BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC PHENIKAA**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**ĐỒ ÁN TÍCH HỢP LIÊN NGHÀNH**

**ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO**

|  |  |  |  |
| --- | --- | --- | --- |
| Sinh viên thực hiện: | Phạm Anh Toàn | 20010650 | K14 ĐTVT |
|
|  |  |  |

Giảng viên hướng dẫn: TS. Nguyễn Thị Thanh Quỳnh

Khoa: Điện – Điện tử

***Hà Nội, 26 tháng 11 năm 2024***

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC PHENIKAA**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**ĐỒ ÁN TÍCH HỢP LIÊN NGHÀNH**

**ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO**

|  |  |  |  |
| --- | --- | --- | --- |
| Nhóm thực hiện: | Đặng Ngọc Lâm | 20010646 | K14 ĐTVT |
| Phạm Anh Toàn | 20010650 | K14 ĐTVT |
|  |  |  |

Giảng viên hướng dẫn: TS. Nguyễn Thị Thanh Quỳnh

Khoa: Điện – Điện tử

***Hà Nội, 26 tháng 11 năm 2024***

**LỜI CAM ĐOAN**

Chúng tôi xin cam đoan công trình thực hiện: “ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO” là một công trình nghiên cứu độc lập không có sự sao chép, thuê mượn, copy của người khác. Đề tài là một sản phẩm mà chúng tôi đã nỗ lực nghiên cứu trong quá trình học tập tại trường. Trong quá trình viết bài có sự tham khảo một số tài liệu có nguồn gốc rõ ràng, dưới sự hướng dẫn của TS. Nguyễn Thị Thanh Quỳnh - Giảng viên Khoa Điện – Điện tử, Trường Đại học Phenikaa. Chúng tôi xin cam đoan sẽ chịu hoàn toàn trách nhiệm nếu có vấn đề xảy ra.

**Nhóm sinh viên thực hiện**

*(Ký, ghi rõ họ tên)*

**TIẾN ĐỘ THỰC HIỆN**

**I) Thành viên nhóm**

|  |  |  |
| --- | --- | --- |
| **STT** | **Hình ảnh** | **Thông tin cá nhân** |
| 1 |  | - Họ và tên: Đặng Ngọc Lâm  - Mã SV: 20010646  - Lớp: K14 – ĐTVT  - Trường Đại học Phenikaa.  - SĐT: 0865589502  - Nơi ở: Hà Nội |
| 2 |  | - Họ và tên: Phạm Anh Toàn  - Mã SV: 20010650  - Lớp: K14 ĐTVT  - Trường Đại học Phenikaa.  - SĐT: 0862220302  - Nơi ở: Hà Nội |

**II) Phân công nhiệm vụ**

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Nhiệm vụ được giao** | **Nhóm đánh giá** |
| Đặng Ngọc Lâm | Lập trình Arduino, thiết kế mô hình, làm báo cáo | Đạt |
| Phạm Anh Toàn | Lập trình Arduino, thiết kế phần cứng, làm báo cáo | Đạt |

**III) Tiến độ thực hiện**

|  |  |  |
| --- | --- | --- |
| **Nội dung** | **Thời gian thực hiện** | **Kết quả** |
| Lên ý tưởng, tìm hiểu các loại linh kiện, vi điều khiển,… | 12/10/2024 - 31/10/2024 | Hoàn thành |
| Vẽ mạch nguyên lý và mạch PCB, chạy thử nghiệm trên Board trắng. | 01/11/2024 - 08/11/2024 | Hoàn thành |
| Làm mạch in thủ công. | 09/11/2024 - 19/11/2024 | Hoàn thành |
| Hoàn thiện và viết báo cáo. | 20/11/2024 - 24/11/2024 | Hoàn thành |

**Nhóm sinh viên thực hiện**

*(Ký, ghi rõ họ tên)*

**IV) Đánh giá tiến độ thực hiện**

Điểm chấm tiến độ: ….. Điểm (Dưới 4 điểm là không đạt, các tiêu chí còn lại mặc định chấm 0 điểm).

**GIẢNG VIÊN**

*(Ký, ghi rõ họ tên)*

**MỤC LỤC**

[LỜI NÓI ĐẦU 10](#_Toc162593434)

[CHƯƠNG I: TỔNG QUAN HỆ THỐNG 13](#_Toc162593435)

[1.1. TỔNG QUAN ARDUINO UNO 14](#_Toc162593436)

[1.1.1. Thông số kỹ thuật của Arduino 19](#_Toc162593438)

[1.1.2. Sơ đồ chân của Arduino 20](#_Toc162593439)

[1.2. VI ĐIỀU KHIỂN ESP32 17](#_Toc162593437)

[1.2.1. Cấu hình chân ESP32 dưới dạng GPIO 19](#_Toc162593438)

[1.2.2. Sơ đồ khối ESP32 20](#_Toc162593439)

[1.3. IC 74HC165N 20](#_Toc162593440)

[1.4. IC 74HC595 22](#_Toc162593441)

[1.5. MẠCH ỔN ÁP 23](#_Toc162593442)

[1.6. LED 24](#_Toc162593443)

[1.7. LCD 25](#_Toc162593444)

[1.8. IIC 27](#_Toc162593445)

[1.9. CLOUD SERVER 27](#_Toc162593445)

[1.10. KẾT LUẬN CHƯƠNG 27](#_Toc162593446)

[CHƯƠNG II: THIẾT KẾ HỆ THỐNG 28](#_Toc162593447)

[2.1. THIẾT KẾ HỆ THỐNG 28](#_Toc162593448)

[2.2. THIẾT KẾ MẠCH NGUYÊN LÝ 28](#_Toc162593449)

[2.3. CHẾ TẠO MẠCH XỬ LÝ 32](#_Toc162593456)

[2.4. KẾT LUẬN CHƯƠNG 38](#_Toc162593457)

[CHƯƠNG III: KẾT LUẬN ĐỒ ÁN 39](#_Toc162593458)

[3.1. SẢN PHẨM ĐẠT ĐƯỢC 40](#_Toc162593459)

[3.2. NHẬN XÉT 42](#_Toc162593460)

[3.2.1. Ưu điểm 42](#_Toc162593461)

[3.2.2. Nhược điểm 42](#_Toc162593462)

[3.2.3. Giải pháp khắc phục 43](#_Toc162593463)

[3.3. KẾT LUẬN 43](#_Toc162593464)

[3.4. HƯỚNG PHÁT TRIỂN 44](#_Toc162593465)

[TÀI LIỆU THAM KHẢO 45](#_Toc162593466)

[PHỤ LỤC 46](#_Toc162593467)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1: Hình dạng thực tế của Board Arduino Uno 15](#_Toc162593218)

[Hình 1.2: Sơ đồ nguồn điện 16](#_Toc162593219)

[Hình 1.3: Analog IN 17](file:///C:\Users\HP%20ZBOOK\Desktop\project_liennganh\final-report.docx#_Toc162593220)

[Hình 1.4: Chân kỹ thuật số 18](file:///C:\Users\HP%20ZBOOK\Desktop\project_liennganh\final-report.docx#_Toc162593221)

[Hình 1.5: Sơ đồ chân của ESP32 với 34 chân GPIO[2] 19](file:///C:\Users\HP%20ZBOOK\Desktop\project_liennganh\final-report.docx#_Toc162593222)

[Hình 1.6: Sơ đồ cho package QFN 19](#_Toc162593223)

[Hình 1.7: Cấu hình chân ESP32 dưới dạng GPIO 20](#_Toc162593224)

[Hình 1.8: Sơ đồ khối cấu trúc ESP32 21](#_Toc162593225)

[Hình 1.9: IC 74HC165 21](#_Toc162593226)

[Hình 1.10: Sơ đồ chân IC 74HC165N 22](#_Toc162593227)

[Hình 1.11: IC 74HC595N 23](#_Toc162593228)

[Hình 1.12: Sơ đồ chân 74HC595N 23](#_Toc162593229)

[Hình 1.13: IC ổn áp L7805 24](#_Toc162593226)

[Hình 1.14: Hình dạng thực tế đèn 25](#_Toc162593227)

[Hình 1.15: LCD 16x2 26](#_Toc162593228)

[Hình 1.16: IIC cho lcd 16x2 27](#_Toc162593227)

[Hình 2.1: Sơ đồ khối 30](file:///C:\Users\HP%20ZBOOK\Desktop\project_liennganh\final-report.docx#_Toc162593230)

[Hình 2.2: Sơ đồ nguyên lý trong phần mềm proteus 30](#_Toc162593231)

[Hình 2.3: Sơ đồ nguyên lý mạch ổn áp nguồn vào 31](file:///C:\Users\HP%20ZBOOK\Desktop\project_liennganh\final-report.docx#_Toc162593232)

[Hình 2.4: Sơ đồ nguyên lý mạch và hiển thị trong thang máy 31](#_Toc162593233)

[Hình 2.5: Sơ đồ nguyên lý mạch hiển thị ngoài thang máy 32](file:///C:\Users\HP%20ZBOOK\Desktop\project_liennganh\final-report.docx#_Toc162593232)

[Hình 2.6: Mạch PCB của nguồn đầu vào 33](#_Toc162593233)

[Hình 2.7: Mạch PCB của phần ngoài thang 34](file:///C:\Users\HP%20ZBOOK\Desktop\project_liennganh\final-report.docx#_Toc162593232)

[Hình 2.8: Mạch PCB của phần trong thang 34](#_Toc162593233)

[Hình 3.1: Mạch điều khiển khi chưa cấp nguồn 40](#_Toc162593234)

[Hình 3.2: Mạch điều khiển khi đã cấp nguồn 40](#_Toc162593235)

**DANH MỤC BẢNG BIỂU**

[Bảng 1.1: Danh mục linh kiện sử dụng 35](#_Toc134213065)

[Bảng 1.2: Thông số kỹ thuật của Arduino 35](#_Toc134213065)

[Bảng 1.3: Thông số kỹ thuật IC 74HC165N 21](#_Toc134213065)

[Bảng 1.4: Thông số kỹ thuật 74HC595N 22](#_Toc134213065)

[Bảng 1.5: Sơ đồ chân LCD 16x2 25](#_Toc134213065)

[Bảng 1.6: Sơ đồ chân của IIC 26](#_Toc134213065)

[Bảng 2.1: Các bước thực hiện làm mạch cứng 35](#_Toc134213065)

# LỜI NÓI ĐẦU

**LÝ DO CHỌN ĐỀ TÀI**

Lý do đầu tiên chọn đề tài "ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO" là để tìm hiểu và ứng dụng các kiến thức về vi điều khiển vào thực tế. Arduino là một nền tảng mã nguồn mở phổ biến, dễ tiếp cận với nhiều thư viện hỗ trợ mạnh mẽ, giúp người học có thể dễ dàng lập trình và điều khiển các thiết bị điện tử. Việc xây dựng hệ thống điều khiển thang máy không chỉ giúp hiểu rõ cách làm việc của vi điều khiển mà còn giúp người thực hiện nắm vững cách tích hợp các thành phần như cảm biến, động cơ, và mạch điều khiển vào một hệ thống hoàn chỉnh, phục vụ cho mục tiêu học tập và thực hành kỹ năng lập trình nhúng.

Ngoài ra, đề tài này có tính ứng dụng cao trong thực tế, góp phần nâng cao kiến thức về tự động hóa và điều khiển thông minh. Thang máy là một phần quan trọng trong hệ thống hạ tầng của các tòa nhà, vì vậy việc hiểu được cách thức hoạt động và khả năng tự thiết kế một hệ thống thang máy thu nhỏ sẽ mang lại trải nghiệm thực tiễn cho người học. Điều này không chỉ giúp phát triển tư duy logic và kỹ năng giải quyết vấn đề mà còn mở ra tiềm năng phát triển các dự án phức tạp hơn trong lĩnh vực tự động hóa và điều khiển công nghiệp.

Trong những năm gần đây, ngành công nghiệp thang máy tại Việt Nam đã có những bước phát triển mạnh mẽ, phản ánh nhu cầu ngày càng tăng về hệ thống vận hành tự động trong các công trình hiện đại. Sự phát triển này không chỉ đáp ứng nhu cầu di chuyển thuận tiện và an toàn, mà còn góp phần vào quá trình nâng cao chất lượng cuộc sống và tối ưu hóa không gian sử dụng trong các tòa nhà cao tầng. Một ví dụ điển hình về hệ thống thang máy hiện đại tại Việt Nam là dự án Landmark 81, một tòa nhà cao nhất Việt Nam nằm tại TP. Hồ Chí Minh. Thang máy trong dự án này được thiết kế với công nghệ tiên tiến, tích hợp các tính năng như điều khiển tự động thông minh và hệ thống an toàn tối ưu, mang lại trải nghiệm di chuyển nhanh chóng và an toàn cho cư dân và khách hàng, minh chứng cho sự phát triển và ứng dụng của công nghệ thang máy hiện đại trong các công trình xây dựng, giúp nâng cao chất lượng cuộc sống và hiệu quả vận hành cho các tòa nhà cao cấp.

Với sự chỉ bảo của giáo viên hướng dẫn của TS. Nguyễn Thị Thanh Quỳnh, chúng tôi đã hoàn thành đồ án " ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO " cho học phần Đồ án tích hợp liên ngành.

**ĐỐI TƯỢNG NGHIÊN CỨU**

Ở đồ án, nội dung nghiên cứu chủ yếu tập trung vào:

* Tìm hiểu về bo mạch Arduino Uno và ESP32.
* Nghiên cứu và chế tạo mạch điều khiển.
* Xây dựng bài toán, mô hình thang máy.

**MỤC TIÊU**

Mục tiêu khi thực hiện đồ án này:

* Hiểu được cách thức và chế độ hoạt động của Arduino.
* Giao tiếp không dây với Arduino thông qua Esp32
* Thiết kế, chế tạo được mạch mô phỏng điều khiển thang máy dùng trong hộ gia đình.
* Giám sát và điều khiển được vị trí của thang qua thiết bị như điện thoại, máy tính thông qua Blynk.

**NỘI DUNG NGHIÊN CỨU**

Công trình nghiên cứu “ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO” có các nội dung chính như sau:

* + Tìm hiểu các hoạt động của Arduino Uno cùng với ESP32.
  + Viết thuật toán cho thang máy gồm có 5 tầng.
  + Viết chương trình cho Arduino Uno
  + Viết chương trình cho ESP32
  + Thiết kế mô hình hệ thống.
  + Chỉnh sửa lỗi.
  + Đánh giá kết quả thực hiện.
  + Viết báo cáo.
  + Bảo vệ đồ án.

**TỔNG QUAN THIẾT KẾ HỆ THỐNG**

Hệ thống điều khiển thang máy trong đồ án được thiết kế sử dụng Arduino kết hợp với các IC 74HC595 và 74HC165 để quản lý đầu ra và đầu vào một cách hiệu quả.

Bên trong thang máy: Hệ thống bao gồm các nút bấm chọn tầng từ 1 đến 5, nút Open (mở cửa), Close (đóng cửa), và Warning (cảnh báo). Các tín hiệu từ nút bấm được xử lý thông qua IC 74HC165, giúp giảm số lượng chân cần thiết trên Arduino.

Bên ngoài thang máy: Tại mỗi tầng, có một nút gọi tầng tương ứng. Các tín hiệu này cũng được truyền qua IC 74HC165, cho phép thu thập tín hiệu hiệu quả từ nhiều tầng.

Hiển thị trạng thái: Các LED báo tầng, LED di chuyển, và LED cảnh báo được điều khiển thông qua IC 74HC595, giúp mở rộng số lượng chân điều khiển mà Arduino có thể quản lý.

Hệ thống hoạt động dựa trên các thuật toán điều khiển, đảm bảo thang máy xử lý các yêu cầu từ bên trong và bên ngoài theo thứ tự ưu tiên hợp lý. Các trạng thái như vị trí hiện tại của thang, trạng thái di chuyển, và cảnh báo được cập nhật và hiển thị chính xác qua LED.

Sự kết hợp giữa Arduino và các IC mở rộng như 74HC595 và 74HC165 không chỉ giúp tiết kiệm tài nguyên phần cứng mà còn đảm bảo hệ thống dễ dàng mở rộng hoặc nâng cấp, đồng thời mô phỏng đầy đủ các tính năng của một thang máy thực tế.

**PHƯƠNG PHÁP NGHIÊN CỨU**

Do đây là một đồ án sản phẩm, nên chúng tôi đã áp dụng phương pháp nghiên cứu lý thuyết, nghiên cứu thực nghiệm trực tiếp trên sản phẩm mô phỏng để tối ưu thuật toán, chạy thử và hoàn thiện sản phẩm.

**PHẠM VI ÁP DỤNG**

Đề tài “ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO” được áp dụng rộng rãi trong rất nhiều lĩnh vực và môi trường khác nhau. Phạm vi áp dụng bao gồm:

Nhà ở thông minh: Thiết bị chế tạo này có thể được dùng trong các gia đình để dễ dàng di chuyển giữ các tầng khí trong nhà, đồ đạc cồng kềnh, nặng nề cũng có thể dung thang máy để di chuyển thay vì phải vác đi lên từ thang bộ. Thang có thể điều khiển, giám sát từ xa giúp người dung có thể điều khiển thang kể cà khi không có ở nhà.

Tương tự như ở nhà, đối với các căn hộ, cơ sở y tế, cơ quan quy mô nhỏ cũng có thể áp dụng đề tài này.

**GIỚI HẠN**

Hiện nay, vi điều khiển, vi xử lí được ứng dụng rộng rãi trong nhiều lĩnh vực tại Việt Nam. Trên cơ sở lý thuyết được học trong các môn học như: Linh kiện điện tử, kỹ thuật cảm biến, hệ thống nhúng,… đã học trên trường và trong khuôn khổ của đồ án, chúng tôi đã thực hiện công trình nghiên cứu có đề tài là " ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO" dưới sự hướng dẫn của TS. Nguyễn Thị Thanh Quỳnh. Mục tiêu chính của chúng tôi là có thể thiết kế được một mạch giám sát và điều khiển mô hình thang máy sử dụng Arduino Uno và ESP32.

