới hoặc từ xa qua App Blynk.
* Nhận thông tin và gửi lệnh điều khiển qua chuẩn UART.
* Chạy được chương trình đã được lập trình trước.
* **Giới thiệu NODEMCU esp8266**
* Với rất nhiều những bạn sinh viên hiện nay, việc bỏ ra một số tiền kha khá để có một thiết bị để điều khiển các thiết bị điện tử là một điều đáng để trăn trở. Điển hình cho những thiết bị như vậy thì chúng ta có: Raspberry Pi, Alexa,..v..v.. Để có một con Raspberry Pi thì chúng ta có thể mất đến 50$, thậm chí 200-300$ cho một con Alexa.Vì vậy, em xin giới thiệu một thiết bị sẽ đáp ứng được những yêu cầu trên và giá thành phù hợp với các bạn sinh viên, đó là ESP8266 nodeMCU với giá thành trên thị trường có lúc chỉ một trăm nghìn.
* NodeMCU được phát triển dựa trên Chip WiFi ESP8266EX bên trong Module ESP-12E dễ dàng kết nối WiFi với một vài thao tác. Board còn tích hợp IC CP2102, giúp dễ dàng giao tiếp với máy tính thông qua Micro USB để thao tác với board, và có sẳn nút nhấn, led để tiện qua quá trình học, nghiên cứu.
* **Thông số kỹ thuật**:
* Chip: ESP8266EX
* WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
* Điện áp hoạt động: 3.3V
* Số chân I/O: 11
* Số chân Analog Input: 1
* Bộ nhớ Flash: 4MB
* Giao tiếp: Cable Micro USB
* Hỗ trợ bảo mật: WPA/WPA2
* Tích hợp giao thức TCP/IP
* Lập trình trên các ngôn ngữ: C/C++, Micropython, NodeMCU



*Hình 3.1.1a*

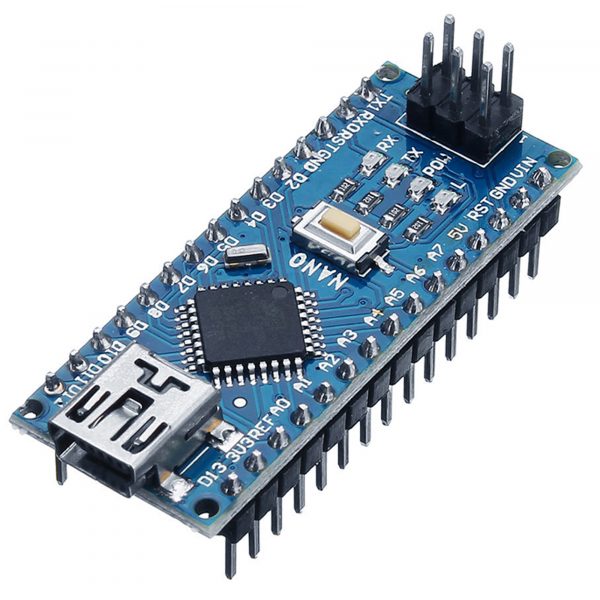
* **Kết luận**
* Ưu điểm: NodeMCU có module nhỏ gọn, sử dụng điện áp nhỏ từ 3.3 – 5V và hỗ trợ kết nối với nhiều thiết bị khác (I2c, SPI, Uart). Ngoài ra cài đặt thư viện nhanh, gọn, nhẹ trên Arduino IDE và các thư viện cũ và mới được Update liên tục. Module ESP8266 được sử dụng phổ biến trong các hệ thống IOT nhà thông minh tương tự.
* Nhược điểm: Thiếu RAM nên gây khó khăn cho quá trình mở rộng hệ thống.

***Nhóm quyết định chọn NodeMCU ESP8266 do có kinh nghiệm, nguồn tài liệu với nó và qua tham khảo giá cả ổn nhất.***

1. **Arduino Nano**

**Yêu cầu kỹ thuật**

* Đọc thông số từ các cảm biến
* Nhận tín hiệu điều khiển tại chỗ từ nút nhấn
* Nhận thông tin và gửi lệnh điều khiển qua chuẩn UART.
* Chạy được chương trình đã được lập trình trước, tự động tự động bật, tắt relay từ các thông số môi trường.
* ***Cấu trúc chung***

******

*Hình 3.1.1b*

* Arduino Nano là board mạch sử dụng vi điều khiển ATmega328P hoặc ATmega168 tích hợp các chân I/O đơn giản nhỏ gọn dựa trên mã nguồn mở được phát triển bởi Arduino.cc, có lợi thế lớn về kích thước so với phiên bản Arduino Uno và Arduino Mega. Arduino Nano có thể hoạt động độc lập và tương tác hiệu quả với các thiết bị điện tử, cũng có thể giúp những người mới tìm hiểu về Arduino có thể kết nối với PC, phối hợp với Flash, Xử lý, Max / Msp, PD,  và các phần mềm khác một cách dễ dàng. Điều này giúp Arduino Nano là sự lựa chọn ưa thích khi muốn thực hiện một projects mà yếu cầu kết nối với các thiết bị ngoại vi ít và đơn giản. Các chức năng rất giống giống với phiên bản Arduino Uno nhưng kích thước nhỏ gọn hơn. Mỗi chân Digital và Analog có thể thực hiện với nhiều chức năng khác nhau nhưng chức năng chính vẫn là được mặc định cấu hình làm đầu vào (Input) hoặc đầu ra (Output). Khi giao tiếp với cảm biến các chân Digital / Analog đóng vai trò chân Input, Và khi sử dụng để điều khiển động cơ, tạo xung, kích dẫn relay, thiết bị chuyển mạch thì các chân Digital / Anologs đống vai trò Output. Arduino Nano tích hợp [*Thạch anh dao động*](https://www.vietnic.vn/thach-anh-dao-dong) với tần số 16 MHz. Nếu sử dụng Arduino Nano để thực hiện một projects liên quan đến đồng hồ số thì độ chính xác trong dao động tần số rất cao.
* **Khối nguồn**

Arduino Nano hoạt động với điện áp 5V. Tuy nhiên nguồn cấp điện áp đầu vào có thể thay đổi từ 7 đến 12V.Có một hạn chế khi sử dụng Arduino Nano đó là không đi kèm giắc nguồn DC, có nghĩa là không thể cung cấp nguồn điện bên ngoài thông qua các chân Arduino Nano như các phiên bản Arduino Uno và Arduino Mega. Phải cắm trực tiếp thông qua cổng USB Mini. Board mạch Arduino Nano không sử dụng cổng USB chuẩn thường dùng để kết nối với máy tính, thay vào đó Arduino Nano chỉ hỗ trở cổng USB Mini để kết nối

* Chân Vin:  Đây là chân cung cấp điện áp đầu vào cho mạch Arduino nano khi sử dụng nguồn ngoài từ 7VDC đến 12 VDC.
* Chân 5V: Là mức điện áp cung cấp quy định của Arduino được sử dụng để cấp nguồn nuôi cho vi điều khiển và các bộ phận linh kiện khác trên board Arduino.
* Chân 3,3V :  Đây là một mức điện áp tối thiểu được tạo ra bởi bộ điều chỉnh điện áp trên board ( sử dụng Lm1117 - 3.3V)
* Chân GND : Chân mass cho Arduino, có nhiều chân GND trên board Arduino cho mục đích dễ dàng kết nối với thiết bị ngoại vi sử dụng dây testboard
* ***Bộ nhớ***

Bộ nhớ flash là 16KB hoặc 32KB, phụ thuộc vào[*Vi điều khiển*](https://www.vietnic.vn/vi-dieu-khien) Atmega, tức là Atmega168 đi có bộ nhớ flash 16KB trong khi Atmega328 đi có bộ nhớ flash là 32KB. Bộ nhớ flash được sử dụng để lưu trữ mã hóa dữ liệu. Bộ nhớ 2KB trong tổng số bộ nhớ flash được sử dụng cho chương trình khởi động gọi là Bootloader.

* Bộ nhớ SRAM có thể thay đổi từ 1KB hoặc 2KB và EEPROM tương ứng là 512 byte hoặc 1KB đối với Atmega168 và Atmega328.
* ***Thông số kỹ thuật***

Các hàm cơ bản như pinMode() và digitalWrite() được sử dụng để điều khiển hoạt động của các chân Digital trong khi analogRead() được sử dụng để điều khiển các chân Analog.Arduino Nano bo gồm tất cả 14 chân Digital, 8 chân Analog, 2 chân Reset và 6 chân Nguồn.

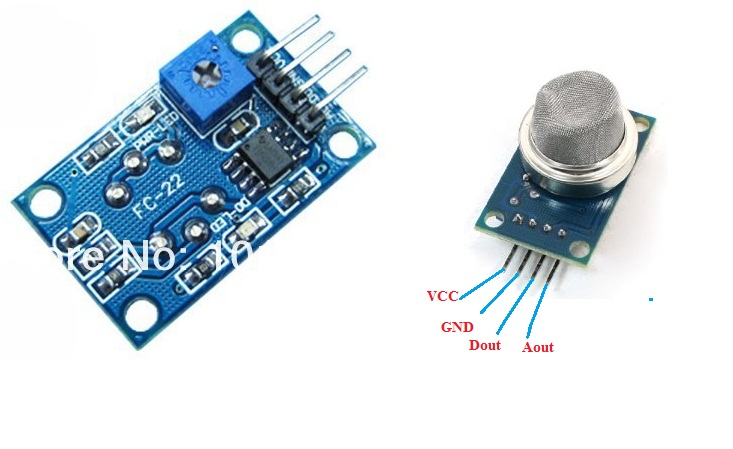
* Chân Reset: Khi tác động nút nhấn reset, Arduino được trả về lại chương trình ban đầu. Rất hữu ích khi chạy chương trình phức tạp và bị treo Vi điều khiển ATmega. Mức tích cực LOW được thiết lập sẽ reset lại Arduino Nano.
* Các chân Analogs: Có 8 chân Analog trên board mạch Arduino Nano được ký hiệu là A0 đến A7. Được sử dụng để đo điện áp tương tự trong khoảng từ 0V đến 5V.
* Chân Rx, Tx: Được sử dụng cho giao tiếp nối tiếp trong đó Tx là truyền dữ liệu và Rx là nhận dữ liệu.
* Chân 13: Để thực hiện bật tắt LED trên board Arduino Nano, sử dụng để quan sát, kiểm tra chương trình cần thiết.
* Chân AREF: Chân này được sử dụng lấy điện áp tham chiếu cho điện áp đầu vào.
* Chân xung PWM: Bao gồm 6 chân là chân 3,5,6,9,10,11 được sử dụng để cung cấp đầu ra 8-bit xung PWM.
* Giao tiếp SPI: Chân 10(SS), Chân 11(MOSI), Chân 12(MISO), Chân 13(SCK) được sử dụng cho SPI Giao diện ngoại vi nối tiếp. SPI được sử dụng chủ yếu để truyền dữ liệu giữa các bộ vi điều khiển và các thiết bị ngoại vi khác như cảm biến, thanh ghi và thẻ nhớ SD.
* Ngắt ngoài (External Interrupts): Chân 2 và 3 sử dụng làm ngắt ngoài được thiết lập trong trường hợp khẩn cấp khi chúng ta cần dừng chương trình chính và tác động các cảnh báo hướng dẫn tại thời điểm đó. Chương trình chính sẽ tiếp tục lại sau khi lệnh ngắt được loại bỏ.
* Giao tiếp I2C: Giao tiếp I2C sử dụng các chân A4 (SDA) và A5 (SCL).

