**ĐẠI HỌC PHENIKAA**

**TRƯỜNG KỸ THUẬT PHENIKAA**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

A logo on a black background

AI-generated content may be incorrect.

**BÁO CÁO BÀI TẬP LỚN HỌC PHẦN: HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG DCS VÀ SCADA**

**ĐỀ TÀI**

**LẬP TRÌNH ĐIỀU KHIỂN GIÁM SÁT HỆ THỐNG ĐIỀU KHIỂN THANG MÁY 4 TẦNG**

Sinh viên thực hiên: Trần Sơn Dương – 21011224

Giảng viên hướng dẫn: TS. Vũ Quốc Đông

Khoa : Điện – Điện tử

*Hà Nội, tháng 7 năm 2025*

**ĐẠI HỌC PHENIKAA**

**TRƯỜNG KỸ THUẬT PHENIKAA**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

A logo on a black background

AI-generated content may be incorrect.

**BÁO CÁO BÀI TẬP LỚN HỌC PHẦN: HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG DCS VÀ SCADA**

**ĐỀ TÀI**

**LẬP TRÌNH ĐIỀU KHIỂN GIÁM SÁT HỆ THỐNG ĐIỀU KHIỂN THANG MÁY 4 TẦNG**

Sinh viên thực hiên: Trần Sơn Dương – 21011224

Nguyễn Văn Hải – 21011657

Trần Văn Hợp – 22010119

Giảng viên hướng dẫn: TS. Vũ Quốc Đông

Khoa : Điện – Điện tử

*Hà Nội, tháng 7 năm 2025*

**LỜI CAM ĐOAN**

Nhóm em xin cam đoan rằng đề tài “L*ập trình điều khiển giám sát hệ thống điều khiển thang máy 4 tầng*” là kết quả của quá trình nghiên cứu độc lập, không sao chép, thuê mượn hay copy từ bất kỳ nguồn nào. Đây là sản phẩm mà nhóm em đã nỗ lực thực hiện trong suốt quá trình học tập tại trường. Trong quá trình nghiên cứu và hoàn thiện bài viết, nhóm em có tham khảo một số tài liệu có nguồn gốc rõ ràng, dưới sự hướng dẫn của *Giảng viên Khoa Điện – Điện tử, Trường Kỹ thuật Phenikaa*. Nhóm em xin chịu hoàn toàn trách nhiệm nếu có bất kỳ vấn đề nào liên quan đến tính trung thực của đề tài.

Nhóm sinh viên thực hiện

*(Ký, ghi rõ họ tên)*

**TIẾN ĐỘ THỰC HIỆN**

**1.Thành viên nhóm**

|  |  |
| --- | --- |
| STT | Thông tin cá nhân |
| 1 | Họ và tên: Trần Sơn Dương  Mã SV: 21011224 |
| 2 | Họ và tên: Nguyễn Văn Hải  Mã SV: 21011657 |
| 3 | Họ và tên; Trần Văn Hợp  Mã SV: 22010119 |

**2.Phân công nhiệm vụ**

|  |  |  |
| --- | --- | --- |
| Thành viên | Nhiệm vụ được giao | Nhóm đánh giá |
| Trần Sơn Dương | Code PLC, Thiết kế App, HMI,WinCC, làm báo cáo | 100% |
| Nguyễn Văn Hải | Code PLC, Thiết kế App, HMI, làm slide | 100% |
| Trần Văn Hợp | Code PLC, Thiết kế WinCC,HMI, làm báo cáo | 100% |

**3.Tiến độ thực hiện**

|  |  |  |
| --- | --- | --- |
| Nội dung | Thời gian thực hiện | Kết quả |
| Lên kế hoạch thiết kế cho toàn bộ hệ thống điều khiển thang máy 4 tầng | 21/04/2025 – 10/05/2025 | Hoàn thành |
| Thực hiện vẽ các bản vẽ lưu đồ thuật toán, sơ đồ đấu nối điện | 11/05/2025 – 20/05/2025 | Hoàn thành |
| Lập trình cho hệ thống, chạy test hệ thống trên phòng thực hành | 21/05/2025- 25/06/2025 | Hoàn thành |
| Lập trình App Android, thiết kế giao diện WinCC, HMI | 26/06/2025 – 17/07/2025 | Hoàn thành |
| Viết báo cáo và làm slide thuyết trình | 18/07/2025 – 29/07/2025 | Hoàn thành |

**4.Đánh giá tiến độ thực hiện**

......................................................................................................................................................................................................................................................................................................................................................................................................

|  |  |
| --- | --- |
| Nhóm sinh viên thực hiện | GIẢNG VIÊN |
| *(Ký, ghi rõ họ và tên)* | *(Ký, ghi rõ họ tên)* |

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc204682473)

[I.TỔNG QUAN MÔ HÌNH THANG MÁY 4 TẦNG 2](#_Toc204682474)

[1.1.Tổng quan về mô hình thang máy 4 tầng 2](#_Toc204682475)

[1.1.1.Sơ đồ khối hệ thống 3](#_Toc204682476)

[1.1.2.Chức năng của hệ thống 3](#_Toc204682477)

[1.2.Các thiết bị sử dụng trong mô hình 4](#_Toc204682478)

[1.2.1.Khối nguồn 4](#_Toc204682479)

[1.2.2.Khối giám sát 5](#_Toc204682480)

[1.2.3.Khối đầu vào 5](#_Toc204682481)

[1.2.4.Khối điều khiển PLC S7-1200 8](#_Toc204682482)

[1.2.5.Khối cơ cấu chấp hành 9](#_Toc204682483)

[1.2.7.Giới thiệu về màn hình Weintek 11](#_Toc204682484)

[II.THIẾT KẾ BẢN VẼ ĐIỆN 12](#_Toc204682485)

[2.1.Giới thiệu phần mềm Autocad Electrical 2023 12](#_Toc204682486)

[2.2.Bản vẽ sơ đồ đấu nối điện 13](#_Toc204682487)

[III.LẬP TRÌNH PLC VÀ VẬN HÀNH HỆ THỐNG 17](#_Toc204682488)

[3.1.Lưu đồ thuật toán 17](#_Toc204682489)

[3.2.Cấu hình phần cứng trên TIA Portal 17](#_Toc204682490)

[3.2.1.Cấu hình các thông số cần thiết cho module truyền thông CM1241 18](#_Toc204682491)

[3.2.2.Cấu hình MODBUS\_COMM\_LOAD 19](#_Toc204682492)

[3.2.3.Cấu hình Modbus\_Master 20](#_Toc204682493)

[3.2.4.Chuyển đổi tần số hoạt động từ real sang hex 20](#_Toc204682494)

[3.2.5.Thiết lập Speed Setpoint 21](#_Toc204682495)

[3.2.6.Lập trình chức năng điều khiển chạy thuận, chạy ngược và stop 21](#_Toc204682496)

[3.3.Chương trình điều khiển trên TIA Portal 22](#_Toc204682497)

[3.3.1.Function block, Data block 22](#_Toc204682498)

[3.3.2.Lập trình trên hàm Main 22](#_Toc204682499)

[3.4.Thiết kế màn hình HMI 24](#_Toc204682500)

[3.5.Thiết kế giao diện mô phỏng WinCC 24](#_Toc204682501)

[IV.ĐIỀU KHIỂN THANG MÁY THÔNG QUA APP 26](#_Toc204682502)

[4.1.Giới thiệu về phần mềm PyCharm và Android Studio 26](#_Toc204682503)

[4.1.1.PyCharm 26](#_Toc204682504)

[4.1.2.Android Studio 26](#_Toc204682505)

[4.2.Xây dựng chương trình Pythong trung gian kết nối PLC với ứng dụng Android 27](#_Toc204682506)

[4.2.1.Khai báo thư viện sử dụng 27](#_Toc204682507)

[4.2.2.Thiết lập cấu hình kết nối 27](#_Toc204682508)

[4.2.3.Hàm kết nối PLC tự động 28](#_Toc204682509)

[4.2.4.Hàm ghi bit vào PLC (bit trong DB) 28](#_Toc204682510)

[4.2.5.Hàm đọc giá trị WORD từ PLC 29](#_Toc204682511)

