

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA ĐIỆN - ĐIỆN TỬ



**HCMUTE**

**BÁO CÁO CUỐI KỲ**

**MÔN HỌC: CƠ SỞ VÀ ỨNG DỤNG IOT**

**HỆ THỐNG NHÀ THÔNG MINH**

**GVHD:** Th.S Trương Quang Phúc

**SVTH:**

- |    |                  |          |
|----|------------------|----------|
| 1. | Nguyễn Phi Hùng  | 20119075 |
| 2. | Trần Thanh Tùng  | 20119386 |
| 3. | Nguyễn Võ Anh Tú | 20119381 |
| 4. | Lương Công Hoàn  | 20119100 |
| 5. | Lê Văn Thành     | 20119375 |

Thành phố Hồ Chí Minh, tháng 12 năm 2022

## PHÂN CÔNG NHIỆM VỤ

### 1. Thông tin sinh viên và phân công nhiệm vụ

| Họ và tên        | Nhiệm vụ   | Email                          |
|------------------|--|--------------------------------|
| Nguyễn Phi Hùng  | Thiết kế và tích hợp chức năng điều khiển quạt, đèn.     | 20119075@student.hcmute.edu.vn |
| Trần Thanh Tùng  | Thiết kế và tích hợp chức năng khóa cửa bằng RFID.       | 20119386@student.hcmute.edu.vn |
| Nguyễn Võ Anh Tú | Thiết kế và xây dựng ứng dụng điều khiển.                | 20119381@student.hcmute.edu.vn |
| Lương Công Hoàn  | Thiết kế và tích hợp chức năng thu thập nhiệt độ, độ ẩm. | 20119100@student.hcmute.edu.vn |
| Lê Văn Thành     | Thiết kế và tích hợp chức năng thu thập dữ liệu khí gas. | 20119375@student.hcmute.edu.vn |

### 2. Thông tin đề tài

Tên của đề tài: **HỆ THỐNG NHÀ THÔNG MINH**

Mục đích của đề tài: Thiết kế, lắp đặt và ứng dụng IOT vào các thiết bị trong nhà.

Báo cáo cuối kỳ môn học Cơ sở và ứng dụng IOT được thực hiện tại khoa Điện – Điện tử, Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh.

### 3. Các nhiệm vụ tổng quát của đề tài

- Tìm hiểu lý thuyết ESP32, RFID, LCD, cảm biến DHT11, MQ4, module L298N.
- Tìm hiểu và thiết kế ứng dụng điều khiển thông qua MIT App Inventor.
- Tích hợp các chức năng và xây dựng mô hình nhà thông minh.

### 4. Lời cam đoan của sinh viên

Sinh viên thực hiện đề tài cam đoan số liệu và kết quả nghiên cứu trong báo cáo này là trung thực. Mọi sự tham khảo sử dụng trong báo cáo đều được trích dẫn các nguồn tài liệu trong báo cáo và danh mục tài liệu tham khảo.

Thành phố Hồ Chí Minh, tháng 12 năm 2022

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên : Nguyễn Phi Hùng

MSSV: 20119075

Trần Thanh Tùng

MSSV: 20119386

Nguyễn Võ Anh Tú

MSSV: 20119381

Lương Công Hoàn

MSSV: 20119100

Lê Văn Thành

MSSV: 20119375

Tên đề tài: Hệ thống nhà thông minh

Giáo viên hướng dẫn: Th.S Trương Quang Phúc

## NHẬN XÉT

[illegible]

## **LỜI CẢM ƠN**

Để hoàn thành báo cáo đề tài cuối kỳ này, nhóm chúng em xin gửi lời cảm ơn chân thành đến thầy Trương Quang Phúc, giảng viên bộ môn Cơ sở dữ liệu và ứng dụng IOT. Nhờ những tiết học và các bài Lab của thầy, chúng em đã tích lũy được nhiều kiến thức về một hệ thống IOT cơ bản, được làm quen và thực hành thu thập dữ liệu từ các loại cảm biến, lưu trữ dữ liệu trên Firebase và điều khiển thiết bị thông qua các platform. Từ đó tạo tiền đề vững chắc để chúng em có thể tìm hiểu và thực hiện tích hợp các cảm biến, các ứng dụng thông minh lên mô hình. Và hơn thế nữa, nhờ thầy đã tạo cơ hội để chúng em mới có thể tiếp cận và thực hiện các đề tài, thông qua đó chúng em có thể hiểu rõ hơn, nắm chắc hơn về kiến thức đã được học trên lớp.

Do vốn kiến thức còn hạn hẹp, bài báo cáo không tránh khỏi những thiếu sót, chúng em rất mong nhận được ý kiến đóng góp của thầy để chúng em có thể rút kinh nghiệm, hoàn thiện và thực hiện tốt hơn cho những đề tài tiếp theo.

Nhóm chúng em xin chân thành cảm ơn.

## **Mục Lục**

|  |              |
|--|--------------|
| <b>DANH MỤC HÌNH .....</b>   | <b>.....</b> |
| <b>DANH MỤC BẢNG .....</b>   | <b>.....</b> |
| <b>CHƯƠNG 1 GIỚI THIỆU.....</b>  | <b>1</b>     |
| <b>1.1 Tổng quan đề tài thực hiện .....</b>                                    | <b>1</b>     |
| <b>1.1.1 Đặt vấn đề .....</b>  | <b>1</b>     |
| <b>1.1.2 Mục tiêu .....</b>  | <b>2</b>     |
| <b>1.1.3 Nhiệm vụ cụ thể của đề tài .....</b>                                  | <b>2</b>     |
| <b>1.2 Các sản phẩm hoặc hệ thống tương tự .....</b>                           | <b>3</b>     |
| <b>1.2.1 Hệ thống giám sát điều khiển hệ thống lạnh IOT .....</b>              | <b>3</b>     |
| <b>1.2.2 Hệ thống giám sát và điều khiển thiết bị từ xa cho ngôi nhà .....</b> | <b>3</b>     |
| <b>1.2.3 Phương pháp và hệ thống trong công nghệ nhà thông minh .....</b>      | <b>4</b>     |
| <b>CHƯƠNG 2 CƠ SỞ LÝ THUYẾT .....</b>  | <b>5</b>     |
| <b>2.1 ESP32.....</b>  | <b>5</b>     |
| <b>2.1.1 Tổng quan.....</b>  | <b>5</b>     |
| <b>2.1.2 Thông số kỹ thuật của ESP32.....</b>                                  | <b>5</b>     |
| <b>2.1.3 Sơ đồ khối và sơ đồ chân của ESP32 .....</b>                          | <b>8</b>     |
| <b>2.1.4 Các loại chip ESP32 .....</b>   | <b>13</b>    |
| <b>2.1.5 Module ESP32-WROOM-32 .....</b>                                       | <b>14</b>    |
| <b>2.2 RFID .....</b>  | <b>15</b>    |
| <b>2.2.1 Công nghệ RFID .....</b>  | <b>15</b>    |
| <b>2.2.2 Module đọc thẻ RFID RC522 và thẻ RFID (RFID tag) .....</b>            | <b>15</b>    |
| <b>2.2.3 Nguyên lý hoạt động.....</b>  | <b>17</b>    |
| <b>2.3 LCD.....</b>  | <b>18</b>    |

|   |           |
|---|-----------|
| 2.3.1 LCD là gì? .....                          | 18        |
| 2.3.2 Màn hình LCD 16x2 và module I2C.....      | 18        |
| 2.4 Servo SG90 .....                            | 22        |
| 2.5 L298N.....                                  | 23        |
| 2.6 Buzzer .....                                | 24        |
| 2.7 Cảm biến DHT11 .....                        | 26        |
| 2.8 Cảm biến MQ4.....                           | 28        |
| 2.9 App Inventor .....                          | 30        |
| 2.9.1 Giới thiệu chung .....                    | 30        |
| 2.9.2 Ưu nhược điểm của App Inventor .....      | 30        |
| <b>CHƯƠNG 3 THIẾT KẾ .....</b>                  | <b>32</b> |
| 3.1 Thiết kế tổng quan.....                     | 32        |
| 3.2 Thiết kế mặt ngoài .....                    | 33        |
| 3.2.1 Sơ đồ khối.....                           | 34        |
| 3.2.2 Sơ đồ nối dây .....                       | 35        |
| 3.2.3 Lưu đồ code.....                          | 37        |
| 3.2.4 Giải thích code .....                     | 38        |
| 3.3 Thiết kế mặt trong .....                    | 42        |
| 3.3.1 Sơ đồ khối.....                           | 43        |
| 3.3.2 Khối thu thập nhiệt độ, độ ẩm DHT11 ..... | 43        |
| 3.3.3 Khối thu thập khí gas MQ4 .....           | 46        |
| 3.3.4 Khối điều khiển Led .....                 | 49        |
| 3.3.5 Khối điều khiển quạt.....                 | 55        |

|   |    |
|---|----|
| 3.4 Chương trình chính .....                          | 60 |
| 3.4.1 Lưu đồ.....                                     | 60 |
| 3.4.2 Giải thích chương trình điều khiển chính .....  | 61 |
| 3.5 Ứng dụng điều khiển .....                         | 63 |
| 3.5.1 Tính năng .....                                 | 63 |
| 3.5.2 Chức năng của từng khối.....                    | 64 |
| CHƯƠNG 4: KẾT QUẢ .....                               | 72 |
| 4.1 Chức năng điều khiển khóa cửa dùng thẻ RFID ..... | 72 |
| 4.2 Chức năng đo độ ẩm và nhiệt độ .....              | 76 |
| 4.3 Chức năng đo khí gas .....                        | 77 |
| 4.4 Chức năng điều khiển đèn.....                     | 78 |
| 4.5 Chức năng điều khiển quạt.....                    | 80 |
| CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....           | 81 |
| 5.1 Tổng kết đề tài.....                              | 81 |
| 5.2 Định hướng phát triển.....                        | 82 |
| Tài liệu tham khảo.....                               | 83 |

## DANH MỤC HÌNH

|  |    |
|--|----|
| Hình 2. 1: Sơ đồ khối của ESP32 .....            | 8  |
| Hình 2. 2: Sơ đồ chân của ESP32.....             | 9  |
| Hình 2.3: ESP32-WROOM-32.....                    | 14 |
| Hình 2.4: RFID RC522.....                        | 15 |
| Hình 2.5: Sơ đồ chân của RFID RC522 .....        | 16 |
| Hình 2. 6: Thẻ RFID.....                         | 17 |
| Hình 2. 7: Nguyên lý hoạt động của RFID.....     | 17 |
| Hình 2. 8: Màn hình LCD 16x2.....                | 19 |
| Hình 2. 9: Module I2C.....                       | 21 |
| Hình 2. 10: Sơ đồ kết nối ESP32 với .....        | 21 |
| Hình 2. 11: Hình ảnh thực tế Servo.....          | 22 |
| Hình 2. 12: Sơ đồ nối chân.....                  | 22 |
| Hình 2. 13: Hình ảnh thực tế L298N.....          | 23 |
| Hình 2. 14: Sơ đồ chân L298N.....                | 24 |
| Hình 2. 15: Sơ đồ chân Buzzer.....               | 25 |
| Hình 2. 16: Sơ đồ chân của DHT11 .....           | 27 |
| Hình 2. 17: Hình ảnh sơ đồ chân MQ4 .....        | 28 |
| Hình 2. 18: Sơ đồ mạch của MQ4.....              | 29 |
| Hình 2. 19: Biểu tượng của App Inventor.....     | 30 |
|  |    |
| Hình 3. 1: Thiết kế tổng quan.....               | 32 |
| Hình 3. 2: Thiết kế mặt ngoài mô hình.....       | 33 |
| Hình 3. 3: Sơ đồ khối mặt ngoài.....             | 34 |
| Hình 3. 4: Sơ đồ nối dây mặt ngoài.....          | 35 |
| Hình 3. 5: Lưu đồ code điều khiển khóa cửa ..... | 37 |
| Hình 3. 6: Thiết kế mặt trong mô hình.....       | 42 |
| Hình 3. 7: Sơ đồ khối mặt trong.....             | 43 |



|   |    |
|---|----|
| Hình 3. 8: Sơ đồ khối thu thập nhiệt độ, độ ẩm.....     | 43 |
| Hình 3. 9: Sơ đồ nối dây DHT11.....                     | 44 |
| Hình 3. 10: Lưu đồ khối thu thập nhiệt độ, độ ẩm .....  | 44 |
| Hình 3. 11: Sơ đồ khối thu thập khí gas .....           | 46 |
| Hình 3. 12: Sơ đồ nối dây khối thu thập khí gas .....   | 47 |
| Hình 3. 13: Lưu đồ khối thu thập khí gas.....           | 47 |
| Hình 3. 14: Sơ đồ khối điều khiển led.....              | 49 |
| Hình 3. 15: Sơ đồ nối dây khối điều khiển led.....      | 50 |
| Hình 3. 16: Lưu đồ chương trình điều khiển led 1 .....  | 51 |
| Hình 3. 17: Lưu đồ chương trình điều khiển led 2 .....  | 52 |
| Hình 3. 18: Sơ đồ khối điều khiển quạt.....             | 55 |
| Hình 3. 19: Sơ đồ nối dây.....                          | 56 |
| Hình 3. 20: Lưu đồ chương trình điều khiển quạt.....    | 57 |
| Hình 3. 21: Lưu đồ chương trình chính.....              | 60 |
| Hình 3. 22: Các tính năng của ứng dụng điều khiển ..... | 63 |
|   |    |
| Hình 4. 1: Thiết kế mặt ngoài.....                      | 72 |
| Hình 4. 2: Quét thẻ đúng, mở cửa .....                  | 73 |
| Hình 4. 3: Quét thẻ đúng, đóng cửa.....                 | 74 |
| Hình 4. 4: Quét thẻ sai.....                            | 75 |
| Hình 4. 5: Chức năng đo độ ẩm và nhiệt độ.....          | 76 |
| Hình 4. 6: Chức năng đo khí gas .....                   | 77 |
| Hình 4. 7: Cảnh báo khí gas .....                       | 77 |
| Hình 4. 8: Bật đèn 1.....                               | 78 |
| Hình 4. 9: Bật đèn 2.....                               | 78 |
| Hình 4. 10: Tắt đèn 1 .....                             | 79 |
| Hình 4. 11: Tắt đèn 2.....                              | 79 |
| Hình 4. 12: Bật quạt.....                               | 80 |

|  |    |
|--|----|
| Hình 4. 13: Điều chỉnh tốc độ và tắt quạt..... | 80 |
|--|----|

## **DANH MỤC BẢNG**

|   |    |
|---|----|
| Bảng 2. 1: Chi tiết các chân ESP32.....                   | 9  |
| Bảng 2. 2: Phân biệt các loại chip ESP32 .....            | 13 |
| Bảng 2. 3: Chức năng từng chân của màn hình LCD 16x2..... | 19 |
| Bảng 2. 4: Sơ đồ kết nối với ESP32 .....                  | 21 |
| Bảng 2. 5: Chức năng từng chân của L298N .....            | 24 |
| Bảng 2. 6: Chức năng các chân DHT11 .....                 | 26 |
| Bảng 2. 7: Chức năng các chân DHT11 .....                 | 27 |
| Bảng 2. 8: Chi tiết các chân MQ4 .....                    | 28 |
|   |    |
| Bảng 3. 1 : Bảng kết nối dây mặt ngoài .....              | 35 |
| Bảng 3. 2: Kết nối chân khối thu thập khí gas .....       | 46 |
| Bảng 3. 3: Kết nối Led với ESP32 .....                    | 49 |
| Bảng 3. 4: Kết nối chân L298N với ESP32.....              | 55 |

## CHƯƠNG 1 GIỚI THIỆU

### 1.1 Tổng quan đề tài thực hiện

#### 1.1.1 Đặt vấn đề

Trong những năm gần đây, đi cùng với sự phát triển mạnh mẽ của các lĩnh vực, thiết bị điện tử, Internet of Things (IoT) hay còn gọi là kết nối vạn vật đang ngày càng phát triển và mở rộng trên nhiều khu vực.

Với sự phát triển của IoT như trên, những ứng dụng sản phẩm cũng đi kèm theo đó mà được tạo ra. Chúng không chỉ được áp dụng cho quá trình sản xuất, điều khiển thiết bị hệ thống công nghiệp, cho các mô hình kinh doanh, dịch vụ,... mà còn cho các nhu cầu sinh hoạt của con người. Một ứng dụng đang cực kỳ phổ biến hiện nay, được tạo ra để phục vụ cho nhu cầu cơ bản cũng như nâng cao của con người có thể kể đến chính là mô hình nhà thông minh.

Mô hình nhà thông minh được hiểu tổng quát là một hệ thống giúp kết nối các thiết bị cơ bản trong một ngôi nhà bình thường lại với nhau, đồng thời thêm các thiết bị nâng cao nhằm phục vụ thêm cho các yêu cầu phức tạp hơn của người dùng. Thông qua chúng, từ một ngôi nhà bình thường với các thiết bị rời rạc không liên kết với nhau, giờ đây chúng được liên kết cũng như thêm nhiều tiện ích để giúp người dùng sinh hoạt trong ngôi nhà tiện lợi hơn, đáp ứng các nhu cầu một cách nhanh chóng.

Không chỉ trong việc sinh hoạt bình thường trong ngôi nhà, người dùng còn có thể tương tác với các thiết bị ở khoảng cách xa, thiết lập thêm các chế độ bảo vệ như cảm biến khí metan, cảm biến trộm,...

Với tính năng như trên, nhà thông minh dần hiện hữu nhiều hơn trong đời sống xã hội, được xem như là một đề tài nghiên cứu phổ biến với các nhà kỹ sư, cũng như học sinh sinh viên.

### **1.1.2 Mục tiêu**

Mục tiêu của đề tài này được hướng đến như sau:

- Điều khiển các thiết bị cơ bản như đèn, quạt thông qua nút nhấn trên mô hình và ứng dụng.
- Hiển thị trạng thái các thiết bị lên ứng dụng.
- Thu thập và hiển thị nhiệt độ, độ ẩm lên ứng dụng
- Mở khoá cửa bằng RFID thông qua đầu đọc RC522
- Hiển thị đóng, mở cửa, đúng sai của thẻ RFID qua màn LCD 16X2 và thông báo qua 2 led, 1 buzzer.
- Thu thập và hiển thị cảm biến khí ga lên ứng dụng, tiến hành cảnh báo khi khí ga vượt ngưỡng thông qua buzzer trên mô hình và ứng dụng trên thiết bị.

### **1.1.3 Nhiệm vụ cụ thể của đề tài**

- Tìm hiểu về lý thuyết ESP32, RFID, LCD, cảm biến DHT11, MQ4, module L298N.
- Tìm hiểu và thiết kế ứng dụng điều khiển thông qua MIT App Inventor.
- Tích hợp các chức năng:
  - + Chức năng điều khiển khóa cửa dùng thẻ RFID
  - + Chức năng điều khiển đèn
  - + Chức năng điều khiển quạt
  - + Chức năng độ nhiệt độ, độ ẩm
  - + Chức năng đo khí ga
- Xây dựng mô hình nhà thông minh.

## **1.2 Các sản phẩm hoặc hệ thống tương tự**

### **1.2.1 Hệ thống giám sát điều khiển hệ thống lạnh IOT**

Nguyên lý hoạt động của hệ thống giám sát điều khiển hệ thống lạnh bằng công nghệ IOT được thể hiện như sau:

Các cảm biến nhiệt độ, độ ẩm được lắp trong các kho lạnh sao cho phù hợp với diện tích từng kho. Ngoài ra, tại điểm cung cấp nguồn điện cho các thiết bị của kho lạnh sẽ lắp các đèn, còi thông báo sự cố. Khi nhiệt độ, độ ẩm trong kho thay đổi, các cảm biến sẽ ghi nhận dữ liệu này và đưa dữ liệu này đến bộ điều khiển trung tâm. Tại bộ điều khiển trung tâm, dữ liệu sẽ được thu nhận và tiến hành xử lý. Sau đó, những dữ liệu này sẽ được gửi để hiển thị lên các ứng dụng có trong thiết bị như điện thoại, máy tính do các kỹ sư sử dụng, từ đó họ có thể giám sát hệ thống một cách gián tiếp nhưng vẫn đảm bảo độ tin cậy [1].

Trường hợp có sự cố xảy ra do nhiệt độ hoặc độ ẩm trong kho vượt ngưỡng quy định, các thiết bị thông báo sự cố như đèn, còi sẽ được bật để cảnh báo, đồng thời, bộ điều khiển trung tâm sẽ ghi nhận lại sự việc này, sau đó gửi cảnh báo đến ứng dụng có trong các thiết bị của các kỹ sư giám sát, nhờ vậy họ có thể phát hiện cảnh báo, tiến hành đưa ra phương án xử lý kịp thời mà không cần phải giám sát trực tiếp kho lạnh, từ đó đảm bảo hiệu quả tối đa trong việc vận hành hệ thống [1].

### **1.2.2 Hệ thống giám sát và điều khiển thiết bị từ xa cho ngôi nhà**

Hệ thống giám sát và điều khiển thiết bị từ xa cho ngôi nhà sẽ bao gồm các hệ thống sau: Hệ thống quản lý chiếu sáng giúp quản lý các thiết bị như đèn, đèn ngủ, đèn trang trí,... Hệ thống kiểm soát ra vào giúp tăng được độ bảo vệ cho ngôi nhà. Hệ thống cảm biến và báo động, báo cháy nhằm mục đích giúp gia chủ kiểm soát các thông số có trong ngôi nhà cũng như đưa ra phương án xử lý kịp thời với các sự cố. Hệ thống các công tắc và bảng hiển thị sẽ cung cấp thông tin cũng như nhận lệnh điều khiển từ gia chủ. Cuối cùng là hệ thống mạng, xử lý trung tâm giúp các hệ thống khác liên kết lại với nhau và vận hành một cách nhịp nhàng [2].

### **1.2.3 Phương pháp và hệ thống trong công nghệ nhà thông minh**

Bài nghiên cứu: “Applications, Systems and Methods in Smart Home Technology: A Review”. Rosslin John Robles and Tai-hoon Kim (2010) được đăng bởi tạp chí International Journal of Advanced Science and Technology.

Trong ngôi nhà bình thường hiện nay, các thiết bị thường được điều khiển bằng các công tắc hoặc một số là thông qua các thiết bị điều khiển từ xa. Tuy nhiên, việc dùng công tắc thông qua các đường dây điện điều khiển các thiết bị thường không phải lúc nào cũng ổn định, một số trường hợp có thể phát sinh sự cố do quá trình cấp nguồn cho thiết bị thông qua dây. Ngày nay, một số hệ thống đã dần sử dụng sóng vô tuyến thay cho hệ thống dây điện để điều khiển thiết bị, có thể kể đến như WiFi hoặc bluetooth. Ngoài ra, cũng có hai mạng vô tuyến nổi bật trong mảng tự động hoá cho ngôi nhà là ZigBee và Z-Wave [3].

Đi kèm với các công nghệ trên, một số xu hướng đang nổi lên trong lĩnh vực nhà thông minh có thể kể đến hiện nay như là hệ thống đỗ xe tự động, hệ thống giám sát nhà thông qua các thiết bị máy ảnh và micro,.. [3]

## CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

### 2.1 ESP32

#### 2.1.1 Tổng quan

ESP32 là một dòng các vi điều khiển tích hợp Wi-Fi và Bluetooth. Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng. Chip ESP32 được thiết kế bằng công nghệ 40 nm của TSMC (một tập đoàn chuyên về chế tạo chất bán dẫn lớn nhất thế giới) để có thể đạt công suất và hiệu suất một cách tốt nhất, nhờ đó thể hiện sự mạnh mẽ, tính linh hoạt và độ tin cậy trong nhiều ứng dụng khác nhau [4].