**Tổng kết**

Vì mỗi phòng trong căn nhà không quá 2 cảm biến và 3 cơ cấu chấp hành, trong đó chỉ có cảm biến khí gas MQ2 là có tín hiệu ra dạng tương tự (analog), còn lại chủ yếu là có tín hiệu ra dạng số (digital) nên ở đây chỉ cần sử dụng module Arduino Nano là có thể xử lý được tất cả tín hiệu. Ngoài ra, Arduino Nano tiết kiệm năng lượng, công suất và rẻ hơn so với các Arduino Uno, STM, 8051, …

* + 1. **Cảm biến khí gas MQ2**
* **Yêu cầu kỹ thuật:**
* Đo được thông số cụ thể chính xác.
* Chính xác, tin cậy, hoạt động lâu dài trong môi trường làm việc.
* Kết nối được với khối xử lý đã chọn ở trên.

1. **Giới thiệu module cảm biến khí gas MQ2**

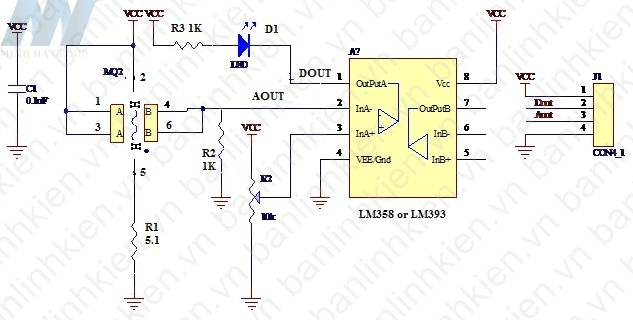


*Hình 3.1.2a. Module cảm biến khí gas MQ2*

* Điện áp hoạt động 3 – 5 V.
* Kết nối 4 chân với 2 chân cấp nguồn (VCC và GND) và 2 chân tín hiệu ngõ ra.
* Hổ trợ cả 2 dạng tín hiệu ra Analog và TTL. Ngõ ra Analog 0 – 4.5V tỷ lệ thuận với nồng độ khí gas, ngõ TTL tích cực mức thấp.
* MQ2 là cảm biến khí, dùng để phát hiện các khí có thể gây cháy. Nó được cấu tạo từ chất bán dẫn SnO2. Chất này có độ nhạy cảm thấp với không khí sạch. Nhưng khi trong môi trường có chất ngây cháy, độ dẫn của nó thay đổi ngay. Chính nhờ đặc điểm này người ta thêm vào mạch đơn gian để biến đổi từ độ nhạy này sang điện áp.
* Khi môi trường sạch điện áp đầu ra của cảm biến thấp, giá trị điện áp đầu ra càng tăng khi nồng độ khí gây cháy xung quanh cảm biến MQ2 càng cao.
* Cảm biến MQ2 hoạt động rất tốt trong môi trường khí hóa lỏng LPG, H2, và các chất khí gây cháy khác. Nó được sử dụng rộng rãi trong công nghiệp và dân dụng do mạch đơn giản và chi phí thấp.
* MQ2 hoạt động rất tốt trong môi trường khí hóa lỏng LPG, H2, và các chất khí gây cháy khác. Nó được sử dụng rộng rãi trong công nghiệp và dân dụng do mạch đơn giản và chi phí thấp.

1. **Sơ đồ cấu tạo và nguyên lí hoạt động của module**

* MQ2 là cảm biến khí, dùng để phát hiện các khí có thể gây cháy. Nó được cấu tạo từ chất bán dẫn SnO2. Chất này có độ nhạy cảm thấp với không khí sạch. Nhưng khi trong môi trường có chất gây cháy, độ dẫn của nó thay đổi lập tức. Chính nhờ đặc điểm này người ta thêm vào mạch đơn giản để biến đổi từ độ nhạy này sang điện áp. Khi môi trường sạch điện áp đầu ra của cảm biến thấp, giá trị điện áp đầu ra càng tăng khi nồng độ khí gây cháy xung quang MQ2 càng cao.
* Khi phát hiện khí gas bị rò rỉ module sẽ xuất tín hiệu ở hai dạng DOUT\_dạng số (digital) và AOUT\_dạng tương tự (analog). Người sử dụng có thể tùy vào mục đích sử dụng để lựa chọn tín hiệu phù hợp.
* Trong mạch có 2 chân đầu ra là Aout và Dout. Trong đó:
* Aout: điện áp ra tương tự. Nó chạy từ 0.3 đến 4.5V, phụ thuộc vào nồng độ khí xung quang MQ2.
* Dout: điện áp ra số, giá trị 0 hay 1 phụ thuộc vào điện áp tham chiếu và nồng độ khí mà MQ2 đo được.



*Hình 3.1.2b. Cấu tạo module cảm biến khí gas MQ2*

Việc có chân ra số Dout rất tiện cho ta mắc các ứng dụng đơn giản, không cần đến vi điều khiển. Khi đó ta chỉ cần chỉnh giá trị biến trở tới giá trị nồng độ ta muốn cảnh báo. Khi nồng độ MQ2 đo được thấp hơn mức cho phép thì Dout = 1, còi sẽ giữ nguyên trạng thái chờ. Khi nồng độ khí gây cháy đo được lớn hơn nồng độ cho phép thì Dout =0, còi sẽ kêu.

* Ta có thể ghép nối vào mạch Realy để điều khiển bật tắt đèn, còi, hoặc thiết bị cảnh báo khác.
* Một điều khó khăn khi làm việc với MQ2 là chúng ta khó có thể quy từ điện áp Aout về giá trị nồng độ ppm. Rồi từ đó hiển thị và cảnh báo theo ppm. Do giá trị điện áp trả về từng loại khí khác nhau, lại bị ảnh hưởng nhiệt độ, độ ẩm nữa.
* Trong thiết bị của mình, có thể xác định điểm cảnh báo một cách thủ công:
* Đầu tiên đo trạng thái không khí sạch, giá trị thu được Vout1.

Cho khí ga từ bật lửa rò rỉ ra. Ta thấy giá trị Aout tăng lên. Khi đạt khoảng cách khí ga từ bật lửa hợp lý rồi tương ứng với nồng độ khí bắt đầu nguy hiểm, ta ghi lại giá trị Vout2. Ta chọn giá trị Vout2 là giá trị ngưỡng cảnh báo. Nếu giá trị đo được lớn hơn ta sẽ cảnh báo.

* Chỉnh chân biến trở để điện áp đo tại chân 3 của L358 = Vout2.

1. **Ứng dụng của module**

Cảm biến khí gas có ứng dụng rất lớn trong đời sống:

* Phát hiện rò rỉ khí gas trong nước.
* Trong công nghiệp dùng đề phát hiện chất dễ cháy.
* Máy phát hiện khí dễ cháy.
  + 1. **Relay**

**Sử dụng mạch 1 Relay Opto Chọn Mức Kích**

Mạch 1 Relay Opto chọn mức kích High/Low (5/12/24VDC) được sử dụng để bật, tắt thiết bị AC/DC qua Relay, mạch có thể tùy chọn kích bằng mức cao hoặc thấp (High/Low) qua Jumper, ngoài ra mạch còn bổ sung thêm Opto cách ly cho độ an toàn và chống nhiễu vượt trội (một số mạch trên thị trường không có Opto), thích hợp với các ứng dụng bật tắt, điều khiển thiết bị qua Relay.

* **Thông số kỹ thuật sản phẩm:**
* Điện áp sử dụng: có ba loại 5/12/24VDC
* Dòng tiêu thụ: khoảng 200mA /1Relay
* Tín hiệu kích: Tùy chọn mức cao High (5/12/24VDC theo loại Relay) hoặc thấp Low (0VDC) qua Jumper.
* Tiếp điểm đóng ngắt Relay trên mạch: Max 250VAC-10A hoặc 30VDC-10A (Để an toàn nên dùng cho tải có công suất **<100W**).
* Kích thước: 1.97 in x 1.02 in x 0.75 in (5.0 cm x 2.6 cm x 1.9 cm)

****

*Hình 3.1.3*

* + 1. **Nguồn**

**Yêu cầu kỹ thuật:**

Sau khi họn các thiết bị như trên ta có thể thấy hệ thồng cần được cấp nguồn có có giá trị từ 3,3V tới 5V, các modul cảm biến, và servo có thể sử dụng nguồn đưa ra từ Arduino Nano. Do đó hệ thống cần hai nguồn từ ngoài vào đáp ứng được hoạt động của NODEMCU esp8266 và Arduino Nano.

**🢥 Sử dụng nguồn pin sạc.**

Nhóm tận dụng sẵn pin sạc điện thoại dự phòng để cung cấp nguồn cho hệ thống.

**Thông số kỹ thuật:**

* Pin sạc Xiaomi 10000 mAh / micro USB
* Nguồn ra: USB: 5V - 2.6A, 9V - 2.1A, 12V - 1.5AUSB: 5V - 2.6A
* Kích thước: 148 x 74 x 15 mm



*Hình 3.1.4*

* **Ưu điểm**: Nguốn sạc dự phòng hoàn chỉnh kết nối micro usb do đó không phải lo lắng về tính ổn định của hệ thống, sử dụng trực tiếp vào hệ thống.
* **Khuyết điểm**: Phải sạc khi hết pin. Tuy không cần thướng xuyên lắm.
* **KẾT LUẬN**
* Nhóm tin tưởng vào việc sử dụng 1 bộ nguồn đã được hoàn thiện bởi Xiaomi hơn là tự mình hoàn thiện
* Tiết kiệm thời gian hơn cho nhóm.
* Tận dụng sẵn cục sạc dự phòng của thành viên trong nhóm, không phải mua mới.
  + 1. **Module Buzzer:**

**Yêu cầu kỹ thuật:**

* Phát ra tiếng kêu rõ ràng.
* Chính xác, tin cậy, hoạt động lâu dài trong môi trường làm việc.
* Kết nối được với khối xử lý đã chọn ở trên.