Do kiến thức còn hạn hẹp cùng với dự án chỉ dừng ở mức mô phỏng bằng các đèn led nên công trình nghiên cứu của chúng tôi còn nhiều thiếu sót. Mặc dù có thể thiết kế được mạch điện nhưng vẫn còn mang tính lý thuyết và tính mô hình nhiều, để áp dụng thực tế cần nhiều thời gian và kiến thức để phát triển hơn nữa. Chúng tôi mong sự ghi nhận đóng góp và sửa chữa của các thầy cô để đề tài này để có thể được hoàn thiện hơn. Chúng tôi xin chân thành cảm ơn TS. Nguyễn Thị Thanh Quỳnh đã hướng dẫn giúp chúng tôi hoàn thành được đồ án này.

# CHƯƠNG I: TỔNG QUAN HỆ THỐNG

Với sự phát triển của công nghệ vi điều khiển và hệ thống tự động hóa, việc thiết kế một hệ thống thang máy hiệu quả đã trở nên khả thi hơn bao giờ hết. Chương một sẽ nghiên cứu và phân tích hệ thống điều khiển thang máy sử dụng Arduino, giúp điều khiển các trang thái của thang như vị trí, nút bấm chọn tầng, cũng như đèn LED hiển thị trạng thái thông qua các mạch điện tích hợp. Những công nghệ này có thể giúp xây dựng một hệ thống thang máy nhỏ gọn, linh hoạt, có khả năng đáp ứng yêu cầu tự động hóa cơ bản, phù hợp cho việc học tập và thử nghiệm trong các dự án mô phỏng thực tế.

**Lý do chọn đề tài** "Điều khiển thang máy dùng Arduino" xuất phát từ mong muốn tìm hiểu và ứng dụng các kiến thức về vi điều khiển vào thực tế, cũng như nâng cao kỹ năng lập trình và tích hợp hệ thống trong lĩnh vực tự động hóa. Arduino là một nền tảng mã nguồn mở phổ biến với nhiều thư viện hỗ trợ mạnh mẽ, giúp người học dễ dàng tiếp cận và lập trình để điều khiển các thiết bị điện tử. Việc nghiên cứu và xây dựng hệ thống điều khiển thang máy không chỉ giúp người thực hiện nắm vững cách hoạt động của vi điều khiển mà còn tạo cơ hội thực hành tích hợp các thành phần như cảm biến, động cơ và mạch điều khiển vào một hệ thống hoàn chỉnh. Đây là nền tảng quan trọng giúp phát triển kỹ năng lập trình nhúng, tư duy logic và khả năng giải quyết vấn đề.

Hơn nữa, đề tài này có tính ứng dụng cao trong thực tiễn khi thang máy đóng vai trò quan trọng trong hệ thống hạ tầng hiện đại, phục vụ nhu cầu di chuyển an toàn, thuận tiện trong các tòa nhà cao tầng. Nghiên cứu và mô phỏng hệ thống thang máy sẽ giúp người học không chỉ hiểu sâu hơn về nguyên lý vận hành mà còn mở rộng kiến thức về tự động hóa và điều khiển thông minh, đồng thời mang lại những trải nghiệm thực tế trong việc phát triển hệ thống tích hợp.

Ngoài ra, ngành công nghiệp thang máy tại Việt Nam đang trên đà phát triển mạnh mẽ, phản ánh nhu cầu ngày càng tăng về các hệ thống vận hành tự động trong các công trình xây dựng hiện đại. Những ví dụ thực tế như hệ thống thang máy tiên tiến trong các tòa nhà cao tầng như Landmark 81 cho thấy công nghệ điều khiển thông minh không chỉ nâng cao hiệu quả vận hành mà còn cải thiện chất lượng cuộc sống. Việc nghiên cứu đề tài này không chỉ đóng vai trò quan trọng trong việc học tập và thực hành mà còn tạo tiền đề cho các ứng dụng và dự án tự động hóa công nghiệp lớn hơn trong tương lai.

**Lịch sử phát triển**: Thang máy đã có lịch sử phát triển lâu dài, phản ánh sự tiến bộ của công nghệ và sự thay đổi trong nhu cầu của con người qua các thời kỳ. Từ thời Hy Lạp cổ đại, khoảng năm 236 TCN, nhà toán học Archimedes đã thiết kế thiết bị nâng đầu tiên, sử dụng ròng rọc và sức người hoặc động vật để di chuyển các vật nặng. Đây được coi là tiền thân của các hệ thống thang máy hiện đại. Đến thời Trung Cổ, các lâu đài và tu viện ở châu Âu bắt đầu sử dụng các thiết bị nâng thủ công để vận chuyển thực phẩm, hàng hóa, nước sạch. Những hệ thống này thường được vận hành bằng tay quay hoặc sức kéo của động vật, đóng vai trò quan trọng trong đời sống hàng ngày nhưng vẫn còn hạn chế về mặt hiệu quả.

Bước sang thế kỷ 18, khi công nghệ cơ giới hóa bắt đầu phát triển, các thiết bị nâng được cải tiến với việc sử dụng năng lượng từ nước và hơi nước. Một ví dụ điển hình là "ghế bay" được lắp đặt tại cung điện Versailles năm 1743, chủ yếu phục vụ cho mục đích tiện nghi và giải trí trong giới quý tộc. Tuy nhiên, các thiết bị nâng trong giai đoạn này vẫn chưa đủ an toàn để vận chuyển con người một cách rộng rãi.

Thế kỷ 19 đánh dấu một bước ngoặt lớn trong lịch sử thang máy. Năm 1853, Elisha Otis phát minh hệ thống phanh an toàn, giúp thang máy tự động khóa lại khi dây cáp bị đứt. Đây là phát minh mang tính đột phá, giúp giảm thiểu rủi ro và tạo tiền đề cho việc ứng dụng thang máy trong các tòa nhà cao tầng. Vào năm 1857, thang máy chạy bằng hơi nước đầu tiên dành cho hành khách được lắp đặt tại một cửa hàng bách hóa ở New York, Mỹ. Tiếp đó, vào cuối thế kỷ 19, với sự ra đời của động cơ điện, thang máy chạy điện trở thành hiện thực. Năm 1880, Werner von Siemens chế tạo thành công thang máy chạy điện đầu tiên, mở ra kỷ nguyên mới cho ngành công nghiệp thang máy.

Trong thế kỷ 20 và 21, thang máy tiếp tục được cải tiến mạnh mẽ về cả thiết kế và công nghệ. Các hệ thống điều khiển tự động, thang máy tốc độ cao, và các tính năng an toàn tiên tiến được tích hợp, giúp thang máy trở thành một phần không thể thiếu trong các tòa nhà cao tầng và các công trình hiện đại trên toàn thế giới. Ngày nay, thang máy không chỉ phục vụ mục đích di chuyển mà còn được thiết kế để tối ưu hóa không gian, tiết kiệm năng lượng và nâng cao trải nghiệm người dùng. Những tiến bộ này minh chứng cho sự phát triển không ngừng nghỉ của công nghệ thang máy từ những ngày đầu sơ khai cho đến hiện tại.

**Ứng dụng**: Thang máy ngày nay đóng vai trò quan trọng trong nhiều lĩnh vực, đáp ứng nhu cầu di chuyển theo chiều dọc một cách nhanh chóng và hiệu quả. Trong các tòa nhà dân dụng và thương mại, thang máy trở thành yếu tố không thể thiếu, phục vụ cư dân, nhân viên và khách hàng trong các chung cư cao tầng, văn phòng, trung tâm thương mại. Thang máy trong những không gian này thường là thang tải khách hoặc thang máy gia đình, được thiết kế để tối ưu hóa trải nghiệm người dùng với các tính năng như điều khiển tự động, tốc độ cao, và tiết kiệm năng lượng. Trong lĩnh vực công nghiệp, thang máy tải hàng được sử dụng rộng rãi trong các nhà máy, kho bãi, và cơ sở sản xuất để vận chuyển hàng hóa, máy móc hoặc nguyên vật liệu nặng giữa các tầng. Chúng thường có tải trọng lớn và được tích hợp vào các hệ thống tự động hóa nhằm tăng hiệu suất làm việc.

Ngoài ra, trong lĩnh vực y tế, thang máy chuyên dụng tại các bệnh viện và cơ sở y tế được thiết kế để vận chuyển giường bệnh, thiết bị y tế và nhân viên một cách an toàn, nhanh chóng. Loại thang máy này thường có kích thước rộng, vận hành êm ái, và trang bị các tính năng khẩn cấp để đảm bảo an toàn cho bệnh nhân trong các tình huống đặc biệt. Trong hạ tầng giao thông công cộng, thang máy được lắp đặt tại các nhà ga, bến xe, sân bay và tàu điện ngầm để hỗ trợ hành khách, đặc biệt là người già, người khuyết tật và trẻ em. Chúng giúp cải thiện khả năng tiếp cận, đảm bảo sự tiện lợi và an toàn cho mọi người trong các không gian công cộng đông đúc.

Hơn thế nữa, thang máy còn được ứng dụng trong các khu vực đặc thù như tòa nhà thông minh, nơi chúng không chỉ đảm nhận nhiệm vụ di chuyển mà còn tích hợp các công nghệ tiên tiến như trí tuệ nhân tạo, quản lý thông minh theo dòng người, và giao diện điều khiển bằng cảm ứng hoặc giọng nói. Trong các công trình kiến trúc hiện đại, như tháp cao tầng hoặc công trình biểu tượng, thang máy tốc độ cao và thang máy kính được sử dụng để tăng tính thẩm mỹ và tối ưu hóa hiệu suất vận hành. Những ứng dụng này minh chứng cho vai trò không thể thiếu của thang máy trong việc nâng cao chất lượng cuộc sống, hiệu quả công việc và khả năng tiếp cận trong xã hội hiện đại.

**Công nghệ thang máy hiện nay**: Thang máy ngày nay sử dụng nhiều công nghệ hiện đại để đáp ứng yêu cầu ngày càng cao về hiệu suất, an toàn và trải nghiệm người dùng. Một trong những công nghệ phổ biến là hệ thống điều khiển thông minh (AI), giúp tối ưu hóa việc điều phối thang máy, giảm thiểu thời gian chờ và nâng cao hiệu quả vận hành, đặc biệt trong các tòa nhà đông người. Công nghệ không phòng máy (Machine Room-Less - MRL) cũng được áp dụng rộng rãi, cho phép lắp đặt động cơ không hộp số gọn nhẹ, tiết kiệm không gian và giảm tiêu hao năng lượng. Các hệ thống tiết kiệm năng lượng, chẳng hạn như phanh tái tạo, chuyển đổi năng lượng dư thừa thành điện năng tái sử dụng, góp phần giảm chi phí vận hành và thân thiện với môi trường. Bên cạnh đó, Internet vạn vật (IoT) được tích hợp để giám sát và bảo trì thang máy từ xa, giúp phát hiện sớm và khắc phục kịp thời các sự cố kỹ thuật. Ngoài ra, công nghệ nhận diện khuôn mặt và điều khiển không chạm cũng được triển khai trong các tòa nhà cao cấp, cho phép người dùng gọi thang mà không cần chạm vào nút bấm, tăng cường sự tiện lợi và vệ sinh. Đối với các tòa nhà siêu cao tầng, thang máy tốc độ cao sử dụng động cơ và cáp vật liệu tiên tiến, đạt tốc độ trên 10m/s, đáp ứng nhu cầu di chuyển nhanh chóng.

Tuy nhiên, Arduino – một nền tảng vi điều khiển mã nguồn mở – ít khi được sử dụng trong các thang máy thương mại hiện đại. Arduino thường được ứng dụng trong các mô hình thang máy nhỏ, đơn giản, phục vụ mục đích nghiên cứu hoặc học tập. Nhờ tính dễ sử dụng, chi phí thấp và khả năng lập trình linh hoạt, Arduino phù hợp để thử nghiệm các nguyên lý điều khiển cơ bản, tích hợp cảm biến và động cơ trong các mô hình thang máy. Tuy nhiên, trong hệ thống thang máy thực tế, các vi điều khiển chuyên dụng hoặc PLC (Programmable Logic Controller) được ưu tiên sử dụng vì chúng có khả năng xử lý phức tạp, đáng tin cậy hơn và đáp ứng các tiêu chuẩn an toàn khắt khe. Do đó, dù Arduino có ưu điểm về chi phí và tính linh hoạt, nhưng nó không thể đáp ứng đầy đủ các yêu cầu kỹ thuật, độ chính xác, và độ bền cần thiết trong các hệ thống thang máy hiện đại.

**Mục tiêu đồ án**: Mục tiêu của đồ án là thiết kế và chế tạo một mô hình thang máy được điều khiển bằng vi điều khiển Arduino, với đầu vào là các nút bấm mô phỏng yêu cầu di chuyển giữa các tầng. Trạng thái hoạt động của thang máy, bao gồm vị trí tầng hiện tại và quá trình di chuyển, sẽ được hiển thị bằng các đèn LED tương ứng. Hệ thống sẽ bao gồm các chức năng cơ bản như nhận tín hiệu từ các nút bấm để lựa chọn tầng, kiểm soát thứ tự ưu tiên khi có nhiều lệnh đồng thời, và hiển thị trạng thái qua hệ thống LED. Ngoài ra, đồ án cũng chú trọng vào việc lập trình thuật toán xử lý tín hiệu và điều khiển logic để đảm bảo hoạt động ổn định, mô phỏng chân thực cách một thang máy thực tế vận hành. Mô hình này không chỉ giúp hiểu rõ hơn về nguyên lý hoạt động của thang máy mà còn là nền tảng để phát triển các hệ thống điều khiển tự động hóa khác trong thực tế.

Mục tiêu cụ thể của đồ án là thiết kế và chế tạo một mô hình thang máy đơn hoạt động trong phạm vi 5 tầng, đảm bảo khả năng di chuyển chính xác và ổn định giữa các tầng. Thang máy được thiết kế để có thể chuyên chở cả người và hàng hóa, với cabin đủ chắc chắn và kích thước phù hợp, đáp ứng yêu cầu về tải trọng và an toàn. Hệ thống điều khiển trung tâm sử dụng vi điều khiển Arduino, nhận tín hiệu từ các nút bấm tại từng tầng và trong cabin, sau đó xử lý để điều khiển động cơ, đèn LED hiển thị và các thành phần khác. Một thuật toán điều khiển tối ưu được lập trình để xử lý các yêu cầu theo thứ tự hợp lý, giảm thời gian chờ và tối ưu hóa hành trình di chuyển, với các yếu tố như hướng di chuyển hiện tại và thứ tự ưu tiên được tính đến. Hệ thống sẽ tích hợp các đèn LED để hiển thị thông tin trạng thái như tầng hiện tại, hướng di chuyển và các tín hiệu trực quan khác, giúp người sử dụng dễ dàng theo dõi hoạt động. Cuối cùng, một mô hình vật lý được xây dựng, bao gồm các thành phần cơ khí (khung thang, cabin, dây kéo) và linh kiện điện tử (Arduino, cảm biến, động cơ) để kiểm chứng toàn bộ hệ thống. Mô hình này sẽ thể hiện trực quan và chân thực cách thức hoạt động của thang máy trong thực tế.

**Tóm lại**: Đồ án tập trung vào việc xây dựng một mô hình thang máy 5 tầng được điều khiển bằng vi điều khiển Arduino, mô phỏng cách thức hoạt động thực tế của thang máy. Hệ thống thang máy nhận tín hiệu từ các nút bấm, được đặt tại mỗi tầng và trong cabin, để xử lý lệnh yêu cầu di chuyển giữa các tầng. Quá trình hoạt động của thang máy, bao gồm trạng thái hiện tại (tầng đang dừng), hướng di chuyển (lên hoặc xuống), và các lệnh đang xử lý, sẽ được hiển thị thông qua hệ thống đèn LED tương ứng. Vi điều khiển Arduino đảm nhận vai trò trung tâm, thực thi thuật toán điều khiển tối ưu nhằm sắp xếp thứ tự xử lý lệnh, giảm thời gian chờ và đảm bảo hiệu quả hoạt động. Toàn bộ hệ thống sẽ được tích hợp trên một mô hình vật lý với cơ cấu cơ khí (khung, cabin, dây kéo) và linh kiện điện tử (Arduino, động cơ, cảm biến), giúp minh họa chi tiết và trực quan cách thức vận hành của một thang máy trong thực tế.

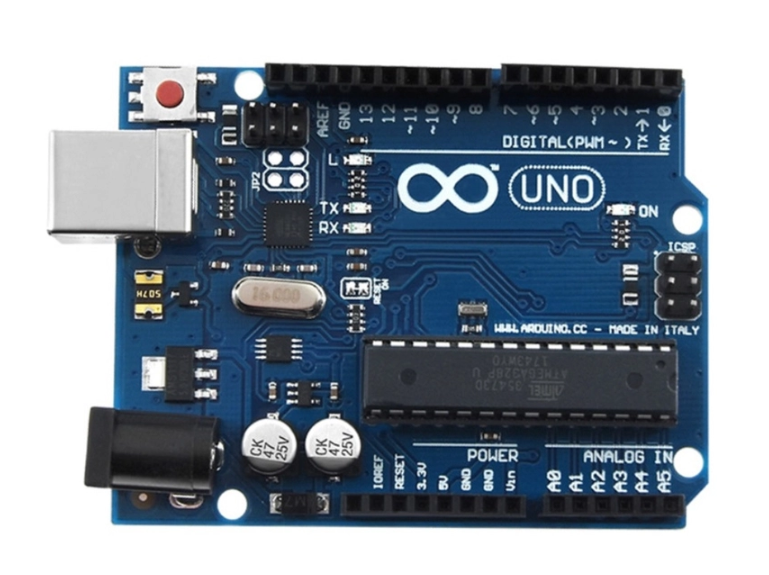
**Thực thi đồ án**:

Các linh kiện, thiết bị được sử dụng trong đề tài để thực hiện là:

Bảng 1.1: Danh mục linh kiện sử dụng

|  |  |  |
| --- | --- | --- |
| **Linh kiện / Thiết bị** | **Số lượng** | **Giá thành (1 cái)** |
| Vi điều khiển ESP32 | 1 | 105.000 vnd |
| Arduino uno | 1 | 500.000 vnd |
| Button | 13 | 500 vnd |
| L7805 | 1 | 3000 vnd |
| LED đỏ | 1 | 300 vnd |
| LED vàng | 2 | 300 vnd |
| LED xanh | 42 | 300 vnd |
| Điện trở 220 Ohm | 37 | 100 vnd |
| Tụ điện | 3 | 1.500 vnd |
| Switch | 1 | 3000 vnd |
| LCD 16x2 | 1 | 30.000 vnd |
| Module IIC | 1 | 15.000 vnd |
| Jump đực, cái |  | 3.000 vnd |
| Điện trở 1k Ohm | 2 | 100 vnd |

## 1.1. TỔNG QUAN ARDUINO UNO

Arduino UNO (Hình 1.1) là một bo mạch vi điều khiển dựa trên **ATmega328P** . Nó có 14 chân đầu vào/đầu ra kỹ thuật số (trong đó 6 chân có thể được sử dụng làm đầu ra PWM), 6 đầu vào tương tự, một bộ cộng hưởng gốm 16 MHz, kết nối USB, giắc cắm nguồn, đầu cắm ICSP và nút đặt lại. Nó chứa mọi thứ cần thiết để hỗ trợ bộ vi điều khiển.  