Dòng vi điều khiển EPS32 qua các năm đã có các phiên bản khác nhau như ESP32-D0WDQ6, ESP32-D0WD, ESP32-D2WD, và ESP32-S0WD [4].

#### 2.1.2 Thông số kỹ thuật của ESP32

##### 2.1.2.1 Bộ xử lý trung tâm – CPU [4]

EPS32 sử dụng một hoặc hai bộ vi xử lý công suất thấp Tensilica Xtensa LX6 tùy theo loại ESP32 áp dụng lõi đơn hay lõi kép.

Đặc trưng:

- Được hỗ trợ để hoạt động trong tần số 240MHz (160 MHz cho ESP32-S0WD và ESP32-D2WD) và hoạt động ở tối đa 600 MIPS (200 MIPS cho ESP32-S0WD).
- Hỗ trợ bộ xử lý dấu phẩy động (FPU).
- Tập lệnh 16/24-bit cung cấp mật độ mã cao, lượng không gian mà một chương trình thực thi chiếm dụng ít đi.
- Hỗ trợ các lệnh xử lý tín hiệu số (DSP), chẳng hạn như bộ nhân 32 bit, bộ chia 32 bit và MAC 40 bit.
- Hỗ trợ 32 vector ngắt từ khoảng 70 nguồn ngắt.

##### 2.1.2.2 Bộ nhớ trong [4]

Bộ nhớ trong của ESP32 bao gồm:



- 448 KB ROM để khởi động và các chức năng cốt lõi.
- 520 KB SRAM trên chip dành cho dữ liệu và các lệnh.
- 16 KB SRAM trong bộ nhớ RTC dùng để lưu trữ dữ liệu và truy cập bởi bộ xử lý xung tâm trong chế độ ngủ sâu.
- Bộ nhớ Flash.

### **2.1.2.3 Bộ nhớ Flash và SRAM ngoài [4]**

ESP32 hỗ trợ nhiều chip Flash QSPI và chip SRAM bên ngoài.

ESP32 có thể truy cập đến Flash QSPI bên ngoài và SRAM thông qua bộ đệm tốc độ cao.

### **2.1.2.4 Hệ thống xung nhịp và bộ định thời [4]**

Hệ thống xung nhịp bao gồm:

- Bộ tạo dao động 8 MHz bên trong.
- Bộ tạo dao động RC.
- Bộ tạo dao động tinh thể 2 MHz ~ 60 MHz bên ngoài.
- Bộ tạo dao động tinh thể 32 kHz bên ngoài cho RTC
- Bộ định thời trong ESP32 có 2 nhóm:
  - Bộ định thời 64 bits: Có 4 bộ định thời 64 bits được nhúng vào ESP32.
  - Bộ định thời giám sát (Watchdog timers): ESP32 có ba bộ định thời giám sát, gồm 2 bộ định thời giám sát chính và một bộ định thời giám sát trong RTC. Bộ định thời giám sát có mục đích phục hồi từ một lỗi không lường trước của chương trình.

### **2.1.2.5 Wifi [4]**

ESP32 triển khai giao thức TCP/IP và giao thức MAC Wi-Fi 802.11 b/g/n.

Sóng vô tuyến và băng tần của Wifi ESP32 hỗ trợ các tính năng sau:

- 802.11b/g/n.
- 802.11n MCS0-7 in both 20 MHz and 40 MHz bandwidth.
- 802.11n MCS32 (RX).
- Tốc độ truyền dữ liệu lên đến 150Mbps

- Hỗ trợ đa dạng ăng-ten để có thể chọn ăng-ten tốt nhất để giảm thiểu ảnh hưởng của hiện tượng mờ dần kênh.

#### **2.1.2.6 Bluetooth [4]**

ESP32 tích hợp bộ điều khiển và băng tần Bluetooth, thực hiện các giao thức như điều chế/giải điều chế, xử lý gói, xử lý luồng bit.

Bluetooth của ESP32 có các đặc trưng sau:

- Tuân thủ các thông số kỹ thuật Bluetooth v4.2 BR/EDR và BLE.
- Công suất phát +12 dBm.
- Kiểm soát công suất nâng cao.
- Máy thu NZIF với độ nhạy -97 dBm BLE.
- Bộ điều khiển chế độ kép Bluetooth 4.2 BR/EDR BLE.
- UART HCI tốc độ cao, lên đến 4 Mbps.

#### **2.1.2.7 Ngoại vi [4]**

34 × GPIO có thể lập trình

12-bit SAR ADC lên đến 18 kênh.

Bộ giải mã 2 × 8-bit.

10 cảm biến điện dung.

4 SPI, 2 I<sup>2</sup>S, 2 I<sup>2</sup>C, 3 UART.

1 host (SD/eMMC/SDIO), 1 slave (SDIO/SPI).

Giao diện Ethernet MAC với hỗ trợ DMA và IEEE 1588 chuyên dụng.

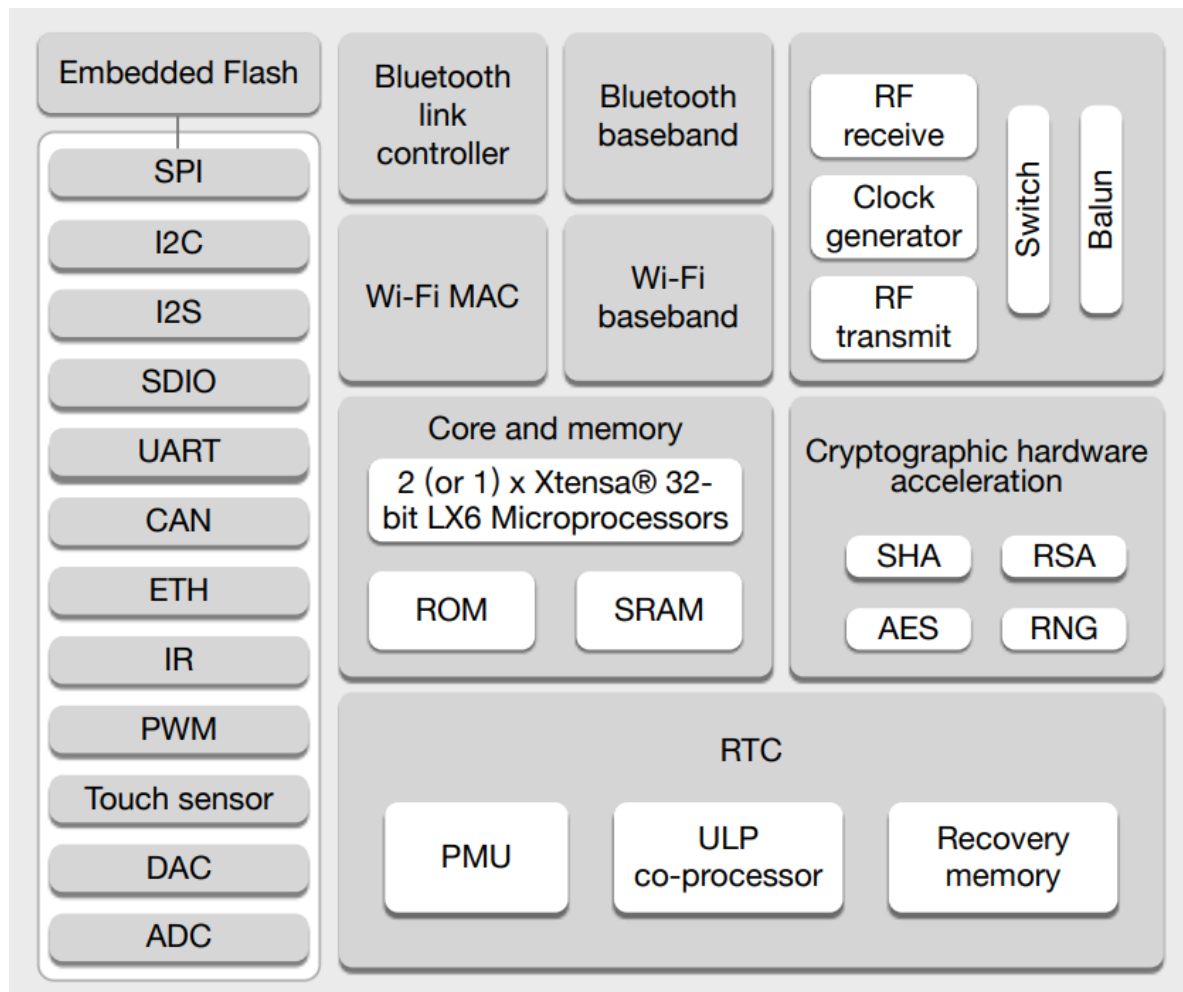
CAN 2.0.

Động cơ PWM.

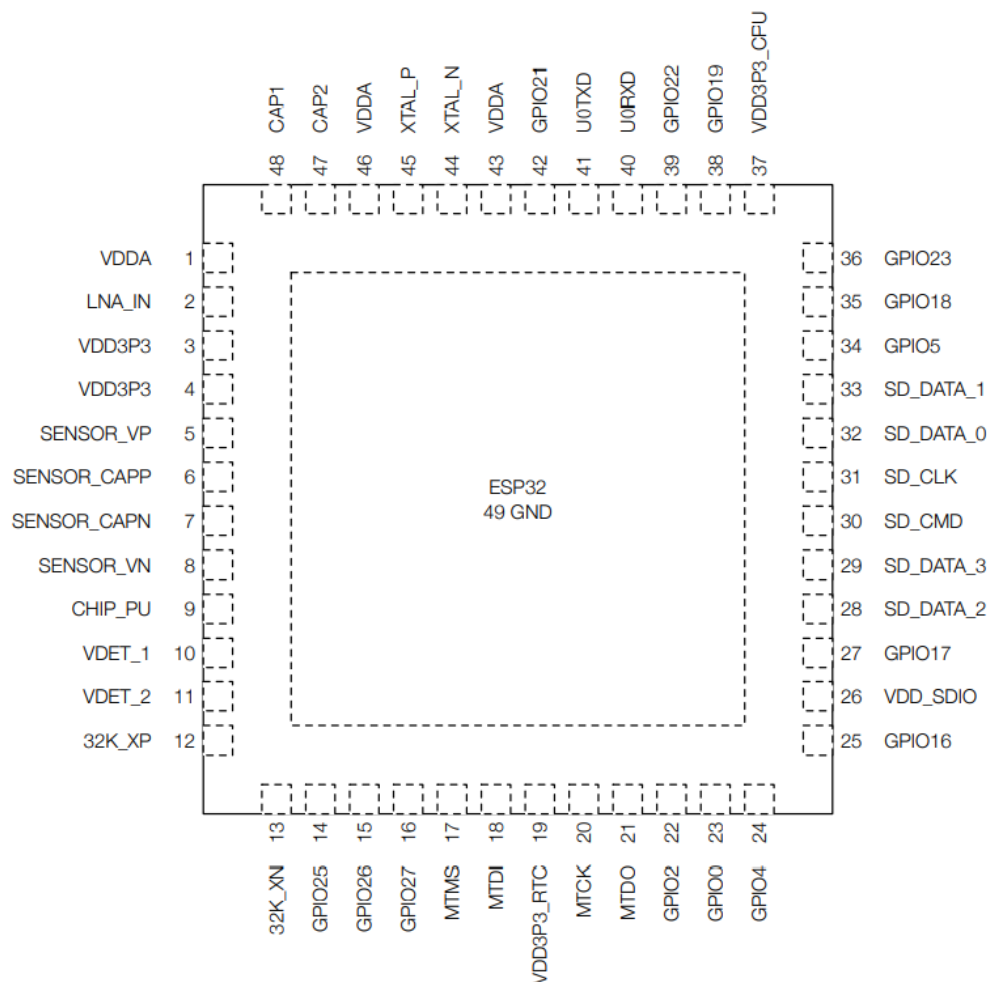
LED PWM lên đến 16 kênh.

Cảm biến từ.

### 2.1.3 Sơ đồ khối và sơ đồ chân của ESP32



Hình 2. 1: Sơ đồ khối của ESP32



Hình 2. 2: Sơ đồ chân của ESP32

Bảng 2. 1: Chi tiết các chân ESP32

| Tên         | Số | Loại         | Chức năng                           |
|-------------|----|--------------|-------------------------------------|
| VDDA        | 1  | Chân         | Analog power supply (2.3 V – 3.6 V) |
| LNA_IN      | 2  | Input/Output | Input và ouput RF                   |
| VDD3P3      | 3  | Chân         | Analog power supply (2.3 V – 3.6 V) |
| VDD3P3      | 4  | Chân         | Analog power supply (2.3 V – 3.6 V) |
| SENSOR_VP   | 5  | Input        | GPIO36, ADC1_CH0, RTC_GPIO0         |
| SENSOR_CAPP | 6  | Input        | GPIO37, ADC1_CH1, RTC_GPIO1         |

|             |    |              |  |
|-------------|----|--------------|--|
| SENSOR_CAPN | 7  | Input        | GPIO38, ADC1_CH2, RTC_GPIO2  |
| SENSOR_VN   | 8  | Input        | GPIO39, ADC1_CH3, RTC_GPIO3  |
| CHIP_PU     | 9  | Input        | Mức cao: Cho phép chip hoạt động.<br>Mức thấp: Ngưng cấp nguồn cho chip.                 |
| VDET_1      | 10 | Input        | GPIO34, ADC1_CH6, RTC_GPIO4  |
| VDET_2      | 11 | Input        | GPIO35, ADC1_CH7, RTC_GPIO   |
| 32K_XP      | 12 | Input/Output | GPIO32, ADC1_CH4, RTC_GPIO9,<br>TOUCH9, 32K_XP   |
| 32K_XN      | 13 | Input/Output | GPIO33, ADC1_CH5, RTC_GPIO8,<br>TOUCH8, 32K_XN   |
| GPIO25      | 14 | Input/Output | GPIO25, ADC2_CH8, RTC_GPIO6,<br>DAC_1, EMAC_RXD0   |
| GPIO26      | 15 | Input/Output | GPIO26, ADC2_CH9, RTC_GPIO7,<br>DAC_2, EMAC_RXD1   |
| GPIO27      | 16 | Input/Output | GPIO27, ADC2_CH7, RTC_GPIO17,<br>TOUCH7, EMAC_RX_DV                                      |
| MTMS        | 17 | Input/Output | GPIO14, ADC2_CH6, RTC_GPIO16,<br>TOUCH6, EMAC_TXD2, HSPICLK,<br>HS2_CLK, SD_CLK, MTMS    |
| MTDI        | 18 | Input/Output | GPIO12, ADC2_CH5, RTC_GPIO15,<br>TOUCH5, EMAC_TXD3, HSPIQ,<br>HS2_DATA2, SD_DATA2, MTDI  |
| VDD3P3_RTC  | 19 | Chân         | Cấp nguồn cho RTC IO(2.3 V – 3.6<br>V)   |
| MTCK        | 20 | Input/Output | GPIO13, ADC2_CH4, RTC_GPIO14,<br>TOUCH4, EMAC_RX_ER, HSPID,<br>HS2_DATA3, SD_DATA3, MTCK |

|           |    |              |  |
|-----------|----|--------------|--|
| MTDO      | 21 | Input/Output | GPIO13, ADC2_CH4, RTC_GPIO14, TOUCH4, EMAC_RX_ER, HSPID, HS2_DATA3, SD_DATA3, MTCK |
| GPIO2     | 22 | Input/Output | GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPIWP, HS2_DATA0, SD_DATA0                   |
| GPIO0     | 23 | Input/Output | GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPIWP, HS2_DATA0, SD_DATA0                   |
| GPIO4     | 24 | Input/Output | GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPIWP, HS2_DATA0, SD_DATA0                   |
| GPIO16    | 25 | Input/Output | GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT   |
| VDD_SDIO  | 26 | Chân         | Đưa ra ngoài nguồn 1.8 V hoặc giống với mức điện áp của VDD3P3_RTC                 |
| GPIO17    | 27 | Input/Output | GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180   |
| SD_DATA_2 | 28 | Input/Output | GPIO9, HS1_DATA2, U1RXD, SD_DATA2, SPIHD   |
| SD_DATA_3 | 29 | Input/Output | GPIO10, HS1_DATA3, U1TXD, SD_DATA3, SPIWP  |
| SD_CMD    | 30 | Input/Output | GPIO11, HS1_CMD, U1RTS, SD_CMD, SPICS0   |
| SD_CLK    | 31 | Input/Output | GPIO6, HS1_CLK, U1CTS, SD_CLK, SPICLK  |
| SD_DATA_0 | 32 | Input/Output | GPIO7, HS1_DATA0, U2RTS, SD_DATA0, SPIQ  |

|            |    |              |   |
|------------|----|--------------|---|
| SD_DATA_1  | 33 | Input/Output | GPIO8, HS1_DATA1, U2CTS, SD_DATA1, SPID                     |
| GPIO5      | 34 | Input/Output | GPIO5, HS1_DATA6, VSPICS0, EMAC_RX_CLK                      |
| GPIO18     | 35 | Input/Output | GPIO18, HS1_DATA7, VSPICLK                                  |
| GPIO23     | 36 | Input/Output | GPIO23, HS1_STROBE, VSPID                                   |
| VDD3P3_CPU | 37 | Chân         | Cấp nguồn cho CPU IO (1.8 V – 3.6 V)                        |
| GPIO19     | 38 | Input/Output | GPIO19, U0CTS, VSPIQ, EMAC_TXD0                             |
| GPIO22     | 39 | Input/Output | GPIO22, U0RTS, VSPIWP, EMAC_TXD1                            |
| U0RXD      | 40 | Input/Output | GPIO3, U0RXD, CLK_OUT2                                      |
| U0TXD      | 41 | Input/Output | GPIO3, U0RXD, CLK_OUT2                                      |
| GPIO21     | 42 | Input/Output | GPIO21, VSPIHD, EMAC_TX_EN                                  |
| VDDA       | 43 | Chân         | Analog power supply (2.3 V – 3.6 V)                         |
| XTAL_N     | 44 | Output       | Ngõ ra giao động thạch anh                                  |
| XTAL_P     | 45 | Input        | Ngõ vào giao động thạch anh                                 |
| VDDA       | 46 | Chân         | Analog power supply (2.3 V – 3.6 V)                         |
| CAP2       | 47 | Input        | Kết nối với tụ 3 nF và trở 20 k $\Omega$ song song với CAP1 |
| CAP1       | 48 | Input        | Kết nối với tụ 10 nF tới đất                                |
| GND        | 49 | Chân         | Nối đất   |

### 2.1.4 Các loại chip ESP32

Qua các năm, có nhiều loại chip ESP32 đã được chế tạo và đưa ra thị trường. Có thể phân biệt các loại chip trong bảng dưới đây [4]:

*Bảng 2. 2: Phân biệt các loại chip ESP32*

| Tên          | Số lõi xử lý | Bộ nhớ Flash            | Kết nối                              | Kích thước đóng gói |
|--------------|--------------|-------------------------|--------------------------------------|---------------------|
| ESP32-D0WDQ6 | 2            | Không                   | Wi-Fi b/g/n +<br>BT/BLE Dual<br>Mode | 6 mm × 6 mm         |
| ESP32-D0WD   | 2            | Không                   | Wi-Fi b/g/n +<br>BT/BLE Dual<br>Mode | 5 mm × 5 mm         |
| ESP32-D2WD   | 2            | Bộ nhớ Flash<br>16-Mbit | Wi-Fi b/g/n +<br>BT/BLE Dual<br>Mode | 5 mm × 5 mm         |
| ESP32-S0WD   | 1            | Không                   | Wi-Fi b/g/n +<br>BT/BLE Dual<br>Mode | 5 mm × 5 mm         |



### 2.1.5 Module ESP32-WROOM-32



*Hình 2.3: ESP32-WROOM-32*

ESP32-WROOM-32 là mô-đun tích hợp Wi-Fi, Bluetooth và Bluetooth LE, được thiết kế để hướng tới nhiều ứng dụng khác nhau, từ các mạng cảm biến công suất thấp đến các tác vụ khắc khe, phức tạp như mã hóa tiếng nói, truyền phát và giải mã âm nhạc [4].

ESP32-WROOM-32 bao gồm chip ESP32-D0WDQ6 là cốt lõi của mô-đun, tích hợp cùng với đó là các thiết bị ngoại vi khác như cảm biến điện dung, cảm biến từ, card SD, Ethernet, SPI, UART, I2S và I2C tốc độ cao, ngoài ra mô-đun còn tích hợp thêm một bộ nhớ Flash ngoài 4 MiB [4].

ESP32-WROOM-32 được công bố trên thị trường lần đầu tiên bởi công ty Espressif [4].

## 2.2 RFID

### 2.2.1 Công nghệ RFID

RFID (Radio Frequency Identification) hay còn gọi là công nghệ nhận dạng qua tần số vô tuyến, nó sử dụng trường điện từ để tự động nhận dạng và theo dõi các thẻ hỗ trợ công nghệ này được gắn vào đối tượng. Công nghệ này cho phép nhận biết các đối tượng thông qua hệ thống thu phát sóng radio, từ đó có thể giám sát, quản lý hoặc lưu vết từng đối tượng. Như trường hợp trên, bạn đi siêu thị, bỏ hàng vào xe đẩy và chỉ đơn giản đẩy thẳng xe qua cổng giám sát. Một thiết bị tự động nhận dạng từng món hàng bạn mua và tự động trừ vào tài khoản thanh toán của bạn. Đó chỉ là một trong rất nhiều ứng dụng tiện ích của công nghệ RFID [5].

Hệ thống RFID gồm hai thành phần chính: thẻ RFID (RFID tag) và đầu đọc (reader). Thẻ RFID có gắn chip silicon và ăng ten radio dùng để gắn vào đối tượng quản lý như sản phẩm, hàng hóa, động vật hoặc ngay cả con người... Thẻ RFID có kích thước rất nhỏ, cỡ vài cm. Bộ nhớ của con chip có thể chứa từ 96 đến 512 bit dữ liệu. Đầu đọc reader cho phép giao tiếp với thẻ RFID qua sóng radio ở khoảng cách trung bình từ 0,5-30 mét, từ đó truyền dữ liệu về hệ thống máy tính trung tâm [5].

### 2.2.2 Module đọc thẻ RFID RC522 và thẻ RFID (RFID tag)

#### 2.2.2.1 Module đọc thẻ RFID RC522



Hình 2.4: RFID RC522

Mạch RFID NFC 13.56MHZ RC522 sử dụng IC MFRC522 của NXP được sử dụng để đọc và ghi dữ liệu cho thẻ RFID NFC tần số 13.56Mhz, mạch có thiết kế nhỏ gọn được sử dụng rất phổ biến hiện nay với Arduino hoặc các loại Vi điều khiển khác trong các ứng dụng cần ghi, đọc thẻ RFID NFC [6].