[4.2.6.Các API điều khiển thang máy (điều khiển bit) 30](#_Toc204682512)

[4.2.7.Chạy server Flask 30](#_Toc204682513)

[4.3.Ứng dụng Android điều khiển và giám sát thang máy 31](#_Toc204682514)

[4.3.1.Giao tiếp giữa App và hệ thống 31](#_Toc204682515)

[4.3.2.Giao diện và chức năng chính của ứng dụng 31](#_Toc204682516)

[4.3.3.Cập nhật trạng thái tự động định kỳ 32](#_Toc204682517)

[4.3.4.Xử lý phản hồi và lỗi kết nối 33](#_Toc204682518)

[V.KẾT LUẬN 35](#_Toc204682519)

[5.1.Các kết quả đạt được 35](#_Toc204682520)

[5.2.Hạn chế 37](#_Toc204682521)

[5.3.Hướng phát triển 37](#_Toc204682522)

[TÀI LIỆU THAM KHẢO 38](#_Toc204682523)

**MỤC LỤC HÌNH ẢNH**

[Hình 1.1: Mô hình điều khiển thang máy 4 tầng 2](#_Toc204682947)

[Hình 1.2: Sơ đồ khối hệ thống 3](#_Toc204682948)

[Hình 1.3: Module nguồn 1 chiều 4](#_Toc204682949)

[Hình 1.4: Nút nhấn trong thang máy 5](#_Toc204682950)

[Hình 1.5: Nút nhấn ngoài thang máy 6](#_Toc204682951)

[Hình 1.6: Công tác hành trình vị trí cabin trong thang máy 7](#_Toc204682952)

[Hình 1.7: Công tắc hành trình cửa thang máy và cảm biến tiệm cận 7](#_Toc204682953)

[Hình 1.8: PLC S7 1200 8](#_Toc204682954)

[Hình 1.9: Đèn báo trạng thái 9](#_Toc204682955)

[Hình 1.10: Động cơ kéo cửa thang máy 10](#_Toc204682956)

[Hình 1.1: Động cơ kéo buồng thang máy 10](#_Toc204682957)

[Hình 1.12: Màn hình HMI Weintek trên mô hình 11](#_Toc204682958)

[Hình 2.1: Autocad Electrical 2023 12](#_Toc204682959)

[Hình 2.2: Giao diện chính của Autocad Electrical 2023 12](#_Toc204682960)

[Hình 2.3: Sơ đồ đấu nối nguồn đầu vào 13](#_Toc204682961)

[Hình 2.4: Sơ đồ đấu nối PLC S7-1200 thang máy 13](#_Toc204682962)

[Hình 2.5: Sơ đồ đấu nối ngõ vào thang máy 14](#_Toc204682963)

[Hình 2.6: Sơ đồ đấu nối ngõ vào thang máy có nút nhấn cabin và công tắc hành trình cửa 14](#_Toc204682964)

[Hình 2.7: Sơ đồ đấu nối ngõ ra thang máy có đèn, còi báo 15](#_Toc204682965)

[Hình 2.8: Sơ đồ đấu nối ngõ ra thang máy hiển thị tầng và board mạch 15](#_Toc204682966)

[Hình 3.1: Lưu đồ thuật toán 17](#_Toc204682967)

[Hình 3.2: Thiết lập khối PLC và các module mở rộng trên TIA PORTAL 18](#_Toc204682968)

[Hình 3.3: Thiết lập thông số cho module CM 1241 18](#_Toc204682969)

[Hình 3.4: Khối MB\_COMM\_LOAD 19](#_Toc204682970)

[Hình 3.5: Khối chức năng MB\_MASTER 20](#_Toc204682971)

[Hình 3.6: Chuyển đổi tần số từ Real sang Hex 21](#_Toc204682972)

[Hình 3.7: Khối MB\_MASTER điều khiển tốc độ 21](#_Toc204682973)

[Hình 3.8: Khối điều khiển biến tầng chạy thuận, ngược và dừng 22](#_Toc204682974)

[Hình 3.9: Function block& Datablock 22](#_Toc204682975)

[Hình 3.10: Main 1 23](#_Toc204682976)

[Hình 3.11: Main 2 23](#_Toc204682977)

[Hình 3.12: Giao diện HMI 24](#_Toc204682978)

[Hình 3.13: Thiết kế giao diện mô phỏng WinCC 25](#_Toc204682979)

[Hình 4.1: PyCharm 26](#_Toc204682980)

[Hình 4.2: Android Studio phiên bản 2023.1.1 27](#_Toc204682981)

[Hình 4.3: Giao diện thiết kế của App Android 32](#_Toc204682982)

[Hình 5.1: Giao diện HMI 35](#_Toc204682983)

[Hình 5.2: Giao diện WinCC khi chạy mô phỏng 36](#_Toc204682984)

[Hình 5.3: Giao diện App khi điều khiển thang máy 36](#_Toc204682985)

**MỤC LỤC BẢNG BIỂU**

[Bảng 2.1: Chú thích một số ký tự trong bản vẽ 13](#_Toc204683001)

[Bảng 2.2:Bảng hiển thị các tầng tương ứng trên tháng máy 16](#_Toc204683002)

[Bảng 4.1: Các API điều khiển được sử dụng 30](#_Toc204683003)

[Bảng 4.2: Các API giám sát 30](#_Toc204683004)

# MỞ ĐẦU

Sau một thời gian học ngành Kỹ thuật điều khiển và Tự động hóa, em nhận thấy tự động hóa không chỉ giảm sức lao động mà còn tối ưu hóa quy trình sản xuất, nâng cao chất lượng và đảm bảo an toàn. Qua các học phần chuyên ngành, đặc biệt là "Hệ thống điều khiển tự động DCS và SCADA", em hiểu rõ hơn về vai trò của các hệ thống điều khiển và giám sát tập trung trong vận hành công nghiệp hiện đại, nơi DCS và SCADA giữ vai trò then chốt.

Từ những kiến thức đã tích lũy, em nhận thấy tiềm năng phát triển lớn của lĩnh vực điều khiển tự động, nhất là mảng thiết kế và lập trình hệ thống giám sát. Với định hướng rõ ràng, em chọn đề tài “Lập trình điều khiển giám sát hệ thống điều khiển thang máy 4 tầng” nhằm vận dụng kiến thức vào thực tế, nâng cao kỹ năng lập trình và chuẩn bị tốt cho công việc sau này.

# I.TỔNG QUAN MÔ HÌNH THANG MÁY 4 TẦNG

## 1.1.Tổng quan về mô hình thang máy 4 tầng

Trong nhịp sống hiện đại và quá trình đô thị hóa mạnh mẽ, thang máy ngày càng đóng vai trò quan trọng trong các công trình xây dựng như nhà cao tầng, trung tâm thương mại, bệnh viện hay khu công nghiệp. Là phương tiện di chuyển theo phương thẳng đứng, thang máy không chỉ giúp vận chuyển người và hàng hóa giữa các tầng một cách nhanh chóng, tiện lợi mà còn góp phần nâng cao chất lượng cuộc sống. Để mô phỏng hoạt động thực tế của thang máy, mô hình thang máy 4 tầng đã được thiết kế với khả năng di chuyển linh hoạt giữa các tầng. Mô hình này sử dụng bộ điều khiển logic khả trình (PLC) làm trung tâm điều khiển, hỗ trợ lập trình các chức năng như dừng đúng tầng, đóng mở cửa tự động và xử lý tín hiệu gọi tầng từ người dùng. Đây là công cụ học tập hiệu quả, giúp người học tiếp cận và hiểu rõ hơn về ứng dụng của PLC trong lĩnh vực tự động hóa, đồng thời là nền tảng để phát triển các hệ thống thang máy tiên tiến trong tương lai.

A machine with buttons and switches

AI-generated content may be incorrect.

Hình 1.1: Mô hình điều khiển thang máy 4 tầng

### 1.1.1.Sơ đồ khối hệ thống

A diagram of a company

AI-generated content may be incorrect.

Hình 1.2: Sơ đồ khối hệ thống

Khối nguồn: Đảm nhiệm vai trò cung cấp điện năng cho toàn bộ hệ thống vận hành.