****

*Hình 3.1.5*

**Thông số kỹ thuật:**

* Điện áp sử dụng: 3.3~5VDC
* Tín hiệu kích: TTL mức thấp LOW 0VDC
* Kích thước: 32x13mm

**Ứng dụng:**

* Sử dụng dùng để phát ra âm thanh khi kích tín hiệu, được dứng dụng trong các hệ thống cảnh báo, chống trộm
* Dùng để cảnh báo cháy, cảnh báo rò rỉ khí gas, khói, …
  + 1. **Cảm biến nhiệt độ, độ ẩm DHT11**

Cảm biến độ ẩm và nhiệt độ DHT11 Temperature Humidity Sensor là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp digital 1 dây truyền dữ liệu duy nhất). Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào.

**Yêu cầu kỹ thuật:**

* Đo được thông số cụ thể chính xác.
* Chính xác, tin cậy, hoạt động lâu dài trong môi trường làm việc.
* Kết nối được với khối xử lý đã chọn ở trên.

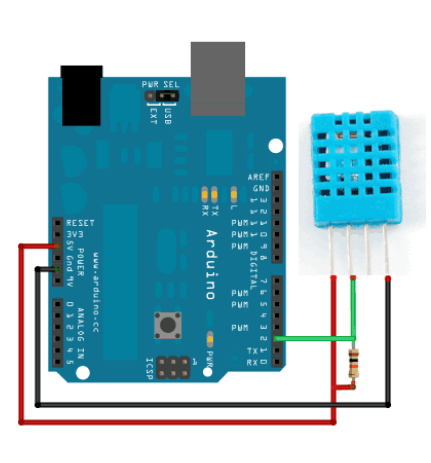
**Thông số kỹ thuật:**

* Nguồn: 3 -> 5 VDC.
* Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
* Đo tốt ở độ ẩm 20 to 70%RH với sai số 5%.
* Đo tốt ở nhiệt độ 0 to 50°C sai số ±2°C.
* Tần số lấy mẫu tối đa 1Hz (1 giây 1 lần).
* Kích thước 15mm x 12mm x 5.5mm.
* chân, khoảng cách chân 0.1''.

**Kết nối với cảm biến DHT11:**

Cảm biến có 4 chân

* **VCC**
* **Data out**
* **Not connected**
* **Ground**

****

*Hình 3.1.6*

* 1. **THI CÔNG PHẦN CỨNG**
     1. **Yêu cầu hệ thống cụ thể**

**Cơ chế hoạt động:**

Hệ thống tự động cập nhật thông tin về khí gas theo thời gian thực sau đó gửi giá trị lên app Blynk, thông báo cho người dùng khi có khí gas/khói qua wifi hoặc mạng di động. Hệ thống nhận lệnh điều khiển đèn, quạt, máy lạnh trực tiếp từ người dùng qua nút ấn tại chỗ hay từ xa qua wifi hoặc mạng di động. Ngoài ra hệ thống còn cho phép người dùng điều khiển đóng mở cửa thông qua nút nhấn.

**Input:**

* Thông số môi trường từ cảm biến khí gas MQ2
* Nút nhấn: Điều khiển bật tắt các đèn, quạt, máy lạnh, điều khiển đóng mở cửa.
* Thông tin điều khiển từ xa của người điều khiển thông qua wifi hoặc mạng di động.

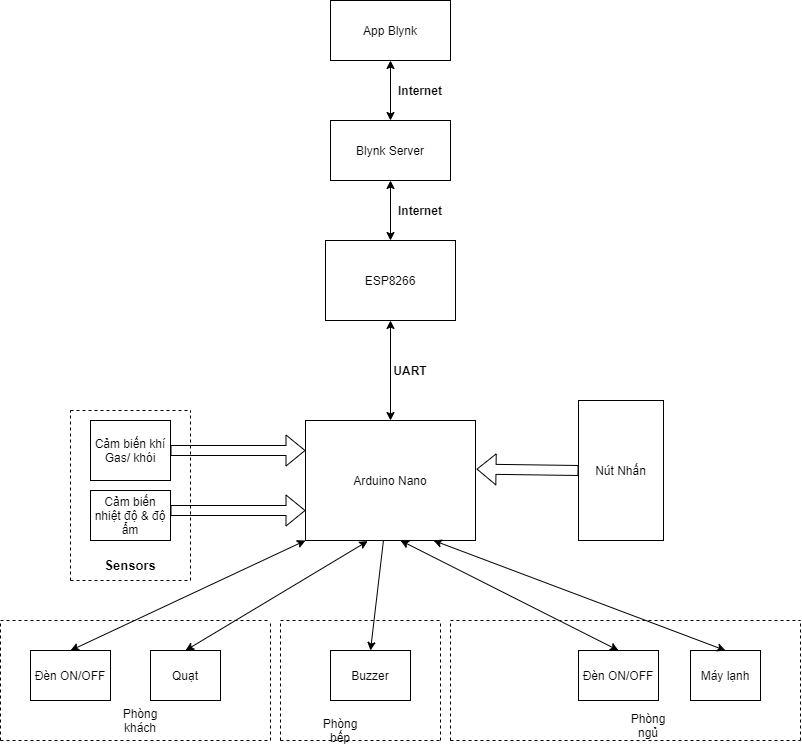
**Output:**

* Relay điều khiển đèn.
* Hiển thị thông số cảm biến đo được lên App Blynk
* Hiển thị trạng thái ON/OFF, relay cho người dùng ở xa qua wifi hoặc mạng di động.

**Kết nối:** Giao tiếp với người dùng từ xa qua wifi hoặc mạng di động.

* + 1. **Sơ đồ khối hệ thống**

Từ yêu cầu cụ thể đặt ra như trên, ta có sơ đồ khối như sau:

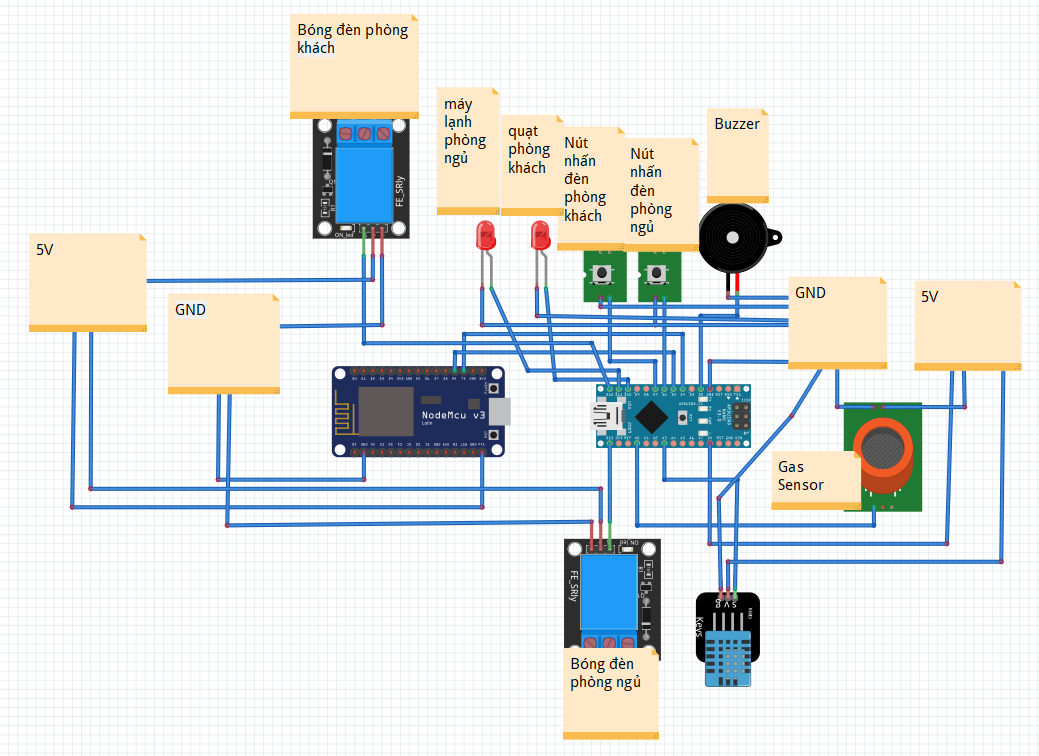
*****Hình 3.2.2*

* Khối cảm biến khí gas/khói:
* Nhiệm vụ: Phát hiện các khí có thể gây cháy và báo thông số theo thời gian thực.
* Yêu cầu: Thông số chính xác, đảm bảo hoạt động ổn định trong thời gian dài.
* Khối Server:
* Nhiệm vụ: Xử lý các tín hiệu từ Blynk Server khi có thay đổi từ App Blynk và truyền tín hiệu điều khiển đến khối Client.
* Yêu cầu: Ổn định, bảo mật, có giải thuật xử lý hợp lý.
* Khối Client:
* Nhiệm vụ: Xử lý và thực hiện các tín hiệu từ khối Server đồng thời xử lý thông số từ cảm biến, tự động thực hiện các điều khiển dữ liệu nhận được, bao gồm từ lập trình trước hay trực tiếp qua nút nhấn và wifi.
* Yêu cầu: Ổn định, bảo mật, có giải thuật xử lý hợp lý.
* Nút nhấn điều khiển:
* Nhiệm vụ: Gửi thông tin điều khiển từ con người đến thiết bị. Nút nhấn cơ học khi điều khiển các thiết bị phải được đồng bộ hóa với nút nhấn điều khiển trên App Blynk.
* Yêu cầu: Ổn định, chính xác.
* Nguồn:

+ Nhiệm vụ: Cung cấp năng lượng cho toàn hệ thống.

+ Yêu cầu: Đủ công suất, đáng tin cậy, ổn định.