Hình 2.5: Sơ đồ chân của RFID RC522

### Thông số kỹ thuật [6]

- Nguồn sử dụng: 3.3VDC.
- Dòng điện: 13~26mA.
- Tần số hoạt động: 13.56Mhz.
- Khoảng cách hoạt động: 0~60mm.
- Chuẩn giao tiếp: SPI.
- Tốc độ truyền dữ liệu: tối đa 10Mbit/s.
- Các loại card RFID hỗ trợ: mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire.
- Kích thước: 40mm × 60mm

#### 2.2.2.1 Thẻ RFID (RFID tag) [7]

Một thẻ RFID bao gồm 3 thành phần: một chip để lưu và xử lý thông tin, điều chế và giải điều chế tín hiệu tần số sóng vô tuyến; một antenna để nhận và gửi tín hiệu; và một substrate (chất nền). Thông tin của thẻ RFID được lưu trong bộ nhớ EEPROM. Các thẻ RFID có thể là loại thẻ chỉ đọc, với số serial từ nhà sản xuất phục vụ cho việc quản lý dữ liệu, hoặc là loại thẻ hỗ trợ đọc/ghi, với các dữ liệu đặc biệt để ghi vào thẻ

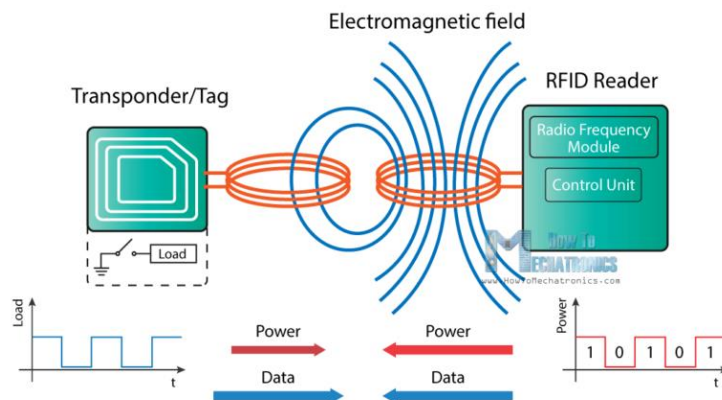
bởi người dùng hệ thống. Các thẻ lập trình được có thể được ghi một lần và đọc nhiều lần, thẻ trống có thể được ghi với một mã code điện tử của sản phẩm bởi người dùng.



Hình 2. 6: Thẻ RFID

Các thẻ RFID nhận yêu cầu truy vấn và phản hồi với thông tin định danh (ID) của nó và nhiều thông tin khác. Đây có thể là số series duy nhất của thẻ, hoặc các thông tin liên quan đến sản phẩm, như số kho hàng hoặc lô hàng, ngày sản xuất hoặc các thông tin đặc thù khác. Do mỗi thẻ RFID có số series riêng, hệ thống RFID có thể được thiết kế để đọc được nhiều thẻ cùng lúc, miễn là chúng nằm trong tầm hoạt động của đầu đọc RFID.

### 2.2.3 Nguyên lý hoạt động



Hình 2. 7: Nguyên lý hoạt động của RFID

Thiết bị RFID reader phát ra sóng điện từ ở một tần số nhất định để truy vấn dữ liệu, khi thiết bị RFID tag trong vùng hoạt động sẽ cảm nhận được sóng điện từ này và thu nhận năng lượng từ đó phát lại cho thiết bị RFID Reader biết mã số của mình [8]. Từ đó thiết bị RFID reader nhận biết được tag nào đang trong vùng hoạt động.

Giá trị trả về từ thẻ RFID này có thể được dùng để theo dõi vật thể, như hàng hóa, thiết bị, ... [8]

## **2.3 LCD**

### **2.3.1 LCD là gì?**

LCD (Liquid Liquid Crystal) là màn hình tinh thể lỏng. Nó là sự kết hợp của hai trạng thái vật chất - chất rắn và chất lỏng. Chúng có cả hai tính chất của chất rắn và chất lỏng và duy trì trạng thái tương ứng của chúng đối với chất khác [9].

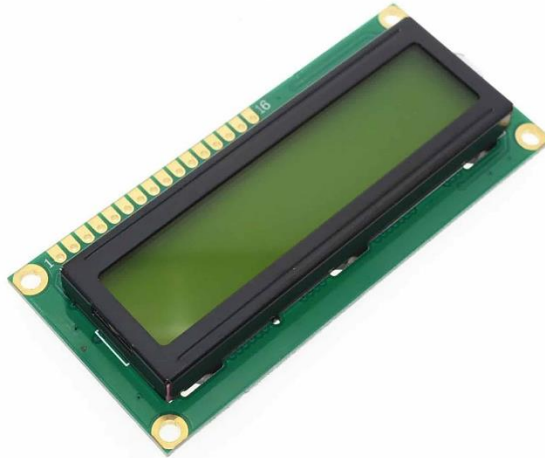
Màn hình tinh thể lỏng có ưu điểm khác biệt so với các loại màn hình khác là có mức tiêu thụ điện năng thấp hơn so với đèn LED. Yêu cầu tiêu thụ điện năng thấp đã làm cho nó tương thích với mạch logic tích hợp MOS. Ưu điểm khác của nó là chi phí thấp, và độ tương phản tốt. Nhược điểm chính của LCD là yêu cầu bổ sung về nguồn sáng, phạm vi hoạt động ở nhiệt độ giới hạn (từ 0 đến 60 ° C), độ tin cậy thấp, tuổi thọ hoạt động ngắn, tầm nhìn kém trong điều kiện ánh sáng xung quanh thấp, tốc độ chậm và cần nguồn AC để hoạt động [9].

LCD có rất nhiều dạng phân biệt theo kích thước từ vài ký tự đến hàng chục ký tự, từ 1 hàng đến vài chục hàng. Ví dụ LCD 16×2 có nghĩa là có 2 hàng, mỗi hàng có 16 ký tự.

### **2.3.2 Màn hình LCD 16x2 và module I2C**

#### **2.3.2.1 Màn hình LCD 16x2**

LCD thường được dùng làm khối hiển thị cho các đề tài đơn giản, giúp thông báo trạng thái, các thông số đọc được từ cảm biến hoặc dùng làm đồng hồ thời gian thực, không yêu cầu độ phân giải cao.



Hình 2. 8: Màn hình LCD 16x2

**Thông số kỹ thuật của màn hình LCD 1602 [10]:**

- Điện áp MAX: 7V.
- Điện áp MIN: - 0,3V.
- Điện áp ra mức thấp: <0.4V.
- Điện áp ra mức cao: > 2.4.
- Hoạt động ổn định: 2.7-5.5V.
- Dòng điện cấp nguồn: 350uA - 600uA.
- Nhiệt độ hoạt động: 30 - 75 độ C.

Bảng 2. 3: Chức năng từng chân của màn hình LCD 16x2

| Chân | Ký hiệu | Mô tả   |
|------|---------|---|
| 1    | VSS     | Đất (0V)  |
| 2    | VCC     | Dương nguồn +5V   |
| 3    | VEE     | Điều khiển độ tương phản của LCD                              |
| 4    | RS      | RS = 0, chọn thanh ghi lệnh<br>RS = 1, chọn thanh ghi dữ liệu |
| 5    | RW      | RW = 1: đọc dữ liệu<br>RW = 0: ghi dữ liệu                    |
| 6    | E       | Cho phép  |

|    |    |                   |
|----|----|-------------------|
| 7  | D0 | Bus dữ liệu 8 bit |
| 8  | D1 | Bus dữ liệu 8 bit |
| 9  | D2 | Bus dữ liệu 8 bit |
| 10 | D3 | Bus dữ liệu 8 bit |
| 11 | D4 | Bus dữ liệu 8 bit |
| 12 | D5 | Bus dữ liệu 8 bit |
| 13 | D6 | Bus dữ liệu 8 bit |
| 14 | D7 | Bus dữ liệu 8 bit |

Để giúp tiết kiệm chân LCD thường dùng chung với mạch giao tiếp **I2C**.

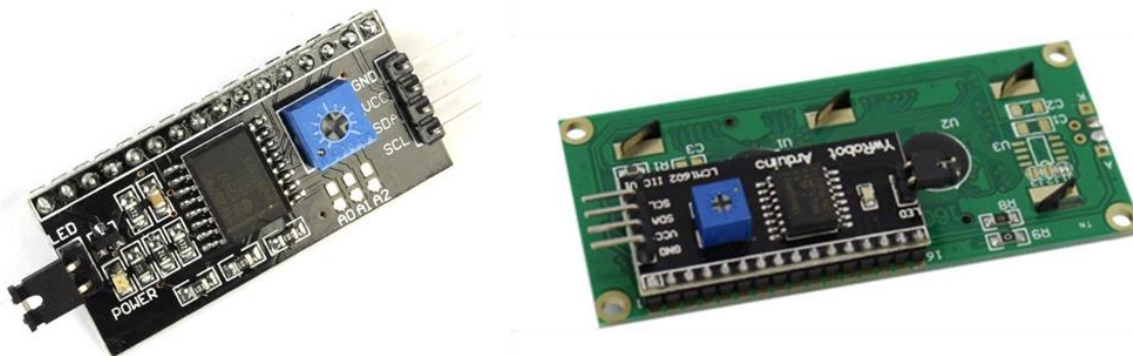
### 2.3.2.2 Module I2C

LCD có quá nhiều chân gây khó khăn trong quá trình kết nối và chiếm dụng nhiều chân của vi điều khiển. Module chuyển đổi I2C cho LCD sẽ giải quyết vấn đề này cho bạn, thay vì sử dụng tối thiểu 6 chân của vi điều khiển để kết nối với LCD (RS, EN, D7, D6, D5 và D4) thì với module chuyển đổi bạn chỉ cần sử dụng 2 chân (SCL, SDA) để kết nối. Module chuyển đổi I2C hỗ trợ các loại LCD sử dụng driver HD44780 (LCD 1602, LCD 2004, ...), kết nối với vi điều khiển thông qua giao tiếp I2C, tương thích với hầu hết các vi điều khiển hiện nay [11].

Ở đây đối với LCD 16x2 dùng module I2C địa chỉ tùy thuộc vào jump cắm trên module I2C. Để biết địa chỉ I2C đang dùng ta vào Arduino và cho chương trình I2C scanner để quét địa chỉ dùng cho I2C đang sử dụng. Và địa chỉ mà quét được cho I2C của nhóm là 0x27.

I2C sử dụng hai đường truyền tín hiệu:

- + Một đường xung nhịp đồng hồ (SCL) chỉ do Master phát đi.
- + Một đường dữ liệu (SDA) theo 2 hướng.



Hình 2. 9: Module I2C

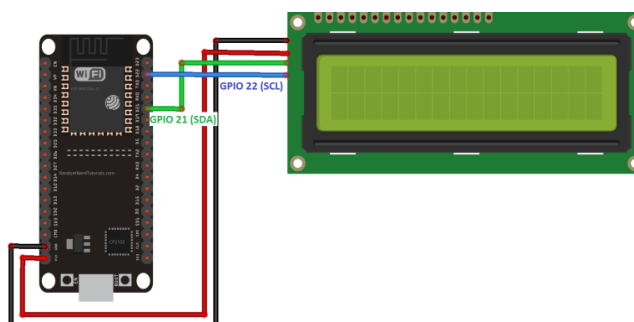
### Thông số kỹ thuật [11]:

- Điện áp hoạt động: 2.5-6V DC.
- Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).
- Giao tiếp: I2C.
- Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).
- Kích thước: 41.5mm(L)x19mm(W)x15.3mm(H).
- Trọng lượng: 5g.
- Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt.
- Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

### Giao tiếp I2C LCD ESP32

Bảng 2. 4: Sơ đồ kết nối với ESP32

| I2C LCD | ESP32   |
|---------|---------|
| GND     | GND     |
| VCC     | VIN     |
| SDA     | GPIO 21 |
| SCL     | GPIO 22 |



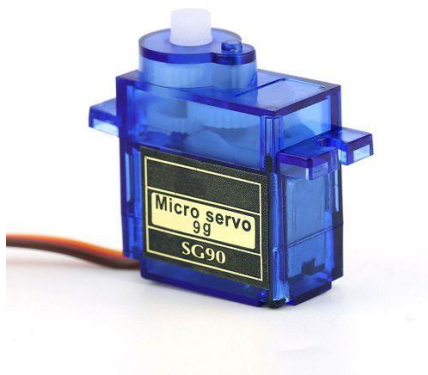
Hình 2. 10: Sơ đồ kết nối ESP32 với module I2C và LCD 16x2



## 2.4 Servo SG90

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ  $0^\circ$  -  $180^\circ$  [12].

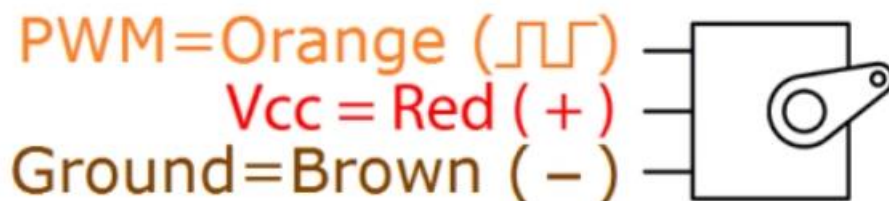
Động Cơ Servo SG90 là loại động cơ được dùng phổ biến trong các mô hình điều khiển nhỏ và đơn giản như cánh tay robot. Động cơ có tốc độ phản ứng nhanh, được tích hợp sẵn Driver điều khiển động cơ, dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM [12].



Hình 2. 11: Hình ảnh thực tế Servo

### Thông số kỹ thuật:

- Động cơ servo SG90 có thể xoay 180.
- Tốc độ hoạt động: 60 độ trong 0.1 giây.
- Điện áp hoạt động: 4.8V(~5V).



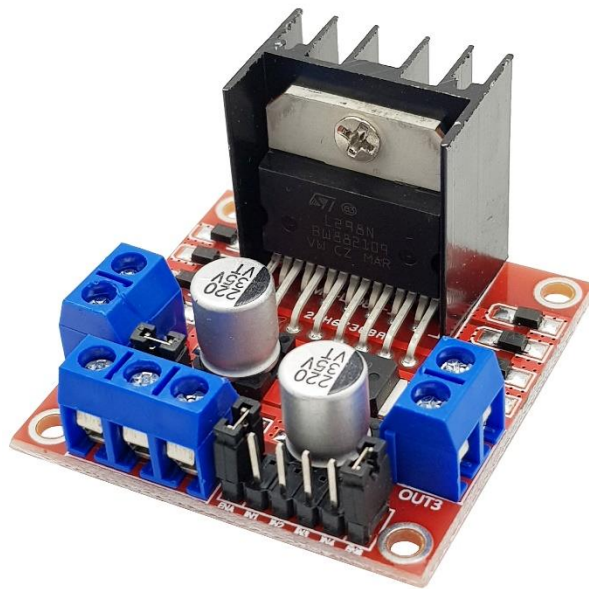
Hình 2. 12: Sơ đồ nối chân

Kết nối dây màu đỏ với 5V, dây màu nâu với mass, dây màu cam với chân phát xung của vi điều khiển.

## 2.5 L298N

L298N là module điều khiển động cơ DC và động cơ bước. Module có một IC điều khiển động cơ L298 và một bộ điều chỉnh điện áp 5V 78M05. Module L298N có thể điều khiển tối đa 4 động cơ DC hoặc 2 động cơ DC với khả năng điều khiển hướng và tốc độ [13].

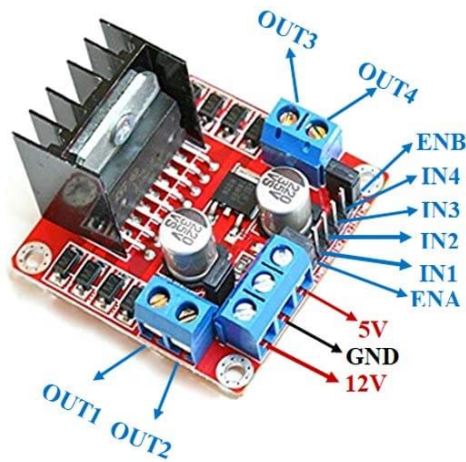
Có gắn tản nhiệt chống nóng cho IC, giúp IC có thể điều khiển với dòng định đạt 2A. IC L298N được gắn với các đi ốt trên board giúp bảo vệ vi xử lý chống lại các dòng điện cảm ứng từ việc khởi động/ tắt động cơ [14].



Hình 2. 13: Hình ảnh thực tế L298N

### 2.5.1 Thông số kỹ thuật:

- Driver: L298N tích hợp hai mạch cầu H.
- Điện áp điều khiển : +5V ~ +12 V.
- Dòng tối đa cho mỗi cầu H là: 2A.
- Điện áp của tín hiệu điều khiển : +5 V ~ +7 V.



Hình 2. 14: Sơ đồ chân L298N

Bảng 2. 5: Chức năng từng chân của L298N

| Chân                   | Chức năng                                 |
|------------------------|---|
| IN1, IN2, IN3, IN4     | Nhận tín hiệu điều khiển                  |
| OUT1, OUT2, OUT3, OUT4 | Nối với động cơ                           |
| ENA, ENB               | Điều khiển mạch cầu H (điều khiển tốc độ) |

## 2.6 Buzzer

### 2.6.1 Giới thiệu

Buzzer là một thiết bị tạo ra tiếng còi hoặc tiếng bíp. Có nhiều loại nhưng cơ bản nhất là buzzer áp điện, là một miếng phẳng của vật liệu áp điện với hai điện cực. Chức năng của buzzer là được dùng ở vị trí cần phát ra âm thanh. Dựa trên các thiết kế khác nhau, nó có thể tạo ra các âm thanh khác nhau như báo thức, nhạc, chuông và còi báo động. Buzzer rẻ và kêu to mà không cần sử dụng nhiều năng lượng [15].

Các loại Buzzer [15]:

- Buzzer áp điện.
- Buzzer điện từ.
- Buzzer cơ khí.

- Buzzer cơ điện.
- Buzzer từ tính.

### 2.6.2 Thông số kỹ thuật [15]

Màu đen.

Dải tần số là 3.300Hz.

Phạm vi nhiệt độ hoạt động từ  $-20^{\circ}\text{C}$  đến  $+60^{\circ}\text{C}$ .

Dải điện áp hoạt động từ 3V đến 24V DC.

Mức áp suất âm thanh là 85dBA hoặc 10cm.

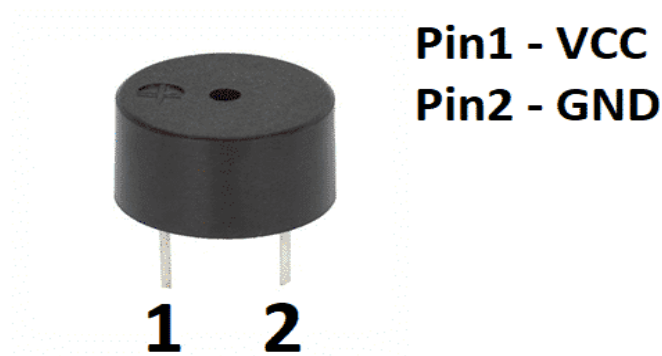
Dòng điện cung cấp dưới 15mA.

### 2.6.3 Nguyên lý hoạt động [15]

Nguyên lý hoạt động của buzzer dựa trên cơ chế: một khi điện áp được cung cấp trên vật liệu áp điện, thì sự chênh lệch áp suất sẽ được tạo ra, trong đó, loại áp điện trên bao gồm các tinh thể áp điện giữa hai dây dẫn.

Sau khi có sự chênh lệch tiềm năng giữa các tinh thể này, chúng sẽ đẩy một dây dẫn và kéo dây dẫn bổ sung thông qua thuộc tính bên trong của chúng. Vì vậy, hành động liên tục này sẽ tạo ra tín hiệu âm thanh sắc nét.

### 2.6.4 Sơ đồ chân [15]



Hình 2. 15: Sơ đồ chân Buzzer

*Bảng 2. 6: Chức năng các chân DHT11*

| Số chân | Tên chân | Mô tả   |
|---------|----------|---------|
| 1       | Negative | Nối VCC |
| 2       | Positive | Nối GND |

## **2.7 Cảm biến DHT11**

### **2.7.1 Giới thiệu [16]**

DHT11 là một cảm biến kỹ thuật số giá rẻ để cảm nhận nhiệt độ và độ ẩm. Cảm biến này có thể dễ dàng giao tiếp với bất kỳ bộ vi điều khiển vi nào như Arduino, Raspberry Pi, ... để đo độ ẩm và nhiệt độ ngay lập tức.

DHT11 là một cảm biến độ ẩm tương đối. Để đo không khí xung quanh, cảm biến này sử dụng một điện trở nhiệt và một cảm biến độ ẩm điện dung.

### **2.7.2 Cấu tạo [16]**

Gồm:

- Cảm biến độ ẩm điện dung và một điện trở nhiệt để cảm nhận nhiệt độ.
- Tự điện cảm biến độ ẩm.
- Cảm biến này sử dụng một nhiệt điện trở có hệ số nhiệt độ âm, làm giảm giá trị điện trở của nó khi nhiệt độ tăng.

### **2.7.3 Thông số kỹ thuật [16]**

Điện áp hoạt động: 3 – 5V DC.

Dòng điện tiêu thụ: 2.5mA.

Phạm vi cảm biến độ ẩm: 20% - 90% RH, sai số  $\pm 5\%$  RH.

Phạm vi cảm biến nhiệt độ: 0°C ~ 50°C, sai số  $\pm 2^\circ\text{C}$ .

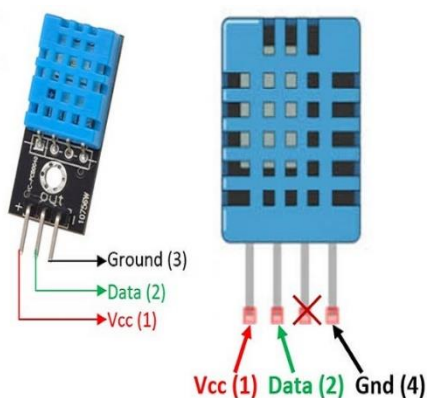
Tần số lấy mẫu tối đa: 1Hz.

Kích thước: 23 \* 12 \* 5 mm.

#### 2.7.4 Nguyên lý chuyển đổi

Nguyên lý chuyển đổi của DHT11: Dữ liệu về độ ẩm và nhiệt độ sẽ được lưu trong DHT11 là dữ liệu số và cảm biến bên trong sẽ nhận nhiệt độ và độ ẩm của môi trường chuyển đổi sang tín hiệu số rồi đưa sang bộ nhớ. Tín hiệu ngõ ra dạng số và có độ dài là 40 bit và 40 bit này được truyền theo chuẩn 1 dây được kết nối từ cảm biến đến bộ đọc giá trị, các bit này được truyền theo thứ tự từng bit và truyền từ bit có trọng số cao đến bit có trọng số thấp, sau khi truyền hết 40 bit này sẽ tiếp tục đọc và truyền 40 bit tiếp theo [16].

#### 2.7.5 Sơ đồ chân [16]



Hình 2. 16: Sơ đồ chân của DHT11.

Bảng 2. 7: Chức năng các chân DHT11

| Số chân | Tên chân | Mô tả                             |
|---------|----------|-----------------------------------|
| 1       | VCC      | Nguồn 3.5V đến 5.5V               |
| 2       | Data     | Đầu ra của nhiệt độ và độ ẩm      |
| 3       | NC       | Không có kết nối và không sử dụng |
| 4       | GND      | Nối đất                           |

## 2.8 Cảm biến MQ4

### 2.8.1 Giới thiệu chung

Cảm biến loại MOS (kim loại oxit bán dẫn), được sử dụng để phát hiện nồng độ khí mêtan trong không khí tại nhà hoặc các khu công nghiệp [17].

### 2.8.2 Nguyên lý chuyển đổi

Vật liệu nhạy cảm của cảm biến khí MQ4 là  $\text{SnO}_2$ , có độ dẫn điện thấp hơn trong không khí sạch [17].