Hình 1.1: Hình dạng thực tế của Board Arduino Uno

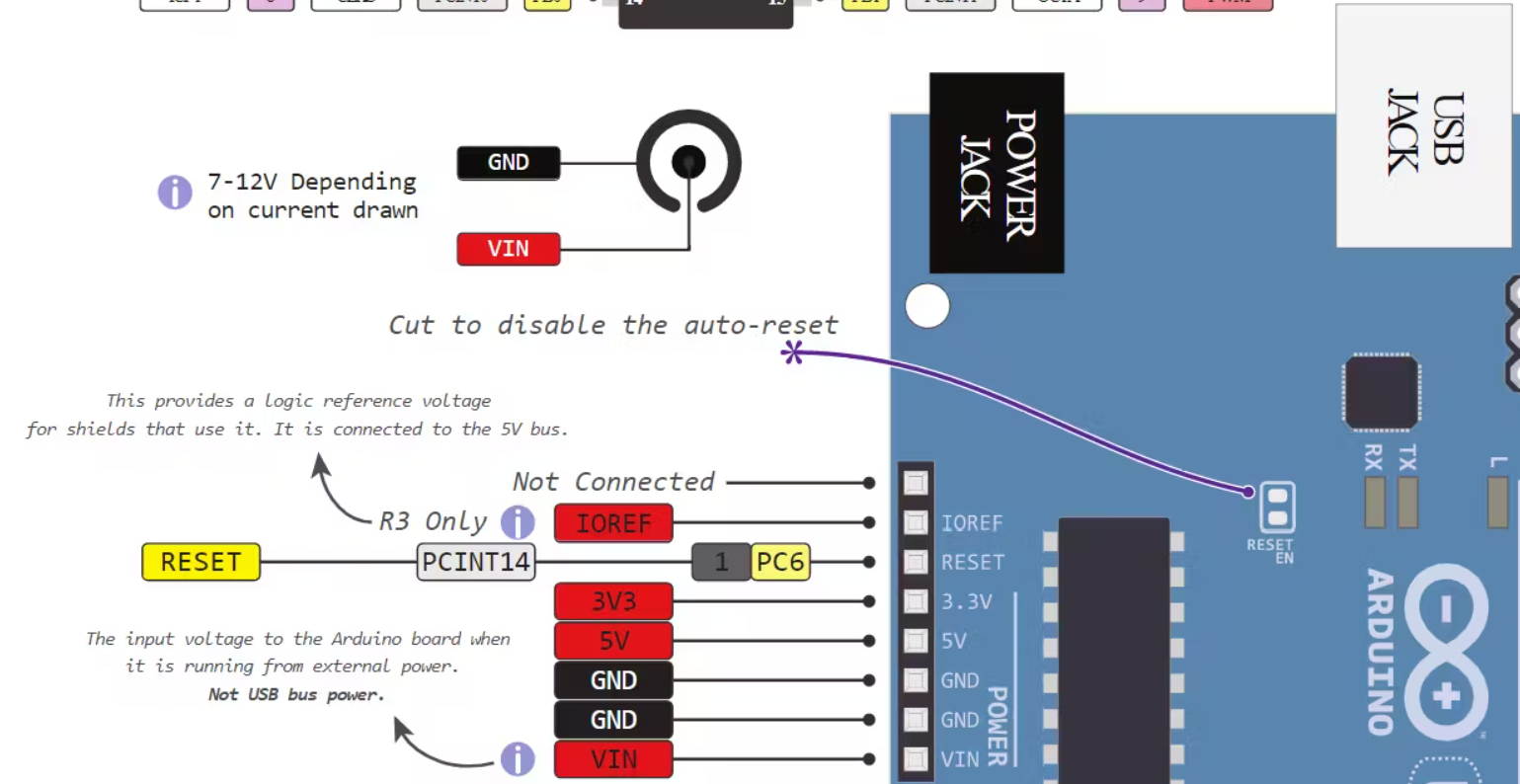
* + 1. **Thông số kĩ thuật của Arduino**

Bảng 1.2: Thông số kỹ thuật của Arduino

|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 |
| Điện áp hoạt động | 5V(cấp qua cổng usb) |
| Điện áp khuyến nghị | 6-9V |
| Số chân digital I/O | 14 chân( 6 chân PWM) |
| Số chân analog | 6 chân |
| Dòng ra tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa(3.3V) | 50 mA |
| Bộ nhớ Flash | 32 KB (ATmega328) với 0.5 KB dùng bởi bootloader |
| SRAM | 2 KB(ATmega328) |
| EEPROM | 1 KB(ATmega328) |
| Giao động của thạch anh | 16 MHz |

* + 1. **Sơ đồ chân của Arduino**

**Nguồn điện**

****

*Hình 1.2: Sơ đồ nguồn điện*

Có 3 cách để cấp nguồn cho Arduino Uno:

Barrel Jack - Barrel jack, hay DC Power Jack có thể được sử dụng để cấp nguồn cho bo mạch Arduino của bạn. Barrel jack thường được kết nối với bộ đổi nguồn gắn tường. Bo mạch có thể được cấp nguồn từ 5-20 vôn nhưng nhà sản xuất khuyến cáo nên giữ ở mức từ 7-12 vôn. Trên 12 vôn, bộ điều chỉnh có thể quá nhiệt và dưới 7 vôn, có thể không đủ.

Chân VIN - Chân này được sử dụng để cấp nguồn cho bo mạch Arduino Uno bằng nguồn điện bên ngoài. Điện áp phải nằm trong phạm vi được đề cập ở trên.

Cáp USB - khi kết nối với máy tính, cung cấp 5 vôn ở mức 500mA.

5v và 3v3

Chúng cung cấp điện áp 5v và 3,3v được điều chỉnh để cấp nguồn cho các linh kiện bên ngoài theo thông số kỹ thuật của nhà sản xuất.

GND

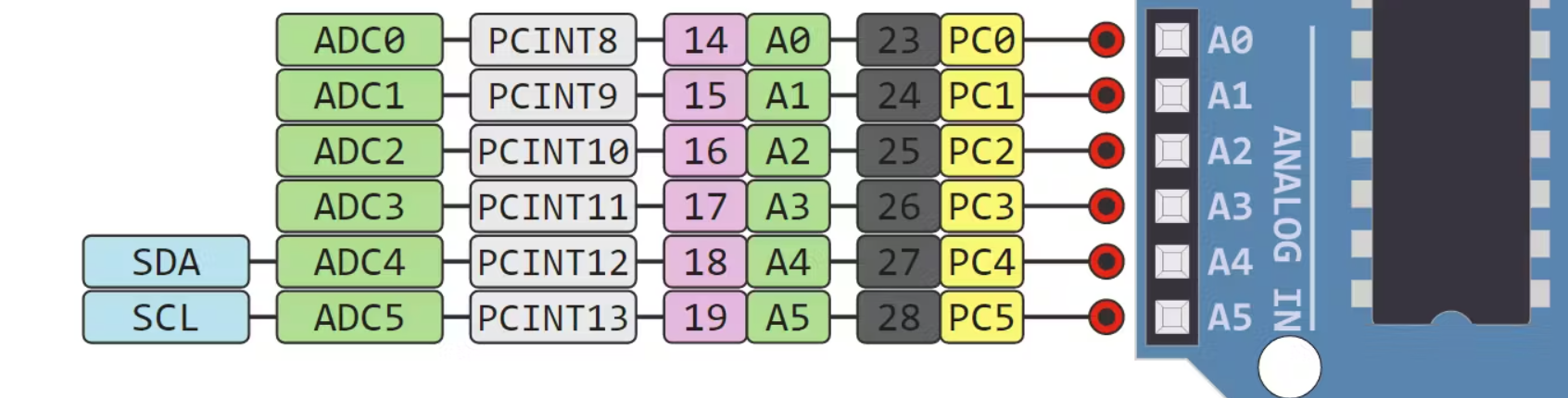
Trong sơ đồ chân Arduino Uno, có 5 chân GND, tất cả đều được kết nối với nhau.(Hình 1.2)

Các chân GND được sử dụng để đóng mạch điện và cung cấp mức tham chiếu logic chung trong toàn bộ mạch của bạn. Luôn đảm bảo rằng tất cả các GND (của Arduino, thiết bị ngoại vi và linh kiện) được kết nối với nhau và có chung một điểm nối đất.

RESET - thiết lập lại Arduino

IOREF - Chân này là chân tham chiếu đầu vào/đầu ra. Nó cung cấp điện áp tham chiếu mà bộ vi điều khiển hoạt động.

**Analog IN**



*Hình 1.3: Analog IN*

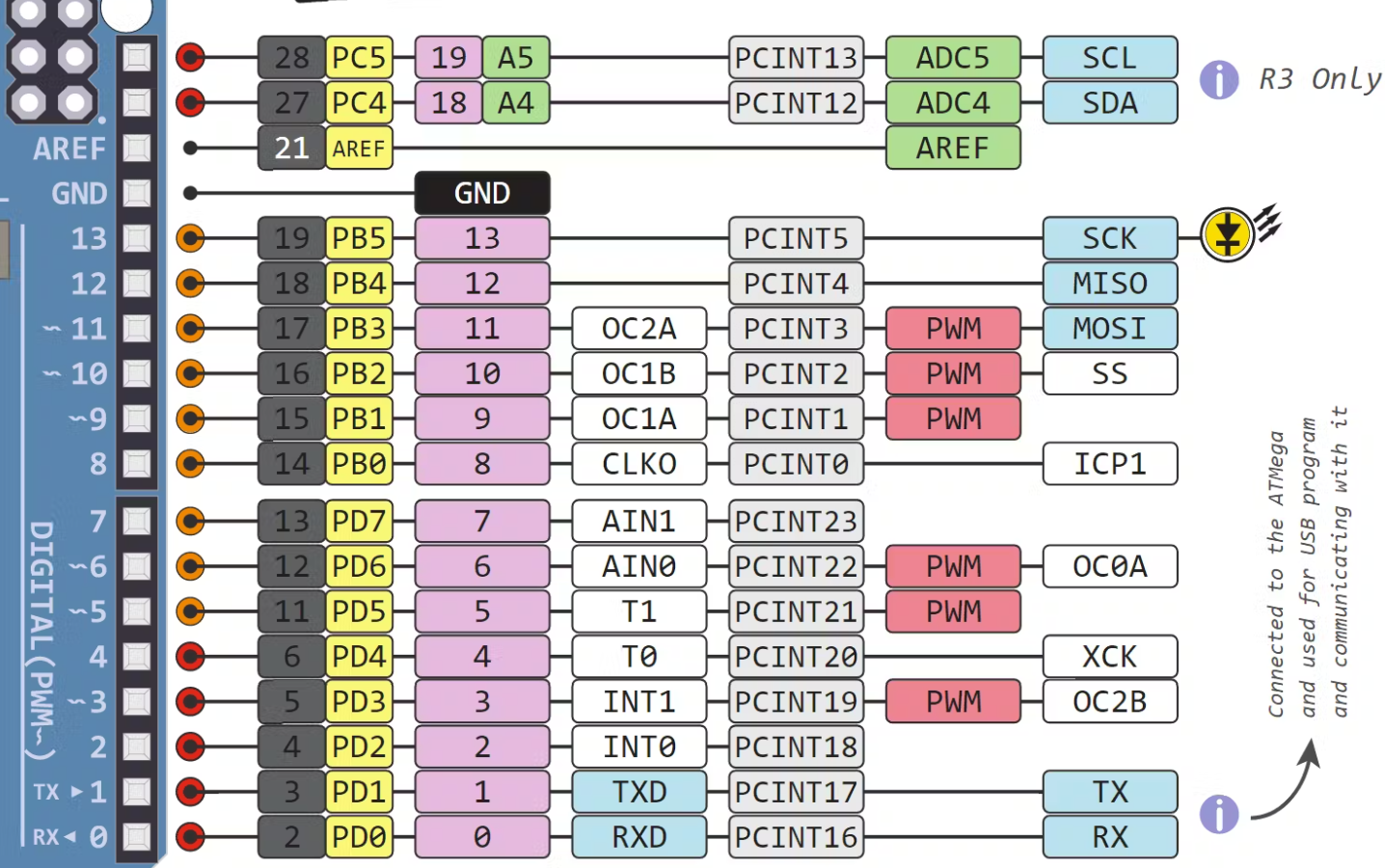
Arduino Uno có 6 chân analog (Hình 1.3) sử dụng bộ chuyển đổi ADC (Analog sang Digital).

Các chân này đóng vai trò là đầu vào tương tự nhưng cũng có thể hoạt động như đầu vào kỹ thuật số hoặc đầu ra kỹ thuật số.

ADC là viết tắt của Analog to Digital Converter. ADC là mạch điện tử dùng để chuyển đổi tín hiệu analog thành tín hiệu số. Biểu diễn số của tín hiệu analog này cho phép bộ xử lý (là thiết bị số) đo tín hiệu analog và sử dụng tín hiệu này thông qua hoạt động của nó.

Các chân Arduino A0-A5 có khả năng đọc điện áp tương tự. Trên Arduino, ADC có độ phân giải 10 bit, nghĩa là nó có thể biểu diễn điện áp tương tự theo 1.024 mức kỹ thuật số. ADC chuyển đổi điện áp thành các bit mà bộ vi xử lý có thể hiểu được.

**Arduino Uno Pinout - Chân kỹ thuật số**

****

*Hình 1.4: Chân kỹ thuật số*

Các chân 0-13 của Arduino Uno(Hình 1.4) đóng vai trò là chân đầu vào/đầu ra kỹ thuật số.

Chân 13 của Arduino Uno được kết nối với đèn LED tích hợp.

Trong Arduino Uno - các chân 3,5,6,9,10,11 có khả năng PWM.

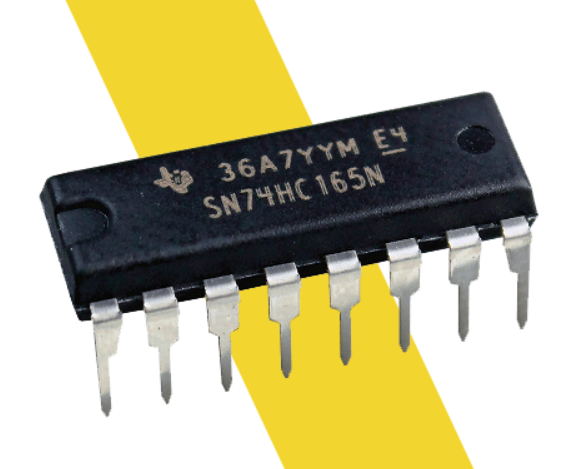
Điều quan trọng cần lưu ý là:

● Mỗi chân có thể cung cấp/hút tối đa 40 mA. Nhưng dòng điện khuyến nghị là 20 mA.

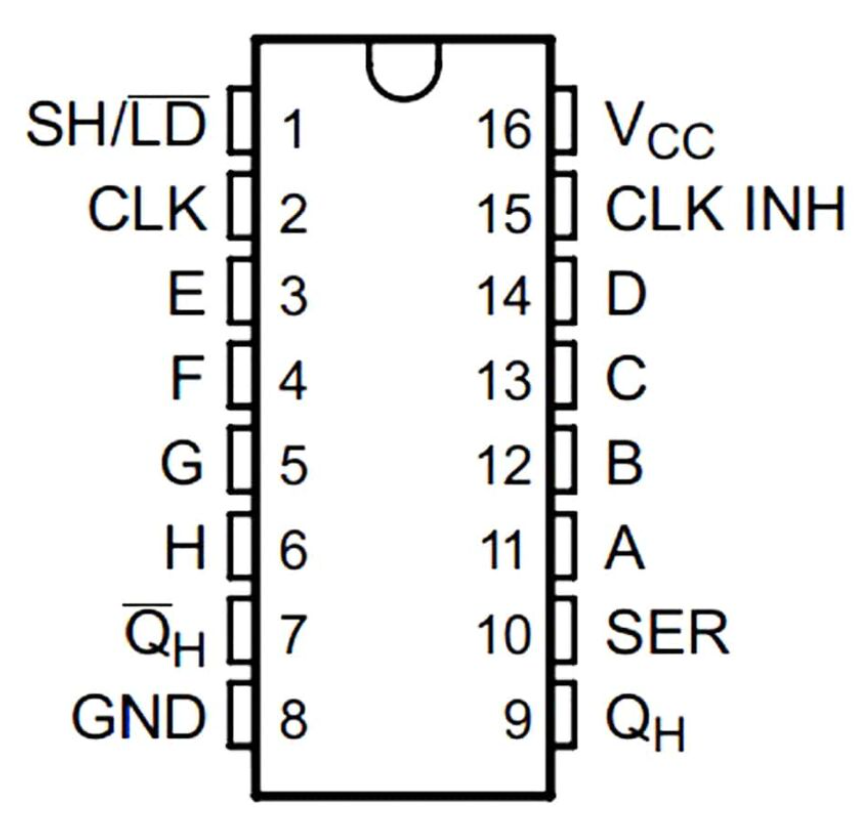
● Dòng điện cực đại tuyệt đối được cung cấp (hoặc chìm) từ tất cả các chân kết hợp là 200mA.