Khối giám sát: Là máy tính có chức năng theo dõi và giám sát hoạt động của toàn hệ thống, tiếp nhận các tín hiệu đầu vào và đầu ra.

Khối đầu vào: Bao gồm các thiết bị như nút nhấn, cảm biến dùng để thu nhận tín hiệu điều khiển từ người dùng hoặc môi trường.

Khối điều khiển: Sử dụng PLC S7-1200 để xử lý tín hiệu nhận được từ khối đầu vào và khối giám sát, từ đó đưa ra tín hiệu điều khiển cho các thiết bị đầu ra.

Khối cơ cấu chấp hành: Có nhiệm vụ nhận tín hiệu điều khiển từ PLC S7-1200 và truyền đến các thiết bị thực thi như động cơ kéo tầng, động cơ một chiều (DC), đèn báo, còi… để thực hiện các thao tác theo yêu cầu hệ thống.

### 1.1.2.Chức năng của hệ thống

Thang máy là phương tiện vận chuyển theo phương thẳng đứng, được sử dụng để di chuyển người và hàng hóa giữa các tầng trong các công trình như tòa nhà, tàu thủy hay các cơ sở xây dựng khác. Trong đời sống hiện đại, đặc biệt là tại các tòa nhà cao tầng, trung tâm thương mại, bệnh viện và công trình công cộng, thang máy đã trở thành một thiết bị không thể thiếu.

Mô hình thang máy là phiên bản thu nhỏ và đơn giản hóa của thang máy thực tế, nhưng vẫn đảm bảo mô phỏng đầy đủ các chức năng cơ bản. Việc thiết kế mô hình này nhằm hỗ trợ quá trình học tập và nghiên cứu, giúp người dùng nắm bắt rõ hơn về cấu trúc, nguyên lý hoạt động và các yếu tố kỹ thuật ảnh hưởng đến quá trình vận hành của thang máy. Sau đây là một số chức năng tiêu biểu của mô hình thang máy 4 tầng:

* Nhận biết vị trí của thang máy nhờ 2 công tác hành trình ở mỗi tầng của thang máy.
* Điều khiển di chuyển thang máy lên xuống giữa các tầng.
* Thực hiện đóng mở cửa thang máy an toàn.
* Có thể thay đổi tốc độ di chuyển của thang máy bằng cách điều khiển biến tần một cách linh hoạt
* Tần ưu tiên, sau khi chọn tầng cần di chuyển đầu tiên các tầng sau đó được chọn sẽ được thang đi tới theo thứ tự ưu tiên.

## 1.2.Các thiết bị sử dụng trong mô hình

### 1.2.1.Khối nguồn

A machine with buttons and wires

AI-generated content may be incorrect.

Hình 1.3: Module nguồn 1 chiều

Thông Số Kỹ Thuật Chung

* Đạt chuẩn TCVN 5699-1:2010 và IEC 60335-1 :2010 về chỉ tiêu công suất và dòng điện, dòng rò và độ bền điện
* Hãng sản xuất: Meanwell
* Điện áp đầu vào: 200-240VAC/50Hz
* Điện áp đầu ra: 24V/10A
* Mạch điện tử: Chân I/O của thiết bị được đưa ra connector trung gian dạng Push-in thuận tiện cho đấu nối.
* Khối đế thiết bị:
* Kích thước: 167 x 49 mm (cao x sâu), chiều rộng là bội số của 20 mm, 2 đầu trên dưới cung bo R
* Chất liệu: Nhôm hợp kim đã được xử lý chống xước, chống dính vết vân tay trên bề mặt
* Kiểu gá lắp: Cơ cấu tháo lắp nhanh (quick fix). [1]

### 1.2.2.Khối giám sát

Bộ phận này đảm nhận vai trò giám sát và quản lý toàn bộ hoạt động của hệ thống thông qua giao diện người dùng. Máy tính cá nhân (PC) đóng vai trò trung tâm điều khiển, hỗ trợ người vận hành trong việc theo dõi trạng thái, thu thập dữ liệu và phân tích thông tin vận hành của hệ thống.

Thông qua kết nối với PLC, hệ thống cho phép hiển thị tình trạng hoạt động của thang máy, các cảm biến và cơ cấu chấp hành. Trong trường hợp xảy ra sự cố, ví dụ như cảm biến gặp trục trặc, hệ thống sẽ ngừng hoạt động theo đúng quy trình và đồng thời đưa ra cảnh báo trên màn hình để người dùng kịp thời xử lý.

### 1.2.3.Khối đầu vào

A close up of a control panel

AI-generated content may be incorrect.

Hình 1.4: Nút nhấn trong thang máy

Nút nhấn là một thiết bị điều khiển điện phổ biến trong các hệ thống tự động hóa công nghiệp, cho phép người dùng can thiệp vào quá trình vận hành bằng cách thay đổi trạng thái của hệ thống một cách nhanh chóng và an toàn.

Trong mô hình thang máy, các nút bấm được sử dụng để người dùng thực hiện thao tác gọi tầng từ bên trong cabin, thậm chí có thể lựa chọn nhiều tầng cùng lúc. Bên cạnh đó, hệ thống còn tích hợp các nút chức năng khác như nút đóng/mở cửa thang máy thủ công và nút chuông dùng để phát tín hiệu cảnh báo khi xảy ra sự cố.

A white box with red buttons

AI-generated content may be incorrect.

Hình 1.5: Nút nhấn ngoài thang máy

Nút bấm bên ngoài thang máy được dùng để người sử dụng gọi thang từ bên ngoài cabin. Tùy vào nhu cầu di chuyển, người dùng có thể lựa chọn gọi thang theo hướng đi lên hoặc đi xuống so với tầng hiện tại.

A close up of a machine

AI-generated content may be incorrect.

Hình 1.6: Công tác hành trình vị trí cabin trong thang máy

Có 2 công tác hành trình mỗi tầng thang máy xác định vị trí thang đến tầng nào để sẵn sàng đóng và mở cửa tầng.

A close-up of a machine

AI-generated content may be incorrect.

Hình 1.7: Công tắc hành trình cửa thang máy và cảm biến tiệm cận

Có 2 công tắc hành trình gần vị trí cửa thang máy xác định động cơ kéo cửa ra ngoài hết và đóng vào hết

Ngoài ra còn có cảm biến tiệm cận xác định có người đang ra, vào trong lúc cửa thang đang đóng không.

### 1.2.4.Khối điều khiển PLC S7-1200

A close-up of several electrical equipment

AI-generated content may be incorrect.

Hình 1.8: PLC S7 1200

Thông Số Kỹ Thuật Chung

* Đạt chuẩn TCVN 5699-1:2010 và IEC 60335-1 :2010 về chỉ tiêu công suất và dòng điện, dòng rò và độ bền điện
* Hãng sản xuất: Siemens
* Loại CPU: 1215C DC/DC/DC
* Nguồn cấp: 20.4 - 28.8 VDC
* Số đầu vào số: 14DI
* Số đầu ra số: 10DO
* Kiểu đầu vào ra số: Transistor.
* Số đầu vào tương tự : 2AI, dải điện áp 0-10V
* Số đầu ra tương tự: 2AO, dải dòng điện 0-20mA
* Bộ nhớ: 125 kbyte
* Cổng giao tiếp: Profinet
* Module mở rộng SM 1223
* Module mở rộng Modbus
* Mạch điện tử: Chân I/O của thiết bị được đưa ra connector trung gian dạng Push-in thuận tiện cho đấu nối.
* Kích thước: 167 x 49 mm (cao x sâu), chiều rộng là bội số của 20 mm, 2 đầu trên dưới cung bo R
* Chất liệu: Nhôm hợp kim đã được xử lý chống xước, chống dính vết vân tay trên bề mặt
* Kiểu gá lắp: Cơ cấu tháo lắp nhanh (quick fix) [2]

### 1.2.5.Khối cơ cấu chấp hành

A close up of a control panel

AI-generated content may be incorrect.