Cảm biến khí MQ4 có độ nhạy cao với khí mê-tan, khi phát hiện khí dễ cháy tồn tại, độ dẫn điện của cảm biến sẽ cao hơn cùng với nồng độ khí tăng [17].

Người dùng có thể chuyển đổi sự thay đổi độ dẫn điện thành tín hiệu đầu ra tương ứng của nồng độ khí thông qua một mạch đơn giản [17].

### 2.8.3 Sơ đồ chân MQ4 [17]

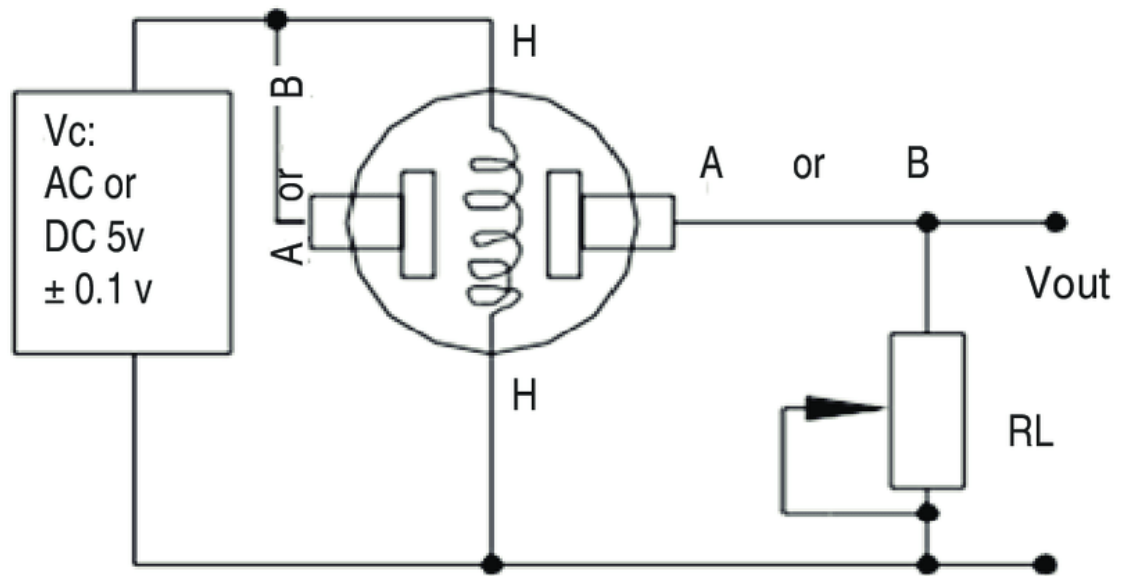


Hình 2. 17: Hình ảnh sơ đồ chân MQ4

Bảng 2. 8: Chi tiết các chân MQ4

| Chân | Mô tả                      |
|------|----------------------------|
| VCC  | Nối nguồn                  |
| GND  | Nối đất                    |
| AO   | Đầu ra tín hiệu số         |
| DO   | • Đầu ra tín hiệu tương tự |

#### 2.8.4 Sơ đồ mạch cảm biến MQ4 [17]



Hình 2. 18: Sơ đồ mạch của MQ4

#### 2.8.5 Thông số kỹ thuật [17]

Nguồn hoạt động: 5V.

Chip chính: cảm biến khí LM393, ZYMQ- 4.

Loại dữ liệu: Analog.

Phạm vi phát hiện rộng.

Tốc độ phản hồi nhanh và độ nhạy cao.

Mạch đơn giản.

Ổn định khi sử dụng trong thời gian dài.

Với chỉ báo đầu ra tín hiệu.

Đầu ra tín hiệu kép (đầu ra analog và đầu ra mức TTL).

TTL đầu ra tín hiệu có giá trị thấp. (Khi đầu ra thấp, đèn tín hiệu bật, nó có thể được kết nối trực tiếp với vi điều khiển).

Đầu ra analog 0 ~ 5V điện áp, nồng độ càng cao, điện áp càng cao.

Độ nhạy cao với khí mê-tan, khí tự nhiên.

Tính năng khôi phục phản hồi nhanh.



## 2.9 App Inventor

### 2.9.1 Giới thiệu chung

App Inventor là một ứng dụng web có mã nguồn mở do Google tạo ra (sau này được quản lý bởi Viện Công nghệ Massachusetts hay còn gọi là MIT) để tạo các ứng dụng cho hệ điều hành Android [18].



*Hình 2. 19: Biểu tượng của App Inventor*

Về cơ bản, các ứng dụng được tạo trên App Inventor sẽ chỉ hoạt động được trên nền tảng Android. Giao diện của App Inventor bao gồm các khối hộp, bên trong là các đoạn mã. Khi sử dụng, người dùng sẽ kéo thả các khối này vào bảng mã để tiến hành lắp ghép thành một ứng dụng hoàn chỉnh [18].

Mục tiêu cốt lõi của App Inventor là giúp những người dùng có thể làm quen với các thao tác tạo ra phần mềm [18].

### 2.9.2 Ưu nhược điểm của App Inventor

#### 2.9.2.1 Ưu điểm [18]

Các ứng dụng được tạo thông qua các khối một cách trực quan mà không cần biết mã lập trình.

Có thể dễ dàng truy cập ở bất cứ đâu chỉ với kết nối Internet.

Dễ dàng sử dụng chỉ với việc kéo thả các khối chứa mã lập trình rồi sắp xếp cho hợp lý.

Hoạt động trên một hệ thống tối giản, không cần bất cứ đoạn mã, thuật toán phức tạp nào để khởi tạo chương trình.

### **2.9.2.2 Nhược điểm [18]**

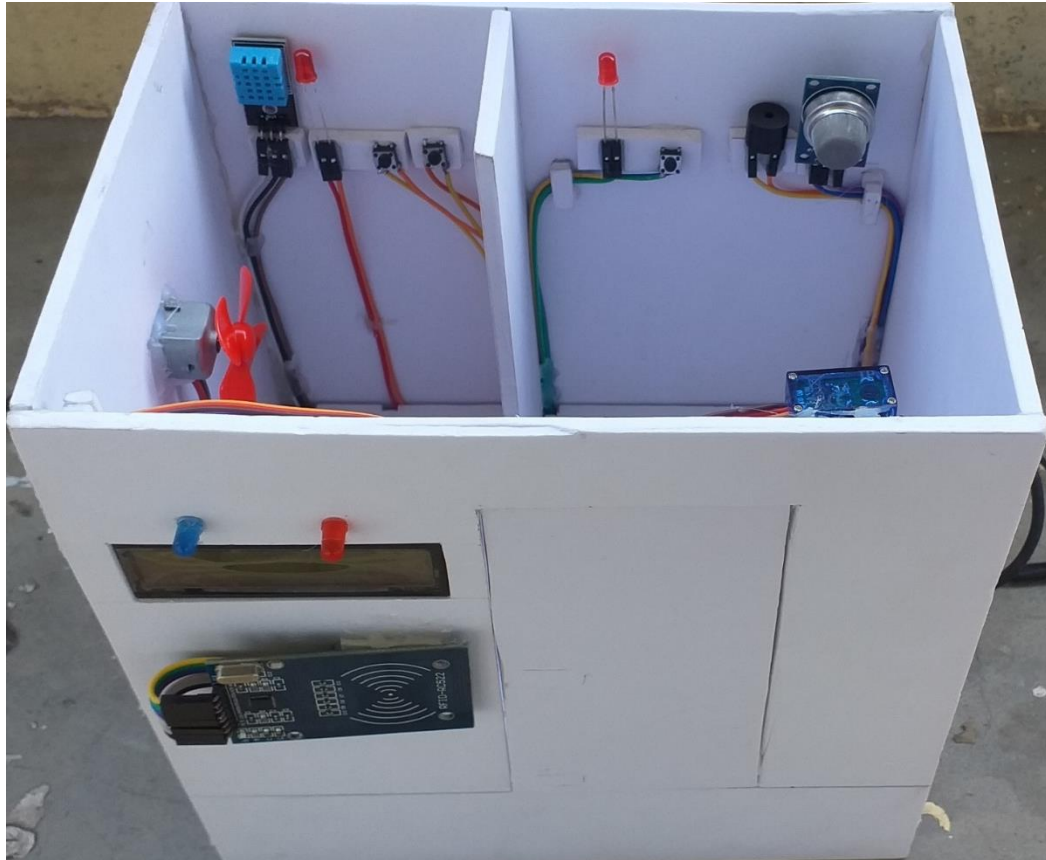
Chỉ dùng để phát triển các ứng dụng Android.

Không tạo mã Java để có thể phát triển sâu hơn.

Do ứng dụng được phát triển trên server của MIT, giới hạn dung lượng của mỗi project chỉ là 5MB.

## CHƯƠNG 3 THIẾT KẾ

### 3.1 Thiết kế tổng quan

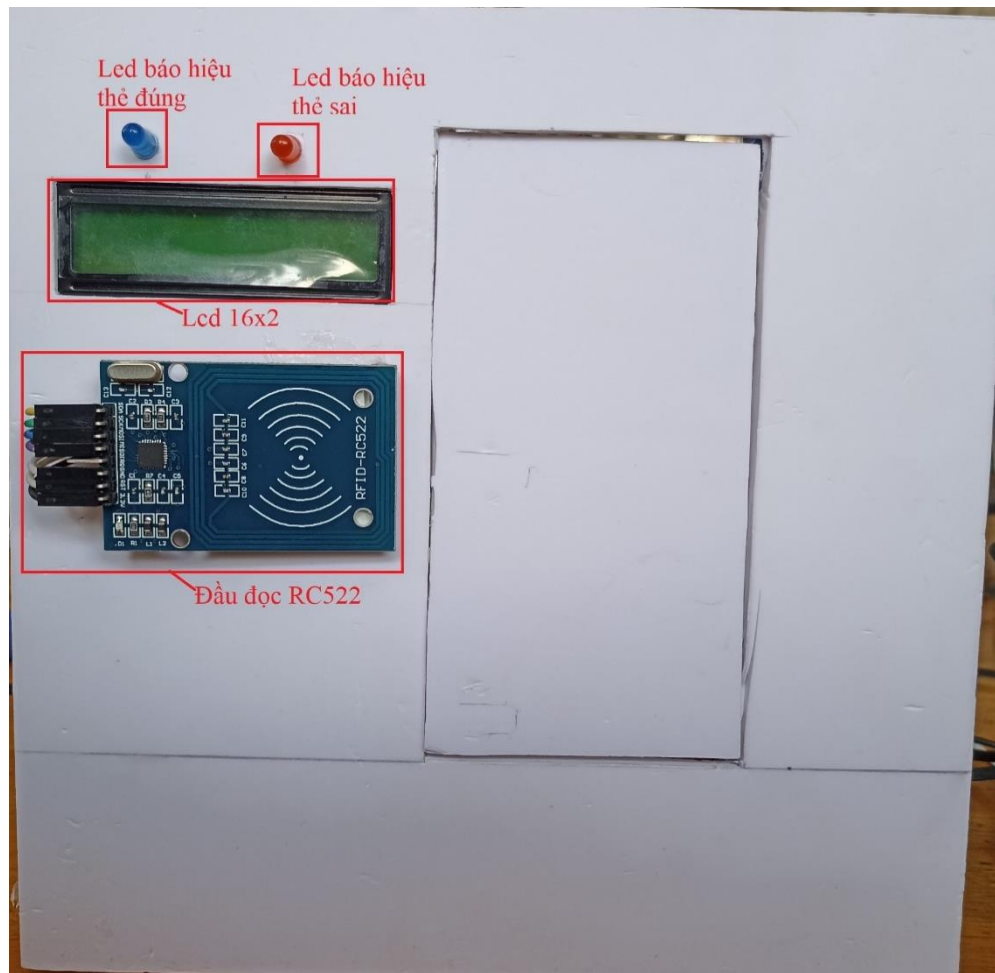


*Hình 3. 1: Thiết kế tổng quan*

Mô hình được thiết kế gồm 2 phần chính:

- Mặt ngoài : Chức năng điều khiển khóa cửa dùng thẻ RFID.
- Mặt trong:
  - + Chức năng đo độ ẩm, nhiệt độ.
  - + Chức năng đo khí gas.
  - + Chức năng điều khiển đèn.
  - + Chức năng điều khiển quạt

### 3.2 Thiết kế mặt ngoài

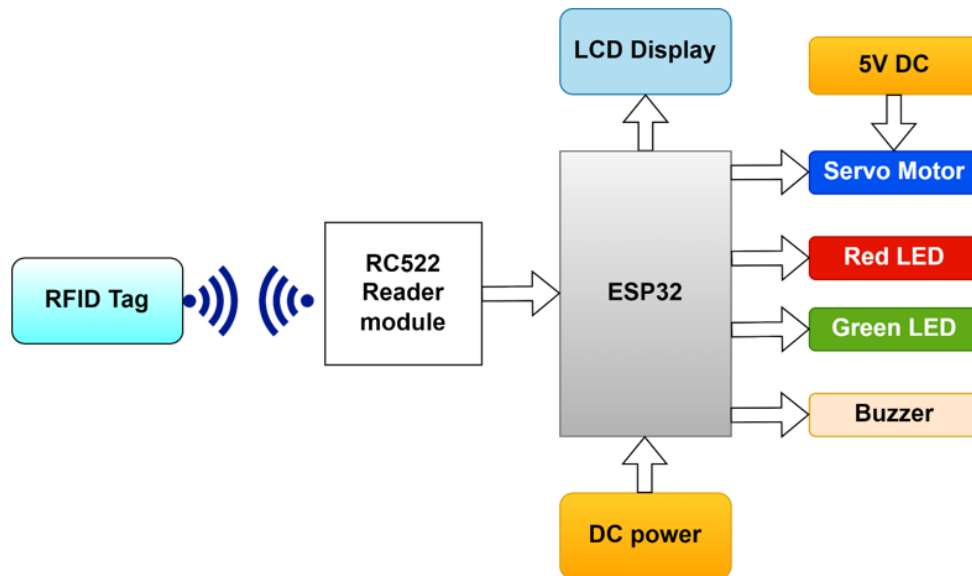


Hình 3. 2: Thiết kế mặt ngoài mô hình

Mặt ngoài gồm có các thiết bị:

- Đầu đọc RC522
- Lcd 16x2
- Led báo hiệu thẻ đúng
- Led báo hiệu thẻ sai

### 3.2.1 Sơ đồ khối



Hình 3. 3: Sơ đồ khối mặt ngoài

#### Chức năng từng khối:

**Khối vi điều khiển ESP32:** là khối xử lý trung tâm cốt lõi của toàn hệ thống, điều khiển mọi hoạt động của mạch. Bao gồm việc tiếp nhận dữ liệu từ RFID, xử lý dữ liệu và điều khiển hoạt động mở cửa, đóng cửa hay cảnh báo nếu thẻ sai, ...

**Khối RFID:** khối quan trọng trong việc thu nhập UID của thẻ và gửi đến khối xử lý trung tâm.

**Khối Servo:** khối này sẽ kết nối trực tiếp với cửa ra vào, khi nhận tín hiệu điều khiển từ khối xử lý trung tâm mà thực hiện việc đóng hay mở cửa.

**Khối LCD:** hiển thị mọi thông báo của hệ thống bao gồm việc yêu cầu quét thẻ, thông báo mở cửa, đóng cửa hay cảnh báo nếu thẻ quét sai.

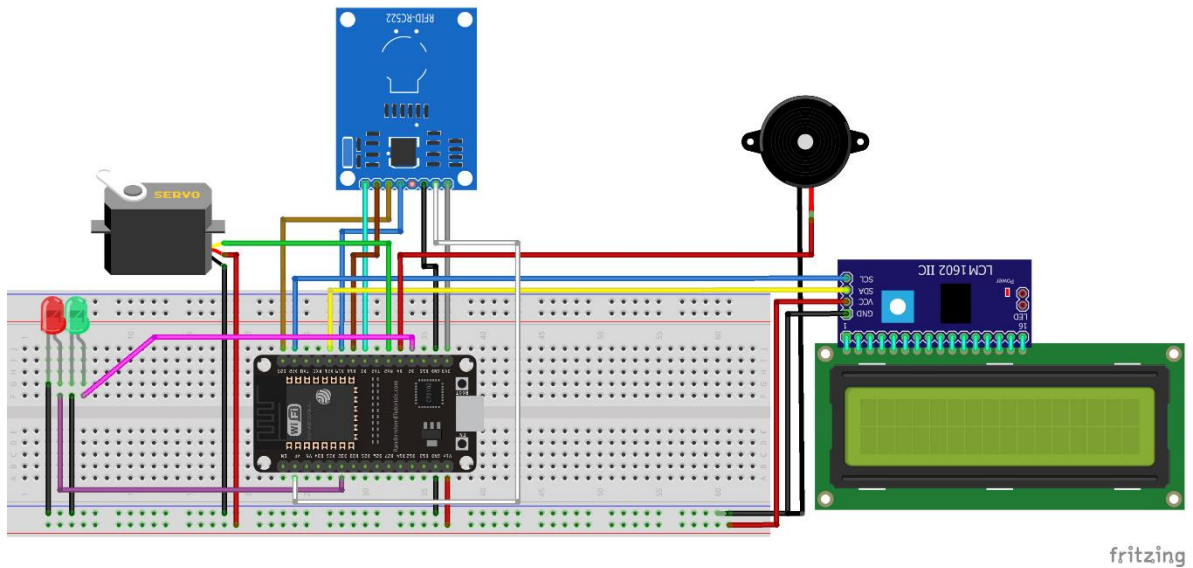
**Khối Buzzer:** thực hiện chức năng phát ra âm thanh cảnh báo nếu thẻ RFID quét sai, cũng như âm thanh phản hồi lại khi phát hiện thẻ đúng.

**Khối LED:** gồm có hai trạng thái LED: xanh và đỏ. Khi quét thẻ đúng thì LED xanh sẽ sáng và ngược lại LED đỏ sáng. Khối này là một phần nhỏ trong chức năng phản hồi lại sự kiện quét thẻ của người dùng.

**Khối nguồn chính (DC power):** cung cấp nguồn cho toàn mạch hoạt động.

**Khối nguồn phụ 5V:** Servo cần phải có một nguồn riêng biệt để hoạt động, điều này làm ổn định nguồn cung cấp cho các phần tử còn lại. Tránh việc hệ thống lỗi khi chỉ dùng chung một nguồn chính.

### 3.2.2 Sơ đồ nối dây



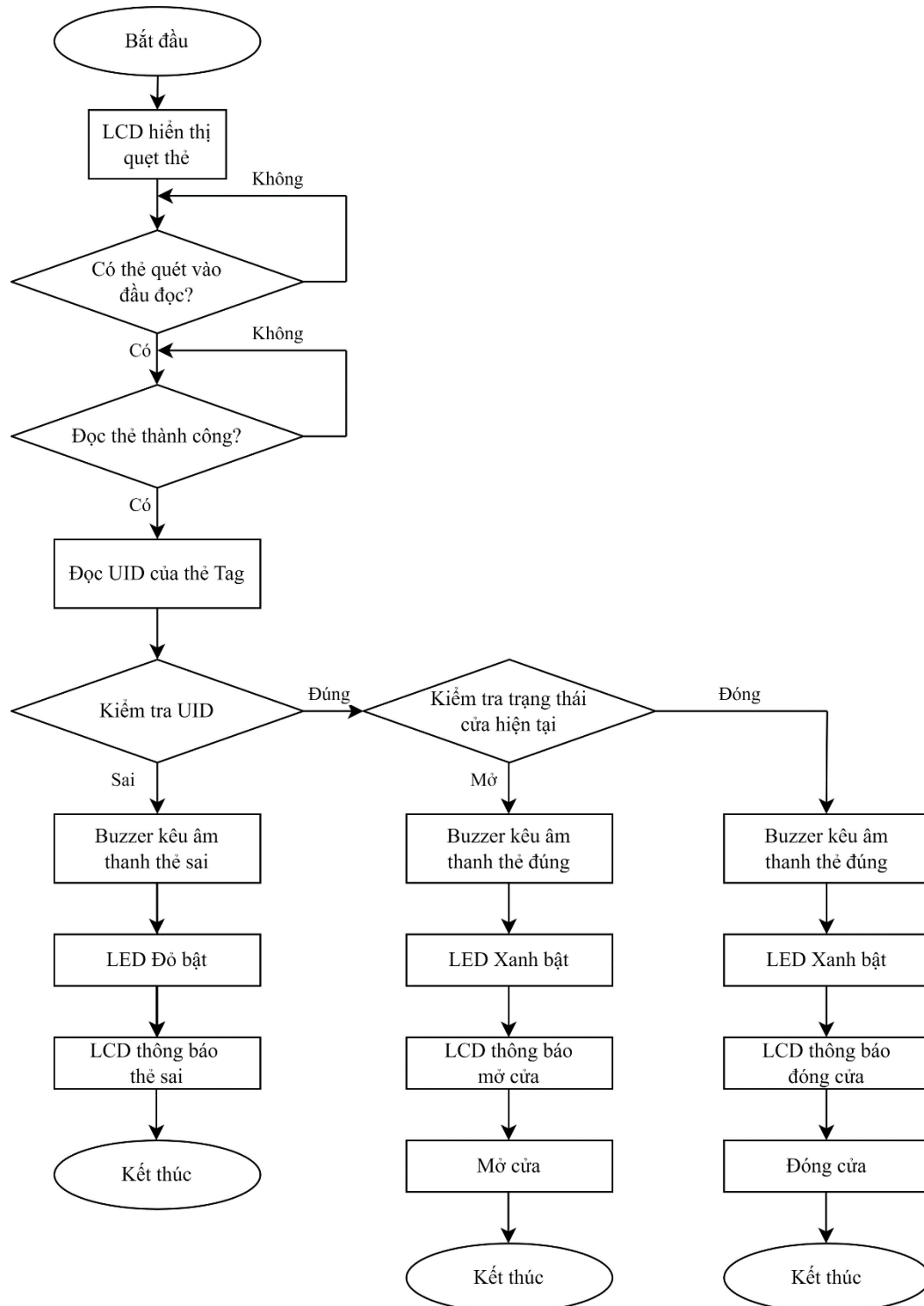
Hình 3. 4: Sơ đồ nối dây mặt ngoài

Bảng 3. 1 : Bảng kết nối dây mặt ngoài

| Thiết bị | Chân thiết bị | Chân của ESP32 |
|----------|---------------|----------------|
| RFID     | SCK           | 18             |
|          | SDA           | 5              |
|          | MOSI          | 23             |
|          | MISO          | 19             |
|          | IRQ           | Không có       |
|          | GND           | GND            |
|          | RST           | VP             |
|          | 3V3           | 3V3            |

|          |        |                       |
|----------|--------|-----------------------|
| SERVO    | VIN    | VIN                   |
|          | GND    | GND                   |
|          | Signal | RX2                   |
| LCD      | SCL    | 22                    |
|          | SDA    | 21                    |
|          | VCC    | Nối nguồn DC 5V riêng |
|          | GND    | GND                   |
| BUZZER   | V+     | 4                     |
|          | V-     | GND                   |
| LED XANH | V+     | 2                     |
|          | V-     | GND                   |
| LED ĐỎ   | V+     | 32                    |
|          | V-     | GND                   |