Arduino Uno được lựa chọn cho hệ thống thang máy nhờ vào tính linh hoạt và dễ sử dụng của nó. Đây là một vi điều khiển phổ biến với môi trường lập trình thân thiện, giúp các kỹ sư và nhà phát triển dễ dàng lập trình và chỉnh sửa. Ngoài ra, Arduino Uno cung cấp đủ số lượng chân I/O để kết nối các thành phần như nút bấm, đèn LED, và cảm biến. Với khả năng xử lý tín hiệu ổn định và tốc độ đáp ứng nhanh, Arduino đảm bảo hệ thống thang máy hoạt động hiệu quả, chính xác và an toàn. Hơn nữa, nó hỗ trợ nhiều thư viện mã nguồn mở, giúp việc lập trình các chức năng điều khiển trở nên đơn giản và tiết kiệm thời gian. Arduino Uno cũng có chi phí thấp, phù hợp với các hệ thống tự động hóa nhỏ gọn như thang máy.

## 1.2. IC 74HC165N



*Hình 1.5:* *IC 74HC165*



*Hình 1.6:* *Sơ đồ chân IC 74HC165N*

IC 74HC165N là một bộ dịch chuyển song song sang nối tiếp (parallel-in, serial-out) thường được sử dụng để mở rộng số lượng đầu vào kỹ thuật số cho Arduino. Khi kết nối với Arduino Uno, IC này cho phép bạn đọc nhiều tín hiệu đầu vào mà chỉ cần sử dụng một vài chân GPIO của vi điều khiển. Các chân quan trọng của 74HC165 bao gồm D0-D7 (đầu vào dữ liệu song song), chân SH/LD (Shift/Load) để chuyển đổi giữa chế độ tải dữ liệu song song và dịch chuyển nối tiếp, chân CLK (xung clock) để điều khiển việc dịch chuyển dữ liệu, chân Q7 (ngõ ra dữ liệu nối tiếp) để xuất dữ liệu đến Arduino, và chân Q7' nếu cần liên kết với các IC 74HC165 khác. Ngoài ra, bạn cần kết nối chân VCC và GND để cấp nguồn cho IC. Trong quá trình kết nối, có thể sử dụng điện trở 10kΩ làm pull-up hoặc pull-down nếu cần thiết để đảm bảo tín hiệu ổn định. Sau khi kết nối phần cứng, Arduino sẽ giao tiếp với 74HC165 thông qua các chân SH/LD, CLK và Q7, cho phép đọc trạng thái của tối đa 8 công tắc hoặc cảm biến qua một chuỗi dữ liệu nối tiếp.

Bảng 1.3 : Thông số kỹ thuật IC 74HC165N

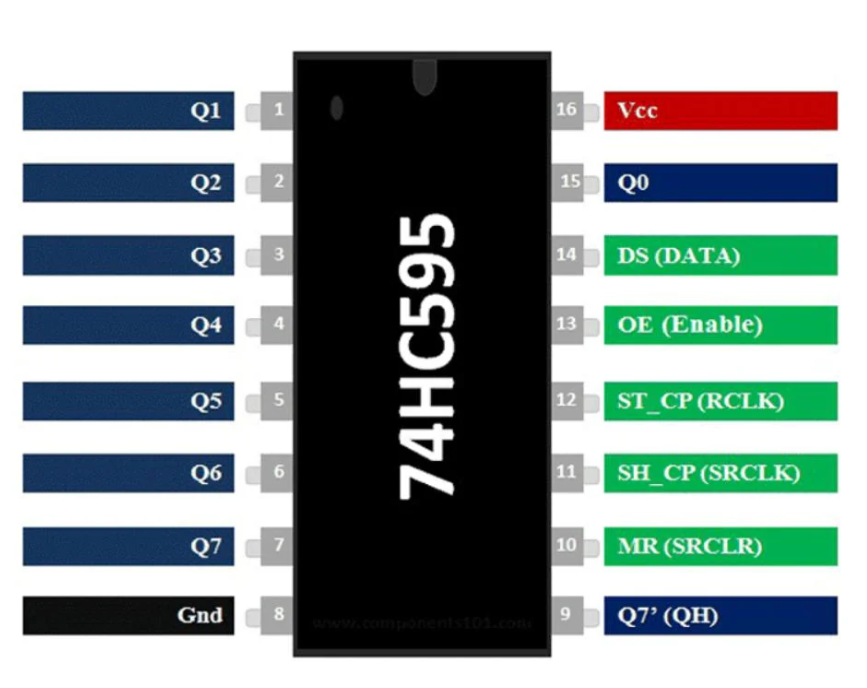
|  |  |  |  |
| --- | --- | --- | --- |
| | **Thông số** | | --- | | | **Giá trị** | | --- | |
| |  | | --- | | Loại IC | | |  | | --- | | 8-bit Parallel-In Serial-Out | |
| |  | | --- | | Điện áp nguồn (Vcc) | | |  | | --- | | 2V đến 6V | |
| |  | | --- | | Dòng điện ngõ ra (I\_OL, I\_OH) | | |  | | --- | | ±4mA tại Vcc = 5V | |
| |  | | --- | | Dòng tĩnh (Icc) | | |  | | --- | | Tối đa 80µA | |
| |  | | --- | | Thời gian trễ truyền tải | | |  | | --- | | ~13 ns tại Vcc = 5V | |
| |  | | --- | | Tần số hoạt động tối đa | | |  | | --- | | 29 MHz tại Vcc = 5V | |
| |  | | --- | | Nhiệt độ hoạt động | | |  | | --- | | -40°C đến 85°C | |
| |  | | --- | | Đóng gói | | |  | | --- | | DIP-16, SOIC-16, TSSOP-16 | |

## 

## 1.3. IC 74HC595N



*Hình 1.7 :* *IC 74HC595N*



*Hình 1.8 : Sơ đồ chân 74HC595N*

IC 74HC595N được sử dụng với Arduino Uno để mở rộng số lượng đầu ra kỹ thuật số mà vi điều khiển có thể điều khiển. IC này là một bộ dịch chuyển nối tiếp sang song song (serial-in, parallel-out), cho phép bạn điều khiển nhiều thiết bị như LED, relay, hoặc hiển thị 7 đoạn chỉ với 3 chân GPIO trên Arduino (SER, RCLK, SRCLK). Thay vì kết nối trực tiếp từng thiết bị với các chân GPIO riêng, IC 74HC595N nhận dữ liệu nối tiếp từ Arduino, sau đó xuất dữ liệu này ra các chân song song, giúp tiết kiệm tài nguyên phần cứng và làm cho thiết kế mạch gọn gàng hơn. Điều này đặc biệt hữu ích trong các ứng dụng cần điều khiển nhiều thiết bị cùng lúc mà Arduino không đủ số chân điều khiển trực tiếp.

Bảng 1.4: Thông số kỹ thuật 74HC595N

|  |  |  |  |
| --- | --- | --- | --- |
| | **Thông số** | | --- | | | **Giá trị** | | --- | |
| |  | | --- | | Loại IC | | |  | | --- | | 8-bit Serial-In Parallel-Out | |
| |  | | --- | | Điện áp nguồn (Vcc) | | |  | | --- | | 2V đến 6V | |
| |  | | --- | | Dòng điện ngõ ra (I\_OL, I\_OH) | | |  | | --- | | ±6mA tại Vcc = 5V | |
| |  | | --- | | Dòng tĩnh (Icc) | | |  | | --- | | Tối đa 80µA | |
| |  | | --- | | Thời gian trễ truyền tải | | |  | | --- | | ~13 ns tại Vcc = 5V | |
| |  | | --- | | Tần số hoạt động tối đa | | |  | | --- | | 25 MHz tại Vcc = 5V | |
| |  | | --- | | Nhiệt độ hoạt động | | |  | | --- | | -40°C đến 85°C | |
| |  | | --- | | Đóng gói | | |  | | --- | | DIP-16, SOIC-16, TSSOP-16 | |

## 

## 1.4. MẠCH ỔN ÁP

L7805



*Hình 1.9: IC ổn áp L7805*

Điện áp đầu ra: 5V (ổn định).

Điện áp đầu vào tối thiểu: 7V (để đảm bảo ổn định).

Điện áp đầu vào tối đa: 25V (tối đa, không khuyến nghị sử dụng lâu dài).

Dòng đầu ra tối đa: 1A

Mạch ổn áp L7805 được sử dụng để cung cấp nguồn điện ổn định 5V cho các thiết bị điện tử. Đây là một linh kiện quan trọng trong các mạch nguồn, đặc biệt khi đầu vào là điện áp không ổn định, như từ pin hoặc nguồn AC/DC chuyển đổi. L7805 có khả năng ổn định điện áp đầu ra ở mức 5V, bất kể điện áp đầu vào dao động (miễn là nằm trong khoảng 7-35V). Điều này giúp bảo vệ các linh kiện nhạy cảm như vi điều khiển, cảm biến, hoặc module giao tiếp khỏi bị hư hỏng do điện áp cao hoặc không ổn định. Ngoài ra, L7805 còn được tích hợp các tính năng bảo vệ như chống quá nhiệt và quá dòng, đảm bảo an toàn cho hệ thống. Với giá thành thấp và dễ sử dụng, L7805 là một lựa chọn phổ biến trong thiết kế mạch nguồn.

## 1.5. LED 3mm



Hình 1.10: Hình dạng thực tế đèn

Điện áp hoạt động (Vf):

LED đỏ: 1.8V - 2.2V

LED xanh lá: 2.0V - 2.4V

LED vàng: 2.0V - 2.2V

Dòng điện hoạt động (If):

10mA - 20mA

Góc phát sáng:

20° - 30°

Độ sáng (mcd):

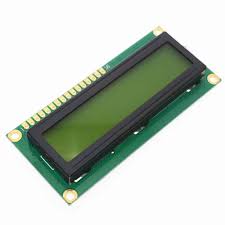
LED đỏ: 50 - 200 mcd

LED xanh lá: 100 - 300 mcd

LED vàng: 50 - 150 mcd

LED 3mm được dùng để hiển thị trạng thái cửa các nút bấm trong và ngoài thang cùng với trạng thái thang máy.

## 1.6. LCD 16x2



*Hình 1.11: LCD 16x2*

Điện áp hoạt động:

Logic: 4.7V - 5.3V (thường 5V).

Đèn nền (Backlight): 4.2V - 4.7V (hoặc phụ thuộc vào model).

Dòng tiêu thụ:

Logic: 1 - 2mA.

Đèn nền: 10 - 20mA (tùy loại).

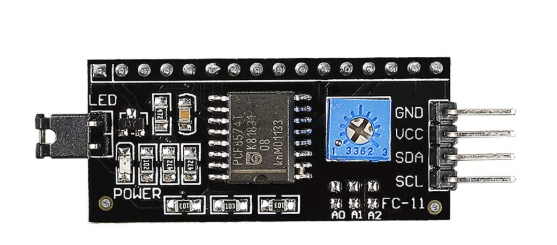
LCD 16x2 được sử dụng trong hệ thống thang máy vì khả năng hiển thị thông tin rõ ràng và dễ đọc. Màn hình này có thể hiển thị các thông tin quan trọng như tầng hiện tại của thang máy, trạng thái hoạt động, và các chỉ dẫn khác cho người sử dụng, giúp họ dễ dàng theo dõi và hiểu được tình trạng của thang máy.

Với khả năng hiển thị 2 dòng, mỗi dòng 16 ký tự, LCD 16x2 đủ để truyền tải thông tin cần thiết mà không chiếm nhiều không gian. Điều này đặc biệt hữu ích trong các thang máy có thiết kế nhỏ gọn, nơi mà không gian hiển thị là một yếu tố quan trọng. Ngoài ra, việc sử dụng LCD 16x2 giúp tiết kiệm chi phí và năng lượng, vì nó tiêu thụ điện năng thấp và dễ dàng tích hợp vào các hệ thống điều khiển của thang máy. Màn hình có đèn nền còn giúp người sử dụng dễ dàng đọc thông tin trong điều kiện ánh sáng yếu, đảm bảo sự tiện dụng và an toàn.

Bảng 1.5: Sơ đồ chân LCD 16x2

|  |  |  |
| --- | --- | --- |
| **Chân** | **Ký hiệu** | **Chức năng** |
| 1 | VSS | Nguồn GND (0V). |
| 2 | VDD | Nguồn VCC (+5V). |
| 3 | VO | Điều chỉnh độ tương phản (qua biến trở). |
| 4 | RS | Chọn thanh ghi (Instruction/Data). |
| 5 | RW | Chọn chế độ đọc (R) hoặc ghi (W). |
| 6 | E | Tín hiệu cho phép (Enable). |
| 7-14 | D0 - D7 | Dữ liệu (8-bit hoặc 4-bit nếu bỏ D0-D3). |
| 15 | LED+ | Chân dương của đèn nền (nếu có). |
| 16 | LED- | Chân âm của đèn nền (nếu có). |

**1.7. IIC**

*****Hình 1.12: IIC cho lcd 16x2*

Điện áp hoạt động:

VCC: 5V (hoặc 3.3V tùy vào loại vi điều khiển).

Giao tiếp:

SDA: Dây dữ liệu (Data).

SCL: Dây xung đồng hồ (Clock).

Sử dụng giao tiếp I2C cho LCD 16x2 mang lại nhiều lợi ích, đặc biệt trong các hệ thống yêu cầu tiết kiệm chân GPIO và đơn giản hóa kết nối. Thông thường, LCD 16x2 cần ít nhất 6 chân GPIO để điều khiển màn hình (4 chân dữ liệu và 2 chân điều khiển), nhưng khi sử dụng I2C, chỉ cần 4 chân: VCC, GND, SDA (dữ liệu) và SCL (xung clock), giúp tiết kiệm nhiều chân GPIO và dễ dàng mở rộng hệ thống. Ngoài ra, giao tiếp I2C còn đơn giản hóa việc kết nối và lập trình, vì các thiết bị ngoại vi có thể chia sẻ cùng một bus dữ liệu. Điều này không chỉ giảm độ phức tạp của mạch mà còn giúp việc điều khiển LCD trở nên dễ dàng hơn, vì chỉ cần một vài dòng lệnh trong phần mềm. I2C cũng cho phép kết nối nhiều thiết bị cùng lúc mà không làm tăng số lượng chân cần thiết trên vi điều khiển, mang lại sự linh hoạt và hiệu quả cho các dự án điện tử.

Bảng 1.6: Sơ đồ chân của IIC

|  |  |  |
| --- | --- | --- |
| **Chân** | **Ký hiệu** | **Chức năng** |
| 1 | VCC | Cung cấp nguồn (+5V hoặc +3.3V tùy vi điều khiển) |
| 2 | GND | Nối đất (Ground). |
| 3 | SDA | Dữ liệu I2C (Serial Data Line). |
| 4 | SCL | Xung I2C (Serial Clock Line). |

## 1.9. KẾT LUẬN CHƯƠNG

Chương I đã trình bày cơ bản về tổng quan hệ thống bao gồm: lý do chọn đề tài, lịch sử phát triển, ứng dụng...và nếu khái quát về hệ thống đề tài cũng như các linh kiện được sử dụng.

# CHƯƠNG II: THIẾT KẾ HỆ THỐNG

Trong chương trước đã thảo luận về sự cần thiết và tiềm năng của việc sử dụng thang máy vào cuộc sống hàng ngày. Chương này sẽ tập trung vào quá trình thiết kế hệ thống thang máy sử dụng Arduino cùng với Esp32, cụ thể là cách chọn lựa và kết hợp các linh kiện, cấu trúc kết nối và phương pháp lập trình để tạo ra một hệ thống hoạt động mạnh mẽ và linh hoạt.

## 2.1. THIẾT KẾ HỆ THỐNG

Hệ thống bao gồm 4 khối cơ bản:

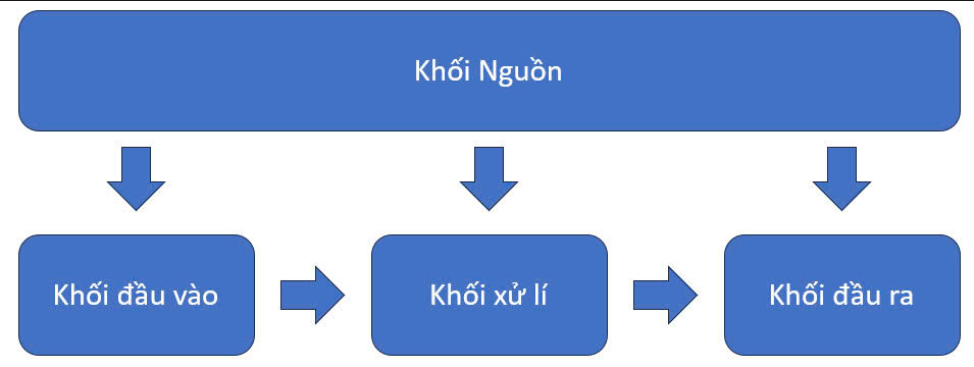
**Khối nguồn**: Trong hệ thống thang máy được thiết kế để cung cấp dòng điện ổn định, đảm bảo các linh kiện điện tử hoạt động chính xác và bền bỉ. Thành phần chính của khối nguồn là IC L7805, một bộ điều chỉnh điện áp tuyến tính, có nhiệm vụ chuyển đổi nguồn điện đầu vào (thường từ 7V đến 12V) xuống mức điện áp cố định 5V. Mức điện áp này phù hợp để cấp nguồn cho các linh kiện chính trong hệ thống, bao gồm vi điều khiển Arduino, đèn LED hiển thị, các nút bấm (button) và các mạch điện phụ trợ khác. Sự ổn định của nguồn điện 5V giúp ngăn ngừa tình trạng dao động điện áp, tránh ảnh hưởng đến độ nhạy của nút bấm, độ sáng của đèn LED, và đảm bảo Arduino hoạt động một cách ổn định. Bên cạnh đó, khối nguồn có thể được bổ sung thêm các tụ lọc để giảm nhiễu điện từ và đảm bảo hiệu suất của toàn hệ thống trong suốt quá trình vận hành.Khối đầu vào: Khối đầu vào bao gồm các nút bấm trong và ngoài thang máy, cho phép người sử dụng yêu cầu thang di chuyển đến các tầng khác nhau. Các nút bấm gửi tín hiệu tới Arduino để xử lý.