* Ngoài ra còn một khối giao tiếp với người dùng qua internet:
* Nhiệm vụ: Giúp người dùng điều khiển hệ thống từ xa thông qua kết nối internet.
* Yêu cầu: Hiển thị được thông số cảm biến, trạng thái relay cho người dùng. Giúp người dùng điều khiển relay bật, tắt các đèn, máy quạt, máy lạnh.
  + 1. **Sơ đồ chi tiết**



*Hình 3.2.3*

* **Giải thích sơ đồ nối chân:**

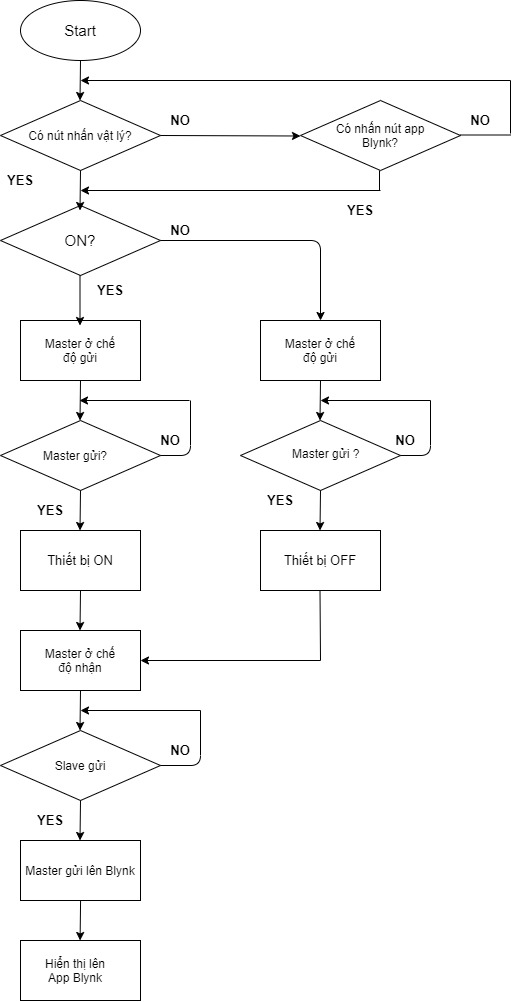
Arduino Nano:

* Pin 2: nối vô chân tín hiệu của còi Buzzer
* Pin 4: nối vô chân Tx của NodeMCU Esp8266
* Pin 5: nối vô chân Rx của NodeMCU Esp8266
* Pin 6: nối vô nút nhấn điều khiển bóng đèn phòng ngủ
* Pin 7: nối vô nút nhấn điều khiển bóng đèn phòng khách
* Pin 10: nối vô chân tín hiệu của quạt phòng khách (mô phỏng là Led 5mm)
* Pin 11: nối vô chân tín hiệu của máy lạnh phòng khách (mô phỏng là Led 5mm)
* Pin 12: nối vô chân data in của Relay điều khiển bóng đèn phòng khách
* Pin 13: nối vô chân data in của Relay điều khiển bóng đèn phòng ngủ
* A0: nối vô chân tín hiệu cảm biến khí Gas/ Khói
* A3: nối vô chân tín hiệu của cảm biến nhiệt độ, độ ẩm DHT11.
* Sử dụng 2 cáp cấp nguồn 5VDC từ pin sạc điện thoại cho Arduino Nano và NodeMCU Esp8266.

1. **THIẾT KẾ VÀ THI CÔNG PHẦN MỀM**
   1. **Lưu đồ giải thuật**

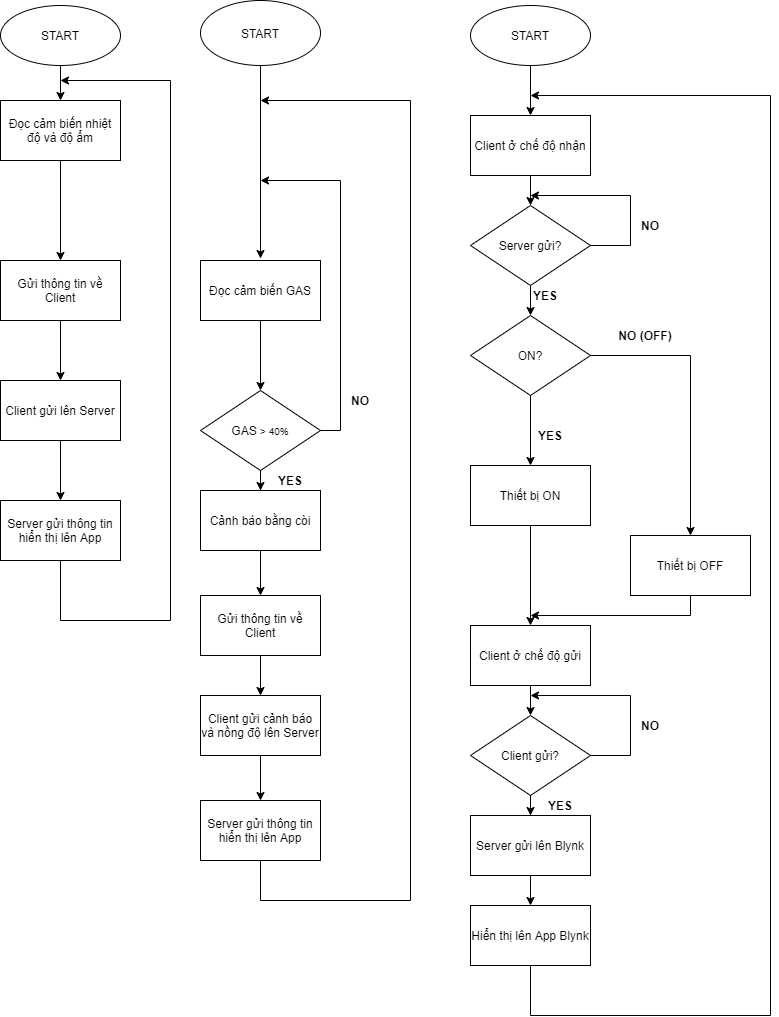
Nhóm sử dụng phần mềm Arduino IDE để lập trình cho NodeMCU ESP8266 dự trên các lưu đồ giải thuật sau:

* ***Phía Server:***



*Hình 4.1a*

* ***Phía Client***:



*Hình 4.1b*

* **Giải thích phía Server:**

Sau khi bắt đầu hệ thống, hệ thống sẽ xem xét người dùng có nhấn nút nhấn vật lý hay nút nhấn trên App Blynk hay không. Nút nhấn vật lý và nút nhấn trên App phải đồng bộ với nhau nhờ giải thuật khi lập trình. Nếu có nút nhấn vật lý hoặc từ Blynk thì tín hiệu ON hoặc OFF của thiết bị sẽ gửi từ Server (blynk server) đến Client để điểu khiển thiết bị theo yêu cầu. Đồng thời, từ thiết bị, phía Client sẽ phản hồi tín hiệu về lại Blynk Server và hiển thị lên App Blynk (chủ yếu là phản hồi các thông số khí Gas/ khói và trạng thái ON/OFF của các đèn).

* **Giải thích phía Client:**
* *Cảnh báo khí Gas/ khói:*

Đối với phần cảnh báo khí Gas/ khói, khi có Gas/ Khói thì cảm biến MQ2 sẽ đọc xem nồng độ Gas/Khói có vượt quá 40% trong môi trường không khí hay không. Nếu vượt qua 40% thì còi báo sẽ kêu để báo động cho chủ nhà cảnh báo có rò rỉ gas hoặc cháy. Đồng thời thông số khí Gas/Khói sẽ liên tục được cập nhật và gửi lên App Blynk để người dùng có thể theo dõi dễ dàng. Sau khi nồng độ Gas/Khói giảm dưới 40% thì tự dộng còi báo sẽ tắt.

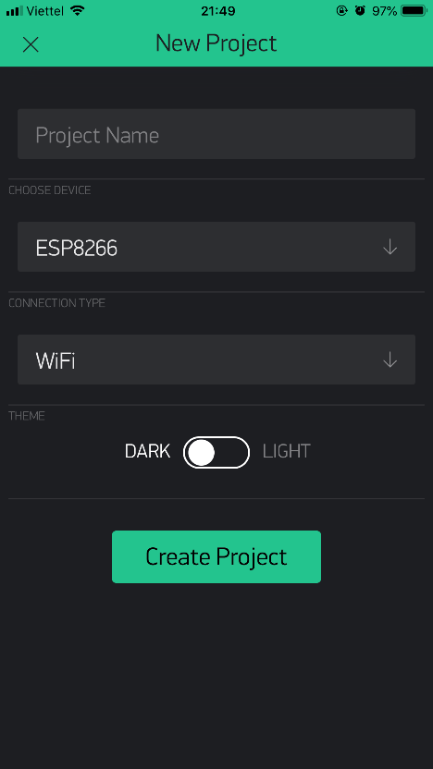
* *Cảm biến nhiệt độ, độ ẩm*

Cảm biến DHT11 sẽ liên tục đọc giá trị nhiệt độ và độ ẩm từ môi trường và gửi lên App Blynk qua các vi điều khiển và Blynk Server để người dùng có thể nhận biết.

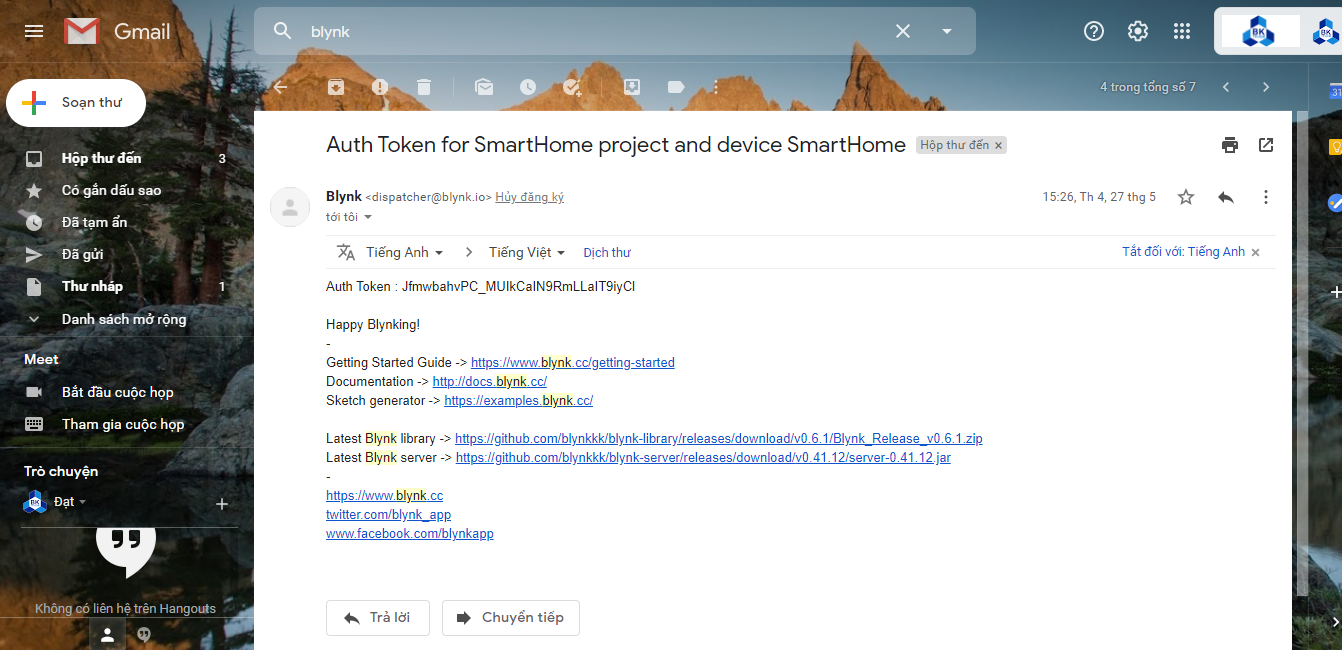
* 1. **Các bước tạo project app Blynk**

**Bước 1**: Tạo project

* Tải app **Blynk** trên **CH Play/App Store**
* Khởi động app, chọn **create new project**
* Đặt tên và chọn **device** sử dụng