### 3.2.3 Lưu đồ code



Hình 3. 5: Lưu đồ code điều khiển khóa cửa



### 3.2.4 Giải thích code

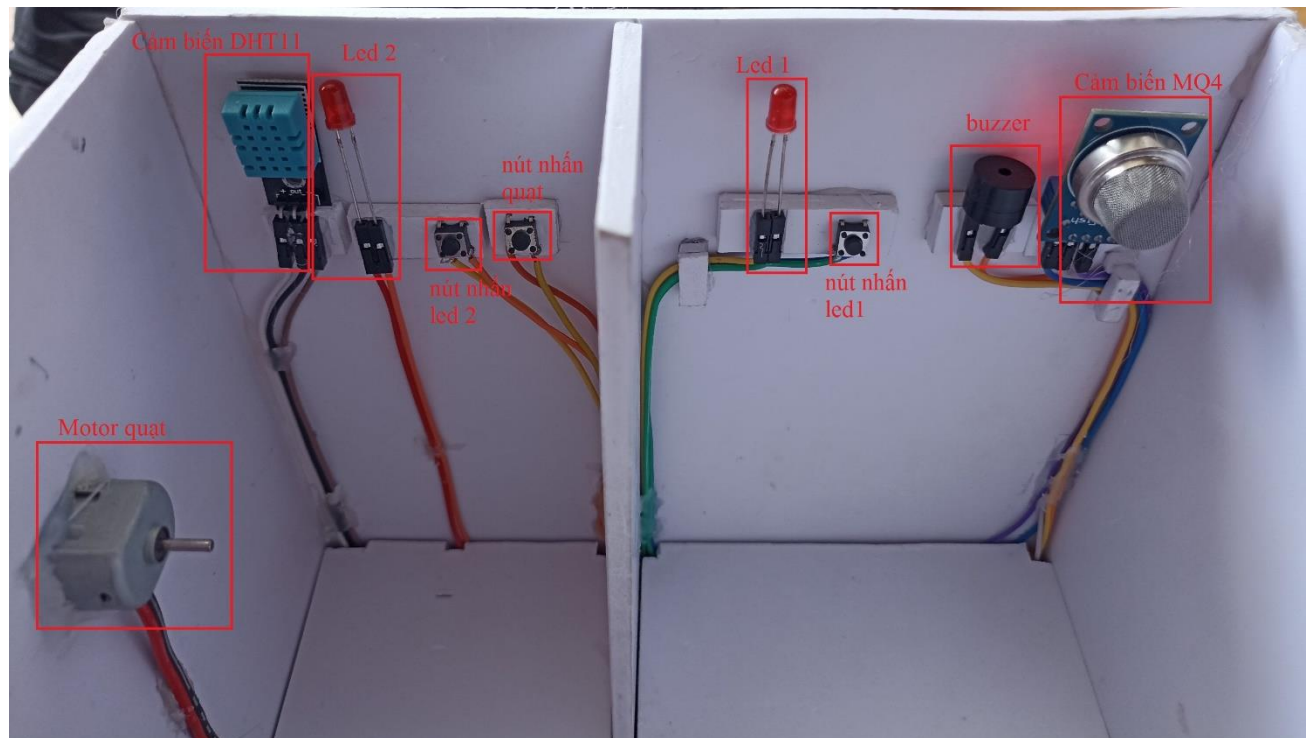
|   |   |
|---|---|
| <pre>#include &lt;LiquidCrystal_I2C.h&gt; #include &lt;SPI.h&gt; #include &lt;MFRC522.h&gt; #include &lt;ESP32Servo.h&gt; #define BUZZER 4 #define RST_PIN 36 #define SDA 5 #define servoPin 16 #define led_xanh 2 #define led_do 32 bool state = false; int i; int unlock = 0; byte user1[4]={0xA0,0x88,0x5D,0x20}; int scan[4] ; Servo myservo; LiquidCrystal_I2C lcd_i2c(0x27, 16, 2); MFRC522 mfrc522(SDA, RST_PIN); void setup(){   Serial.begin(115200);   pinMode(BUZZER,OUTPUT);   pinMode(led_xanh,OUTPUT);   pinMode(led_do,OUTPUT);    myservo.setPeriodHertz(50);   myservo.attach(servoPin);   myservo.write(0);</pre> | <p>Thư viện LCD_I2C</p> <p>Thư viện SPI</p> <p>Thư viện RFID</p> <p>Thư viện Servo</p> <p>Chân của Buzzer</p> <p>Chân reset của RFID</p> <p>Chân dữ liệu của RFID</p> <p>Chân của động cơ Servo</p> <p>Chân LED báo đọc đúng thẻ</p> <p>Chân LED báo đọc sai thẻ</p> <p>Biến báo hiệu trạng thái cửa</p><br><p>Mảng chứa UID của thẻ đúng</p> <p>Mảng chứa UID của thẻ được quét</p> <p>Khởi tạo biến cho Servo</p> <p>I2C address 0x27, 16 column and 2 rows</p> <p>Thiết lập chân dữ liệu và reset của RFID</p><br><p>Khai báo chu kỳ xung PWM</p> <p>Khởi chạy servo</p> |
|---|---|

|   |   |
|---|---|
| <pre> lcd_i2c.init(); lcd_i2c.backlight(); lcd_i2c.setCursor(4, 0); lcd_i2c.print("Quet The"); SPI.begin(); mfrc522.PCD_Init(); mfrc522.PCD_DumpVersionToSerial(); }  void control_rfid(){   if ( ! mfrc522.PICC_IsNewCardPresent()){     return;   }   if ( ! mfrc522.PICC_ReadCardSerial()){     return;   }    digitalWrite(buzzer_rfid,HIGH);   delay(300);   digitalWrite(buzzer_rfid,LOW);    for (i = 0; i &lt; 4; i++) {     scan[i] = mfrc522.uid.uidByte[i];     unlock = 1;     if(scan[i] != user1[i]) unlock = 0;   }   if (unlock == 1){     if(state == false){ </pre> | <p>Khởi tạo màn hình LCD</p> <p>Chọn chế độ bật đèn nền</p> <p>Di chuyển con trỏ sang vị trí thứ 4 của hàng đầu</p> <p>Khởi tạo SPI để giao tiếp với RFID</p> <p>Khởi tạo RFID</p> <p>Kiểm tra xem có thẻ quẹt hay không</p> <p>Kiểm tra xem đọc thẻ có thành công hay không</p> <p>Buzzer báo hiệu đọc thẻ thành công</p> <p>Lấy UID của thẻ đang quét</p> <p>Kiểm tra xem thẻ được quẹt có khớp với thẻ đã thiết lập trước hay không?</p> <p>Kiểm tra trạng thái cửa hiện tại</p> |
|---|---|

|  |   |
|--|---|
| <pre> state = true; digitalWrite(led_xanh,HIGH); digitalWrite(led_do,LOW); myservo.write(90); lcd_i2c.clear(); lcd_i2c.setCursor(4, 0); lcd_i2c.print("The Dung"); lcd_i2c.setCursor(5, 1); lcd_i2c.print("Mo Cua"); delay(1000); digitalWrite(led_xanh,LOW); } else { </pre>                                | <p>Thực hiện mở cửa và hiện thị thông báo lên LCD và hiển thị trạng thái đèn LED báo hiệu</p>                                     |
| <pre> state = false; digitalWrite(led_xanh,HIGH); digitalWrite(led_do,LOW); myservo.write(0); lcd_i2c.clear(); lcd_i2c.setCursor(4, 0); lcd_i2c.print("The Dung"); lcd_i2c.setCursor(4, 1); lcd_i2c.print("Dong Cua"); delay(1000); digitalWrite(led_xanh,LOW); } } else { digitalWrite(led_do,HIGH); </pre> | <p>Thực hiện đóng cửa và hiện thị thông báo lên LCD và hiển thị trạng thái đèn LED báo hiệu</p> <p>Thông báo nếu thẻ quét sai</p> |

|  |  |
|--|--|
| <pre>digitalWrite(led_xanh,LOW);<br/>lcd_i2c.clear();<br/>lcd_i2c.setCursor(4, 0);<br/>lcd_i2c.print("The Sai");<br/>lcd_i2c.setCursor(4, 1);<br/>lcd_i2c.print("Quet lai");<br/>delay(1000);<br/>digitalWrite(led_do,LOW);<br/>}<br/>lcd_i2c.clear();<br/>lcd_i2c.setCursor(4, 0);<br/>lcd_i2c.print("Quet The");<br/>}</pre> |  |
|--|--|

### 3.3 Thiết kế mặt trong

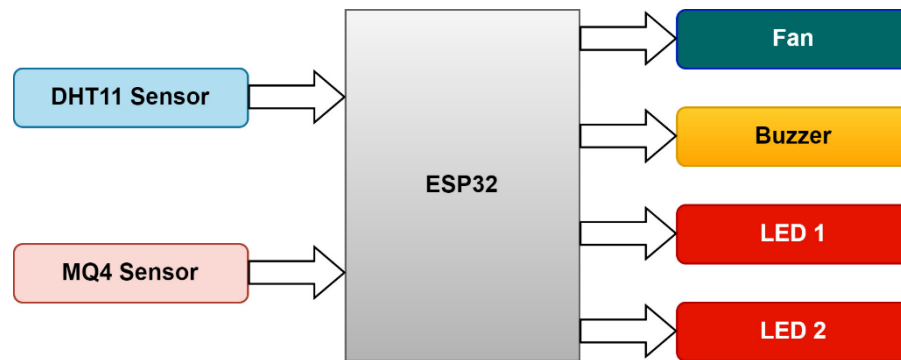


Hình 3. 6: Thiết kế mặt trong mô hình

Mặt trong gồm các thiết bị:

- Cảm biến DHT11
- Cảm biến MQ4
- Buzzer
- Led 1
- Led 2
- Nút nhấn led 1
- Nút nhấn led 2
- Motor quạt
- Nút nhấn quạt

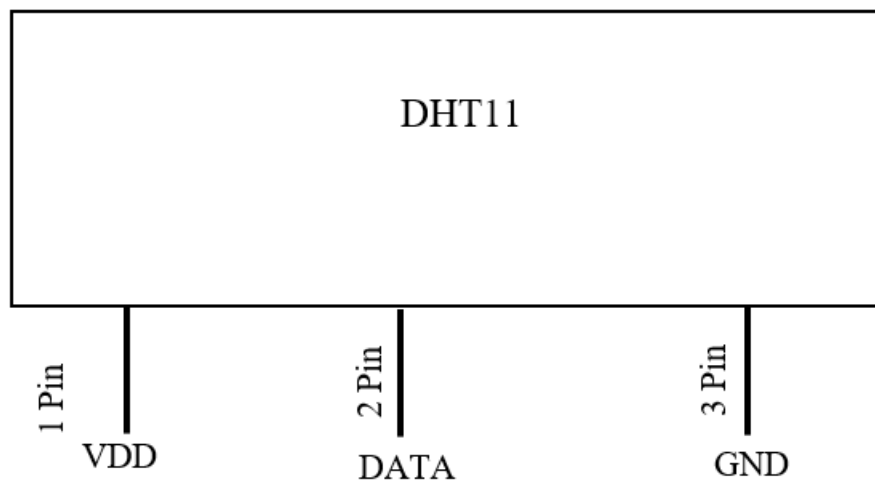
### 3.3.1 Sơ đồ khối



Hình 3. 7: Sơ đồ khối mặt trong

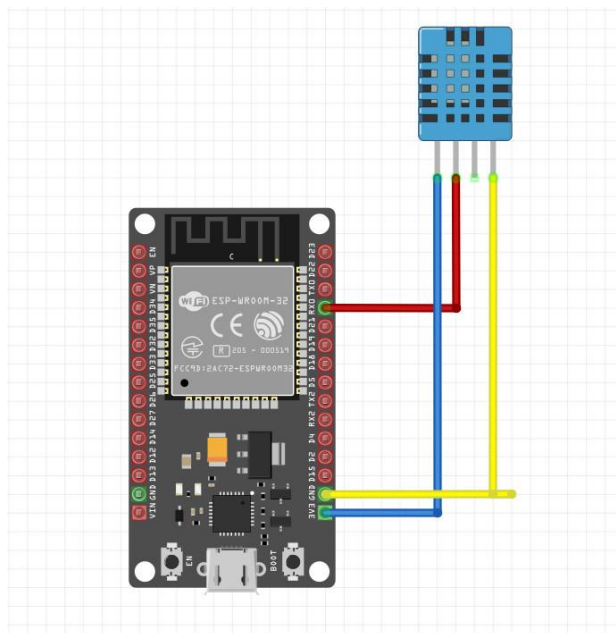
### 3.3.2 Khối thu thập nhiệt độ, độ ẩm DHT11

#### 3.3.2.1 Sơ đồ khối



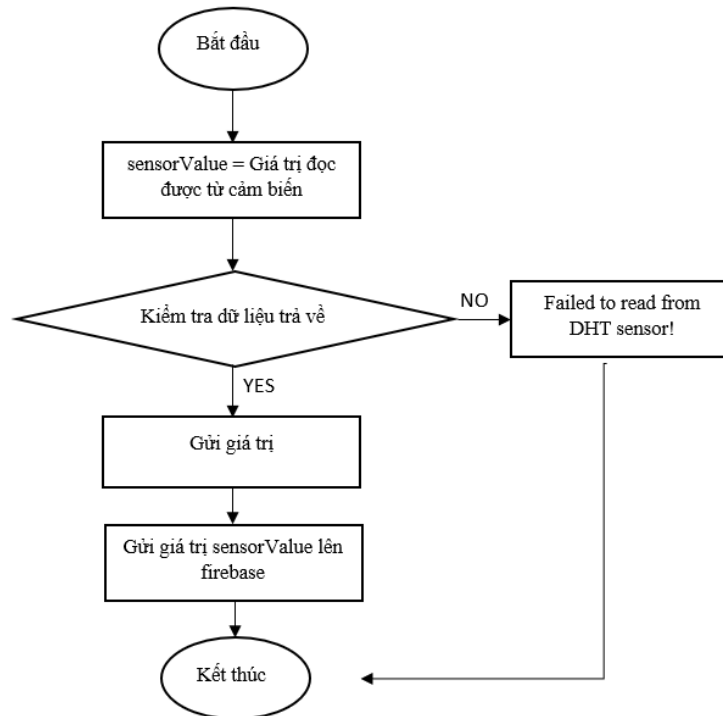
Hình 3. 8: Sơ đồ khối thu thập nhiệt độ, độ ẩm

### 3.3.2.2 Sơ đồ nối dây



Hình 3. 9: Sơ đồ nối dây DHT11

### 3.3.2.3 Lưu đồ



Hình 3. 10: Lưu đồ khối thu thập nhiệt độ, độ ẩm

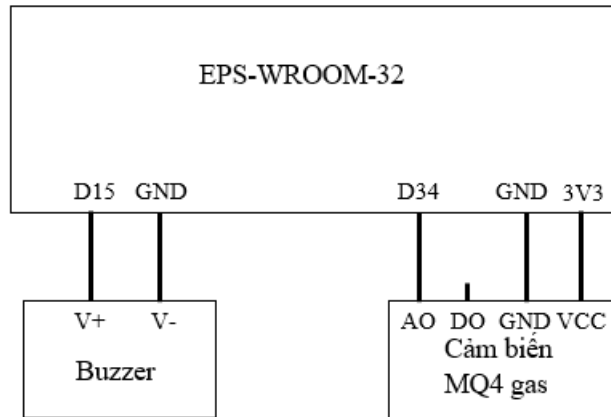
### 3.3.2.4 Giải thích chương thu thập nhiệt độ, độ ẩm

|   |   |
|---|---|
| <pre>#include "DHT.h" #define DHTPIN 3 #define DHTTYPE DHT11 DHT dht(DHTPIN, DHTTYPE); void setup() {     Serial.begin(9600);     dht.begin(); }  void control_DHT11() {     delay(1000);      float h = dht.readHumidity();     float t = dht.readTemperature();      if (isnan(h)    isnan(t))     {         Serial.println("Failed to read from DHT sensor!");         return;     }     Firebase.setFloat(firebaseData, "/humidity",h);     Firebase.setFloat(firebaseData,"/temperature",t); }</pre> | <pre>// khai báo thư viện DHT11 // // Chân dữ liệu của DHT 11 // DHT 11  // bắt đầu đọc dữ liệu  //Đọc dữ liệu trả về vào 3 biến: // Đọc độ ẩm // Đọc nhiệt độ theo độ C  //Kiểm tra dữ liệu trả về, nếu dữ liệu đọc được là nan (not a number) thì in ra màn hình: Failed to read from DHT sensor! .  //Cập nhật giá trị độ ẩm lên firebase //Cập nhật giá trị nhiệt độ lên firebase</pre> |
|---|---|



### 3.3.3 Khối thu thập khí gas MQ4

#### 3.3.3.1 Sơ đồ khối

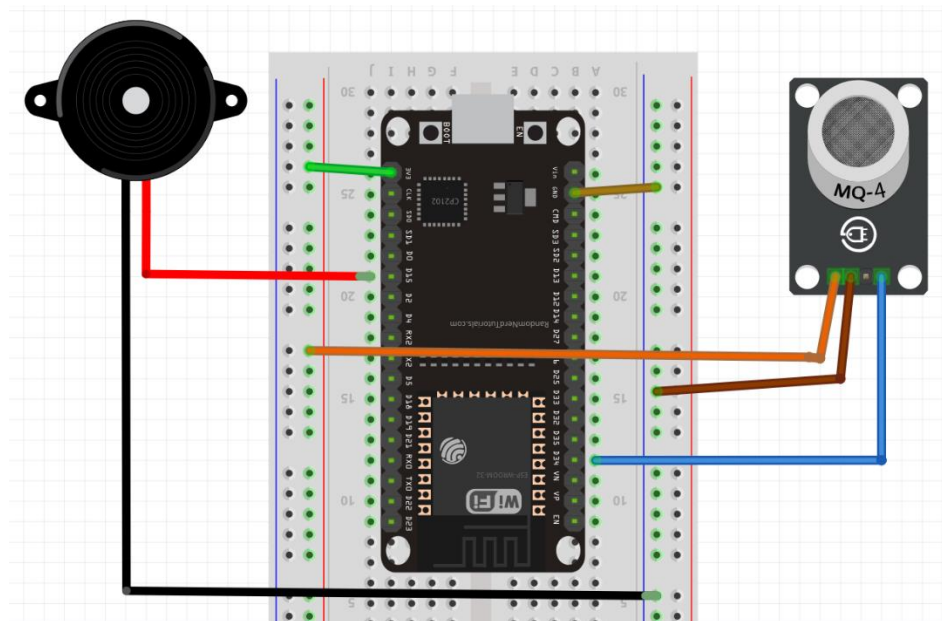


Hình 3. 11: Sơ đồ khối thu thập khí gas

Bảng 3. 2: Kết nối chân khối thu thập khí gas

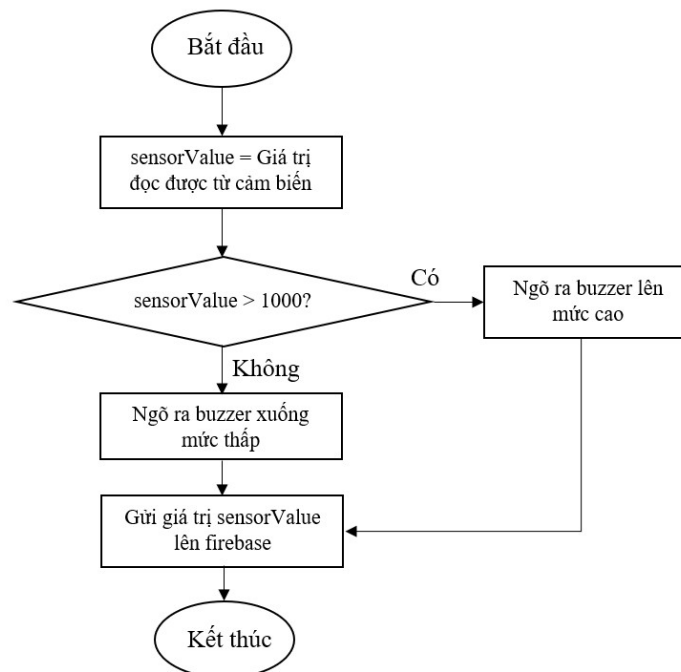
| Tên thiết bị | Chân thiết bị | Chân ESP32 |
|--------------|---------------|------------|
| MQ4 gas      | VCC           | 3V3        |
|              | GND           | GND        |
|              | DO            | Không có   |
|              | AO            | D34        |
| Buzzer       | V+            | D15        |
|              | V-            | GND        |

### 3.3.3.2 Sơ đồ nối dây



Hình 3. 12: Sơ đồ nối dây khối thu thập khí gas

### 3.3.3.3 Lưu đồ



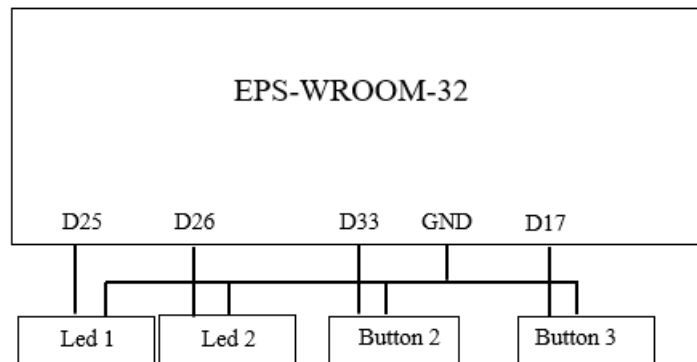
Hình 3. 13: Lưu đồ khối thu thập khí gas

### 3.3.3.4 Giải thích chương thu thập dữ liệu khí gas

|  |   |
|--|---|
| <pre>#define SENSOR 34 #define buzzer_gas 15  void control_gas() { int sensorValue = 0;  pinMode(buzzer_gas, OUTPUT);  sensorValue = analogRead(SENSOR);  if(sensorValue &gt; 1000) digitalWrite(buzzer_gas,HIGH); else digitalWrite(buzzer_gas,LOW);  Firebase.setFloat(firebaseData, "/gas",sensorValue);  delay(500); }</pre> | <p>Khai báo SENSOR cho chân 34<br/>Khai báo buzzer_gas cho chân 15</p> <p>Gọi hàm control_gas</p> <p>Khai báo và đặt giá trị bằng 0 cho biến sensorValue</p> <p>Khai báo chân buzzur_gas là output</p> <p>Đọc và đưa giá trị từ cảm biến vào sensorValue.</p> <p>Nếu sensorValue &gt; 1000 thì đưa ngõ ra buzzer_gas lên mức cao, buzzer sẽ kêu, nếu không thì đưa xuống mức thấp, buzzer không kêu.</p> <p>Đưa giá trị của sensorValue lên firebase kèm theo dòng “/gas”</p> |
|--|---|

### 3.3.4 Khối điều khiển Led

#### 3.3.4.1 Sơ đồ khối



Hình 3. 14: Sơ đồ khối điều khiển led

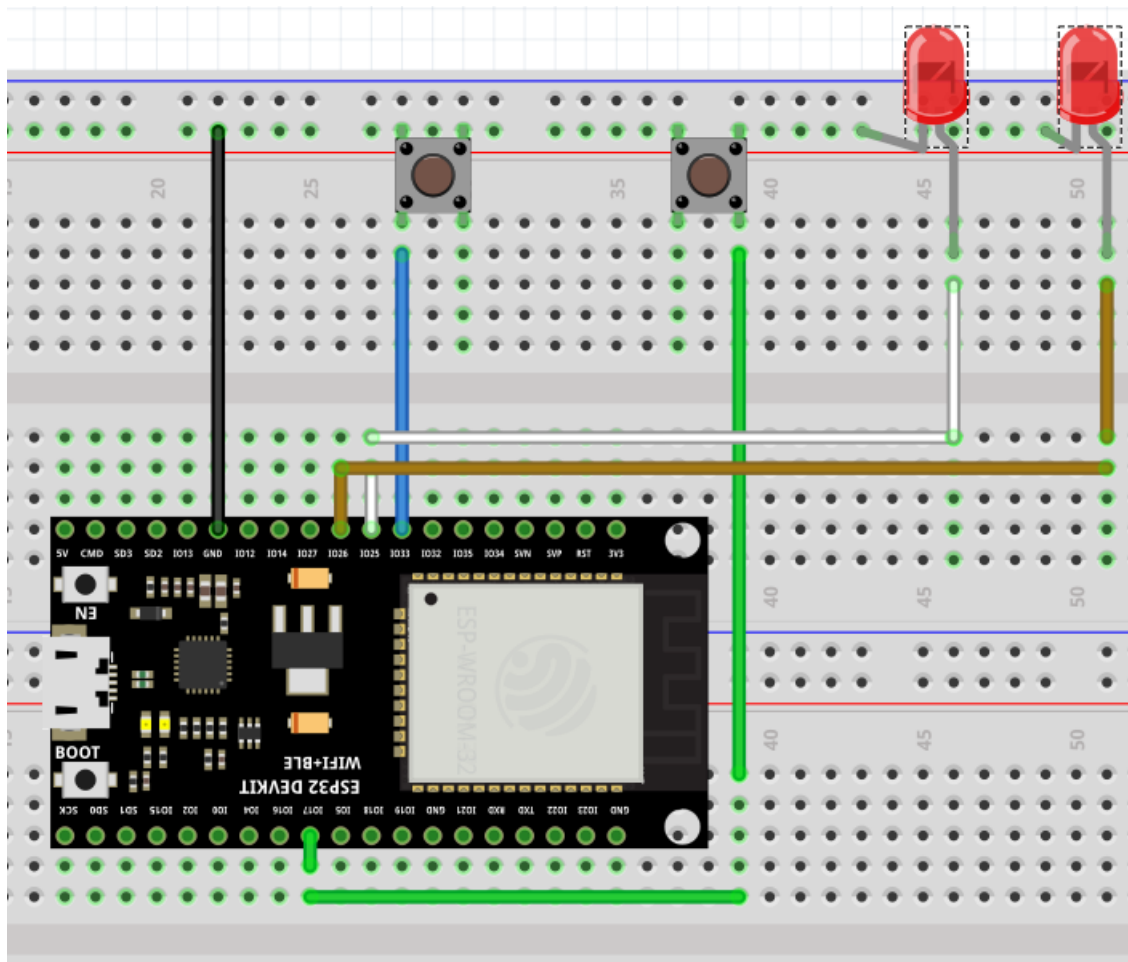
Bảng 3. 3: Kết nối Led với ESP32

| Thiết bị | Chân    | ESP32 |
|----------|---------|-------|
| Led 1    | Anode   | D25   |
|          | Cathode | GND   |
| Led 2    | Anode   | D26   |
|          | Cathode | GND   |

Kết nối Button với ESP32: 1 chân Button 2 nối với GPIO 33, 1 chân nối GND.