**Khối đầu vào**: Khối đầu vào trong hệ thống thang máy là thành phần đảm nhiệm việc nhận tín hiệu từ người sử dụng. Thành phần này bao gồm các nút bấm bên trong cabin thang máy, cho phép người dùng chọn tầng mà họ muốn đến, và các nút bấm bên ngoài thang máy, thường được lắp ở mỗi tầng để gọi thang đi lên hoặc đi xuống. Khi người dùng nhấn nút, tín hiệu từ các nút bấm này sẽ được gửi đến Arduino, nơi tín hiệu được xử lý để thực hiện các yêu cầu như gọi thang hoặc điều khiển thang dừng tại tầng mong muốn. Điều này giúp hệ thống thang máy hoạt động theo đúng yêu cầu của người dùng.

**Khối xử lý**: Khối xử lý trung tâm trong hệ thống thang máy sử dụng Arduino, một loại vi điều khiển được lập trình để thực hiện các nhiệm vụ cốt lõi. Khối này có vai trò tiếp nhận tín hiệu từ các nút bấm do người dùng thao tác, phân tích các tín hiệu đó và đưa ra quyết định để điều khiển hoạt động của thang máy. Cụ thể, Arduino đảm nhiệm việc xử lý dữ liệu, điều khiển chuyển động lên xuống của thang máy, và đáp ứng các yêu cầu như dừng tại tầng được chọn hoặc mở cửa khi cần thiết. Tất cả hoạt động này đảm bảo thang máy vận hành đúng cách và an toàn theo nhu cầu của người sử dụng.

**Khối đầu ra**: Khối đầu ra trong hệ thống thang máy là bộ phận chịu trách nhiệm cung cấp thông tin về trạng thái hoạt động của thang máy cho người sử dụng. Thành phần này bao gồm các đèn LED hiển thị, được lắp đặt ở cả bên trong và bên ngoài thang máy. Các LED này hiển thị thông tin như tầng hiện tại của thang máy, hướng di chuyển (lên hoặc xuống), hoặc thông báo rằng thang đã đến tầng được chọn. Ngoài ra, khối đầu ra cũng có thể báo hiệu các trạng thái khác như "thang máy đang bận" hoặc "đang trong quá trình di chuyển". Thông qua các thông tin trực quan này, người sử dụng dễ dàng nắm bắt trạng thái của thang máy và biết khi nào có thể sử dụng.

## 2.2. THIẾT KẾ MẠCH NGUYÊN LÝ



Hình 2.1: Sơ đồ khối

Hình 2.1 mô tả tổng quan một sơ đồ khối của hệ thống thang máy bao gồm 4 khối chính như sau: Khối nguồn cung cấp nguồn điện ổn định 5V nhờ IC L7805, đảm bảo hoạt động chính xác cho các linh kiện như Arduino, đèn LED, nút bấm và các mạch phụ trợ, đồng thời giảm nhiễu để duy trì hiệu suất hệ thống. Khối đầu vào bao gồm các nút bấm trong và ngoài thang máy, cho phép người dùng gửi tín hiệu yêu cầu như chọn tầng hoặc gọi thang, các tín hiệu này được gửi đến Arduino để xử lý. Khối xử lý là trung tâm điều khiển, sử dụng vi điều khiển Arduino để tiếp nhận, phân tích tín hiệu từ khối đầu vào, sau đó đưa ra lệnh điều khiển thang máy di chuyển, dừng đúng tầng, hoặc thực hiện các yêu cầu khác. Khối đầu ra gồm các đèn LED hiển thị trạng thái của thang máy như vị trí hiện tại, hướng di chuyển, hoặc thông báo thang đã đến tầng yêu cầu, giúp người dùng dễ dàng theo dõi và sử dụng hệ thống.



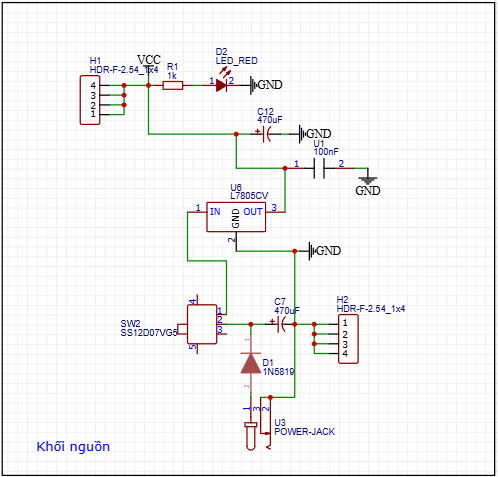
Hình 2.2: Sơ đồ nguyên lý trong phần mềm proteus

Hình 2.2 là sơ đồ mạch được mô phỏng trên phần mềm Proteus trong đó đươc chia thành các khu chính sau:

**Khu vực trong thang:** Gồm có các Button biểu thị các nút lệnh ấn tầng muốn đến ( tầng 1 – tầng 5) và 3 nút ấn để mở cửa, đóng cửa và gửi cảnh báo khi người trong thang gặp sự cố. Ngoài ra mạch ngoài thang còn có 1 IC74hc165N và 1 IC 74HC595N dùng để nối với các button và led và từ đó gửi thông tin để xử lí về cho Arduino. Khi nhận các giá trị của các button được gửi về qua IC74HC165N, Arduino sẽ xử lí và đưa ra các trạng thái của led tương ứng qua IC 74HC595.

**Khu vực ngoài thang:** Tương tự mạch mô phỏng của thang trong mạch mô phỏng bên ngoài thang cũng gồm có các button đóng vai trò là đầu vào khi có người ấn ở tầng nào đó. Ở bên ngoài mõi thang sẽ có 1 nút gọi tầng duy nhất. Mạch ngoài thang sử dụng 1 IC74HC165N và 2 IC 74HC595N. Các led ở đây hiển thị trạng thái vị trí của thang, Với mỗi tầng sẽ có 5 led biểu thị cho 5 tầng tương ứng, Khi thang đang di chuyển đến tầng nào thì led tầng đó sẽ sáng và ngược lại. Ngoài ra còn có 5 led dùng để biểu thị trạng thái của button.

**Khu vực hiển thị qua LCD 16x2**: Trong đề tài này LCD được sử dụng để biểu thị trạng thái tầng của thang, trang thái di chuyển của thang và trạng thái đóng cửa thang máy.

  
Hình 2.3: Sơ đồ nguyên lý mạch ổn áp nguồn vào

Trong mạch nguồn vào (Hình 2.3) có các linh kiện được sử dụng gồm:

Jack DC: Dùng để nhận chân cắm nguồn vào từ bên ngoài

IC 7805: Được dùng vì nó cung cấp điện áp ổn định là 5V, dễ sử dụng có tích hợp bảo vệ khi quá dòng, quá nhiệt.

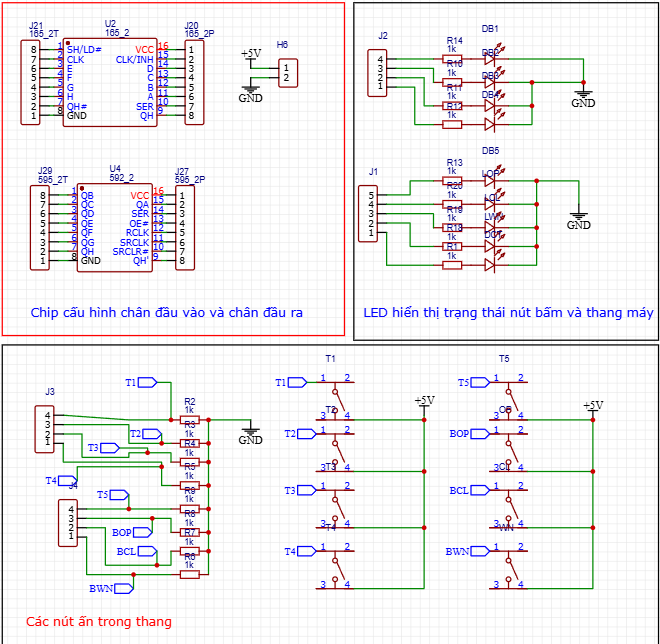
Diode: Chống ngược nguồn

Tụ điện dùng để giảm thiểu các giao động điện áp do sự thay đổi nhanh của dòng điện trong mạch. Nó giúp cố định mạch ổn áp.

Điện trở: Dùng để giảm điện áp vào cho led và cho IC ổn áp L7805

LED: sử dụng led đỏ có kích thước 5mm để hiển thị trạng thái nguồn vào của mạch ổn áp.

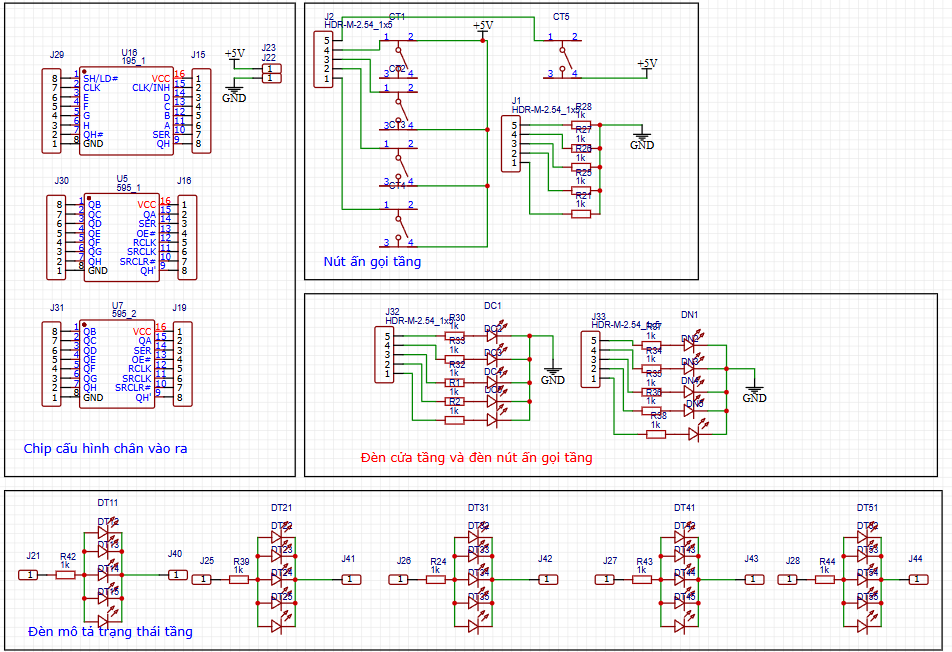
Switch: Sử dụng loại switch gạt 5 chân để làm công tác bật tắt cho mạch nguồn vào này.



Hình 2.4: Sơ đồ nguyên lý mạch và hiển thị trong thang máy

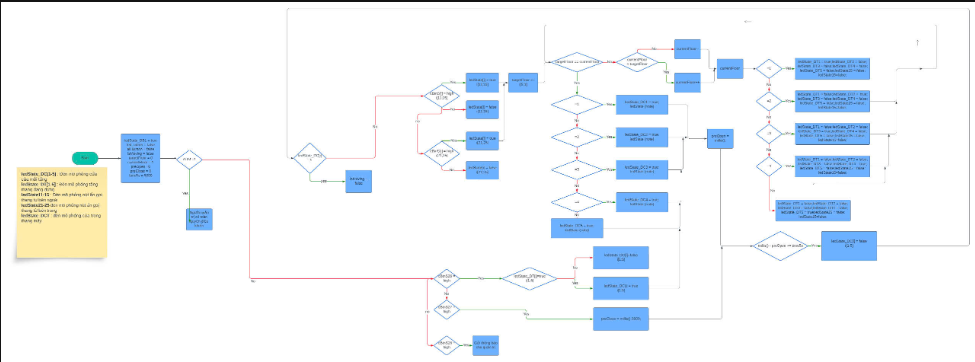
Trong mạch mô phỏng cho bên trong thang máy (Hình 2.4) sẽ có những linh kiện chính như button biểu thị các nút ấn tầng muốn đến(từ 1-5) và 3 nút lần lượt là Open, Close, Warning. Led ở đây gồn có các led biểu thị trạng thái của nút bấm tương ứng.

Tương tự như khi mô phỏng trên Proteus, mạch biểu thị bên trong thang máy được bố trí các led, button, IC74HC165N và IC 74HC95N. Do khi làm mạch thực tế, với một lớp đi dây thì khó có thể kiểm tra được nên ở đây được sử dụng thêm các jack đực ra chân.



Hình 2.5: Sơ đồ nguyên lý mạch hiển thị ngoài thang máy

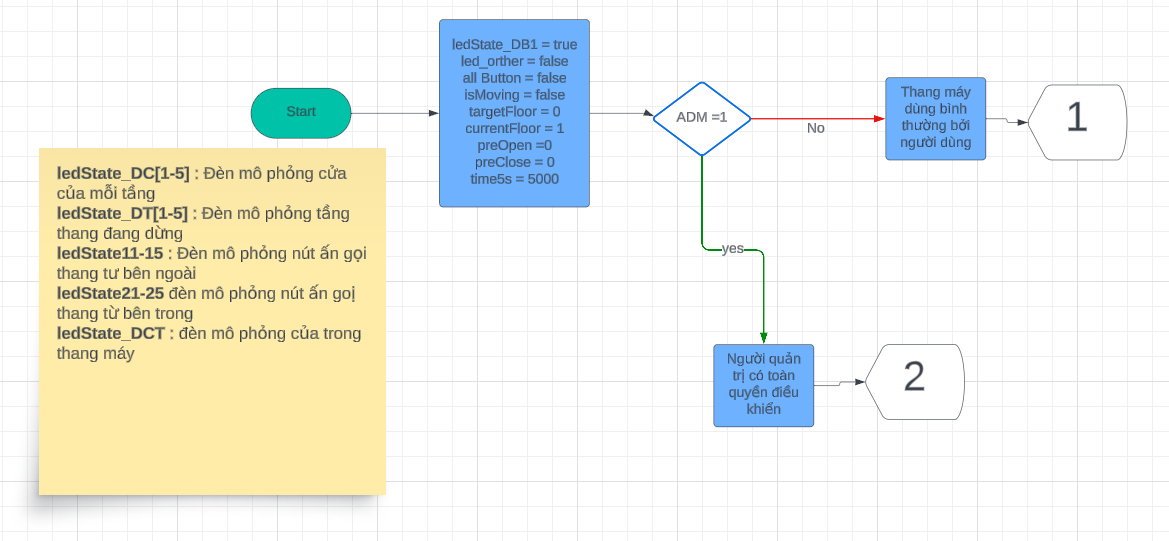
Mạch nguyên lí mô phỏng phía biên ngoài thang máy (Hình 2.5) cũng gồm các loại linh kiện như mạch mô phỏng phía trong thang nhưng với số lương nhiều hơn chút. Ở đây sử dụng 1 IC74HC165N và 2 IC 74HC595N do số led ở đây có số lượng nhiều.



Hình 2.6: Lưu đồ thuật toán của hệ thống

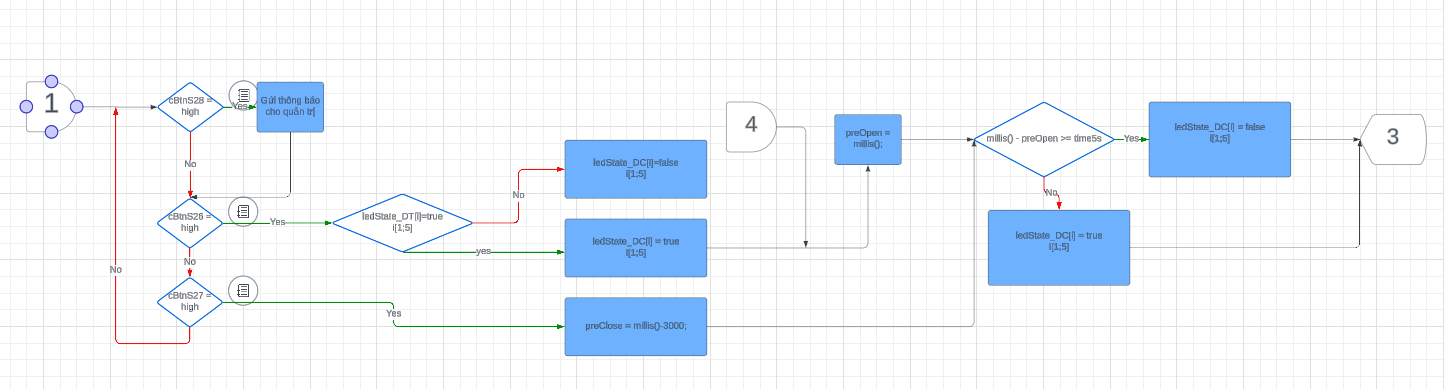
Thuật toán để sử lí các tín hiệu đầu vào từ tất cả nút ấn bên trong cũng như bên ngoài thang để cho ra các tín hiệu led và hiển thị phù hợp lên LCD được vẽ trên phần mềm Flowchart với sơ đồ đầy đủ như Hình 2.6

Trong lưu đồ thuật toán được chia làm 3 phần chính như sau: Phần kiểm tra tín hiệu điều khiển thang từ người quản lí thang máy như bảo vệ …. , Phần kiểm tra trạng thái của cửa thang máy và cửa của mỗi tầng và cuối cùng là phần nhận tín hiệu các nút gọi thang cũng như gọi tầng muốn đến từ người dùng.