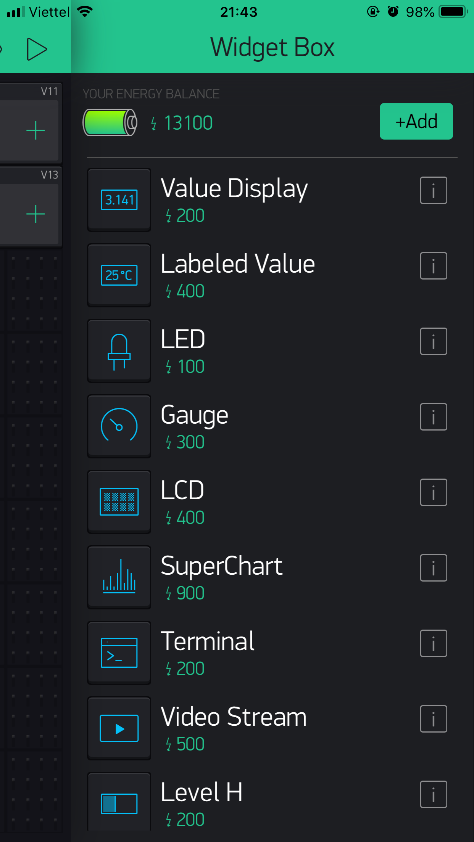


* Sau khi được tạo, app sẽ gửi một **token** đến mail đã đăng ký.



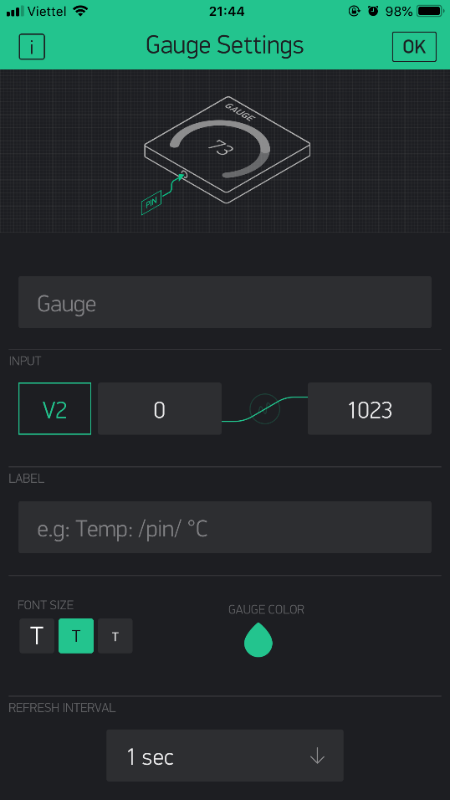
**Bước 2**: Thêm giao diện hiển thị thông số.

* Nhấn  / bên phải màn hình, vào mục Display chọn **Gauge**
* Nhấn vào giao diện hiển thị sẽ xuất hiện mục **Gauge Settings**
* Đặt tên và cài đặt các thông số hiển thị nồng độ khí gas

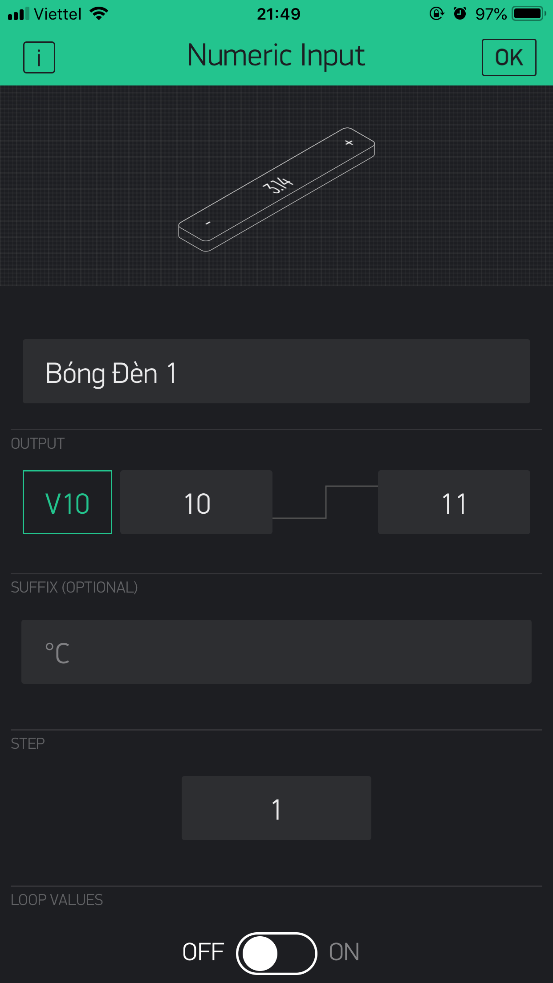


**Bước 3**: Chỉnh thông số hiển thị

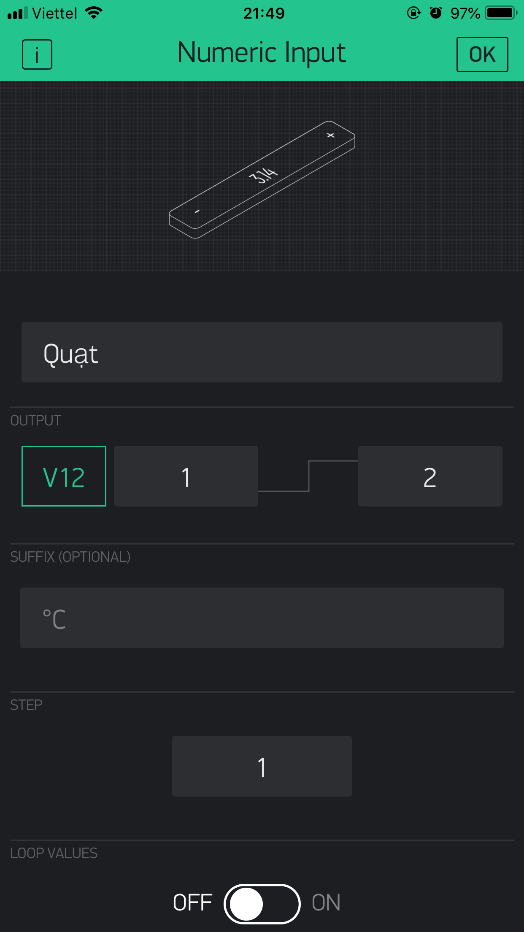
* Nồng độ giá trị từ: 0 – 1023
* Thời gian cập nhật 1 giây
* Input: V2