1 chân Button 3 nối với GPIO 17, 1 chân nối GND.

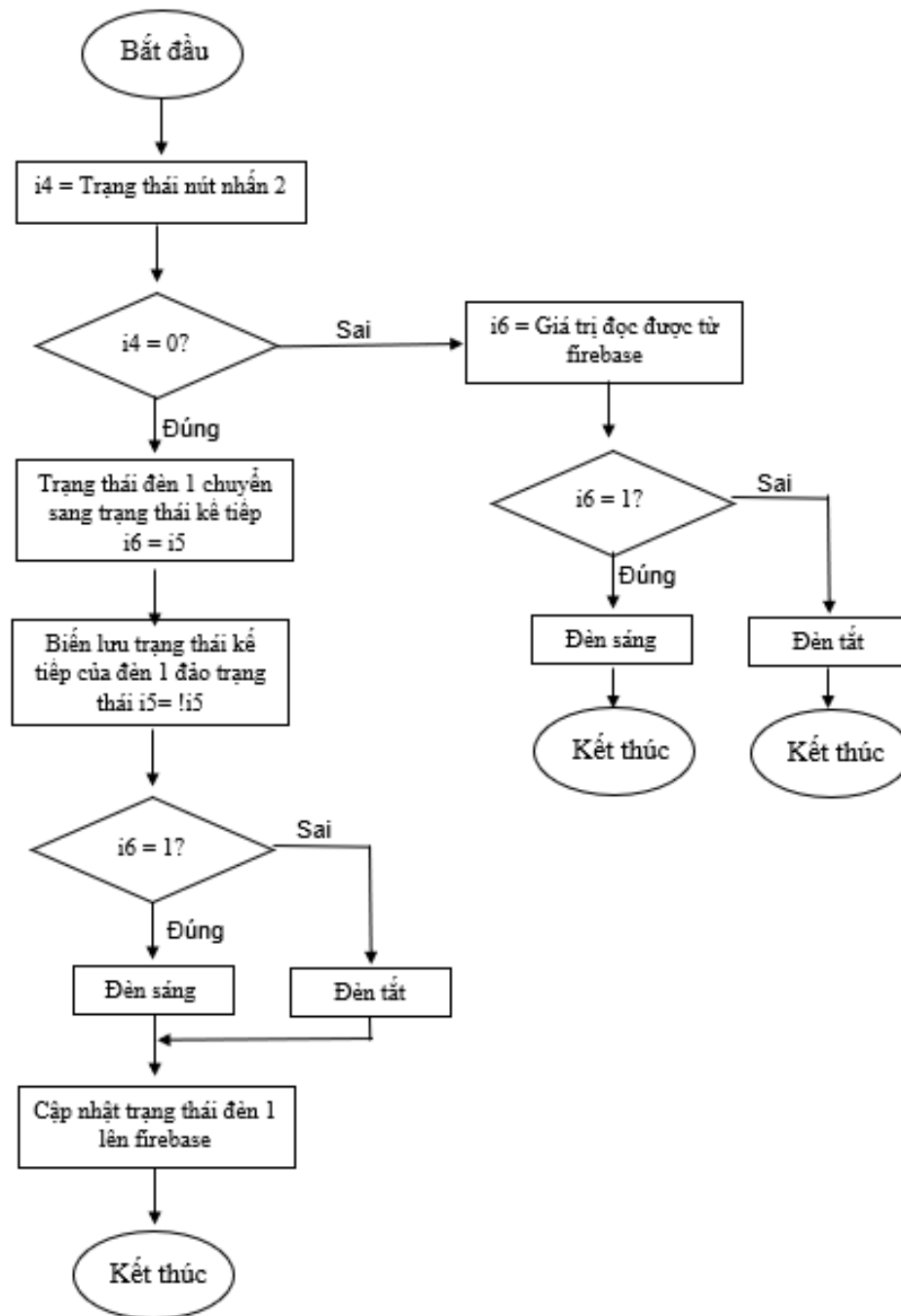
### 3.3.4.2 Sơ đồ nối dây



Hình 3. 15: Sơ đồ nối dây khối điều khiển led

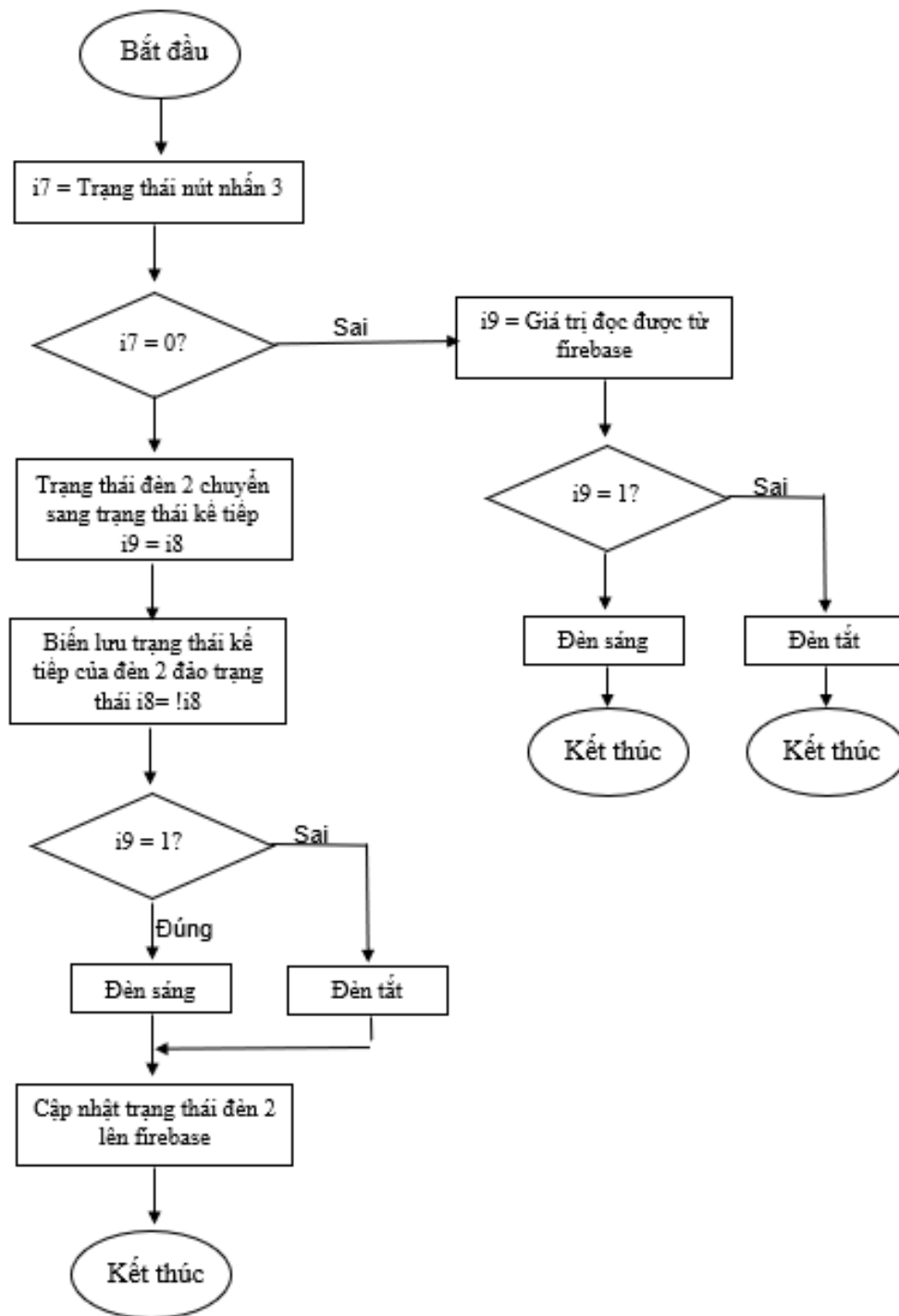
### 3.3.4.3 Lưu đồ

Led 1:



Hình 3. 16: Lưu đồ chương trình điều khiển led 1

## Led 2:



Hình 3. 17: Lưu đồ chương trình điều khiển led 2

### 3.3.4.4 Giải thích chương trình điều khiển đèn

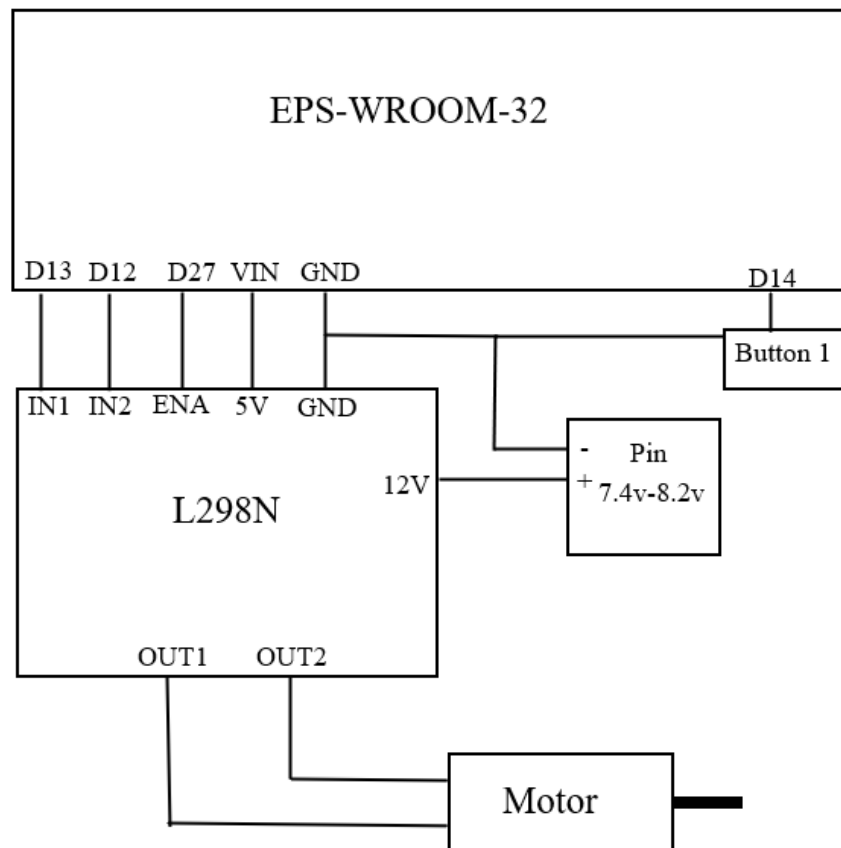
|   |  |
|---|--|
| <pre> #define light1 25 #define light2 26 #define button2 33 #define button3 17 //light1 bool i4=1; bool i5 =1; bool i6 ; //light2 bool i7=1; bool i8 =1; bool i9 ; void control_light1() {   pinMode(light1, OUTPUT);   pinMode(button2,INPUT_PULLUP);   i4= digitalRead(button2);   if(i4 == 0)   {     delay(50);     i6 = i5;     i5 =!i5;     if(i6 ==1) digitalWrite(light1, HIGH);     else digitalWrite(light1, LOW);      Firebase.setString(firebaseData,"/light1",String(i6));   }   else   if(Firebase.getString(firebaseData,"/light1"))   {     i6 = (firebaseData.stringData()).toInt();     if(i6 ==1) digitalWrite(light1, HIGH);     else digitalWrite(light1, LOW);   } } void control_light2()   pinMode(light2, OUTPUT);   pinMode(button3,INPUT_PULLUP);   i7= digitalRead(button3);   if(i7 == 0) </pre> | <p>         Khai báo đèn 1 là chân 25<br/>         Khai báo đèn 2 là chân 26<br/>         Khai báo nút nhấn 2 là chân 33<br/>         Khai báo nút nhấn 3 là chân 17<br/>         //đèn 1<br/>         i4: trạng thái của nút nhấn 2<br/>         i5: trạng thái đèn 1 kế tiếp<br/>         i6: trạng thái đèn 1 hiện tại<br/>         //đèn 2<br/>         i7: trạng thái của nút nhấn 3<br/>         i8: trạng thái đèn 2 kế tiếp<br/>         i9: trạng thái đèn 2 hiện tại<br/>         Hàm điều khiển đèn 1       </p> <p>         Đọc trạng thái của nút nhấn 2<br/>         Nếu nhấn nút<br/>         Trạng thái hiện tại bằng trạng<br/>         thái tiếp theo được gán trước<br/>         đó. Đảo trạng thái i5, được<br/>         trạng thái kế tiếp.<br/>         Nếu i6=1 đèn sáng<br/>         i6=0 đèn tắt       </p> <p>         Cập nhật trạng thái đèn 1 lên<br/>         firebase       </p> <p>         Nếu nút không được nhấn,<br/>         kiểm tra giá trị trên firebase<br/>         Chuyển đổi sang kiểu số<br/>         Nếu i6=1 đèn sáng<br/>         Nếu i6=0 đèn tắt       </p> <p>         Hàm điều khiển đèn 2       </p> <p>         Đọc trạng thái của nút nhấn 3<br/>         Nếu nhấn nút       </p> |
|---|--|



|   |  |
|---|--|
| <pre> {   delay(50);   i9 = i8;   i8 =!i8;   if(i9 == 1 ) digitalWrite(light2, HIGH);   else digitalWrite(light2, LOW);    Firebase.setString(firebaseData, "/light2",String(i9)); } else if(Firebase.getString(firebaseData, "/light2")) {   i9 = (firebaseData.stringData()).toInt();   if(i9 ==1) digitalWrite(light2, HIGH);   else digitalWrite(light2, LOW); } } </pre> | <p>Trạng thái hiện tại bằng trạng thái tiếp theo được gán trước đó. Đảo trạng thái i8, được trạng thái kế tiếp.<br/>Nếu i9=1 đèn sáng<br/>i9=0 đèn tắt</p> <p>Cập nhật trạng thái đèn 2 lên firebase</p> <p>Nếu nút không được nhấn, kiểm tra giá trị trên firebase<br/>Chuyển đổi sang kiểu số<br/>Nếu i9=1 đèn sáng<br/>Nếu i9=0 đèn tắt</p> |
|---|--|

### 3.3.5 Khối điều khiển quạt

#### 3.3.5.1 Sơ đồ khối



Hình 3. 18: Sơ đồ khối điều khiển quạt

Bảng 3. 4: Kết nối chân L298N với ESP32

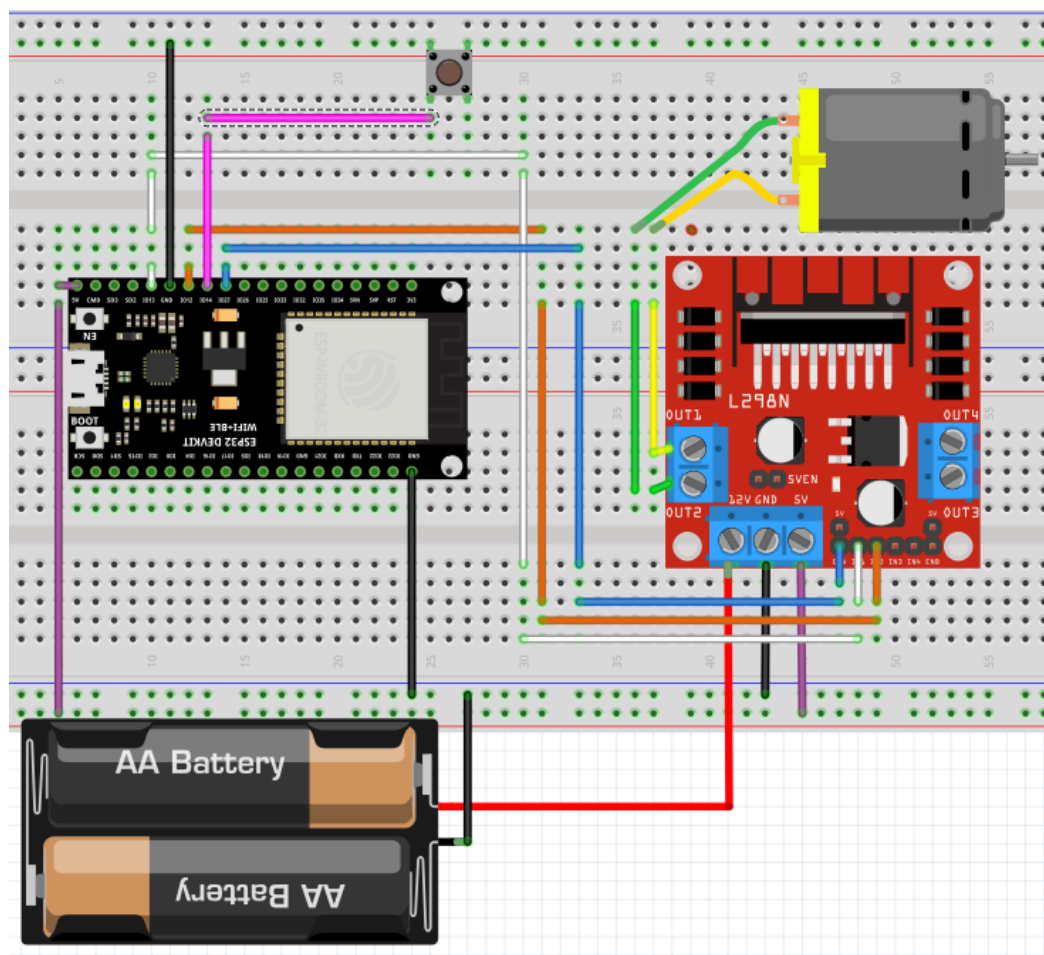
| L298N | ESP32 |
|-------|-------|
| IN1   | D13   |
| IN2   | D12   |
| ENA   | D27   |
| 5V    | VIN   |
| GND   | GND   |

Kết nối chân Button với ESP32: Kết nối 1 chân button với GPIO 14 và 1 chân nối GND.

Kết nối Motor với L298N: Nối lần lượt 2 chân của Motor với OUT1 và OUT2.

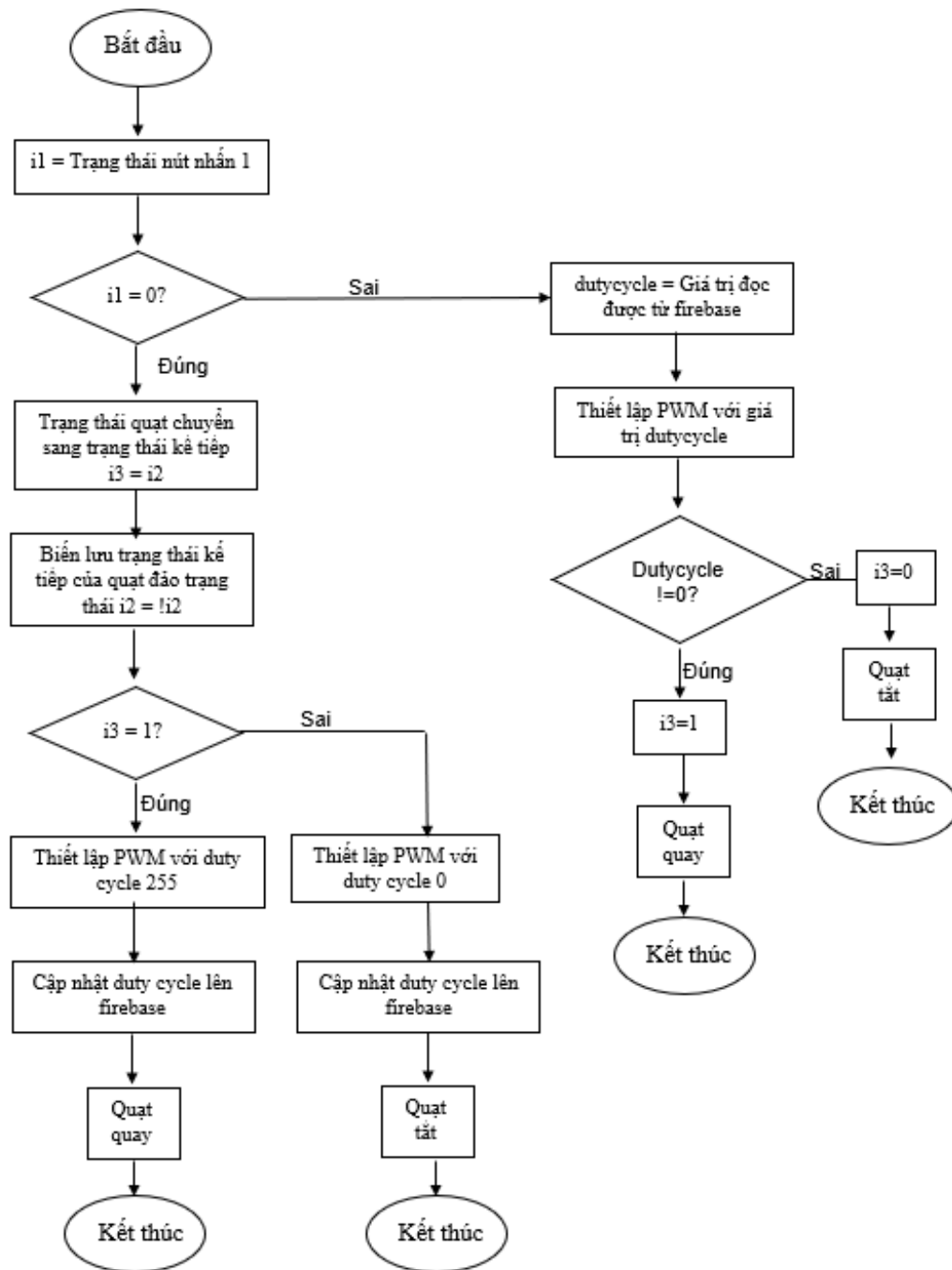
Sử dụng PIN có mức điện áp từ: 7.4V-8.2V để cấp nguồn cho L298N.