 Hình 2.7: Lưu đồ thuật toán của hệ thống

Như Hình 2.7 ta có trạng thái ban đầu khi mới cấp nguồn cho hệ thống hay khi vừa reset hệ thống thì thang đang ở tầng 1 tương ứng với đèn mô phỏng ràng thang đang ở tầng 1 đang sáng (ledState\_DC1 = true) và các led còn lại đang tắt. Khi có tín hiệu chiếm quyền điểu khiên thang máy từ người quản lí (ADM ==1) thì thang máy sẽ không lên xuống theo người dùng khi họ ấn bất kì nút ấn nào bên trong hay ngoài thang. Ngược lại nếu không có tín hiệu muốn điều khiển thang từ người quản lí thì người dùng sẽ ấn được các nút trong và ngoài thang máy như bình thường.

Điều kiện này là điều kiện ban đầu để khi có trường hợp khẩn cấp như hỏng hóc , ưu tiên người nào đó,… mà người quản lí có thể chuyển quyền điều khiển chỉ cho bản thân mình phòng trường hợp người ngoài sử dụng thang đang có sự cố hoặc được dùng cho những mục đích riêng.

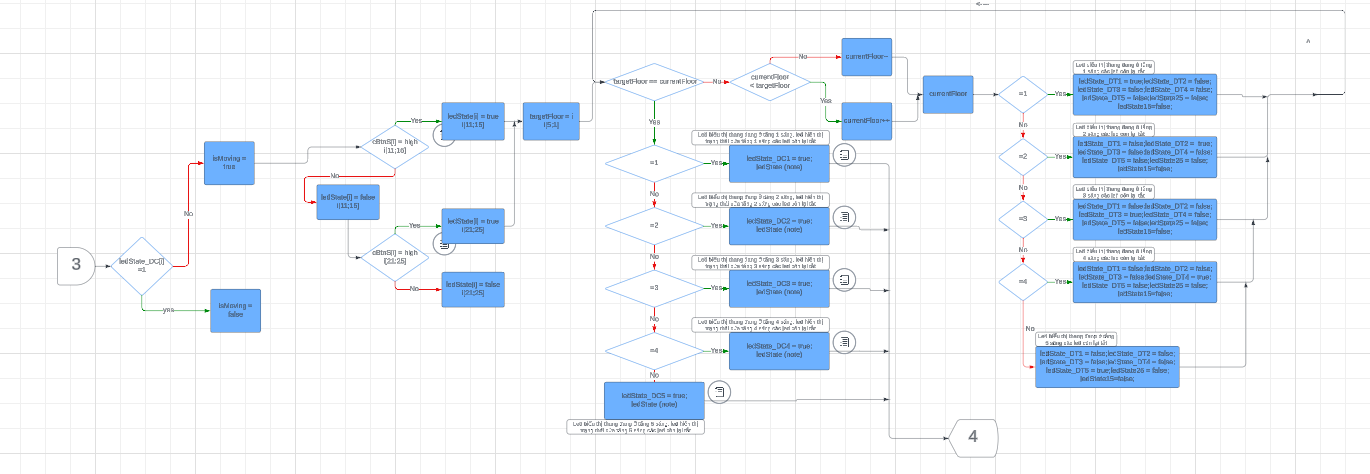


*Hình 2.7: Lưu đồ thuật toán của hệ thống*

Nếu người điều khiển không có đưa ra quyền muốn sử dụng thang máy thì hệ thống sẽ kiểm tra trạng thái của cuwat thang máy để phòng trường hợp thang máy di chuyển khi cửa mở gây là các tai nạn không đáng có cho người dùng. Trong Hình 2.7 hệ thống sẽ xét xem button Warning ở trong thang có được ấn không, Nếu nó được ấn thì có nghĩa rằng trong thang đang có người gặp rắc rối hoặc sự cố nào đó và đang cần sự trợ giúp của người bên ngoài thang. Khi đó sẽ có một cảnh báo được gửi về máy cho người quản lí để cho người quản lí kiểm tra và xem xét tình hình của thang máy rồi đưa ra các các quyết định phù hợp cho các tình hướng cần trợ giúp.

Sau khi kiểm tra nút ấn Warning, hệ thống sẽ kiểm tra lần lượt tiếp đến 2 nút ấn đóng mở cửa trong thang máy. Khi có người bên trong ấn nút mở cửa thì cửa thang máy sẽ mở, đồng thời cửa tầng tương ứng cũng sẽ mở ra. Khi cửa mở, một biến thời gian (preOpen) sẽ được gán bằng mốc thời gian lúc đó. Khi thời gian thực (millis) trừ đi preOpen >=5 thì cửa thang và cửa tầng tương ứng sẽ đóng ( Cửa sẽ đóng sau khi mở 5s).

Khi nút đóng cửa được ấn cửa thang và cửa tầng tương ứng sẽ chỉ được mở thêm 2s nữa rồi mới đóng.



*Hình 2.8: Lưu đồ thuật toán của hệ thống*

Phần lưu đồ thuật toán này (Hình 2.8) là phần được coi là phần quan trọng nhất trong đề tài này. Đó là thuật toán để hệ thống xử lí các đầu vào từ trong và ngoài thang máy của người dùng.

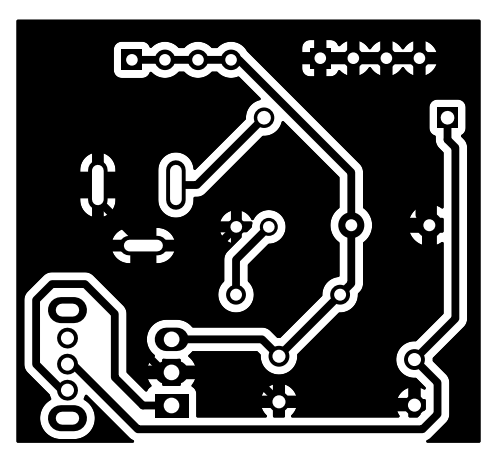
Việc đầu tiên trong phân đoạn này là kiểm tra trạng thái đóng mở cửa của thang máy để đề phòng trường hợp thang di chuyển khi cửa vẫn còn đang mở gây nguy hiểm cho người sử dụng. Nếu các cửa đều đóng, thang máy sẽ có thể di chuyển (isMoving = true) ngược lại nếu cửa đang mở thang máy sẽ không nhận bất kì lệnh gọi thang hay gọi tầng nào.

Sau khi kiểm tra cửa xong, nếu cửa đóng thang máy sẽ có thể di chuyển. Khi này nó sẽ nhận lệnh ở ngoài thang máy nếu nút ở tầng nào được bấm thì sẽ in ra biến targetfloor tương ứng của tầng. giả sử người dùng đứng ở tầng 4 ấn thì ngày lâp tức targetfloor = 4. Nếu ngoài thang không có ai ấn gì thì hệ thống sẽ nhận lệnh từ những nút bấm gọi tầng từ tầng 1 đến tầng 5 của mọi người. Khi đó vẫn in ra targetfloor tương ứng với tầng được gọi.

Tiếp đến ta so sánh biến targetfloor với biến curentfloor ( trang thái tầng hiện tại của thang máy).Nếu targetfloor == curentfloor thì có nghĩa người dùng đang ở tầng đích rồi nên cửa thang cùng với cửa tầng tương ứng sẽ mở, Nếu targetfloor < curentfloor nghĩa rằng thang đang ở trên tầng cao hơn so với tầng của người gọi. Khi này currentfloor – ( thang đang di chuyển xống). thang đi đến tầng nào thì led hiển thị tương ứng của cửa đó sẽ sáng để mô phỏng cho quá trình di chuyển của thang, Nếu targetfloor > curentfloor thì currentfloor ++ biểu thị trạng thái thang đang đi lên. Khi targetfloortăng hoặc giảm lần lượt để bằng với curentfloor Nghĩa rằng thang đang ở tầng đích ( tầng có người nhu cầu tới) thì này cửa thang sẽ mở để người dùng đi qua. Khi cửa thang được mở vẫn sẽ gán biến thời gian (preOpen) sẽ được gán bằng mốc thời gian lúc đó. Khi thời gian thực (millis) trừ đi preOpen >=5 thì cửa thang và cửa tầng tương ứng sẽ đóng ( Cửa sẽ đóng sau khi mở 5s).

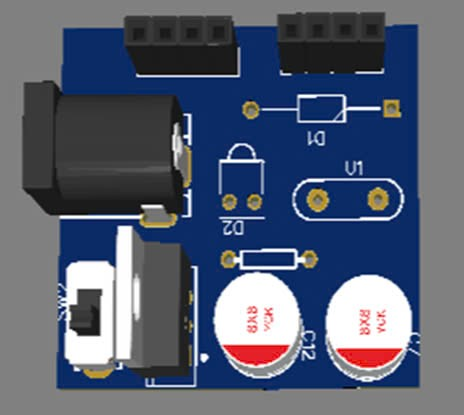
## 2.3. CHẾ TẠO MẠCH XỬ LÝ

Mạch được vẽ và hoàn thiện trên phần mềm EasyEDA.



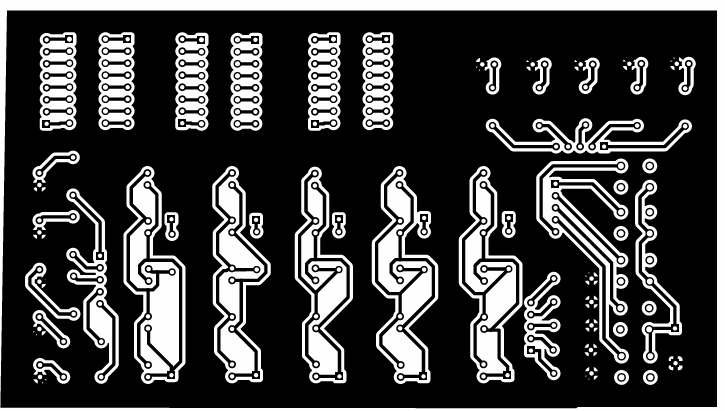
*Hình 2.7: Mạch PCB của nguồn đầu vào*

Hình 2.7 là hình vẽ mạch PCB khối nguồn đầu vào của hệ thống thang máy trên ứng dụng EDA dùng để in lên mạch đồng cho việc thiết kế mạch thủ công.



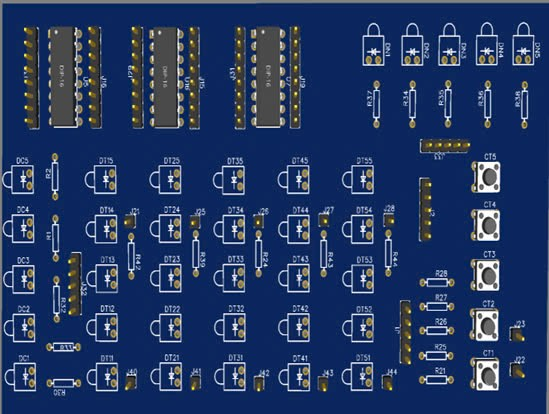
*Hình 2.8: Mạch layout 3D của nguồn đầu vào*

Hình 2.8 là hình vẽ mạch layout 3D khối nguồn đầu vào của hệ thống thang máy trên ứng dụng EDA dùng để quan sát linh kiện một cách trực quan.



*Hình 2.9: Mạch PCB của phần mạch ngoài thang*

Hình 2.9 là hình vẽ mạch PCB phần mạch ngoài của hệ thống thang máy trênứng dụng EDA dùng để in lên mạch đồng cho việc thiết kế mạch thủ công.



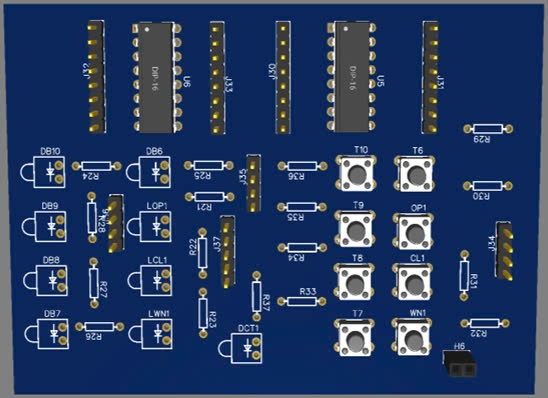
*Hình 2.10: Mạch layout 3D của phần mạch ngoài thang*

Hình 2.10 là hình vẽ mạch layout 3D khối nguồn đầu vào của hệ thống thang máy trên ứng dụng EDA dùng để quan sát linh kiện một cách trực quan.



*Hình 2.11: Mạch PCB của phần trong thang máy*

Hình 2.11 là hình vẽ mạch PCB phần mạch trong của hệ thống thang máy trênứng dụng EDA dùng để in lên mạch đồng cho việc thiết kế mạch thủ công.



*Hình 2.12: Mạch layout 3D của phần mạch trong thang*

Hình 2.10 là hình vẽ mạch layout 3D khối nguồn đầu vào của hệ thống thang máy trên ứng dụng EDA dùng để quan sát linh kiện một cách trực quan.

Bảng 2.1: Các bước thực hiện làm mạch cứng

|  |  |  |
| --- | --- | --- |
| Bước | Nội dung thực hiện | Hình ảnh quá trình làm mạch |
| 1 | Thiết kế mạch in và in mạch in lên giấy |  |
| 2 | In mạch lên board đồng bằng cách là nhiệt ở nhiệt độ phù hợp với lớp giấy in ở trên lớp đồng. |  |
| 3 | Loại bỏ lớp đồng không có mực in bằng dung dịch muối ăn mòn FeCl3 |  |
| 4 | Làm sạch lớp mực để chỉ còn lại lớp mạch đồng sáng bóng |  |
| 5 | Khoan mạch thủ công:  Chọn mũi khoan phù hợp với lỗ cắm chân linh kiện. Khoan từ từ đi xuống không có bên kia sẽ bị rỗ ra. |  |
| 6 | Hàn linh kiện:  Chuẩn bị mỏ hàn, thiếc, mỡ hàn. Vệ sinh mỏ hàn trước khi hàn. Hàn gắn từng linh kiện vào mạch. |  |
| 7 | Thiết kế mô hình và vị trí đặt mạch. |  |

## 

## 2.5. KẾT LUẬN CHƯƠNG

Chương II đã trình bày về nội dung Thiết kế hệ thống. Tiếp theo chương III sẽ trình bày về kết quả sau khi hoàn thiện hệ thống.

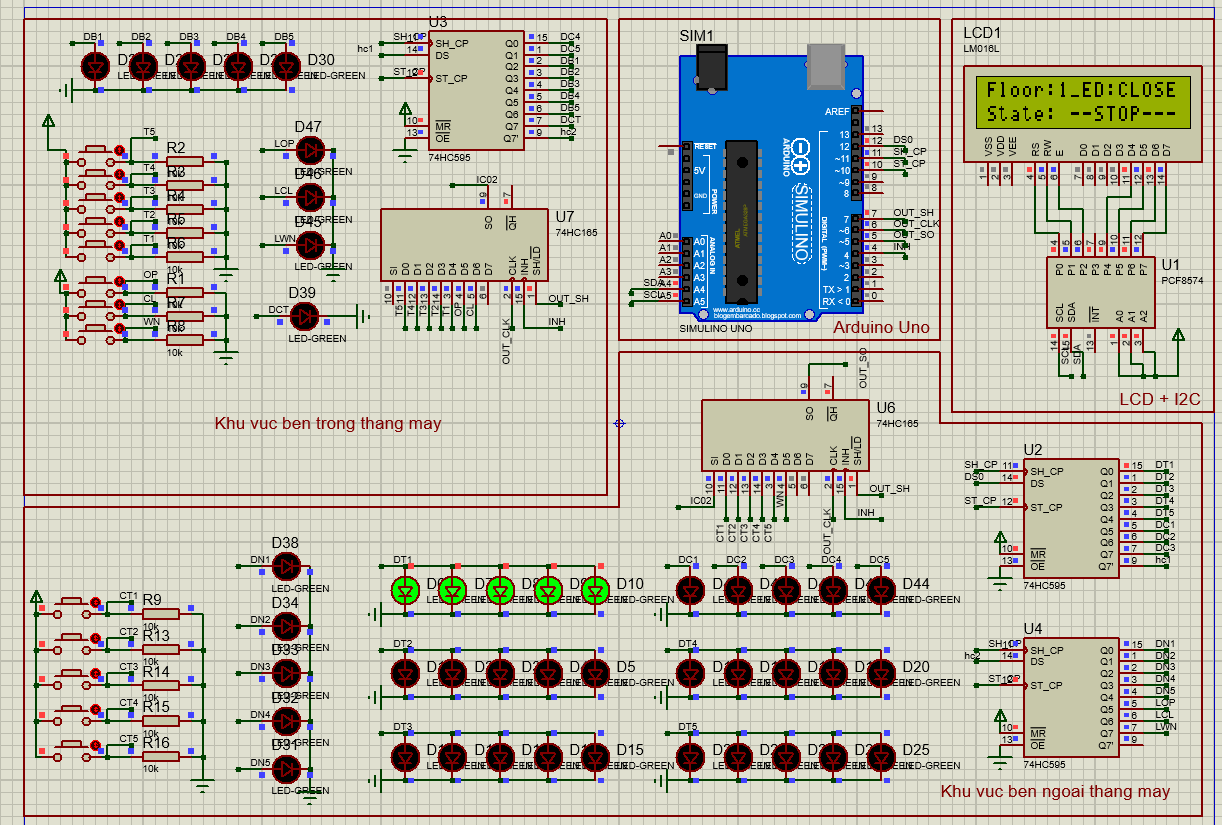
# CHƯƠNG III: KẾT LUẬN ĐỒ ÁN

Chương 3 đánh dấu phần kết thúc của quá trình nghiên cứu về hệ thống thang máy sử dụng Arduino. Trong chương này tổng kết lại những điểm nổi bật và những hạn chế của hệ thống, đồng thời đề xuất các hướng phát triển tiềm năng.