Hình 1.9: Đèn báo trạng thái

Đèn báo có chức năng hiển thị các tín hiệu trạng thái của hệ thống tủ điện hoặc bảng điện thông qua việc phát sáng. Nhờ đó, đèn báo góp phần nâng cao mức độ an toàn cho hệ thống, hỗ trợ nhận biết tình trạng pha điện, cảnh báo lỗi và giúp phát hiện, xử lý kịp thời các sự cố trong quá trình vận hành.

Trong mô hình thang máy này, hệ thống bao gồm các đèn báo từ tầng 1 đến tầng 4 nhằm thể hiện các tầng đã được lựa chọn bên trong cabin. Ngoài ra, các nút chức năng "Open" và "Close" được dùng để điều khiển việc đóng mở cửa thang máy một cách trực tiếp.

A machine with gears and chains

AI-generated content may be incorrect.

Hình 1.10: Động cơ kéo cửa thang máy

Động cơ DC 24V: Được sử dụng để thực hiện chức năng kéo đóng và mở cửa thang máy một cách linh hoạt và ổn định.

Công tắc hành trình (NO): Bao gồm công tắc hành trình đóng cửa và mở cửa, thuộc loại thường hở (Normally Open), giúp xác định chính xác trạng thái cửa đang đóng hay mở.

Cảm biến phát hiện vật cản (Autonics): Có nhiệm vụ nhận diện khi có người hoặc vật cản tại khu vực cửa thang máy, từ đó hỗ trợ hệ thống đưa ra phản hồi kịp thời để đảm bảo an toàn cho người sử dụng.

A machine with a metal cylinder

AI-generated content may be incorrect.

Hình 1.1: Động cơ kéo buồng thang máy

Nguồn điện 3 pha 220VAC là một mức điện áp tiêu chuẩn được sử dụng phổ biến tại một số quốc gia, đặc biệt là ở một số khu vực thuộc châu Á hoặc trong các hệ thống điện đặc thù. Mức điện áp này khác biệt so với nguồn 3 pha 380–400VAC – vốn phổ biến tại Việt Nam và nhiều quốc gia khác. Động cơ sử dụng điện áp 3 pha 220VAC có thể được cấp nguồn trực tiếp từ lưới điện tương ứng, hoặc thông qua bộ biến tần nếu cần kiểm soát và thay đổi tốc độ vận hành.

### 1.2.7.Giới thiệu về màn hình Weintek

A machine with buttons and wires

AI-generated content may be incorrect.

Hình 1.12: Màn hình HMI Weintek trên mô hình

Thông số kỹ thuật chung

* Đạt chuẩn TCVN 5699-1:2010 và IEC 60335-1 :2010 về chỉ tiêu công suất và dòng điện, dòng rò và độ bền điện
* Hãng sản xuất: Weintek
* Thiết bị không bao gồm phần mềm.
* Màn hình dòng MT8000
* Loại màn hình: cảm ứng, 7” TFT
* Độ phân giải màn hình: 800x480 pixel
* Cổng (I/ O): USB 2.0 x 1, Ethernet: 10/100 Base-T x 1, RS-232, RS-485
* Mạch điện tử: Chân I/O của thiết bị được đưa ra connector trung gian dạng Push in thuận tiện cho đấu nối. [3]

# II.THIẾT KẾ BẢN VẼ ĐIỆN

## 2.1.Giới thiệu phần mềm Autocad Electrical 2023

Autocad Electrical là phần mềm thiết kế hệ thống điện chuyên nghiệp trong ngành điện công nghiệp. Sở hữu nhiều công cụ hỗ trợ mạnh mẽ cho thiết kế điện, phần mềm giúp các kỹ sư điện thiết kế, mô phỏng và quản lý dự án một cách hiệu quả.

A black rectangular object with white text

AI-generated content may be incorrect.

Hình 2.1: Autocad Electrical 2023

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.2: Giao diện chính của Autocad Electrical 2023

## 2.2.Bản vẽ sơ đồ đấu nối điện

Chú thích một vài kí tự viết tắt trong bảng vẽ:

|  |  |  |
| --- | --- | --- |
| CN | Connecter | Giắc kết nối |
| RL | Relay | Rơ le |
| PS | Power Supply | Nguồn cấp |

Bảng 2.1: Chú thích một số ký tự trong bản vẽ

A diagram of a motor

AI-generated content may be incorrect.

Hình 2.3: Sơ đồ đấu nối nguồn đầu vào

A diagram of a computer

AI-generated content may be incorrect.

Hình 2.4: Sơ đồ đấu nối PLC S7-1200 thang máy

A diagram of a machine

AI-generated content may be incorrect.

Hình 2.5: Sơ đồ đấu nối ngõ vào thang máy

A screen shot of a diagram

AI-generated content may be incorrect.

Hình 2.6: Sơ đồ đấu nối ngõ vào thang máy có nút nhấn cabin và công tắc hành trình cửa

Đối với sơ đồ đấu nối ngõ ra của thang máy bao gồm những phần chính như đèn báo mở/đóng cửa, còi báo dùng trong các trường hợp có sự cố và thông báo khi đến tầng. Ngoài ra còn có đầu ra hiển thị tầng trên led 7 thanh cùng với các board mạch đóng mở cửa thang máy.

A diagram of a circuit board

AI-generated content may be incorrect.

Hình 2.7: Sơ đồ đấu nối ngõ ra thang máy có đèn, còi báo

A diagram of a wiring diagram

AI-generated content may be incorrect.

Hình 2.8: Sơ đồ đấu nối ngõ ra thang máy hiển thị tầng và board mạch

Bảng hiển thị các tầng tương ứng:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q2.4 | Q2.5 | Q0.7 | Led Segment | Tầng |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 | 2 |
| 1 | 1 | 0 | 3 | 3 |
| 0 | 0 | 1 | 4 | 4 |

Bảng 2.2:Bảng hiển thị các tầng tương ứng trên tháng máy

# III.LẬP TRÌNH PLC VÀ VẬN HÀNH HỆ THỐNG

## 3.1.Lưu đồ thuật toán

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 3.1: Lưu đồ thuật toán

Lưu đồ mô tả hệ thống điều khiển thang máy tự động sử dụng cảm biến và bộ xử lý để xử lý yêu cầu gọi tầng, đảm bảo di chuyển cabin chính xác và hiệu quả.

## 3.2.Cấu hình phần cứng trên TIA Portal

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.2: Thiết lập khối PLC và các module mở rộng trên TIA PORTAL

### 3.2.1.Cấu hình các thông số cần thiết cho module truyền thông CM1241

Sau khi khai báo cấu hình phần cứng, ta sẽ bắt đầu thiết lập các thông số truyền thông tại module CM1241:

* Baud rate: 9.6 kbps
* Parity: No parity
* Data bits: 8 bits per character
* Stop bits: 1
* Wait time: 5000ms

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.3: Thiết lập thông số cho module CM 1241

### 3.2.2.Cấu hình MODBUS\_COMM\_LOAD

Khối MB\_COMM\_LOAD: là một khối hàm dùng trong các hệ thống PLC để thiết lập cấu hình truyền thông MODBUS cho các giao thức truyền thông nối tiếp như RS232, RS485 hoặc MODBUS TCP/IP. Khối này chịu trách nhiệm tải các thông số truyền thông cần thiết vào PLC để chuẩn bị cho việc giao tiếp giữa PLC và các thiết bị ngoại vi (như cảm biến, HMI, hoặc các thiết bị điều khiển khác) thông qua giao thức MODBUS.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.4: Khối MB\_COMM\_LOAD

Các tham số chính của khối:

* PORT (ID của cổng truyền thông): đây là tham số xác định cổng truyền thông nào trên PLC sẽ được sử dụng để giao tiếp với MODBUS.
* BAUD (baud rate – tốc độ truyền dữ liệu): 9600kps
* PARITY (kiểm tra chẵn lẻ): xác định phương thức kiểm tra lỗi (0 – None – không kiểm tra, 1 – Even – chẵn, 2 – Odd – lẻ). Thông số này cũng cần đồng bộ với thiết bị Slave để đảm bảo kiểm tra lỗi khi truyền thông
* MB\_DB: tham chiếu đến khối dữ liệu thể hiện của lệnh MB\_MASTER hoặc MB\_SLAVE.
* REQ: lệnh sẽ được thực hiện khi phát hiện tín hiệu sườn lên ở chân này.
* DONE: thực hiện lệnh hoàn tất mà không có lỗi.
* ERROR (lỗi): 0 – không có lỗi; 1 – đã phát hiện lỗi và mã lỗi được đưa ra trong tham số STATUS.
* STATUS: Mã lỗi