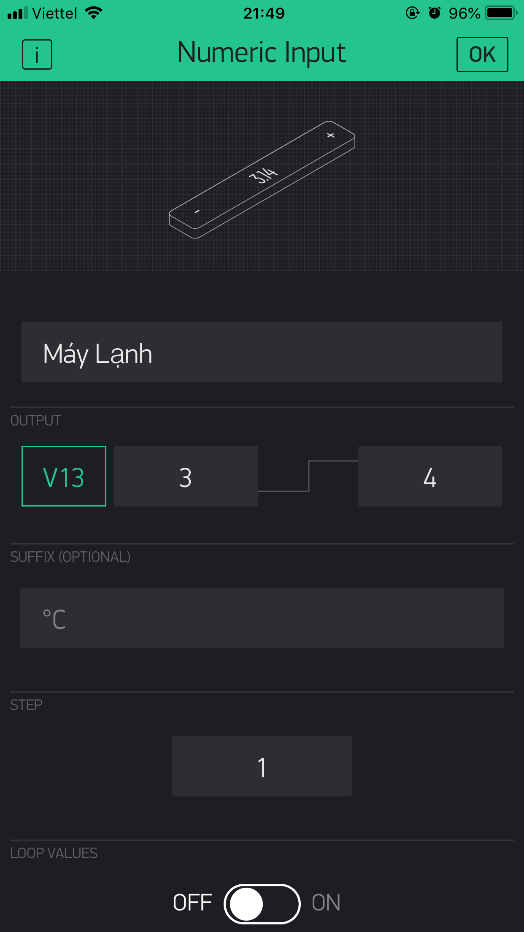


**Bước 4**: Thêm Widget “Numeric Input” cho 2 bóng đèn ON/OFF ở phòng khách và phòng ngủ, máy quạt và máy lạnh.

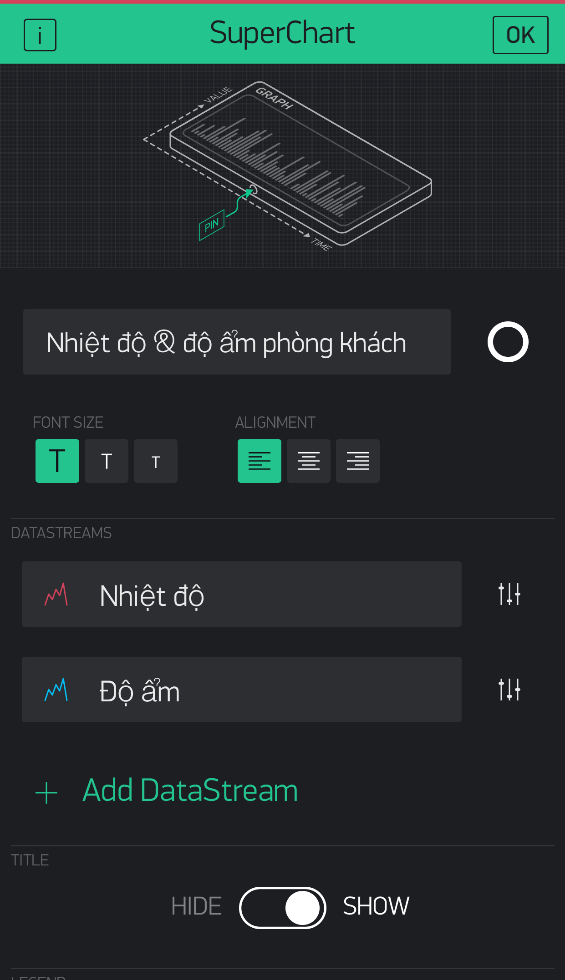


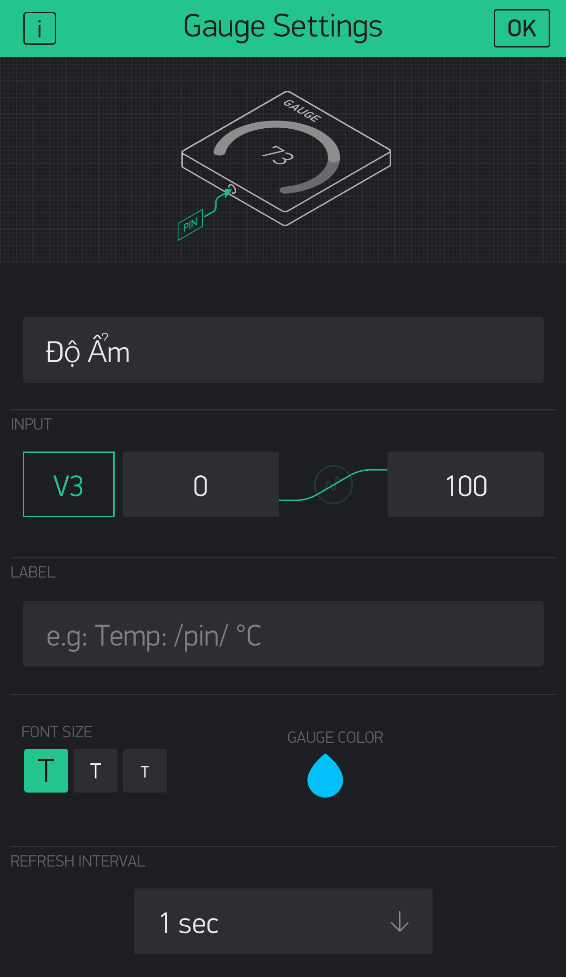


****

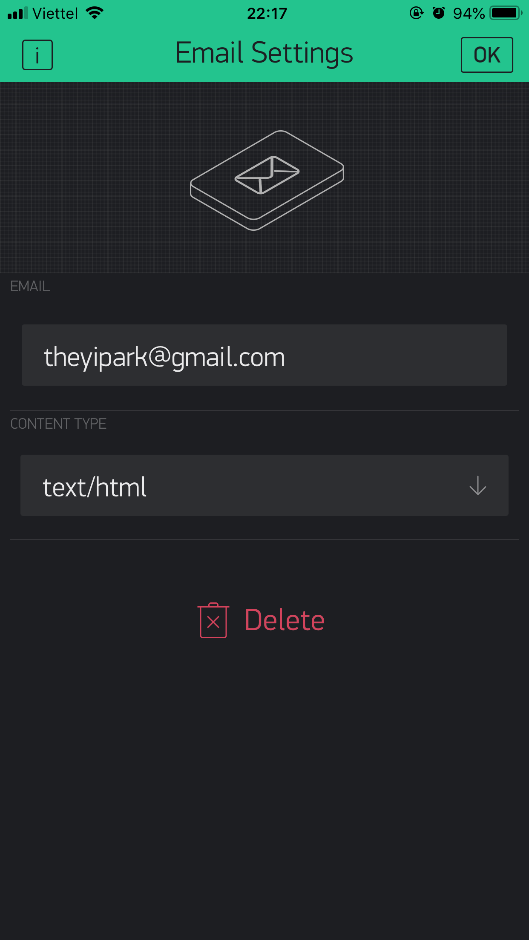
****

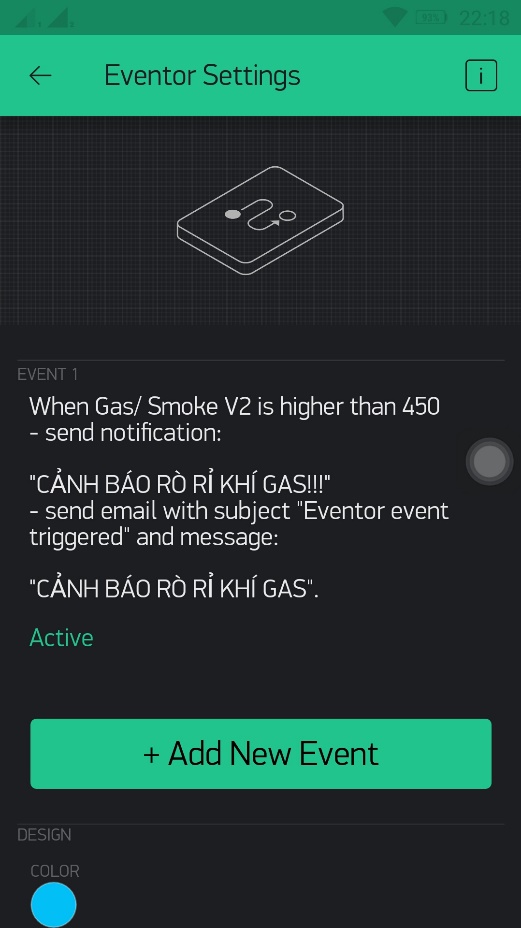
**Bước 5:** Cài thêm giao diện của “Nhiệt độ”, “Độ ẩm” và Biểu đồ nhiệt độ, độ ẩm theo thời gian





**Bước 6:** Cài các Widget cảnh báo



****

**Giao diện sau khi hoàn thành:**

****

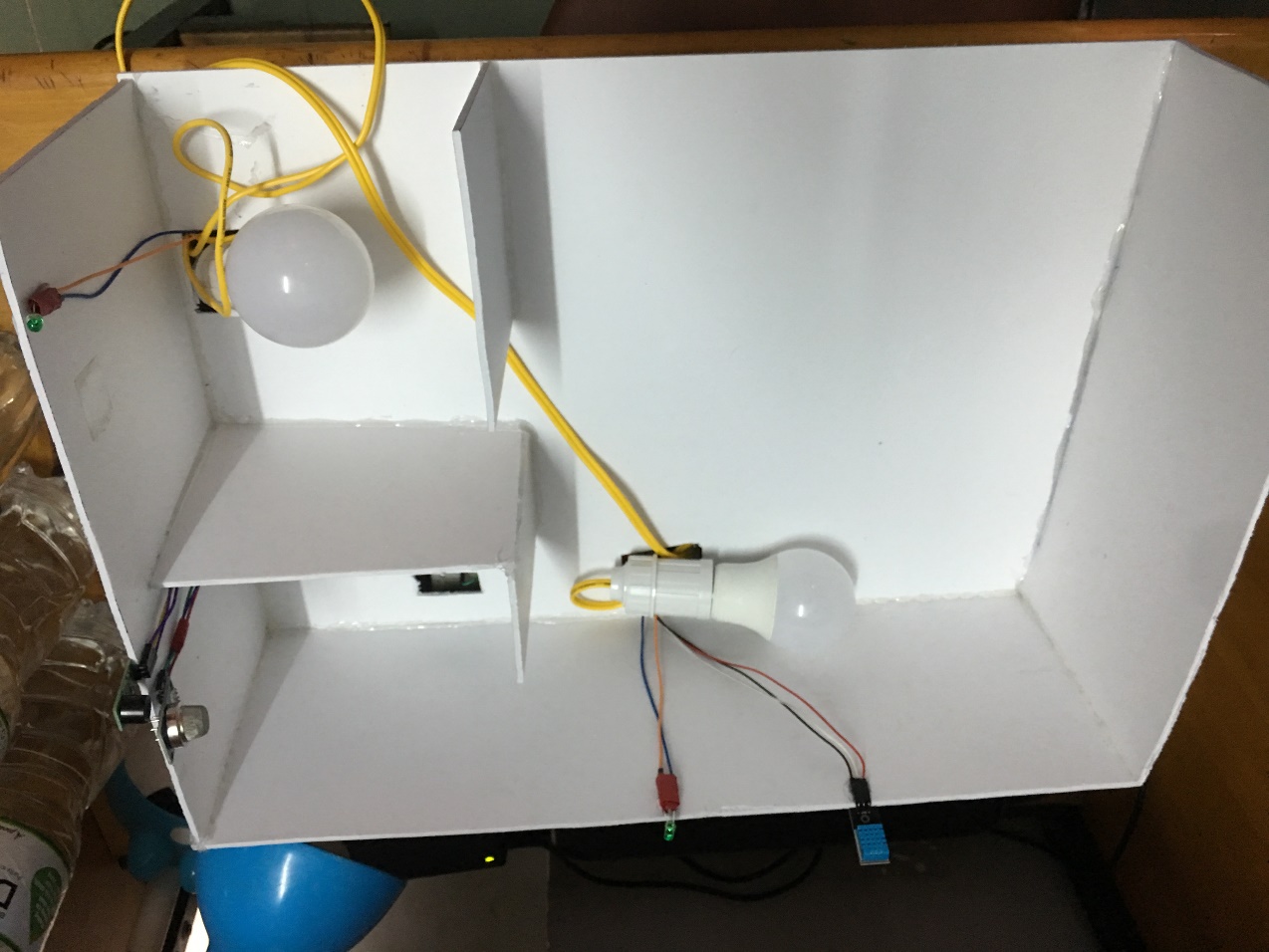
* Giải thích:
* Bóng đèn 1: Numeric Input khi nhấn dấu trừ (-) về giá trị 10, bóng đèn ở phòng ngủ sẽ tắt. Khi nhấn dấu cộng (+) về giá trị 11, bóng đèn ở phòng ngủ sẽ sáng.
* Bóng đèn 2: Numeric Input khi nhấn dấu trừ (-) về giá trị 12, bóng đèn ở phòng khách sẽ tắt. Khi nhấn dấu cộng (+) về giá trị 13, bóng đèn ở phòng khách sẽ sáng.
* Quạt: Numeric Input khi nhấn dấu trừ (-) về giá trị 14, quạt ở phòng khách sẽ tắt. Khi nhấn dấu cộng (+) về giá trị 15, quạt ở phòng khách sẽ được bật.
* Máy lạnh: Numeric Input khi nhấn dấu trừ (-) về giá trị 16, máy lạnh ở phòng ngủ sẽ tắt. Khi nhấn dấu cộng (+) về giá trị 17, máy lạnh ở phòng ngủ sẽ được bật.
* Gas/ Smoke: Widget này sẽ liên tục cập nhật và giám sát nồng độ Gas/khói của phòng bếp.
* Nhiệt độ: Widget này sẽ liên tục cập nhật và giám sát nhiệt độ môi trường của phòng khách.
* Độ ẩm: Widget này sẽ liên tục cập nhật và giám sát độ ẩm môi trường của phòng khách.