### 3.3.5.2 Sơ đồ nối dây



Hình 3. 19: Sơ đồ nối dây

### 3.3.5.3 Lưu đồ



Hình 3. 20: Lưu đồ chương trình điều khiển quạt

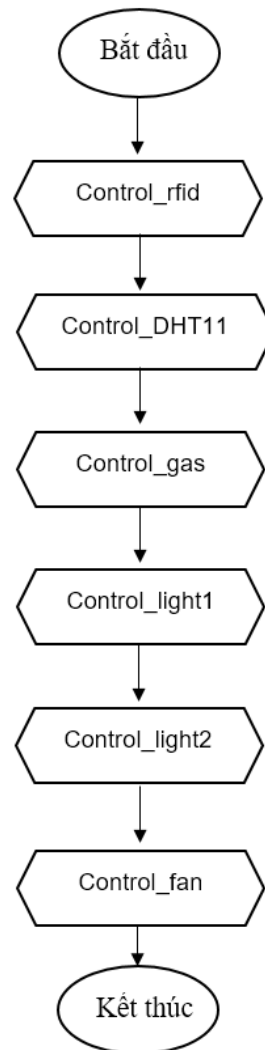
### 3.3.5.4 Giải thích chương trình chức năng điều khiển quạt

|   |  |
|---|--|
| <pre> #define fan_pin1 13 #define fan_pin2 12 #define control_speed 27 const int freq = 5000; const int ledChannel_2 = 2; const int resolution = 8;  bool i1=1; bool i2 =1; bool i3 ; int dutycycle =0; void setup {   ledcSetup(ledChannel_2, freq, resolution);   ledcAttachPin(control_speed, ledChannel_2); } void control_fan() {   pinMode(fan_pin1, OUTPUT);   pinMode(fan_pin2, OUTPUT);   pinMode(button1,INPUT_PULLUP);   pinMode(control_speed, OUTPUT);   i1= digitalRead(button1);   if(i1 == 0)   {     delay(250);     i3 = i2;     i2 =!i2;     if( i3 ==1)     {       ledcWrite(ledChannel_2, 255);        Firebase.setString(firebaseData,"/fan",255);     }else     {       ledcWrite(ledChannel_2, 0);       Firebase.setString(firebaseData,"/fan",0);     }   }else   { </pre> | <p>Khai báo chân IN1 L298N<br/> Khai báo chân IN2 L298N<br/> Khai báo chân ENA L298N<br/> Cấu hình thông số pwm: tần số 5000hz, kênh 2, độ phân giải 8 bit</p> <p>Khao báo i1: biến đọc trạng thái nút nhấn. i2: trạng thái quạt kế tiếp<br/> i3: trạng thái của quạt<br/> Khai báo chu kì mức cao mặc định là 0</p> <p>Cấu hình PWM<br/> Chọn GPIO cần xuất tín hiệu PWM</p> <p>Tạo hàm điều khiển quạt</p> <p>Thiết lập 2 chân điều khiển là output<br/> Thiết lập nút nhấn là input_pullup<br/> Thiết lập chân điều khiển tốc độ</p> <p>Đọc giá trị từ nút nhấn<br/> Nếu nhấn nút thì giá trị i1=0<br/> Khi nhấn nút i3 được gán bằng i2<br/> Và đảo giá trị i2 chuẩn bị cho trạng thái kế tiếp.</p> <p>Nếu i3 =1 thì thiết lập dutycycle=255</p> <p>Đồng bộ dutycycle lên firebase</p> <p>Nếu i3=0 thì thiết lập dutycycle=0<br/> Đồng bộ tốc độ quạt lên firebase</p> |
|---|--|

|   |  |
|---|--|
| <pre>         if(Firebase.getString(firebaseData,"/fan")) dutyCycle = (firebaseData.stringData()).toInt();         ledcWrite(ledChannel_2, dutyCycle);         if(dutyCycle!= 0) i3=1;         else i3 =0;     }     if(i3 ==1)     {         digitalWrite(fan_pin1, LOW);         digitalWrite(fan_pin2, HIGH);     }else     {         digitalWrite(fan_pin1, LOW);         digitalWrite(fan_pin2, LOW);     } } </pre> | <p>Nếu nhận giá trị thay đổi từ firebase.Tiến hành đọc và gán giá trị vào biến dutyCycle. Và Thiết lập xung PWM</p> <p>Cập nhật trạng thái quạt</p> <p>Nếu i3 =1 thiết lập chân tín hiệu điều khiển chiều quay của quạt</p> <p>Nếu i3=0 tắt quạt</p> |
|---|--|

### 3.4 Chương trình chính

#### 3.4.1 Lưu đồ



*Hình 3. 21: Lưu đồ chương trình chính*

Control\_rfid : Chương trình điều khiển khóa cửa dùng thẻ RFID.

Control\_DHT11: Chương trình đo độ ẩm, nhiệt độ.

Control\_gas: Chương trình đo khí gas.

Control\_light1: Chương trình điều khiển đèn 1.

Control\_light2: Chương trình điều khiển đèn 2.

Control\_fan: Chương trình điều khiển quạt.

### 3.4.2 Giải thích chương trình điều khiển chính

|   |   |
|---|---|
| <pre>#include &lt;FirebaseESP32.h&gt; #include &lt;WiFi.h&gt; #define FIREBASE_HOST "https://smart-home- a8b72-default-rtdb.firebaseio.com/" #define FIREBASE_Authorization_key "vvHhSMACxAXEavaTW3ly7uSyeJ4bOyZrhcv2 2Gr" #define WIFI_SSID "VNPT Hung" #define WIFI_PASSWORD "12345789" FirebaseData firebaseData; void setup() {   Serial.begin(9600);   WiFi.begin (WIFI_SSID, WIFI_PASSWORD);   Serial.print("Connecting...");   while (WiFi.status() != WL_CONNECTED)   {     Serial.print(".");     delay(300);   }   Serial.println();   Serial.print("IP Address: ");   Serial.println(WiFi.localIP());   Serial.println();   Firebase.begin(FIREBASE_HOST,FIREBASE_ Authorization_key); }</pre> | <p>Khai báo các thư viện</p> <p>Khai báo liên kết database</p> <p>Khai báo key secret</p> <p>Tên Wifi</p> <p>Mật khẩu Wifi</p> <p>Thiết lập kết nối wifi</p> <p>Khởi tạo kết nối firebase</p> |
|---|---|



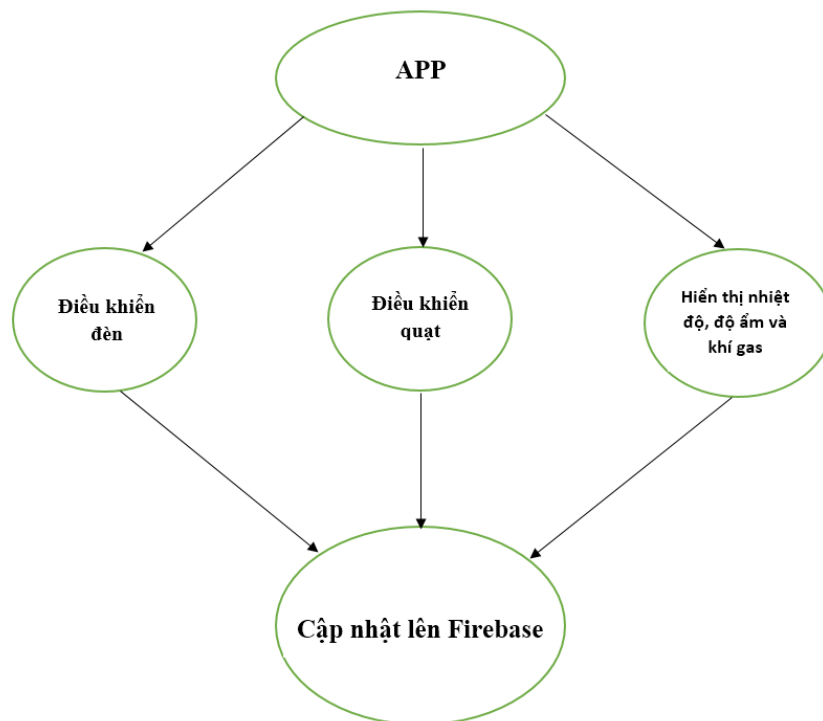
|   |   |
|---|---|
| <pre> void loop() {   control_rfid();   control_DHT11();   control_gas();   control_light1();   control_light2();   control_fan(); } </pre> | <p>Gọi hàm điều khiển RFID</p> <p>Gọi hàm điều khiển DHT11</p> <p>Gọi hàm điều khiển gas</p> <p>Gọi hàm điều khiển đèn 1</p> <p>Gọi hàm điều khiển đèn 2</p> <p>Gọi hàm điều khiển quạt</p> |
|---|---|

### 3.5 Ứng dụng điều khiển

#### 3.5.1 Tính năng

Ứng dụng mà nhóm chúng em làm bao gồm 3 chức năng:

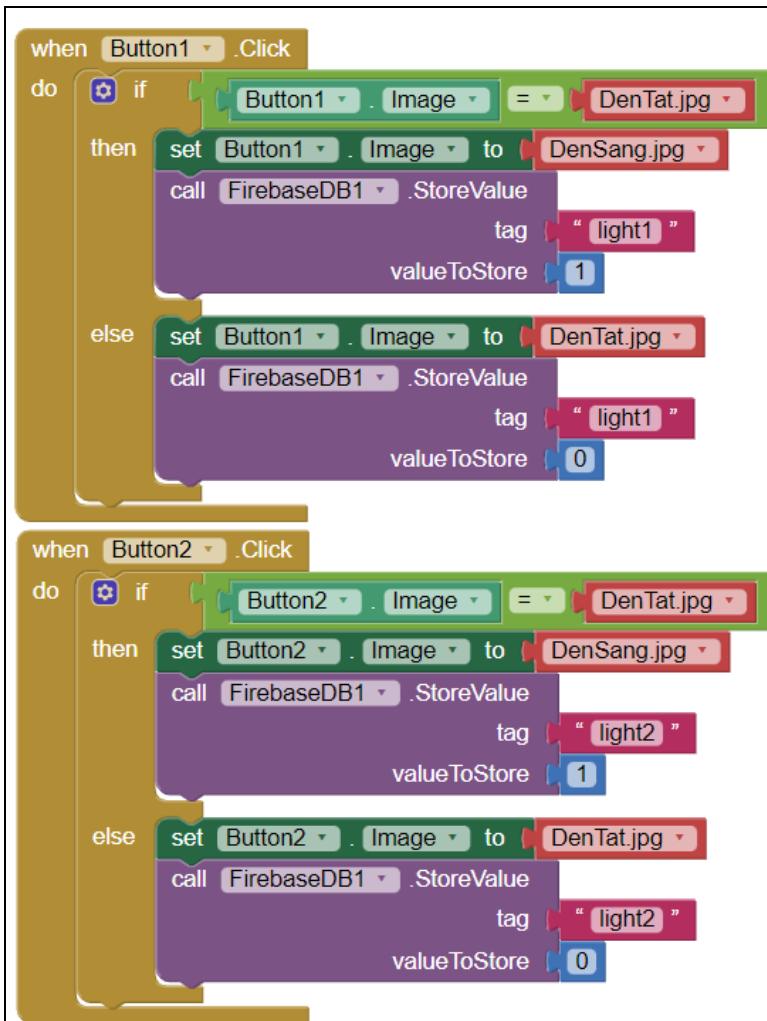
- Điều khiển đèn.
- Điều khiển quạt.
- Hiển thị nhiệt độ, độ ẩm và khí gas.



*Hình 3. 22: Các tính năng của ứng dụng điều khiển*

### 3.5.2 Chức năng của từng khối

#### 3.5.2.1 Khối điều khiển đèn



Các khối bên thực hiện chức năng điều khiển đèn khi các nút tương ứng được nhấn.

Khi nhấn nút, ứng dụng sẽ kiểm tra hình ảnh của nút nhấn:

- Nếu ảnh của nút nhấn đang là ảnh đèn tắt sẽ đặt lại thành ảnh đèn sáng và cập nhật giá trị 1 cho biến "light1"/"light2" trên Firebase.
- Ngược lại, nếu ảnh nút nhấn khác ảnh đèn tắt thì sẽ đặt lại thành ảnh đèn tắt và cập nhật giá trị 0 cho biến "light1"/"light2" trên Firebase.

### 3.5.2.2 Khối điều khiển quạt

```
when ButtonFan . Click
do
  if ButtonFan . Image = buttonOFF.png
  then
    set ButtonFan . Image to buttonON.png
    set fan . Picture to Quatchay.jpg
    set Slider1 . ThumbPosition to Slider1 . MinValue
    call FirebaseDB1 . StoreValue
      tag "fan"
      valueToStore Slider1 . ThumbPosition
  else
    set ButtonFan . Image to buttonOFF.png
    set fan . Picture to Quattat.jif
    set Slider1 . ThumbPosition to 0
    call FirebaseDB1 . StoreValue
      tag "fan"
      valueToStore 0

when Slider1 . PositionChanged
thumbPosition
do
  if Slider1 . ThumbPosition ≠ Slider1 . MinValue
  then
    set fan . Picture to Quatchay.jpg
    set ButtonFan . Image to buttonON.png
    call FirebaseDB1 . StoreValue
      tag "fan"
      valueToStore Slider1 . ThumbPosition
  else
    call FirebaseDB1 . StoreValue
      tag "fan"
      valueToStore Slider1 . ThumbPosition
```

MaxValue

MinValue

Các khối bên thực hiện chức năng điều khiển quạt

**\*Khởi nút nhấn**

Khi nút điều khiển được ấn, ứng dụng sẽ kiểm tra hình ảnh của nút nhấn:

- Nếu hình ảnh của nút nhấn là OFF thì sẽ đặt lại thành ON, đặt hình ảnh của quạt thành quạt chạy, đặt thanh trượt điều khiển tốc độ quạt về vị trí nhỏ nhất và cập nhật vị trí của thanh trượt lên biến “fan” trên Firebase.

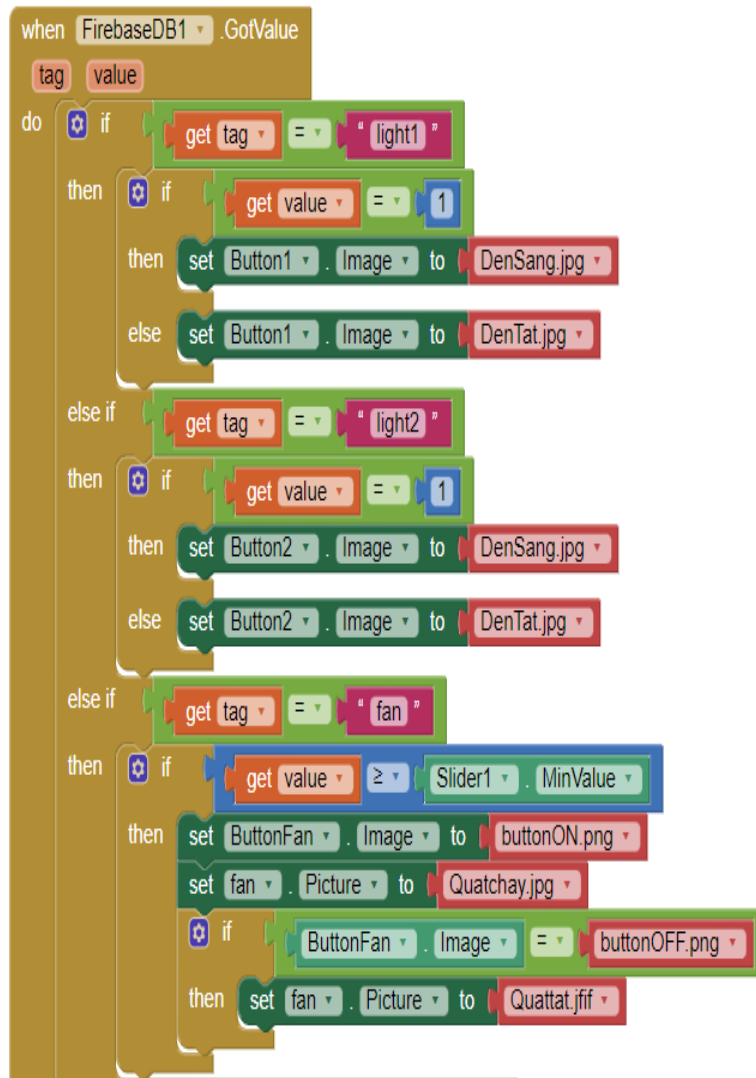
- Ngược lại, nếu hình ảnh nút nhấn là ON thì sẽ đặt lại thành OFF, đặt hình ảnh của quạt thành quạt tắt, đặt thanh trượt điều khiển tốc độ quạt về 0 và cập nhật giá trị 0 lên biến “fan” trên Firebase.

**\*Khởi thanh trượt**

Khi vị trí thanh trượt thay đổi, ứng dụng sẽ kiểm tra vị trí của nó:

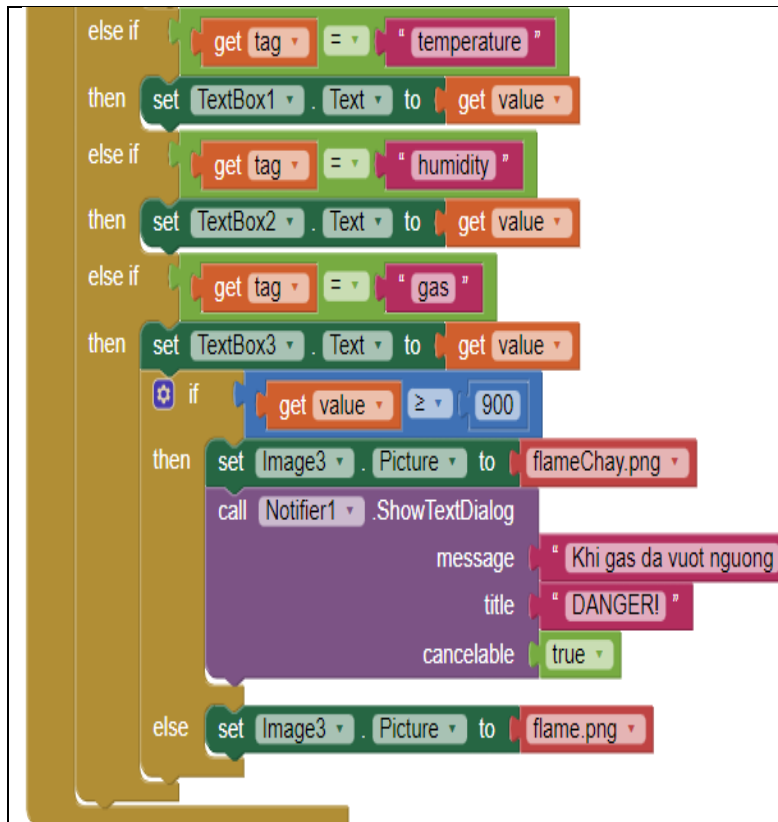
- Nếu vị trí thanh trượt khác vị trí nhỏ nhất của nó thì sẽ đặt hình ảnh của quạt thành quạt

|  |   |
|--|---|
|  | <p>chạy, nút nhấn điều khiển quạt sẽ thành ON và vị trí của thanh trượt sẽ được cập nhật lên biến “fan” trên Firebase.</p> <ul style="list-style-type: none"> <li>- Ngược lại thì cập nhật vị trí của thanh trượt lên biến “fan” trên Firebase.</li> </ul> <p>*Lưu ý:</p> <ul style="list-style-type: none"> <li>- Vị trí lớn nhất của thanh trượt hay tốc độ quạt lớn nhất ứng với vị trí 255 của thanh trượt.</li> <li>- Vị trí nhỏ nhất sẽ ứng với vị trí 100 của thanh trượt.</li> <li>- Khi quạt tắt thì vị trí thanh trượt là 0.</li> </ul> |
|--|---|



Firestore sau khi cập nhật giá trị mà ứng dụng gửi lên sẽ tiến hành thực hiện các câu lệnh bên dưới

- Nếu Firestore nhận giá trị của biến “light1”/”light2” là 1 thì sẽ đặt hình ảnh cho đèn thành đèn sáng và ngược lại khi nhận giá trị 0 cho hai biến trên thì đặt cho đèn thành đèn tắt.
- Nếu Firestore không nhận giá trị nào cho biến “light1”/”light2” thì sẽ thực hiện tiếp câu lệnh bên dưới.
- Nếu Firestore nhận giá trị của biến “fan” (vị trí của thanh trượt) thì sẽ kiểm tra vị trí thanh trượt. Nếu vị trí thanh trượt lớn hơn hoặc bằng vị trí nhỏ nhất thì sẽ đặt hình ảnh cho nút điều khiển quạt thành ON và đặt hình ảnh của quạt thành quạt chạy, ngược lại nếu vị trí thanh



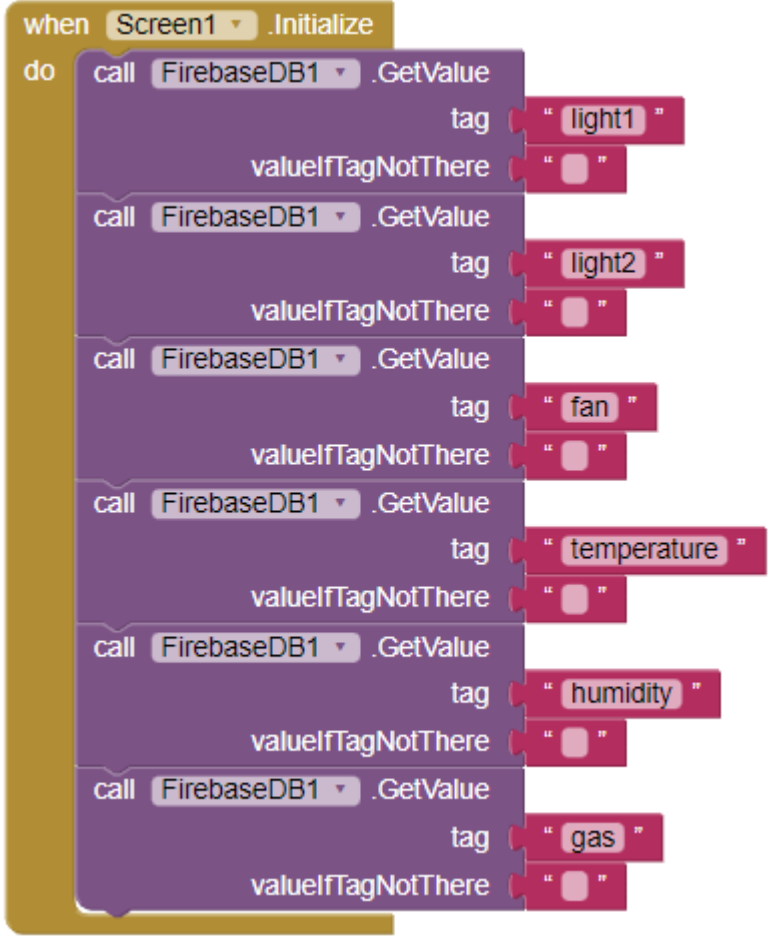
trượt bé hơn vị trí nhỏ nhất của nó thì sẽ đặt nút thành OFF và hình ảnh quạt thành quạt tắt.