### 3.2.3.Cấu hình Modbus\_Master

Các thông số cơ bản trong khối:

* REQ: Bắt đầu thực thi lệnh khi lệnh này “TRUE”
* MB\_ADDR: 1 - Địa chỉ trạm biến tần cần điều khiển đã được cấu hình trước đó trên biến tần
* MODE: Lựa chọn chế độ Read, write, or diagnostics “1” (đang ở chế độ Write)
* DATA\_ADDR: Địa chỉ bắt đầu điều khiển trong Slave “40100”
* DATA\_LEN: Độ dài dữ liệu cần điều khiển
* DATA\_PTR: Dữ liệu điều khiển được thiết lập từ PLC

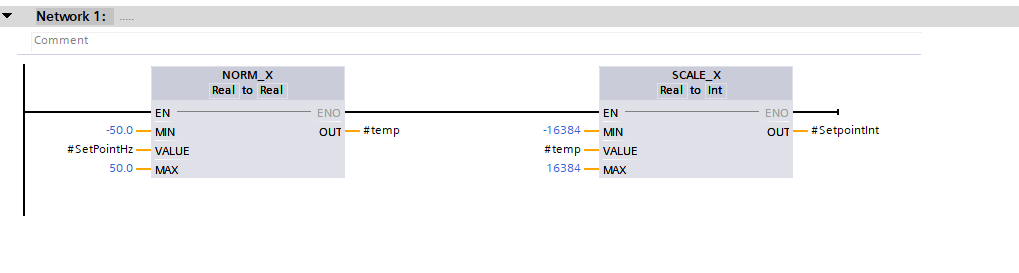
A diagram of a computer program

AI-generated content may be incorrect.

Hình 3.5: Khối chức năng MB\_MASTER

### 3.2.4.Chuyển đổi tần số hoạt động từ real sang hex

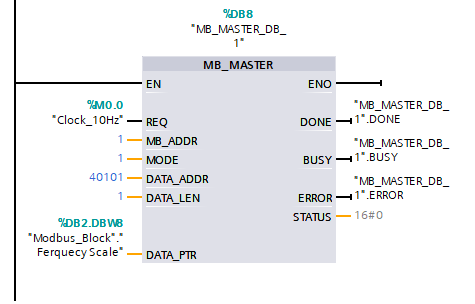
Sử dụng khối chức năng function để lập trình chuyển đổi dữ liệu tần số ( sử dụng 2 lệnh NORM\_X và SCALE\_X). Biến tần có tần số tối đa 50Hz tương ứng với 4000 trong hệ hexa, chuyển sang hệ thập phân là 16384.Mà biến tần điều khiển động cơ cả thuận và nghịch nên giá trị tần số từ -50 đến 50, chuyển sang hệ thập phân là -16384 đến 16384.



Hình 3.6: Chuyển đổi tần số từ Real sang Hex

### 3.2.5.Thiết lập Speed Setpoint

Khối MB\_MASTER thứ 2 này có nhiệm vụ điều khiển tốc độ của biến tần có địa chỉ thanh ghi Modbus là 40101



Hình 3.7: Khối MB\_MASTER điều khiển tốc độ

### 3.2.6.Lập trình chức năng điều khiển chạy thuận, chạy ngược và stop

Khi nhấn chạy hoạt động thì MOVE nó sang giá trị 16#047F, còn khi chạy ngược thì MOVE nó sang giá trị 16#0C7F (3199 ở hệ thập phân) , khi stop thì move sang 16#047E

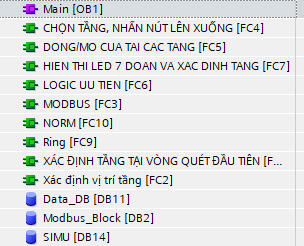
A computer screen shot of a diagram

AI-generated content may be incorrect.

Hình 3.8: Khối điều khiển biến tầng chạy thuận, ngược và dừng

## 3.3.Chương trình điều khiển trên TIA Portal

### 3.3.1.Function block, Data block



Hình 3.9: Function block& Datablock

Các Function block và Data block chính được sử dung trong chương trình

### 3.3.2.Lập trình trên hàm Main

A screenshot of a computer

AI-generated content may be incorrect.

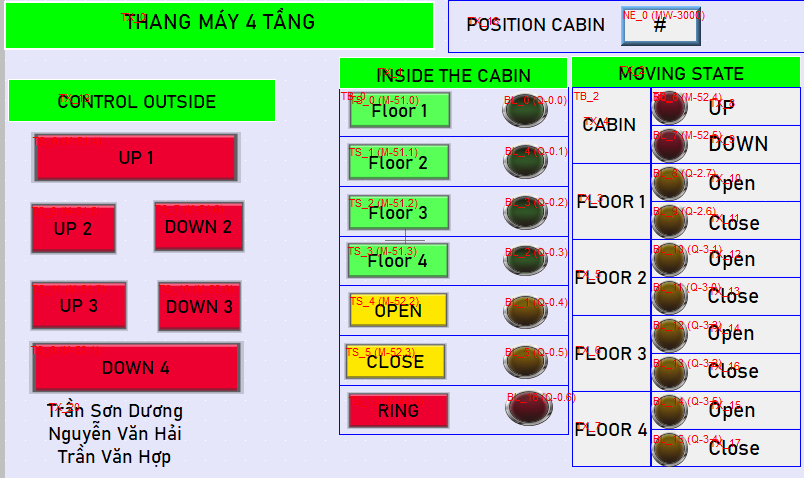
Hình 3.10: Main 1

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.11: Main 2

## 3.4.Thiết kế màn hình HMI

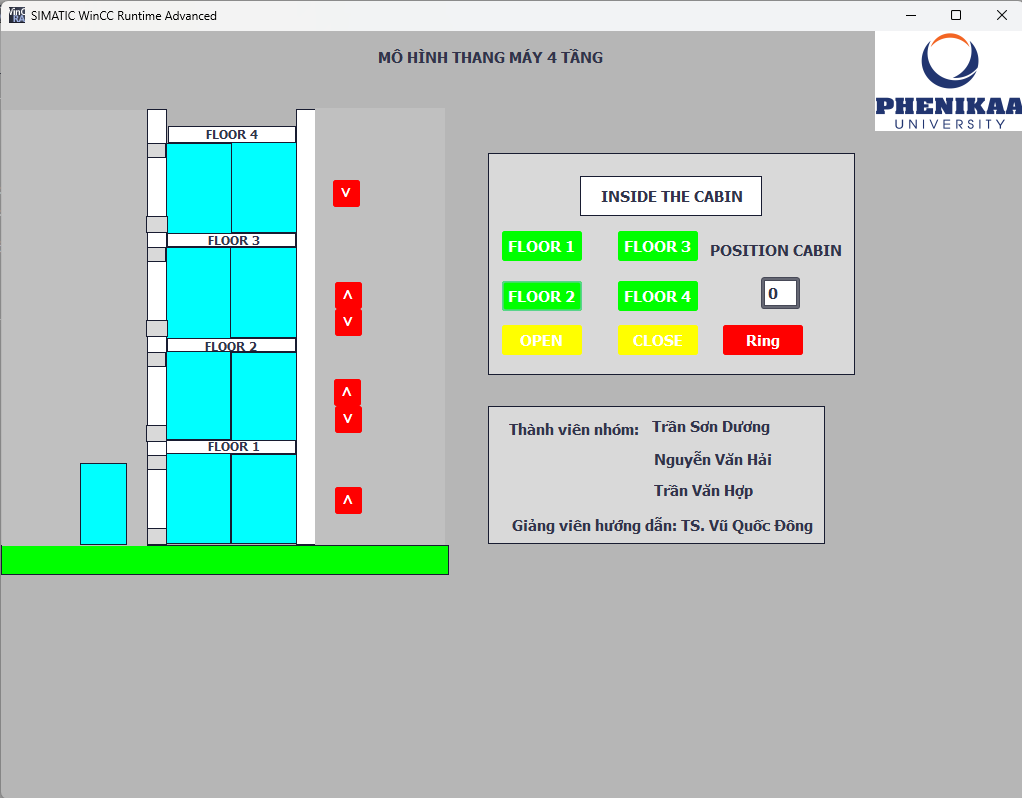


Hình 3.12: Giao diện HMI

Giao diện HMI được thiết kế bao gồm:

* Trạng thái :RUN/STOP hệ thống, cabin di chuyển tới các tầng, trạng thái đóng mở cửa các tầng
* Hiển thị tầng hiện tại của cabin, tầng đang đến(tầng ưu tiên), số tầng đang đợi(số tầng đã nhấn nhưng chưa di chuyển tới)
* Các nút nhấn gồm START/STOP chương trình, gọi các tầng 1,2,3,4, đóng mở cửa các tầng OPEN/CLOSE

## 3.5.Thiết kế giao diện mô phỏng WinCC



Hình 3.13: Thiết kế giao diện mô phỏng WinCC

Giao diện mô phỏng WinCC hiển thị mô phỏng việc di chuyển của thang máy, tác vụ thực hiện nhấn điều khiển trong cabin và ngoài cabin, hiển thị vị trí tầng.

# IV.ĐIỀU KHIỂN THANG MÁY THÔNG QUA APP

## 4.1.Giới thiệu về phần mềm PyCharm và Android Studio

### 4.1.1.PyCharm

PyCharm là một môi trường phát triển tích hợp (IDE) giúp đơn giản hóa quá trình phát triển Python được tạo ra bởi JetBrains. PyCharm cung cấp một loạt công cụ và tính năng được gói gọn trong một giao diện duy nhất.

PyCharm hỗ trợ đầy đủ các giai đoạn phát triển phần mềm, từ viết mã, gỡ lỗi, kiểm thử cho đến triển khai. Đối với lập trình viên Python, phần mềm PyCharm đóng vai trò là một trung tâm điều khiển mạnh mẽ, giúp nâng cao năng suất đáng kể. [4]



Hình 4.1: PyCharm

Tại đề tài này, nhóm sử dụng PyCharm làm IDE để lập trình code Python làm trung gian kết nối giữa PLC với Android App.

### 4.1.2.Android Studio

Android Studio là môi trường tích hợp IDE để viết code và phát triển ứng dụng trên nền tảng hệ điều hành Android, dành cho Google Android Development được ra mắt vào ngày 16 tháng 5 năm 2013, trong sự kiện I/O 2013 của Google.

Android Studio chứa tất cả các công cụ để hỗ trợ việc phát hành ứng dụng Android như: thiết kế, kiểm tra, gỡ lỗi và cấu hình ứng dụng. Android Studio sử dụng Gradle để quản lý dự án của bạn. [5]



Hình 4.2: Android Studio phiên bản 2023.1.1

Android Studio được sử dụng làm IDE để thiết kế app điều khiển hệ thống thang máy.

## 4.2.Xây dựng chương trình Pythong trung gian kết nối PLC với ứng dụng Android

Chương trình Python sử dụng thư viện snap7 để giao tiếp với PLC Siemens (S7-1200 hoặc S7-1500) thông qua giao thức S7, đồng thời dùng Flask để xây dựng Web API điều khiển và giám sát hệ thống thang máy. Android App có thể gọi API qua HTTP để điều khiển thang máy hoặc lấy trạng thái hiện tại.

### 4.2.1.Khai báo thư viện sử dụng

from flask import Flask, jsonify

import snap7

from snap7 import Area

from snap7.util import set\_bool, get\_bool, get\_int

import socket

* flask: Tạo Web server để cung cấp các REST API.
* snap7: Giao tiếp với PLC Siemens.
* set\_bool, get\_int: Hỗ trợ thao tác bit hoặc word.
* socket: In ra địa chỉ IP LAN của máy tính.

### 4.2.2.Thiết lập cấu hình kết nối

Cấu hình thông số PLC Siemens như địa chỉ IP, rack, slot và số DB (Data Block) đang sử dụng, khai báo client = snap7.client.Client() tạo đối tượng để kết nối tới PLC.

# Cấu hình PLC

PLC\_IP = "192.168.0.5"

RACK = 0

SLOT = 1

DB\_NUMBER = 11

# Tạo đối tượng PLC

client = snap7.client.Client()

### 4.2.3.Hàm kết nối PLC tự động

Hàm này sẽ kiểm tra nếu PLC đã ngắt kết nối thì tự động kết nối lại. Được gọi trước mọi thao tác đọc/ghi xuống PLC.

def ensure\_connected():

    """Đảm bảo kết nối tới PLC, kết nối lại nếu mất."""

    if not client.get\_connected():

        try:

            client.connect(PLC\_IP, RACK, SLOT)

            print(f"✅ Đã kết nối lại tới PLC tại {PLC\_IP}")

            return True

        except Exception as e:

            print(f"❌ Lỗi kết nối lại PLC: {e}")

            return False

    return True

### 4.2.4.Hàm ghi bit vào PLC (bit trong DB)

Hàm giúp hỗ trợ ghi vào Data Block (Area.DB), đọc trước 1 byte từ đó sửa bit cần thiết sau đó ghi lại toàn byte. Dùng set\_bool() để sửa đúng vị trí bit trong byte.

def write\_plc\_bit(area\_type, byte\_offset, bit\_offset, value, db\_num=None):

    """

    Ghi một bit cụ thể trong Data Block hoặc Memory Area của PLC.

    Đọc byte hiện tại, thay đổi bit, sau đó ghi lại byte.

    :param area\_type: Loại vùng nhớ (Area.DB, Area.MK, v.v.)

    :param byte\_offset: Byte offset trong vùng nhớ.

    :param bit\_offset: Bit offset trong byte.

    :param value: Giá trị boolean (True/False) để ghi.

    :param db\_num: Số Data Block nếu area\_type là Area.DB. Bỏ qua nếu là Area.MK.

    """

    if not ensure\_connected():

        raise Exception("PLC chưa kết nối.")

    try:

        # Xác định kích thước đọc (1 byte)

        read\_size = 1

        # Đọc 1 byte từ địa chỉ đã cho

        if area\_type == Area.DB:

            if db\_num is None:

                raise ValueError("db\_num phải được cung cấp cho Area.DB")

            data = client.read\_area(area\_type, db\_num, byte\_offset, read\_size)

        else: # Đối với Area.MK (Memory Area)

            data = client.read\_area(area\_type, 0, byte\_offset, read\_size) # Offset 0 cho M-Area, byte\_offset là địa chỉ thực

        byte\_array = bytearray(data)  # Chuyển đổi sang bytearray để có thể sửa đổi

        # Đặt giá trị bit

        set\_bool(byte\_array, 0, bit\_offset, value) # byte\_array, byte\_index\_in\_array, bit\_index\_in\_byte, value

        # Ghi lại byte đã sửa đổi vào PLC

        if area\_type == Area.DB:

            client.write\_area(area\_type, db\_num, byte\_offset, byte\_array)

            print(f"Đã ghi DB{db\_num}.DBX{byte\_offset}.{bit\_offset} = {value}")

        else: # Đối với Area.MK

            client.write\_area(area\_type, 0, byte\_offset, byte\_array) # Offset 0 cho M-Area

            print(f"Đã ghi M{byte\_offset}.{bit\_offset} = {value}")

    except Exception as e:

        area\_str = f"DB{db\_num}.DBX" if area\_type == Area.DB else "M"

        print(f"Lỗi khi ghi {area\_str}{byte\_offset}.{bit\_offset}: {e}")

        raise

### 4.2.5.Hàm đọc giá trị WORD từ PLC

Đọc 2 byte từ PLC rồi dùng get\_int() để lấy giá trị INT.

def read\_plc\_int\_value(int\_offset, db\_num=DB\_NUMBER):

    """

    Đọc giá trị của một INT (WORD) cụ thể từ Data Block của PLC.