# KẾT QUẢ, KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* 1. **Đo đạc và thực nghiệm hệ thống:**

### **5.1.1.** **Mô hình thực tế**

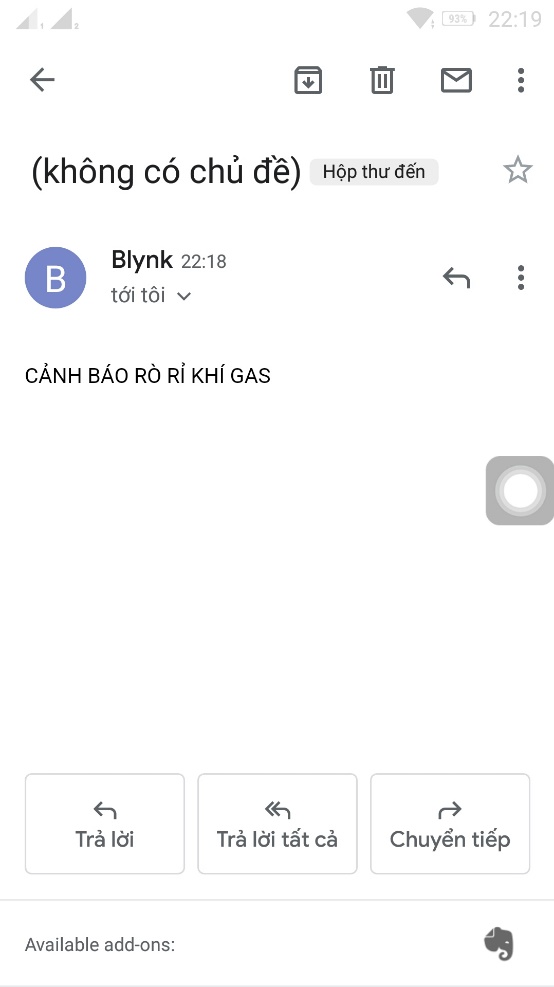
Mô hình ngôi nhà thực tế có kích thước là 370x320 mm, được chia làm 3 phần tương ứng với một phòng khách, một phòng ngủ, một phòng bếp.



*Hình 5.1.1*

* + 1. **. Kết quả giám sát một số thiết bị**
* **Báo lên điện thoại và mail người dung khi khí gas vượt mức cho phép**

****

****

*Hình 5.1.2a*

* **Biểu đồ nhiệt độ, độ ẩm theo thời gian**

****

*Hình 5.1.2b*

* 1. **Đánh giá kết quả làm việc nhóm.**

Nhóm làm việc khá hiệu quả, thành quả đạt được hiện tại tuy hạn chế nhưng đều có sự đóng góp như nhau cũng như khả năng lắng nghe, góp ý của tất cả các thành viện trong nhóm.

|  |  |
| --- | --- |
| **BẢNG PHÂN CÔNG VIẾT BÁO CÁO** | |
| **Công việc** | **Thành viên phụ trách** |
| * Tập hợp phần việc mình đã làm, trình bày cách thức làm, tài liệu tham khảo ở phần phần công công việc. Gửi cho người viết báo cáo ( thô ) | Cả nhóm |
| * Viết báo cáo ( thô ) | Thành Đạt |
| * Dò lỗi chính tả | Tiến Đạt |
| * Hoàn thiện, trình bày theo form LVTN | Cả nhóm |

* 1. **Kết luận**

Do còn hạn chế về kiến thức và còn nhiều sai sót trong lần đầu làm đồ án nên trong quá trình tìm hiểu chúng em đã gặp phải một số khó khăn nhất định. Nhưng với sự đam mê và sự hướng dẫn nhiệt tình của thầy giáo cộng thêm tìm tòi, khám phá cái mới đã giúp em đạt được một số kết quả sau:

* Tổng quan về hệ thống giám sát và điều khiển qua mạng internet, và ngôi nhà điều khiển tự động. Đồng thời em cũng biết được tình hình sử dụng, xu hướng phát triển của Việt Nam và các nước khác trên thế giới.
* Thiết kế được mô hình ngôi nhà mặc dù còn thô và chưa có tính thẩm mỹ.
* Tìm hiểu được cơ sở, ý nghĩa của việc điều khiển thiết bị qua internet.
* Hiểu được nguyên lí làm việc của các module trong hệ thống và cách ghép nối chúng như thế nào.
* Tiến hành chạy thực nghiệm, đánh giá kết quả.

Bên cạnh những kết quả đạt được thì mô hình vẫn tồn tại nhiều giới hạn về kĩ thuật như khả năng mở rộng số lượng biến giám sát và điều khiển, xảy ra lỗi, thời gian đáp ứng còn lâu cũng như nhiều chức năng không được hoàn thiện.

* 1. **Hướng phát triển**

Hệ thống điều khiển và giám sát qua mạng internet có ý nghĩa rất lớn về nhiều mặt trong cuộc sống như: an ninh, quốc phòng, chăn nuôi, y tế, gia đình, trồng trọt….. Việc làm chủ được vấn đề này vẫn còn nhiều hạn chế nhất định, nhất là đối với một đất nước đang phát triển khoa học công nghệ như chúng ta. Qua đề tài này em mong muốn sử dụng kiến thức học được trong thời gian sinh viên để thực hiện việc tiếp cận với công nghệ và xu hướng của thế giới.

Qua đồ án này em xin đề xuất một số hướng phát triển cho hệ thống:

* Lắp đặt hệ thống camera giám sát cho hệ thống.
* Ứng dụng hệ thống giám sát và điều khiển trong y tế, giáo dục.
* Xây dựng một phần mềm quản lí trên các thiết bị di động, phần mềm đó có khả năng tự động thông báo các điều kiện bất lợi cho người dùng mà không cần phụ thuộc vào Blynk.
* Sử dụng những con vi điều khiển khác như Kit STM32F407 (ARM Cortex M4 + DSP Core) để thay cho Arduino Nano.
* Sử dụng một chuyển giao tiếp khác thay cho UART (ví dụ như RS485) để có truyền tín hiệu đi được xa hơn.

# 6. TÀI LIỆU THAM KHẢO

**[1]** <https://www.electroniclinic.com/nodemcu-with-arduino-serial-communication-control-and-monitoring/>

**[2]** <https://www.youtube.com/watch?v=vDm1zhqS-wE&t=298s>

**[3]** <https://lophocvui.com/iot-internet-of-things/smart-home/openhab/tong-quan-ve-esp8266/>

**[4]** <https://nshopvn.com/product/cam-bien-khi-gas-mq-2/?gclid=Cj0KCQjw6ar4BRDnARIsAITGzlDCRh_Yn0J5fXViaX8GhanWmzlHn2Praqx8gvunIi8zQAV6sPJjxpQaAm9pEALw_wcB>

**[5]** <https://nshopvn.com/product/module-1-relay-voi-opto-cach-ly-kich-h-l-5vdc/?gclid=Cj0KCQjw6ar4BRDnARIsAITGzlDjTFfl2hcC3DNzG6VGb0Sn2IKLy2xlUxUe-AkjX7Yc1hdJyCJlqAgaAjdDEALw_wcB>

**[6]** <https://nshopvn.com/product/dong-co-servo-sg90-tower-pro/?gclid=Cj0KCQjw6ar4BRDnARIsAITGzlCZ9h8UcyY4gZ8Ba0AJaqreWOT8AjDWghQ8gECvNH7Oj0W2mfysI3caAi0YEALw_wcB>

**[7]** <https://nshopvn.com/product/arduino-nano-v3-0-atmega328p-khong-kem-day-cap-usb/?gclid=Cj0KCQjw6ar4BRDnARIsAITGzlAs3OZ8GnzqI9T_ufSlNiHBFiGV9JMqEvcFA2kwrWgsOLuPMiRnYe4aAidJEALw_wcB>

**[8]** <https://nshopvn.com/product/module-thu-phat-wifi-esp8266-nodemcu-lua-cp2102/?gclid=Cj0KCQjw6ar4BRDnARIsAITGzlBRKEw32p0d1hoW7V9DO66a2mUWKsarlnhBpm7gGIFFdvr_F-ySTFQaAvVVEALw_wcB>

**[9]** <https://robojax.com/learn/arduino/?vid=robojax-servo-push-button>

**[10]** <https://mechasolution.vn/Blog/blynk-la-gi>

**[11]** <https://www.makerlab.vn/blynk-huong-dan-cai-dat/>

**[12]** <https://forum.arduino.cc/index.php?topic=3800.0>

**[13]** <http://arduino.vn/bai-viet/68-cai-dat-driver-va-arduino-ide>

**[14]** <https://cmcdistribution.com.vn/kien-thuc-cnc/giao-tiep-uart-la-gi-ung-dung-cua-uart-trong-cuoc-song>

**[15]** [http://vn.serial-cable.com/info/difference-of-uart-serial-and-rs232-serial-30244559.html#:~:text=UART%20%C4%91%E1%BB%81%20c%E1%BA%ADp%20%C4%91%E1%BA%BFn%20c%E1%BB%95ng,c%E1%BB%95ng%20n%E1%BB%91i%20ti%E1%BA%BFp%20m%E1%BB%A9c%20RS232.&text=C%C3%A1c%20ph%C6%B0%C6%A1ng%20th%E1%BB%A9c%20truy%E1%BB%81n%20th%C3%B4ng,(transistor%2Dtransistor%20logic).](http://vn.serial-cable.com/info/difference-of-uart-serial-and-rs232-serial-30244559.html#:~:text=UART%20%C4%91%E1%BB%81%20c%E1%BA%ADp%20%C4%91%E1%BA%BFn%20c%E1%BB%95ng,c%E1%BB%95ng%20n%E1%BB%91i%20ti%E1%BA%BFp%20m%E1%BB%A9c%20RS232.&text=C%C3%A1c%20ph%C6%B0%C6%A1ng%20th%E1%BB%A9c%20truy%E1%BB%81n%20th%C3%B4ng,(transistor%2Dtransistor%20logic))

**[16]** <https://qastack.vn/electronics/37814/usart-uart-rs232-usb-spi-i2c-ttl-etc-what-are-all-of-these-and-how-do-th>

**[17]** <https://advancecad.edu.vn/khai-niem-co-ban-ve-truyen-thong-uart-so-do-khoi-ung-dung/>

**[18]** <https://hshop.vn/products/cam-bien-do-am-nhiet-do-dht11>

# 7. PHỤ LỤC

***1. Code lập trình NODEMCU của nhóm:***

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <SoftwareSerial.h>

#include <SimpleTimer.h>

int pinValue1;

int pinValue2;

int pinValue3;

int pinValue4;

char auth[] = "5uz1Fujc2ltaGnHeDla8Ux7T8tHzwp21";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "P519";

char pass[] = "123456789";

SimpleTimer timer;

String myString; // complete message from arduino, which consistors of snesors data

char rdata; // received charactors

//for temperature , pressure and altitude

int firstVal, secondVal,thirdVal; // sensors

// This function sends Arduino's up time every second to Virtual Pin (1).