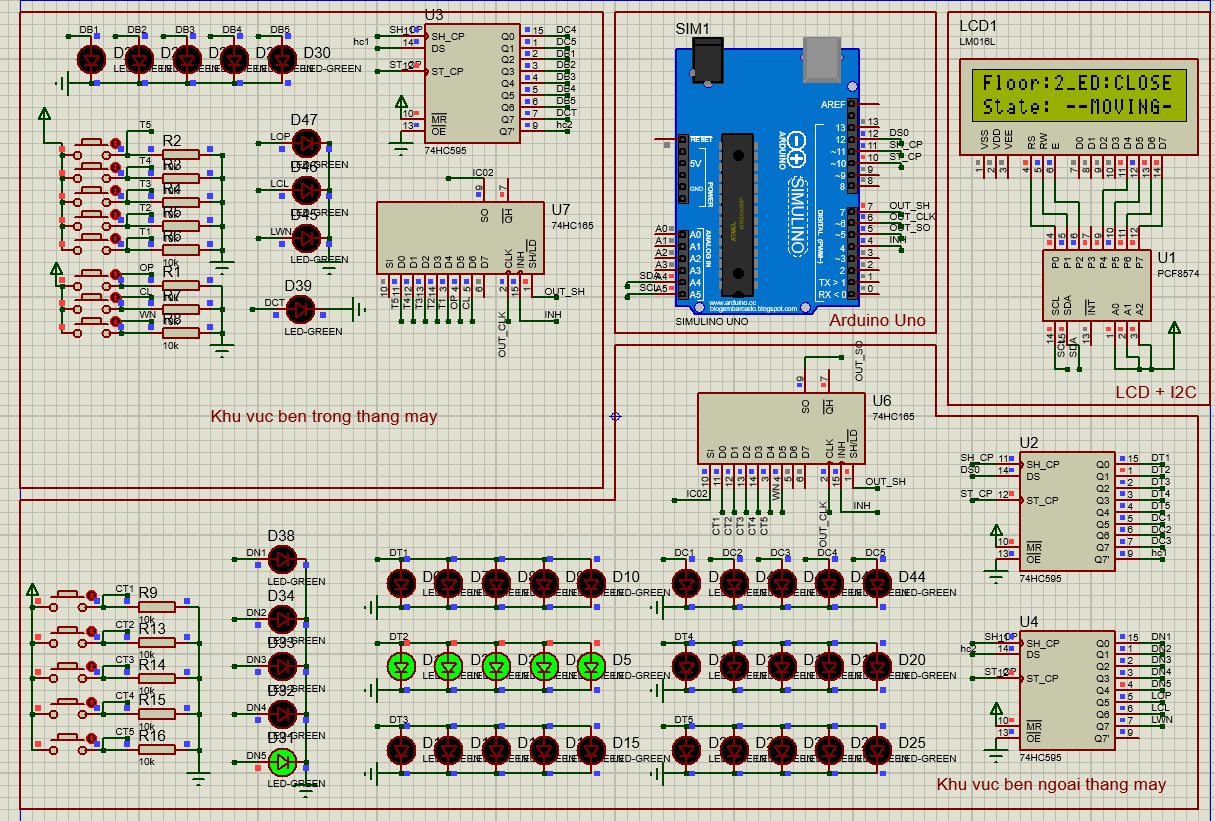
## 3.1. SẢN PHẨM ĐẠT ĐƯỢC

Mô phỏng trên proteus:

Trạng thái ban đầu

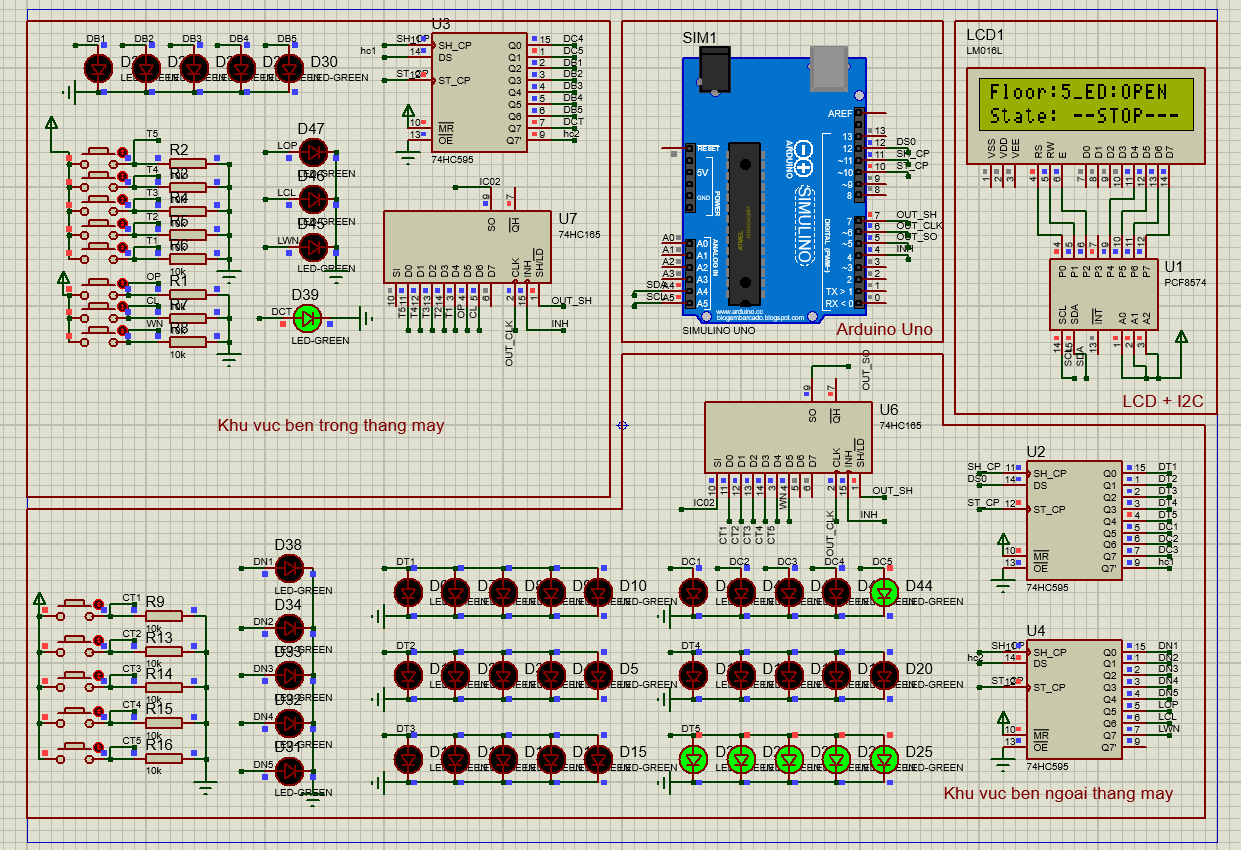
Hình 3.1: Trạng thái ban đầu

Khi bắt đàu cấp nguồn cho hệ thống thang sẽ ở tầng 1 ứng với led mô phỏng thang đang ở tầng 1 của cả 5 tầng sáng lên như trên (Hình 3.1). Khi này LCD cũng hiển thị những trạng thái như tầng hiện tại của thang đang là 1, trạng thái cửa là đang đóng và thang máy đang dừng



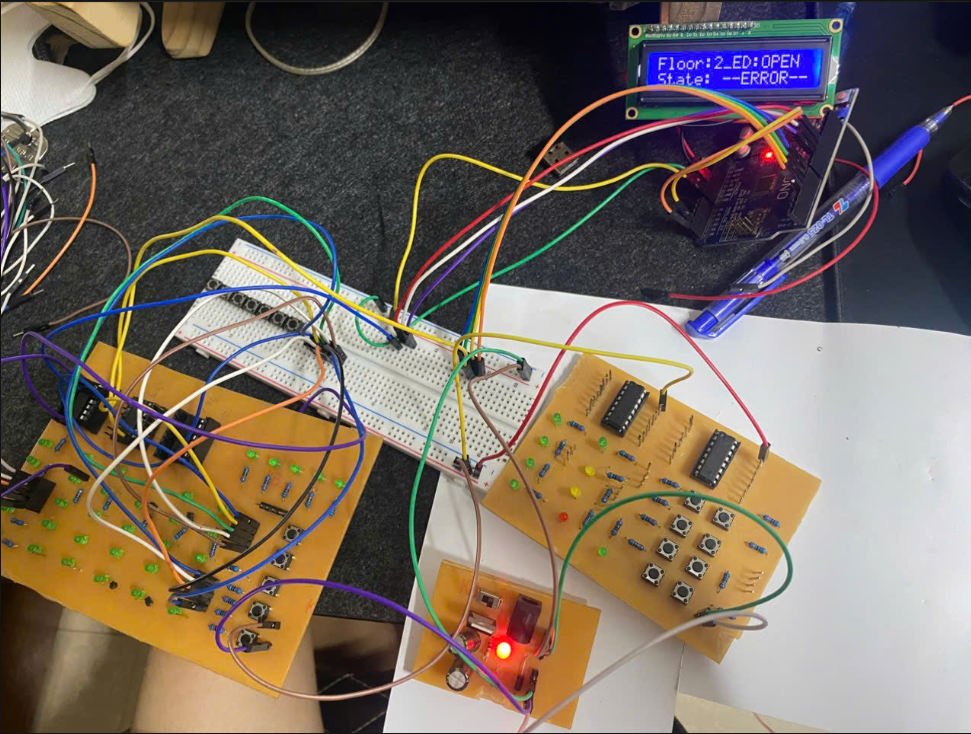
Hình 3.2: Trạng thái khi ấn tầng 5

Gỉa xử có người đang đứng ngoài ở tầng 5 bấm nút gọi thang thì đèn báo của nút gọi đó sẽ sáng lên đến khi nào thang đến tầng đó thì mới tắt. Với trạng thái đâu là thang đang ở tầng 1 (Hình 3,1) khi này thang sẽ di chuyển lên tầng 5 để mô phỏng sự di chuyển đó led biểu hị tầng của thang sẽ chạy lần lượt từ 1,2,3,4, Khi này LCD sẽ hiển thị lần lượt các tầng 1,2,3,4 và trạng thái thang là đang di chuyển. cuối cùng thang đến và dừng ở tầng 5. Khi thang máy đến tầng 5, led mô phỏng cửa thang máy và cửa tầng 5 sẽ sáng lên trong khoảng 5s biểu thị cửa thang và tầng đang mở để khách ra trong 5s rồi lại đóng. Ngoài ra LCD cũng hiển thị của thang máy là STOP (Hình 3.3).



Hình 3.2: Trạng thái khi đến tầng 5

Mạch thực tế:



## 3.2. NHẬN XÉT

Hệ thống thang máy hiện tại đã đạt được những bước tiến lớn trong việc nâng cao hiệu quả hoạt động và tối ưu hóa trải nghiệm người sử dụng. Việc áp dụng các thuật toán điều khiển cơ bản giúp thang máy có thể hoạt động ổn định trong môi trường ít người sử dụng. Thang di chuyển nhanh chóng và dễ dàng phản hồi với lệnh gọi từ các nút bấm. Điều này mang lại sự thuận tiện và tiết kiệm thời gian cho người dùng trong những tòa nhà nhỏ hoặc ít tầng.

Bên cạnh đó, hệ thống cũng đã cải thiện khả năng vận hành trong môi trường có nhiều lệnh gọi đồng thời. Các nút bấm trong thang và ngoài thang đã được tối ưu hóa để giảm thiểu thời gian chờ đợi, giúp thang di chuyển một cách hợp lý và tiết kiệm thời gian cho người sử dụng. Mặc dù chưa áp dụng công nghệ AI hay Machine Learning, nhưng hệ thống đã có thể điều chỉnh hoạt động dựa trên nhu cầu thực tế, mang lại hiệu quả trong những tình huống ít phức tạp.

### 3.2.1. Ưu điểm

Hệ thống thang máy đơn giản và dễ triển khai là lựa chọn lý tưởng cho các tòa nhà nhỏ hoặc các tòa nhà ít tầng. Không cần phải đầu tư vào các công nghệ phức tạp hoặc các thuật toán tối ưu hóa, điều này giúp giảm chi phí ban đầu cũng như chi phí bảo trì trong suốt quá trình vận hành. Việc sử dụng nút bấm truyền thống cũng dễ dàng tiếp cận và sử dụng cho người dân, không đòi hỏi quá nhiều sự thay đổi về thói quen sử dụng.

Trong những tòa nhà có ít người sử dụng hoặc ít lệnh gọi, hệ thống thang máy cơ bản có thể hoạt động một cách nhanh chóng và hiệu quả. Thang máy có thể di chuyển trực tiếp đến các tầng cần thiết mà không gặp phải sự tắc nghẽn hay chậm trễ. Điều này giúp tiết kiệm thời gian cho người dùng, đồng thời tránh được các vấn đề phức tạp khi không có quá nhiều lệnh gọi đồng thời, đảm bảo rằng thang máy hoạt động một cách trơn tru và không tốn quá nhiều tài nguyên.

### 3.2.2. Nhược điểm

Hệ thống thang máy sẽ gặp phải nhiều vấn đề về hiệu quả hoạt động. Đầu tiên, thuật toán điều khiển thang không tối ưu sẽ dẫn đến thời gian di chuyển dài hơn, đặc biệt khi có nhiều lệnh gọi đồng thời từ các tầng khác nhau, cả trong và ngoài thang. Người sử dụng phải chờ lâu hơn, và thang có thể di chuyển qua các tầng không cần thiết, gây lãng phí thời gian và năng lượng. Hơn nữa, việc phản hồi chậm khi bấm nút bên trong và bên ngoài thang cũng làm giảm sự hài lòng của người dùng. Nếu không có mô phỏng thực tế với các yếu tố như buồng thang, cửa thang hay hệ thống cáp treo, việc thiết kế và vận hành thang sẽ thiếu tính linh hoạt và hiệu quả, gây khó khăn cho việc nâng cấp và mở rộng hệ thống khi cần thiết.

Một hệ thống thang máy chưa được tối ưu hóa sẽ không thể kiểm soát tốt tốc độ di chuyển và thời gian đón khách. Trong tình huống có nhiều người sử dụng, thang có thể gặp khó khăn khi xử lý nhiều lệnh gọi cùng lúc, đặc biệt là khi thang gần đến tầng đích hoặc khi có nhiều người ở trong thang. Việc này không chỉ làm giảm hiệu suất mà còn ảnh hưởng đến sự an toàn và thoải mái của người dùng. Nếu không có các giải pháp như tích hợp AI hoặc Machine Learning để tối ưu hóa thời gian đón khách, thang sẽ không thể linh hoạt điều chỉnh hành trình sao cho phù hợp với nhu cầu thực tế. Thêm vào đó, việc mở rộng hệ thống thang máy cho các tòa nhà cao tầng hay các chung cư mà không có những cải tiến này sẽ trở nên khó khăn, dẫn đến việc quản lý và vận hành không hiệu quả.

### 3.2.3. Giải pháp khắc phục

Để tối ưu hóa hiệu quả của hệ thống thang máy trong môi trường có nhiều người sử dụng hoặc nhiều lệnh gọi đồng thời, việc áp dụng thuật toán tối ưu hóa di chuyển là rất cần thiết. Những thuật toán này có thể giúp tính toán lộ trình di chuyển sao cho hợp lý nhất, giảm thiểu việc thang máy phải đi qua các tầng không cần thiết. Điều này không chỉ giúp tiết kiệm thời gian mà còn giúp giảm bớt tắc nghẽn và thời gian chờ đợi của người dùng, nâng cao hiệu quả vận hành và sự hài lòng của khách hàng.

Tích hợp công nghệ AI hoặc Machine Learning vào hệ thống thang máy sẽ giúp nâng cao khả năng tự động hóa và tối ưu hóa quá trình vận hành. AI có thể học hỏi và dự đoán các mẫu hành vi của người dùng, từ đó điều chỉnh tốc độ và lộ trình di chuyển sao cho hợp lý nhất. Thang máy có thể chủ động thay đổi lộ trình dựa trên nhu cầu thực tế, giúp giảm thiểu thời gian chờ đợi, tối ưu hóa thời gian đón khách và phản hồi nhanh chóng hơn trong những tình huống có nhiều người sử dụng hoặc lệnh gọi cùng lúc.

Tổng quát, việc sử dụng các giải pháp kỹ thuật và chiến lược quản lý thông minh có thể giúp giảm thiểu nhược điểm của hệ thống và tạo ra một giải pháp toàn diện và đáng tin cậy cho người dùng.

## 3.3. KẾT LUẬN

Sau quá trình tìm hiểu, nghiên cứu từ các tài liệu tham khảo thì đã giải quyết được tương đối yêu cầu của đề tài thiết kế “ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO” với đầy đủ các tính năng, nội dung và mục tiêu ban đầu đề ra.

## 3.4. HƯỚNG PHÁT TRIỂN

Sau khi hoàn thành đề tài nghiên cứu “ĐIỀU KHIỂN THANG MÁY DÙNG ARDUINO”, một đề tài có tính thực tiễn lớn, quá trình làm việc mang lại cho chúng tôi một chiều sâu về thuật toán của thang máy. Nếu có cơ họi phát triển thì dưới đây là những gì chúng tôi cần nghiên cứu và bổ sung để cho đề tài được trọn vẹn và thuật toán được tối ưu hơn:

* Cải thiện thuật toán giúp tối ưu thời gian di chuyển của người dùng khi cùng lúc có nhiều lệnh gọi cả trong và ngoài thang.
* Tối ưu phẩn hồi khi bấm nút trong và ngoài thang của người dùng.
* Làm them hệ thống thang máy lớn hơn (ở chung cư) có nhiều tầng và nhiều thang.
* Mô hình mô phỏng thang máy gần giống với thực tế nhất. Cũng có buồng thang, cửa thang, hệ thống cáp treo, …
* Kiểm soát tốc độ di chuyển của thang khi có nhiều người ấn, khi gần đến thang đích, khi có nhiều người ở trong, …
* Tích hợp thêm Ai hoặc ML để tối ưu hóa thời gian đón khách.