    INT chiếm 2 bytes.

    """

    if not ensure\_connected():

        raise Exception("PLC chưa kết nối.")

    try:

        # Đọc 2 bytes từ địa chỉ đã cho (vì INT là 2 bytes)

        data = client.read\_area(Area.DB, db\_num, int\_offset, 2)

        read\_value = get\_int(data, 0)  # Lấy giá trị số nguyên từ 2 bytes

        print(f"✅ Đã đọc DB{db\_num}.DBW{int\_offset}: Giá trị = {read\_value}")

        return read\_value

    except Exception as e:

        print(f"❌ Lỗi khi đọc DB{db\_num}.DBW{int\_offset}: {e}")

        raise

### 4.2.6.Các API điều khiển thang máy (điều khiển bit)

Các API cung cấp khả năng điều khiển và giám sát thang máy từ xa, gồm:

#### 4.2.6.1.API điều khiển

|  |  |
| --- | --- |
| **API** | **Chức năng** |
| /elevator/cabin/call/<floor> | Gọi tầng từ cabin |
| /elevator/door/open | Mở cửa thang máy |
| /elevator/door/close | Đóng cửa thang máy |
| /elevator/floor/<floor>/up | Gọi lên từ tầng |
| /elevator/floor/<floor>/down | Gọi xuống từ tầng |

Bảng 4.1: Các API điều khiển được sử dụng

Các API đều ghi trực tiếp vào PLC thôgn qua hàm write\_plc\_bit()

#### 4.2.6.2.API giám sát

|  |  |
| --- | --- |
| **API** | **Chức năng** |
| /elevator/position | Trả về vị trí hiện tại của cabin |
| /status | Trả về trạng thái kết nối và vị trí của cabin |

Bảng 4.2: Các API giám sát

### 4.2.7.Chạy server Flask

Server chạy trên địa chỉ IP của máy tính (sử dụng mạng do laptop phát WiFi), lắng nghe ở port 5000. API được gọi từ ứng dụng Android trong cùng mạng LAN.

if \_\_name\_\_ == "\_\_main\_\_":

    try:

        client.connect(PLC\_IP, RACK, SLOT)

        print(f"✅ Đã kết nối tới PLC tại {PLC\_IP}")

    except Exception as e:

        print(f"❌ Lỗi kết nối ban đầu tới PLC: {e}")

    app.run(host="192.168.137.1", port=5000, debug=True)

## 4.3.Ứng dụng Android điều khiển và giám sát thang máy

Lớp giao diện người dùng (Android) cho phép người dùng giám sát vị trí cabin và điều khiển thang máy từ xa qua mạng LAN.

Ứng dụng Android đóng vai trò như một giao diện điều khiển và giám sát từ xa, cho phép người dùng có thể gửi yêu cầu đến hệ thống (chọn tầng, đóng/mở cửa cabin) cũng như quan sát trạng thái hiện tại của thang máy. Điều này giúp tăng tính linh hoạt và tính thân thiện với người sử dụng cuối.

Ứng dụng Android được thiết kế để giao tiếp với một server trung gian viết bằng Python (Flask), đóng vai trò cầu nối giữa thiết bị di động và PLC điều khiển. Việc tách biệt tầng giao tiếp giúp tăng độ bảo trì, khả năng mở rộng và giảm phụ thuộc nền tảng.

### 4.3.1.Giao tiếp giữa App và hệ thống

Ứng dụng Android sử dụng thư viện OkHttp để thực hiện các yêu cầu HTTP GET đến server Flask trung gian. Server Flask nhận yêu cầu từ ứng dụng và chuyển tiếp điều khiển hoặc truy vấn trạng thái đến PLC thông qua giao thức Modbus RTU.

Ứng dụng Android ⇄ HTTP GET ⇄ Server Python Flask ⇄ Modbus RTU ⇄ PLC

### 4.3.2.Giao diện và chức năng chính của ứng dụng

Giao diện chính bao gồm các nhóm nút chức năng và vùng hiển thị trạng thái:

* Cụm nút điều khiển trong cabin: Gồm các nút chọn tầng (tầng 1 đến tầng tầng 4), nút mở cửa và đóng cửa.
* Cụm nút gọi tầng ngoài cabin: Gồm các nút gọi lên/xuống tại các tầng 1 đến tầng 4.
* Vùng hiển thị vị trí cabin: Cập nhật liên tục vị trí hiện tại của cabin (theo thời gian thực mỗi 1 giây).

A screenshot of a phone

AI-generated content may be incorrect.

Hình 4.3: Giao diện thiết kế của App Android

### 4.3.3.Cập nhật trạng thái tự động định kỳ

Ứng dụng sử dụng Handler kết hợp với Runnable để thực hiện việc gửi yêu cầu cập nhật trạng thái cabin theo chu kỳ 1 giây. Endpoint /status sẽ trả về dữ liệu JSON chứa thông tin về vị trí cabin, từ đó ứng dụng cập nhật lên TextView tương ứng. Việc này đảm bảo khả năng phản hồi gần thời gian thực và tránh tình trạng người dùng phải thao tác thủ công để lấy trạng thái.

Khai báo Handler, Runnable và chu kỳ cập nhật

private Handler handler = new Handler(Looper.*getMainLooper*());  
private Runnable updatePositionRunnable;  
private static final long *UPDATE\_INTERVAL\_MS* = 1000;

Khởi tạo Runnable trong hàm onCreate():

// Khởi tạo Runnable để cập nhật vị trí  
updatePositionRunnable = new Runnable() {  
 @Override  
 public void run() {  
 sendRequest("/status"); // Gửi yêu cầu lấy vị trí  
 handler.postDelayed(this, *UPDATE\_INTERVAL\_MS*); // Lên lịch chạy lại sau UPDATE\_INTERVAL\_MS  
 }  
};

Cơ chế được mô tả như sau:

* Khi Activity được hiển thị (onResume()), tiến trình cập nhật bắt đầu chạy.

@Override  
protected void onResume() {  
 super.onResume();  
 // Bắt đầu cập nhật vị trí khi Activity hiển thị  
 handler.post(updatePositionRunnable); // Chạy lần đầu ngay lập tức (hoặc sau một độ trễ nhỏ nếu muốn)  
 Log.*d*(*TAG*, "Bắt đầu cập nhật vị trí định kỳ.");  
}

* Khi Activity bị ẩn (onPause()), tiến trình cập nhật bị dừng để tiết kiệm tài nguyên.

@Override  
protected void onPause() {  
 super.onPause();  
 // Dừng cập nhật vị trí khi Activity bị ẩn  
 handler.removeCallbacks(updatePositionRunnable);  
 Log.*d*(*TAG*, "Dừng cập nhật vị trí định kỳ.");  
}

### 4.3.4.Xử lý phản hồi và lỗi kết nối

Các phản hồi từ server được phân tích thông qua JSONObject. Trường "display\_text" là chuỗi mô tả vị trí cabin hiện tại, được hiển thị lên TextView. Trong trường hợp phản hồi không hợp lệ hoặc xảy ra lỗi (ví dụ mất kết nối), ứng dụng sẽ hiển thị thông báo lỗi bằng Toast, đồng thời ghi log để hỗ trợ gỡ lỗi.

private void sendRequest(String endpoint) {  
 String fullUrl = SERVER\_URL + endpoint;  
 Log.*d*(*TAG*, "Đang gửi yêu cầu đến: " + fullUrl);  
  