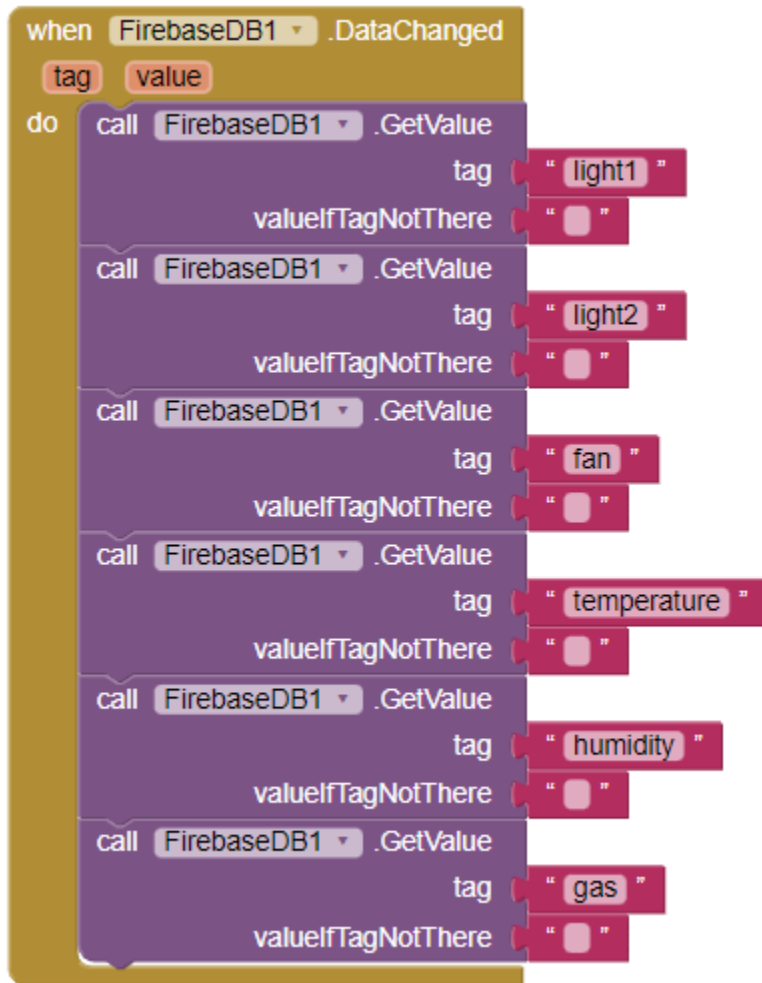
- Nếu Firebase không nhận giá trị nào cho biến “fan” thì sẽ thực hiện tiếp câu lệnh bên dưới.

|  |   |
|--|---|
|  | <p>- Nếu Firebase nhận giá trị của nhiệt độ, độ ẩm và khí gas từ các thiết bị cảm biến thì sẽ đưa giá trị vào ô hiển thị tương ứng cho từng mục.</p> <p>- Riêng đối với biến “gas”, nếu giá trị nhận được lớn hơn hoặc bằng 1500 thì sẽ hiện một thông báo với tiêu đề DANGER! và nội dung thông báo là “Khi gas đã vượt ngưỡng”.</p> |
|--|---|



### 3.5.2.3 Các chức năng khác

|  |  |
|--|--|
|  <pre>when Screen1.Initialize do   call FirebaseDB1.GetValue     tag "light1"     valueIfTagNotThere " "   call FirebaseDB1.GetValue     tag "light2"     valueIfTagNotThere " "   call FirebaseDB1.GetValue     tag "fan"     valueIfTagNotThere " "   call FirebaseDB1.GetValue     tag "temperature"     valueIfTagNotThere " "   call FirebaseDB1.GetValue     tag "humidity"     valueIfTagNotThere " "   call FirebaseDB1.GetValue     tag "gas"     valueIfTagNotThere " "</pre> | <p>Khối này thực hiện chức năng cập nhật các giá trị của các biến bên trong từ Firebase khi vừa khởi tạo hay vừa khởi động ứng dụng.</p> |
|--|--|



Khối này sẽ thực hiện chức năng cập nhật các giá trị của các biến bên trong từ Firebase bất cứ khi nào nhận thấy sự thay đổi.

## CHƯƠNG 4: KẾT QUẢ

### 4.1 Chức năng điều khiển khóa cửa dùng thẻ RFID



*Hình 4. 1: Thiết kế mặt ngoài*

Hình 4.1 là mặt ngoài với đầu đọc RFID-RC522, màn hình LCD với 2 LED báo hiệu.



*Hình 4. 2: Quét thẻ đúng, mở cửa*

Hình 4.2 mô tả khi đầu đọc nhận được thẻ đúng, cửa sẽ được mở ra đồng thời thông báo mở cửa hiện trên LCD cùng LED xanh sáng và buzzer kêu báo hiệu.



*Hình 4. 3: Quét thẻ đúng, đóng cửa*

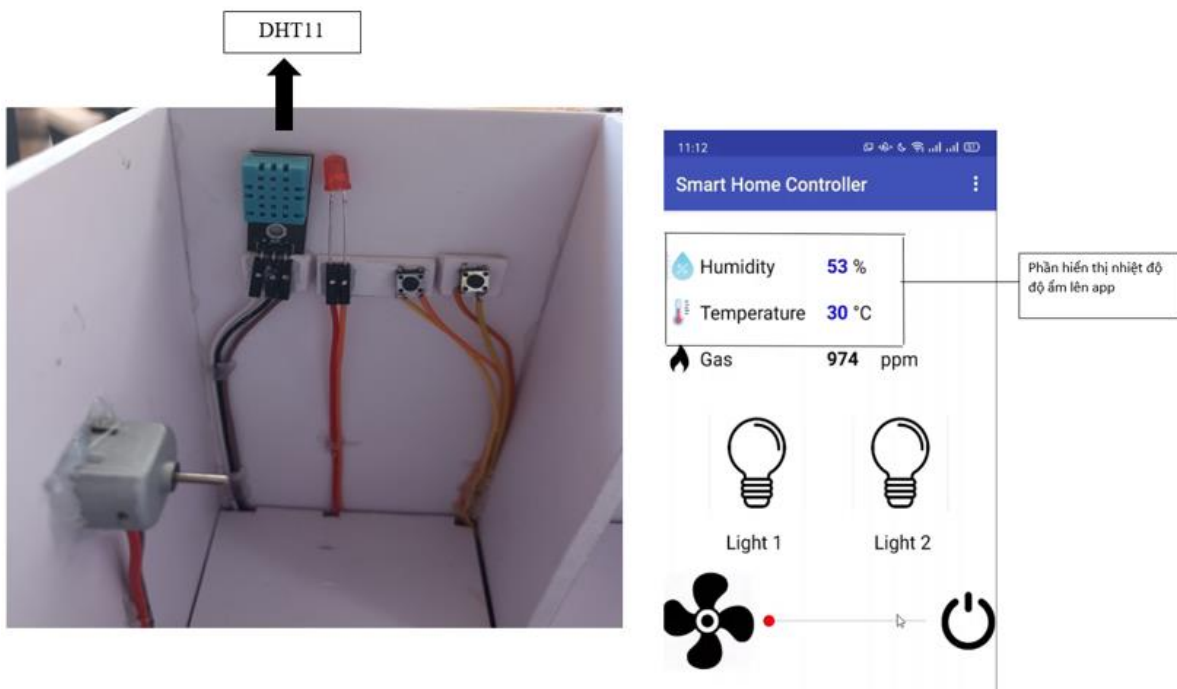
Hình 4.3 mô tả khi đầu đọc nhận được thẻ đúng lần nữa, cửa sẽ được đóng lại đồng thời thông báo đóng cửa hiện trên LCD cùng LED xanh sáng và buzzer kêu báo hiệu.



*Hình 4. 4: Quét thẻ sai*

Hình 4.4 mô tả khi đầu đọc nhận được thẻ sai, cửa sẽ không được mở đồng thời thông báo thẻ sai quét lại hiện trên LCD cùng LED đỏ sáng và buzzer kêu báo hiệu.

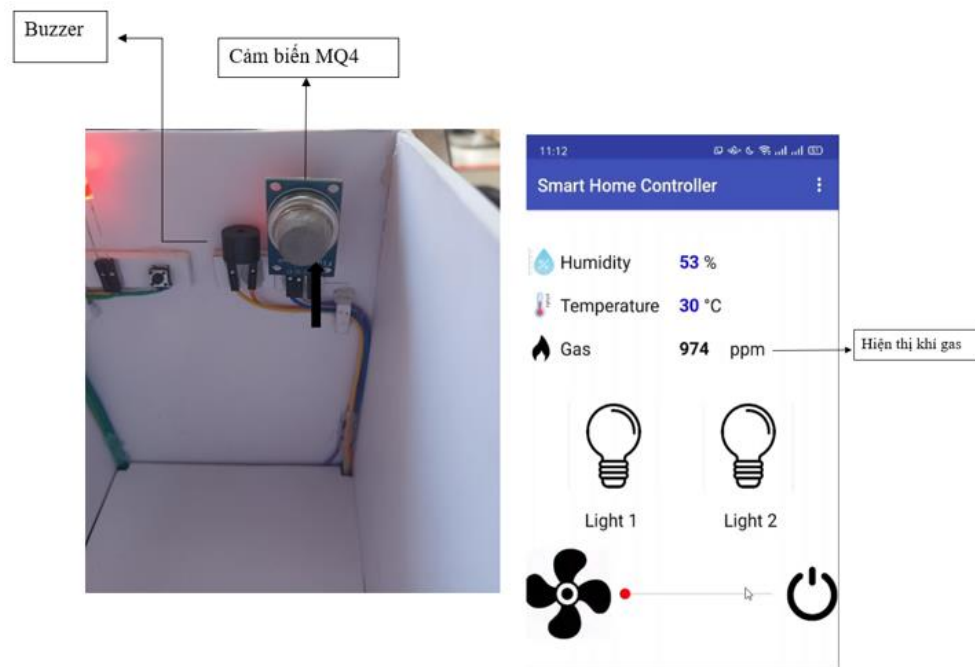
## 4.2 Chức năng đo độ ẩm và nhiệt độ



Hình 4. 5: Chức năng đo độ ẩm và nhiệt độ

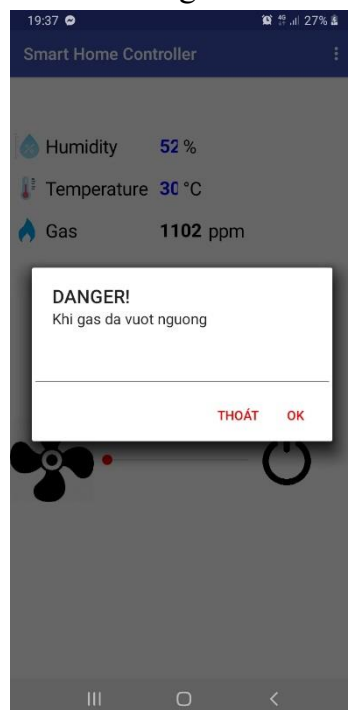
Hình 4.5 mô tả khi nhiệt độ và độ ẩm thay đổi, thì cảm biến nhận dữ liệu và lưu trữ lên firebase. Chương trình điều khiển sẽ tự động cập nhật các giá trị trên.

### 4.3 Chức năng đo khí gas



Hình 4. 6: Chức năng đo khí gas

Hình 4.6 mô tả khi khí gas thay đổi, thì cảm biến nhận dữ liệu và lưu trữ lên firebase. Chương trình điều khiển sẽ tự động cập nhật các giá trị trên.



Hình 4. 7: Cảnh báo khí gas

Hình 4.7 mô tả khi lượng khí ga vượt ngưỡng cho phép thì còi buzzer phát chuông báo động và đồng thời một cảnh báo hiển thị trên app.

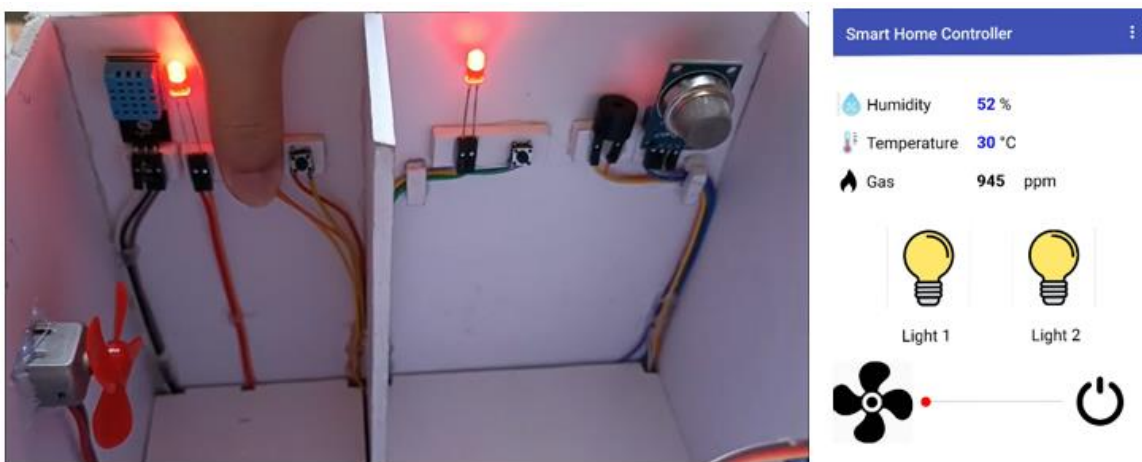


#### 4.4 Chức năng điều khiển đèn



Hình 4. 8: Bật đèn 1

Hình 4.8 mô tả khi nhấn nút điều khiển đèn 1 thì đèn 1 sẽ sáng, tương tự có thể điều khiển trên app.



Hình 4. 9: Bật đèn 2

Hình 4.9 mô tả khi nhấn nút điều khiển đèn 2 thì đèn 2 sẽ , tương tự có thể điều khiển trên app.



*Hình 4. 10: Tắt đèn 1*

Hình 4.10 mô tả khi nhấn nút điều khiển đèn 1 lần nữa thì đèn 1 sẽ tắt và được cập nhật lên app.



*Hình 4. 11: Tắt đèn 2*

Hình 4.11 mô tả khi nhấn nút điều khiển đèn 2 lần nữa thì đèn 2 sẽ tắt và được cập nhật lên app.

## 4.5 Chức năng điều khiển quạt



Hình 4. 12: Bật quạt

Hình 4.12 mô tả khi nhấn nút điều khiển quạt, quạt sẽ được bật ở tốc độ mạnh nhất và hiển thị trên app đồng thời có thể điều khiển tốc độ trên app bằng cách trượt thanh trượt.



Hình 4. 13: Điều chỉnh tốc độ và tắt quạt

Hình 4.13 mô tả khi nhấn nút điều khiển quạt lần nữa, quạt sẽ được tắt và hiển thị trên app.

## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Tổng kết đề tài

Thông qua việc thực hiện đề tài thiết kế nhà thông minh, nhóm chúng em đã tiếp thu thêm nhiều kiến thức cũng như các kỹ năng trong việc xây dựng hệ thống nhà thông minh nói riêng cũng như hệ thống IoT hoàn chỉnh nói chung. Về cơ sở lý thuyết, nhóm đã tìm hiểu và nghiên cứu thêm nhiều kỹ thuật liên quan đến chip ESP32, module ESP32 WROOM, các khả năng đặc trưng của chúng, từ đó áp dụng vào trong hệ thống nhà thông minh. Tiếp theo, nhóm nghiên cứu đến lý thuyết của các thiết bị cảm biến như DHT11, MQ4, thiết bị điều khiển khoá của như RFID, RC522, hiển thị thông báo như LCD, buzzer và cuối cùng là module L298N. Với cơ sở lý thuyết trên của các thiết bị, nhóm đã áp dụng vào trong việc xây dựng mô hình, đề ra kế hoạch nối dây, chọn chân hợp lý cho ESP32, vị trí đặt các thiết bị và những thiết kế khác để có thể đạt mô hình hoàn chỉnh. Cuối cùng, nhóm đã tìm hiểu về MIT APP Inventor, áp dụng vào trong quá trình xây dựng một ứng dụng điều khiển thiết bị thích hợp cho mô hình nhà thông minh của nhóm.

Về quá trình thiết kế, xây dựng, nhóm đã tiến hành thực hiện từng chức năng có trong mô hình nhà thông minh, bao gồm chức năng điều khiển led, chức năng điều khiển quạt, chức năng mở đóng khóa cửa bằng RFID, chức năng đo nhiệt độ độ ẩm, chức năng đo khí ga và cảnh báo khi khí ga vượt ngưỡng. Sau quá trình đó, nhóm tiến hành tích hợp các chức năng đó lại, tiến hành thiết kế mô hình, sau đó đặt các thiết bị ở vị trí hợp lý, đi dây và cuối cùng là nạp code, hoàn thiện sản phẩm. Trong quá trình trên, ứng dụng được xây dựng theo từng chức năng đã được định trước, cuối cùng là hoàn thiện ứng dụng điều khiển. Về tổng quan, mô hình nhà thông minh và ứng dụng thu nhận dữ liệu, điều khiển một số thiết bị có trong ngôi nhà đã được hoàn thành.

Tuy nhiên, đề tài nhóm vẫn còn nhiều vấn đề hạn chế. Các thiết bị trong ngôi nhà vẫn còn độ trễ trong việc phản hồi cao. Mô hình chỉ thực hiện cho một căn phòng chứ chưa là một căn nhà hoàn chỉnh. Về tính năng, đề tài vẫn còn thiếu nhiều tính năng như cảm biến có người, cảm biến mưa, tự động bật quạt khi nhiệt độ cao,...

## **5.2 Định hướng phát triển**

Về định hướng phát triển, nhóm sẽ hướng đến xây dựng một căn nhà hoàn chỉnh thay vì một căn phòng như kết quả hiện tại. Đồng thời, nhóm muốn hướng đến xây dựng thêm nhiều chức năng cho căn nhà như cảm biến người, tự động bật tắt các thiết bị, đặc biệt là hướng đến việc tích hợp hệ điều hành thời gian thực để giảm độ trễ trong việc phản hồi giữa các thiết bị. Cuối cùng là về phần ứng dụng, nhóm hướng đến tăng thêm nhiều tương tác giữa ứng dụng và các thiết bị, thiết kế thêm nhiều chế độ thông minh nhằm phục vụ cho nhu cầu của người dùng.

## Tài liệu tham khảo

- [1] "HIỆU QUẢ CÔNG NGHỆ IOT TRONG VIỆC GIÁM SÁT ĐIỀU KHIỂN HỆ THỐNG LÀM LẠNH," [Online]. Available: <https://iotdaiviet.com/dich-vu/hieu-qua-cong-nghe-iot-trong-viec-giam-sat-dieu-khien-he-thong-lam-lanh/>. [Accessed 12 12 2022].
- [2] "HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN CÁC THIẾT BỊ TRONG NHÀ," [Online]. Available: <https://kctn.vhu.edu.vn/vi/tin-moi-1/he-thong-giam-sat-va-dieu-khien-cac-thiet-bi-trong-nha>. [Accessed 12 12 2022].
- [3] Rosslin John Robles & Tai-hoon Kim, "Applications, Systems and Methods in Smart Home Technology: A Review," *International Journal of Advanced Science and Technology*, vol. 15, 2010.
- [4] Espressif Systems, "ESP32 Series Datasheet," 2019.
- [5] "TÌM HIỂU VỀ RFID," [Online]. Available: <https://maviet.com.vn/tu-van/cong-nghe-rfid.html>. [Accessed 4 12 2022].
- [6] Cửa hàng HSHOP ĐIỆN TỬ VÀ ROBOT, "Mạch RFID NFC 13.56Mhz RC522," [Online]. Available: <https://hshop.vn/products/mach-rfid-rc522-nfc-13-56mhz>. [Accessed 4 12 2022].
- [7] "RFID," [Online]. Available: <https://vi.wikipedia.org/wiki/RFID>. [Accessed 4 12 2022].
- [8] "RFID là gì? Ứng dụng của RFID trong sản xuất," 24 2 2020. [Online]. Available: [https://baoanjsc.com.vn/du-an/rfid-la-gi-ung-dung-cua-rfid-trong-san-xuat\\_2\\_69\\_31600\\_vn.aspx](https://baoanjsc.com.vn/du-an/rfid-la-gi-ung-dung-cua-rfid-trong-san-xuat_2_69_31600_vn.aspx). [Accessed 4 12 2022].
- [9] "MÀN HÌNH TINH THỂ LÔNG (LCD)," [Online]. Available: <https://dientutuonglai.com/man-hinh-tinh-the-long-lcd.html>. [Accessed 4 12 2022].

- [10] CỬA HÀNG IC ĐÂY RỒI, [Online]. Available: <https://icdayroi.com/lcd1602-xanh-la-5v>. [Accessed 4 12 2022].
- [11] "Mạch chuyển đổi I2C cho LCD," [Online]. Available: <https://iotmaker.vn/mach-chuyen-doi-i2c-cho-lcd.html>. [Accessed 4 12 2022].
- [12] Linh Kiện Điện Tử 3M, "Động Cơ Servo SG90, Góc Quay 180 Độ," [Online]. Available: <https://chotroi.vn/dong-co-servo-sg90>. [Accessed 12 12 2022].
- [13] Long Phan, "Module điều khiển động cơ L298N," [Online]. Available: <https://blog.mecsu.vn/module-dieu-khien-dong-co-l298n/>. [Accessed 12 12 2022].
- [14] Công ty Cổ Phần Điện Tử 360, "MODULE ĐIỀU KHIỂN MOTOR L298N," [Online]. Available: <https://dientu360.com/module-dieu-khien-motor-l298n>. [Accessed 12 12 2022].
- [15] Elprocus, "What is a Buzzer : Working & Its Applications," [Online]. Available: <https://www.elprocus.com/buzzer-working-applications/>. [Accessed 12 12 2022].
- [16] "Cảm biến DHT11 và hoạt động của nó," [Online]. Available: <https://vi.jf-parede.pt/dht11-sensor-its-working>. [Accessed 12 12 2022].
- [17] Elprocus, "MQ4 Methane Gas Sensor : Pin Configuration & Its Applications," [Online]. Available: <https://www.elprocus.com/mq4-methane-gas-sensor/>. [Accessed 12 12 2022].
- [18] ZinPro, "Giới Thiệu Về Lập trình App Inventor," [Online]. Available: [https://zinpro.vn/vn/gioi-thieu-ve-lap-trinh-app-inventor\\_1106.html](https://zinpro.vn/vn/gioi-thieu-ve-lap-trinh-app-inventor_1106.html). [Accessed 12 12 2022].