// In the app, Widget's reading frequency should be set to PUSH. This means

// that you define how often to send data to Blynk App.

void myTimerEvent()

{

// You can send any value at any time.

// Please don't send more that 10 values per second.

Blynk.virtualWrite(V1, millis() / 1000);

}

void setup()

{

// Debug console

Serial.begin(9600);

Blynk.begin(auth, ssid, pass);

timer.setInterval(1000L,sensorvalue1);

timer.setInterval(1000L,sensorvalue2);

timer.setInterval(1000L,sensorvalue3);

}

void loop()

{

if (Serial.available() == 0 )

{

Blynk.run();

timer.run(); // Initiates BlynkTimer

}

if (Serial.available() > 0 )

{

rdata = Serial.read();

myString = myString+ rdata;

// Serial.print(rdata);

if( rdata == '\n')

{

// Serial.println(myString);

// Serial.println("fahad");

// new code

String l = getValue(myString, ',', 0);

String m = getValue(myString, ',', 1);

String n = getValue(myString, ',', 2);

firstVal = l.toInt();

secondVal = m.toInt();

thirdVal = n.toInt();

myString = "";

// end new code

}

}

}

// in Blynk app writes values to the Virtual Pin 10

void sensorvalue1()

{

int sdata = firstVal;

// You can send any value at any time.

// Please don't send more that 10 values per second.

Blynk.virtualWrite(V2, sdata);

}

void sensorvalue2(){

int sdata = secondVal;

Blynk.virtualWrite(V3, sdata); //độ ẩm

}

void sensorvalue3(){

int sdata = thirdVal;

Blynk.virtualWrite(V4, sdata); //nhiệt độ

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++) {

if (data.charAt(i) == separator || i == maxIndex) {

found++;

strIndex[0] = strIndex[1] + 1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

BLYNK\_WRITE(V10)

{

pinValue1 = param.asInt(); // assigning incoming value from pin V10 to a variable

Serial.print(pinValue1);

}

// in Blynk app writes values to the Virtual Pin 11

BLYNK\_WRITE(V11)

{

pinValue2 = param.asInt(); // assigning incoming value from pin V10 to a variable

Serial.print(pinValue2);

}

BLYNK\_WRITE(V12)

{

pinValue3 = param.asInt(); // assigning incoming value from pin V10 to a variable

Serial.print(pinValue3);

}

BLYNK\_WRITE(V13)

{

pinValue4 = param.asInt(); // assigning incoming value from pin V10 to a variable

Serial.print(pinValue4);

}

1. ***Code lập trình Arduino Nano của nhóm:***

//Khai báo thư viện

#include <Wire.h>

#include <SoftwareSerial.h>

#include <DHT.h>

//Rx Tx

SoftwareSerial nodemcu(4,5);

#define relay1 13 //đèn led phòng ngủ

#define relay2 12 //đèn led phòng khách

#define button\_led\_1 6 //nút nhấn đk led ngủ

#define button\_led\_2 7 //nút nhấn đk led khách

#define BUZZER 2 //còi

#define sensor1 A0 // Gas sensor

#define sensor2 A3

#define DHTTYPE DHT11

DHT dht(sensor2, DHTTYPE);

#define quat 10 //quạt phòng khách

#define mayLanh 11 //máy lạnh phòng ngủ

#define button\_quat A1 //nút nhấn đk quạt

#define button\_mayLanh A2 //nút nhấn đk máy lạnh

long int data;

int sdata1 = 0; // temperature data

long int sdata2 = 0; // pressure data

int sdata3 = 0; // altitude data

String cdata; // complete data

void setup()

{

Serial.begin(9600);

nodemcu.begin(9600);

pinMode(relay1, OUTPUT); //đèn led phòng ngủ

pinMode(relay2, OUTPUT); //đèn led phòng khách

pinMode(button\_led\_1, INPUT\_PULLUP); //nút nhấn led ngủ

pinMode(button\_led\_2, INPUT\_PULLUP); //nút nhấn led khách

pinMode(BUZZER, OUTPUT); //còi

pinMode(sensor1, INPUT); //gas sensor

pinMode(button\_quat, INPUT\_PULLUP); //nút nhấn quạt

pinMode(button\_mayLanh, INPUT\_PULLUP); //nút nhấn máy lạnh

pinMode(quat, OUTPUT); //quạt

pinMode(mayLanh, OUTPUT); //máy lạnh

dht.begin();

}

void loop()

{

thietbi();

den();

quatVaMayLanh();

sensor();

}

void den(){ //đồng bộ nút nhấn cho 2 bóng đèn

if(digitalRead(relay1)==HIGH){ //nếu đèn led ngủ sáng

if(digitalRead(button\_led\_1)==LOW){//nếu nút nhấn ấn

digitalWrite(relay1, LOW); //tắt đèn led ngủ

}

else{}

}

else{ //nếu đèn tắt

if(digitalRead(button\_led\_1)==LOW){ //nếu nút ấn

digitalWrite(relay1, HIGH); //đèn sáng

}

else{}

}

if(digitalRead(relay2)==HIGH){ //nếu đèn led ngủ sáng

if(digitalRead(button\_led\_2)==LOW){//nếu nút nhấn ấn

digitalWrite(relay2, LOW); //tắt đèn led ngủ

}

else{}

}

else{ //nếu đèn tắt

if(digitalRead(button\_led\_2)==LOW){ //nếu nút ấn

digitalWrite(relay2, HIGH); //đèn sáng

}

else{}

}

}

void quatVaMayLanh(){ //đồng bộ nút nhấn cho quạt và máy lạnh

if(digitalRead(quat)==HIGH){ //nếu đèn led ngủ sáng

if(analogRead(button\_quat)==LOW){//nếu nút nhấn ấn

digitalWrite(quat, LOW); //tắt đèn led ngủ

}

else{}

}

else{ //nếu đèn tắt

if(analogRead(button\_quat)==LOW){ //nếu nút ấn

digitalWrite(quat, HIGH); //đèn sáng

}

else{}

}

if(digitalRead(mayLanh)==HIGH){ //nếu đèn led ngủ sáng

if(analogRead(button\_mayLanh)==LOW){//nếu nút nhấn ấn

digitalWrite(mayLanh, LOW); //tắt đèn led ngủ

}

else{}

}

else{ //nếu đèn tắt

if(analogRead(button\_mayLanh)==LOW){ //nếu nút ấn

digitalWrite(mayLanh, HIGH); //đèn sáng

}

else{}

}

}

void sensor(){

sdata1 = analogRead(sensor1); //đọc cảm biến gas

sdata2 = dht.readHumidity();

sdata3 = dht.readTemperature();

if(sdata1 > 450) //co gas hoac co khoi

{

digitalWrite(BUZZER,LOW);

delay(1000);

}

else{ //không có khói

digitalWrite(BUZZER,HIGH);

}

delay(1000);

cdata = cdata + sdata1+","+sdata2+","+sdata3; // comma will be used a delimeter

Serial.println(cdata);

nodemcu.println(cdata); //gửi về cho nodemcu, về blynk server

delay(1000); // 100 milli seconds

cdata = "";

}

void thietbi(){

if ( nodemcu.available() > 0 )

{

data = nodemcu.parseInt(); //đọc dữ liệu từ blynk

delay(100);

Serial.println(data);

if(data == 10){

digitalWrite(relay1,LOW);

}

if(data == 12){

digitalWrite(relay2,LOW);

}

if(data == 11){

digitalWrite(relay1,HIGH);

}

if(data == 13){

digitalWrite(relay2,HIGH);

}

if(data == 1210){

digitalWrite(relay1,LOW);

digitalWrite(relay2,LOW);

}

if(data == 1012){

digitalWrite(relay1,LOW);

digitalWrite(relay2,LOW);

}

if(data == 1311){

digitalWrite(relay1,HIGH);

digitalWrite(relay2,HIGH);

}

if(data == 1113){

digitalWrite(relay1,HIGH);

digitalWrite(relay2,HIGH);

}

if(data == 1013){

digitalWrite(relay1,LOW);

digitalWrite(relay2,HIGH);

}

if(data == 1310){

digitalWrite(relay1,LOW);

digitalWrite(relay2,HIGH);

}

if(data == 1112){

digitalWrite(relay1,HIGH);

digitalWrite(relay2,LOW);

}

if(data == 1211){

digitalWrite(relay1,HIGH);

digitalWrite(relay2,LOW);

}

if(data == 1010){

digitalWrite(relay1,LOW);

}

if(data == 1111){

digitalWrite(relay1,HIGH);

}

if(data == 1212){

digitalWrite(relay2,LOW);

}

if(data == 1313){

digitalWrite(relay2,HIGH);

}

if(data == 14){

digitalWrite(quat,LOW);

}

if(data == 15){

digitalWrite(quat,HIGH);

}

if(data == 16){

digitalWrite(mayLanh,LOW);

}

if(data == 17){

digitalWrite(mayLanh,HIGH);

}

if(data == 1416){

digitalWrite(quat,LOW);

digitalWrite(mayLanh,LOW);

}

if(data == 1614){

digitalWrite(quat,LOW);

digitalWrite(mayLanh,LOW);

}

if(data == 1517){

digitalWrite(quat,HIGH);

digitalWrite(mayLanh,HIGH);

}

if(data == 1715){

digitalWrite(quat,HIGH);

digitalWrite(mayLanh,HIGH);

}

if(data == 1417){

digitalWrite(quat,LOW);

digitalWrite(mayLanh,HIGH);

}

if(data == 1714){

digitalWrite(quat,LOW);

digitalWrite(mayLanh,HIGH);

}

if(data == 1516){

digitalWrite(quat,HIGH);

digitalWrite(mayLanh,LOW);

}

if(data == 1615){

digitalWrite(quat,HIGH);

digitalWrite(mayLanh,LOW);

}

if(data == 1414){

digitalWrite(quat,LOW);

}

if(data == 1515){

digitalWrite(quat,HIGH);

}

if(data == 1616){

digitalWrite(mayLanh,LOW);

}

if(data == 1717){

digitalWrite(mayLanh,HIGH);

}

}

}