Một lần nữa chúng tôi xin chân thành cảm ơn TS. Nguyễn Thị Thanh Quỳnh đã nhiệt tình hướng dẫn, truyền đạt kiến thức trong suốt quá thực hiện đồ án này.

# TÀI LIỆU THAM KHẢO

[1] https://www.alldatasheet.com/

[2]. https://docs.arduino.cc/tutorials/communication/guide-to-shift-out/

[3] https://forum.arduino.cc/t/serial-data-problem-with-75hc165/207945

[4]https://www.makerguides.com/interfacing-parallel-in-serial-out-shift-register-74hc165-with-arduino/

# PHỤ LỤC

Code cho Arduino Uno:

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

#define NUMBER\_OF\_SHIFT\_CHIPS 2

#define DATA\_WIDTH NUMBER\_OF\_SHIFT\_CHIPS \* 8

#define TotalIC 3

#define TotalICPins TotalIC \* 8

unsigned long preClose = 0;

unsigned long time5s = 5000;

const unsigned long tmove = 2000;

unsigned long preOpen = 0;

unsigned long timeclose = 2000;

int dlbt = 5;

int cr\_Floor = 1;

int LoadPin = 7;

int EnablePin = 4;

int DataPin = 5;

int ClockPin = 6;

int RCLK = 10;//st cp 12

int SER = 12;// data 14

int SRCLK = 11;//clk 11 sh cp

unsigned long pinValues;

unsigned long oldPinValues;

int currentFloor = 1; // Biến để theo dõi tầng hiện tại của thang máy

bool isMoving = false; // Biến theo dõi trạng thái di chuyển

//.....................Trạng thái LED

//..........IC74HC165\_1

bool ledState11 = false; // Trạng thái LED (bật/tắt)đèn ngoài t5

bool ledState12 = false; // Trạng thái LED (bật/tắt)đèn ngoài t4

bool ledState13 = false; // Trạng thái LED (bật/tắt)đèn ngoài t3

bool ledState14 = false; // Trạng thái LED (bật/tắt)đèn ngoài t2

bool ledState15 = false; // Trạng thái LED (bật/tắt)đèn ngoài t1

bool lastButtonState28 = false;

//..........IC74HC165\_2

bool ledState21 = false; // Trạng thái LED (bật/tắt)DB5

bool ledState22 = false; // Trạng thái LED (bật/tắt)DB4

bool ledState23 = false; // Trạng thái LED (bật/tắt)DB3

bool ledState24 = false; // Trạng thái LED (bật/tắt)DB2

bool ledState25 = false; // Trạng thái LED (bật/tắt)DB1

bool ledState26 = false; // Trạng thái LED (bật/tắt)LOP

bool ledState27 = false; // Trạng thái LED (bật/tắt)LCL

bool ledState28 = false; // Trạng thái LED (bật/tắt)LWN

//...............END

//...............Led hiển thị

bool ledState\_DT1 = false, ledState\_DT2 = false, ledState\_DT3 = false, ledState\_DT4 = false, ledState\_DT5 = false;

bool ledState\_DC1 = false, ledState\_DC2 = false, ledState\_DC3 = false, ledState\_DC4 = false, ledState\_DC5 = false;

bool ledState\_DCT = false;

//end

void setup() {

//SET LCD

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("Floor:");

lcd.setCursor(7, 0);

lcd.print("\_ED:");

lcd.setCursor(11, 0);

lcd.print("CLOSE");

lcd.setCursor(6, 0);

lcd.print("1");

lcd.setCursor(0, 1);

lcd.print("State:");

lcd.setCursor(7, 1);

lcd.print("--STOP---");

pinMode(LoadPin, OUTPUT);

pinMode(EnablePin, OUTPUT);

pinMode(ClockPin, OUTPUT);

pinMode(DataPin, INPUT);

digitalWrite(ClockPin, LOW);

digitalWrite(LoadPin, HIGH);

pinMode(SER, OUTPUT);

pinMode(RCLK, OUTPUT);

pinMode(SRCLK, OUTPUT);

digitalWrite(EnablePin, HIGH);

//Trang thai ban dau (LED hien thi tang 1 sang)

ledState\_DT1 = true;

ledState\_DT2 = false;

ledState\_DT3 = false;

ledState\_DT4 = false;

ledState\_DT5 = false;

ledState\_DC1 = false;

ledState\_DC2 = false;

ledState\_DC3 = false;

ledState\_DC4 = false;

ledState\_DC5 = false;

ledState\_DCT = false;

ledState21 = false; ledState22 = false; ledState23 = false; ledState24 = false; ledState25 = false; ledState26 = false; ledState27 = false; ledState28 = false;

ledState11 = false; ledState12 = false; ledState13 = false; ledState14 = false; ledState15 = false;

updateLEDs();

}

void loop() {

pinValues = read\_shift\_regs();

if (pinValues != oldPinValues)

{

oldPinValues = pinValues;

}

digitalWrite(LoadPin, LOW); // Kích hoạt để tải dữ liệu song song

delayMicroseconds(5);

digitalWrite(LoadPin, HIGH); // Đưa PL trở lại HIGH để đọc dữ liệu nối tiếp

// Bắt đầu đọc dữ liệu từ các IC

byte dataIC1 = shiftIn(DataPin, ClockPin, MSBFIRST);

byte dataIC2 = shiftIn(DataPin, ClockPin, MSBFIRST);

//...................Set button

//.................ic74HC165\_1

bool cBtnS11 = bitRead(dataIC1, 5); delay(dlbt);//bt led ngoai tang 5

bool cBtnS12 = bitRead(dataIC1, 4); delay(dlbt);//bt led ngoai tang 4

bool cBtnS13 = bitRead(dataIC1, 3); delay(dlbt);//bt led ngoai tang 3

bool cBtnS14 = bitRead(dataIC1, 2); delay(dlbt);//bt led ngoai tang 2

bool cBtnS15 = bitRead(dataIC1, 1); delay(dlbt);//bt led ngoai tang 1

bool cBtnS26 = bitRead(dataIC2, 6); delay(dlbt);//bt open

bool cBtnS27 = bitRead(dataIC2, 7); delay(dlbt);//bt close

bool cBtnS28 = bitRead(dataIC1, 6); delay(dlbt);//bt warning

bool cBtnS21 = bitRead(dataIC2, 1); delay(dlbt);//bt led trong tang 5

bool cBtnS22 = bitRead(dataIC2, 2); delay(dlbt);//bt led trong tang 4

bool cBtnS23 = bitRead(dataIC2, 3); delay(dlbt);//bt led trong tang 3

bool cBtnS24 = bitRead(dataIC2, 4); delay(dlbt);//bt led trong tang 2

bool cBtnS25 = bitRead(dataIC2, 5); delay(dlbt);//bt led trong tang 1

if (ledState\_DC1 || ledState\_DC2 || ledState\_DC3 || ledState\_DC4 || ledState\_DC5) {

ledState\_DCT = true;

} else {

ledState\_DCT = false;

}

// BT5

if (cBtnS11 == true && currentFloor == 5) {

ledState11 = false;

ledState\_DC5 = true;

preOpen = millis();

updateLEDs();

} else if (cBtnS11 == true && currentFloor != 5) {

ledState11 = true;

}

// BT4

if (cBtnS12 == true && currentFloor == 4) {

ledState12 = false;

ledState\_DC4 = true;

preOpen = millis();

updateLEDs();

} else if (cBtnS12 == true && currentFloor != 4) {

ledState12 = true;

}

// BT3

if (cBtnS13 == true && currentFloor == 3) {

ledState13 = false;

ledState\_DC3 = true;

preOpen = millis();

updateLEDs();

} else if (cBtnS13 == true && currentFloor != 3) {

ledState13 = true;

}

// BT2

if (cBtnS14 == true && currentFloor == 2) {

ledState14 = false;

ledState\_DC2 = true;

preOpen = millis();

updateLEDs();

} else if (cBtnS14 == true && currentFloor != 2) {

ledState14 = true;

}

// BT1

if (cBtnS15 == true && currentFloor == 1) {

ledState15 = false;

ledState\_DC1 = true;

preOpen = millis();

updateLEDs();

} else if (cBtnS15 == true && currentFloor != 1) {

ledState15 = true;

}

// T5

if (cBtnS21 == true && currentFloor == 5) {

ledState21 = false;

} else if (cBtnS21 == true && currentFloor != 5) {

ledState21 = true;

}

// T4

if (cBtnS22 == true && currentFloor == 4) {

ledState22 = false;

} else if (cBtnS22 == true && currentFloor != 4) {

ledState22 = true;

}

// T3

if (cBtnS23 == true && currentFloor == 3) {

ledState23 = false;

} else if (cBtnS23 == true && currentFloor != 3) {

ledState23 = true;

}

// T2

if (cBtnS24 == true && currentFloor == 2) {

ledState24 = false; // Đảo ngược trạng thái của LED

} else if (cBtnS24 == true && currentFloor != 2) {

ledState24 = true;

}

// T1

if (cBtnS25 == true && currentFloor == 1) {

ledState25 = false; // Đảo ngược trạng thái của LED

} else if (cBtnS25 == true && currentFloor != 1) {

ledState25 = true;

}

// .............................OP

if (cBtnS26 == true) {

ledState26 = true;

}

if (cBtnS26 == false) {

ledState26 = false;

}

//............................. CL

if (cBtnS27 == true) {

ledState27 = true;

}

if (cBtnS27 == false) {

ledState27 = false;

}

if (cBtnS26 == true ) {

if (ledState26 == true && ledState\_DT1 == true ) {

ledState\_DC1 = true;

preOpen = millis();

updateLEDs();

}

if (ledState26 == true && ledState\_DT2 == true ) {

ledState\_DC2 = true;

preOpen = millis();

updateLEDs();

}

if (ledState26 == true && ledState\_DT3 == true ) {

ledState\_DC3 = true;

preOpen = millis();

updateLEDs();

}

if (ledState26 == true && ledState\_DT4 == true ) {

ledState\_DC4 = true;

preOpen = millis();

updateLEDs();

}

if (ledState26 == true && ledState\_DT5 == true ) {

ledState\_DC5 = true;

preOpen = millis();

updateLEDs();

}

} else if (cBtnS27 == true ) {

//time5s = timeclose;

preOpen = millis() - 3000;

}

if (ledState\_DC1 && millis() - preOpen >= time5s) {

ledState\_DC1 = false;

updateLEDs();

}

if (ledState\_DC2 && millis() - preOpen >= time5s) {

ledState\_DC2 = false;

updateLEDs();

}

if (ledState\_DC3 && millis() - preOpen >= time5s) {

ledState\_DC3 = false;

updateLEDs();

}

if (ledState\_DC4 && millis() - preOpen >= time5s) {

ledState\_DC4 = false;

updateLEDs();

}

if (ledState\_DC5 && millis() - preOpen >= time5s) {

ledState\_DC5 = false;

updateLEDs();

}

// WN

if (cBtnS28 && !lastButtonState28) {

ledState28 = !ledState28;

} lastButtonState28 = cBtnS28;

///..........END

// Xử lý các logicState và di chuyển thang máy

if

(ledState15 && currentFloor != 1) {

moveToFloor(1);

ledState15 = false;

} else if (ledState14 && currentFloor != 2) {

moveToFloor(2);

ledState14 = false;

} else if (ledState13 && currentFloor != 3) {

moveToFloor(3);

ledState13 = false;

} else if (ledState12 && currentFloor != 4) {

moveToFloor(4);

ledState12 = false;

} else if (ledState11 && currentFloor != 5) {

moveToFloor(5);

ledState11 = false;

}

if (ledState25 && currentFloor != 1) {

moveToFloor(1);

ledState25 = false;

} else if (ledState24 && currentFloor != 2) {

moveToFloor(2);

ledState24 = false;

} else if (ledState23 && currentFloor != 3) {

moveToFloor(3);

ledState23 = false;

} else if (ledState22 && currentFloor != 4) {

moveToFloor(4);

ledState22 = false;

} else if (ledState21 && currentFloor != 5) {

moveToFloor(5);

ledState21 = false;

}

updateLEDs();

//UPDATE LCD

if (ledState28 == true) {

lcd.setCursor(6, 1);

lcd.print(" --ERROR--");

}

if (ledState\_DCT) {

lcd.setCursor(11, 0);

lcd.print("OPEN ");

} else {

lcd.setCursor(11, 0);

lcd.print("CLOSE");

}

}

unsigned long read\_shift\_regs()

{

long bitVal;

unsigned long bytesVal = 0;

digitalWrite(EnablePin, HIGH);

digitalWrite(LoadPin, LOW);

delayMicroseconds(5);

digitalWrite(LoadPin, HIGH);

digitalWrite(EnablePin, LOW);

for (int i = 0; i < DATA\_WIDTH; i++)

{

bitVal = digitalRead(DataPin);

digitalWrite(ClockPin, HIGH);

delayMicroseconds(5);

digitalWrite(ClockPin, LOW);

}

return (bytesVal);

}

void updateLCD( int crFloor) {

if (crFloor == 1 ) {

lcd.setCursor(11, 0);

lcd.print("CLOSE");

lcd.setCursor(6, 0);

lcd.print("1");

} else if (crFloor == 2) {

lcd.setCursor(11, 0);

lcd.print("CLOSE");

lcd.setCursor(6, 0);

lcd.print("2");

} else if (crFloor == 3) {

lcd.setCursor(6, 0);

lcd.print("3");

} else if (crFloor == 4) {

lcd.setCursor(6, 0);

lcd.print("4");

} else if (crFloor == 5) {

lcd.setCursor(6, 0);

lcd.print("5");

}

}

void moveToFloor(int targetFloor) {

if (ledState\_DCT == false) {

if (isMoving || targetFloor == currentFloor) {

//...

//...

return;

}

isMoving = true; // Thang bắt đầu di chuyển

while (currentFloor != targetFloor) {

delay(tmove);

if (currentFloor < targetFloor) {

currentFloor++; // Di chuyển lên

} else {

currentFloor--; // Di chuyển xuống

}

if (currentFloor == 1) {

updateLCD(1);

ledState\_DT1 = true; ledState\_DT2 = false;

ledState\_DT3 = false; ledState\_DT4 = false;

ledState\_DT5 = false; ledState25 = false;

ledState15 = false;

} else if (currentFloor == 2) {

updateLCD(2);

ledState\_DT1 = false;

ledState\_DT2 = true;

ledState\_DT3 = false;

ledState\_DT4 = false;

ledState\_DT5 = false;

ledState24 = false;

ledState14 = false;

} else if (currentFloor == 3) {

updateLCD(3);

ledState\_DT1 = false;

ledState\_DT2 = false;

ledState\_DT3 = true;

ledState\_DT4 = false;

ledState\_DT5 = false;

ledState23 = false;

ledState13 = false;

} else if (currentFloor == 4) {

updateLCD(4);

ledState\_DT1 = false;

ledState\_DT2 = false;

ledState\_DT3 = false;

ledState\_DT4 = true;

ledState\_DT5 = false;

ledState22 = false;

ledState12 = false;

} else if (currentFloor == 5) {

updateLCD(5);

ledState\_DT1 = false;

ledState\_DT2 = false;

ledState\_DT3 = false;

ledState\_DT4 = false;

ledState\_DT5 = true;

ledState21 = false;

ledState11 = false;

}

if (currentFloor == 1 && targetFloor == 1) {

ledState\_DC1 = true;

preOpen = millis();

lcd.setCursor(6, 1);

lcd.print(" --STOP---");

} else if (currentFloor == 2 && targetFloor == 2) {

ledState\_DC2 = true;

preOpen = millis();

lcd.setCursor(6, 1);

lcd.print(" --STOP---");

} else if (currentFloor == 3 && targetFloor == 3) {

ledState\_DC3 = true;

preOpen = millis();

lcd.setCursor(6, 1);

lcd.print(" --STOP---");

} else if (currentFloor == 4 && targetFloor == 4) {

ledState\_DC4 = true;

preOpen = millis();

lcd.setCursor(6, 1);

lcd.print(" --STOP---");

} else if (currentFloor == 5 && targetFloor == 5) {

ledState\_DC5 = true;

preOpen = millis();

lcd.setCursor(6, 1);

lcd.print(" --STOP---");

} else {

lcd.setCursor(6, 1);

lcd.print(" --MOVING-");

}

updateLEDs();

}

} else {

isMoving = false; // Thang dừng di chuyển

}

}

void updateLEDs() {

// Gửi dữ liệu để điều khiển LED qua 74HC595

byte ic595\_1 = (ledState\_DT1 << 0) | (ledState\_DT2 << 1) | (ledState\_DT3 << 2) | (ledState\_DT4 << 3) | (ledState\_DT5 << 4) | (ledState\_DC1 << 5) | (ledState\_DC2 << 6) | (ledState\_DC3 << 7);

byte ic595\_2 = (ledState\_DC4 << 0) | (ledState\_DC5 << 1) | (ledState25 << 2) | (ledState24 << 3) | (ledState23 << 4) | (ledState22 << 5) | (ledState21 << 6) | (ledState\_DCT << 7);

byte ic595\_3 = (ledState11 << 4) | (ledState12 << 3) | (ledState13 << 2) | (ledState14 << 1) | (ledState15 << 0) | (ledState26 << 5) | (ledState27 << 6) | (ledState28 << 7);

digitalWrite(RCLK, LOW);

shiftOut(SER, SRCLK, MSBFIRST, ic595\_3); // 74HC595\_3

shiftOut(SER, SRCLK, MSBFIRST, ic595\_2); //74HC595\_2

shiftOut(SER, SRCLK, MSBFIRST, ic595\_1); //74HC595\_1

digitalWrite(RCLK, HIGH);

delay(5);

}