 Request request = new Request.Builder()  
 .url(fullUrl)  
 .build();  
  
 client.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onResponse(Call call, Response response) throws IOException {  
 String res = response.body() != null ? response.body().string() : "Không có phản hồi";  
 Log.*d*(*TAG*, "Nhận phản hồi từ server: " + res);  
  
 runOnUiThread(() -> {  
 // Cập nhật TextView hiển thị vị trí cabin nếu endpoint là /status  
 if (endpoint.equals("/status") || endpoint.equals("/elevator/position")) {  
 try {  
 JSONObject jsonResponse = new JSONObject(res);  
 if (jsonResponse.has("display\_text")) {  
 String displayText = jsonResponse.getString("display\_text");  
 tvCabinPositionDisplay.setText("Cabin: " + displayText);  
 } else {  
 tvCabinPositionDisplay.setText("Cabin: Phản hồi không có 'display\_text'");  
 }  
 } catch (JSONException e) {  
 Log.*e*(*TAG*, "Lỗi phân tích JSON vị trí cabin: " + e.getMessage());  
 tvCabinPositionDisplay.setText("Cabin: Lỗi JSON");  
 } catch (Exception e) {  
 Log.*e*(*TAG*, "Lỗi xử lý phản hồi vị trí cabin: " + e.getMessage());  
 tvCabinPositionDisplay.setText("Cabin: Lỗi dữ liệu");  
 }  
 } else {  
 //Toast.makeText(MainActivity.this, res, Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }  
  
 @Override  
 public void onFailure(Call call, IOException e) {  
 e.printStackTrace();  
 Log.*e*(*TAG*, "Lỗi kết nối server: " + e.getMessage());  
  
 runOnUiThread(() -> {  
 String errorMessage = "❌ Lỗi kết nối: " + (e.getMessage() != null ? e.getMessage() : "Không xác định");  
 Toast.*makeText*(MainActivity.this, errorMessage, Toast.*LENGTH\_LONG*).show();  
 });  
 }  
 });  
}

# V.KẾT LUẬN

## 5.1.Các kết quả đạt được

Tìm hiểu và học tập kiến thức về lập trình và điều khiển hệ thống DCS và SCADA trong lĩnh vực tự động hóa công nghiệp

Lập trình mô hình hoạt động được đúng với yêu cầu của bài tập lớn, gọi tầng thang máy, đóng mở cửa đúng tru trình, ngoài ra còn thực hiện được gọi tầng ưu tiên.

Thiết kế lập trình điều khiển được tốt trên giao diện HMI.

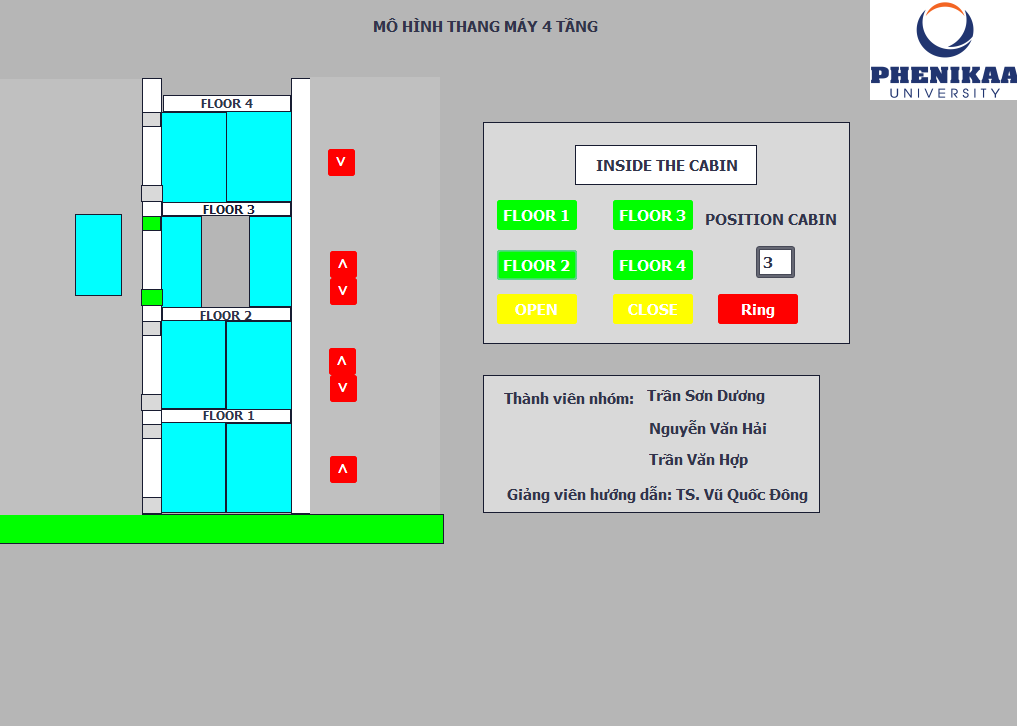
A screenshot of a computer

AI-generated content may be incorrect.

Hình 5.1: Giao diện HMI

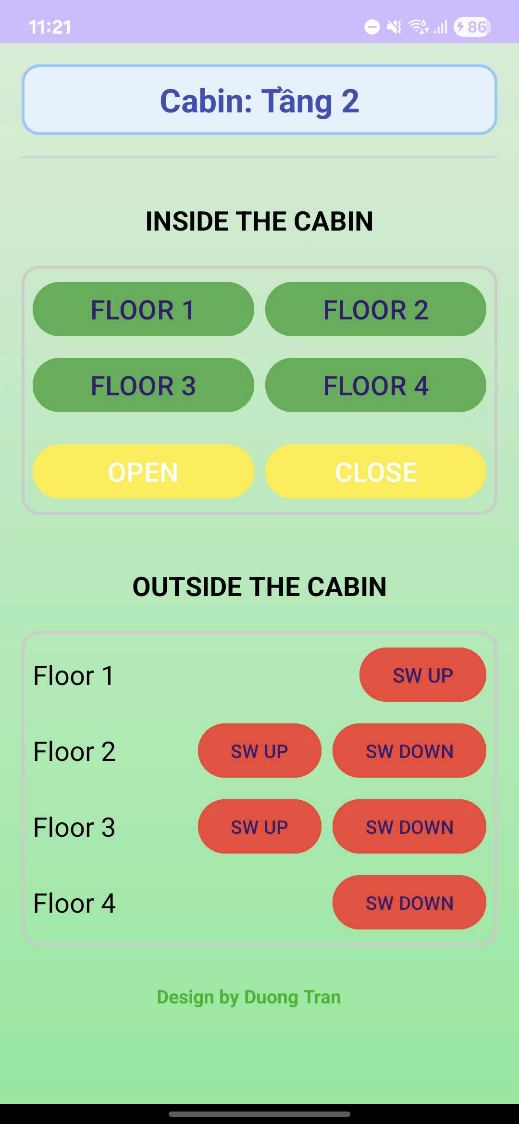
Hiểu và thực hành tốt phần mềm mô phỏng mạch điện AutoCad Electrical.

Thiết kế được giao diện mô phỏng trên WinCC



Hình 5.2: Giao diện WinCC khi chạy mô phỏng

Thiết kế App điều khiển hệ thống thang máy



Hình 5.3: Giao diện App khi điều khiển thang máy

## 5.2.Hạn chế

Cần nâng cao kỹ năng tối ưu hóa chương trình điều khiển PLC.

Cần nâng cao khả năng thiết kế giao diện HMI theo hướng trực quan, dễ sử dụng và thân thiện hơn với người vận hành trong môi trường công nghiệp.

Chưa khai thác hết các tính năng nâng cao của phần mềm mô phỏng.

## 5.3.Hướng phát triển

Tích hợp thêm AI và IoT vào hệ thống điều khiển.

Nghiên cứu các giao thức truyền thông công nghiệp để kết nối PLC với các thiết bị khác.

Thực hành trên nhiều loại PLC, HMI khác nhau để mở rộng kỹ năng.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Hoàng Quốc, “MÔ HÌNH ĐIỀU KHIỂN THANG MÁY 4 TẦNG,” 2025, pp. 8,9. |
| [2] | Hoàng Quốc, “MÔ HÌNH ĐIỀU KHIỂN THANG MÁY 4 TẦNG,” 2025, p. 6. |
| [3] | Hoàng Quốc, “MÔ HÌNH ĐIỀU KHIỂN THANG MÁY 4 TẦNG,” 2025, p. 7. |
| [4] | Mỹ Y, “PyCharm là gì? A-Z về phần mềm IDE số 1 cho lập trình Python,” [Trực tuyến]. Available: https://interdata.vn/blog/pycharm-la-gi/ . [Đã truy cập 29 07 2025]. |
| [5] | Lê Chi, “Android Studio là gì? Hướng dẫn tải và cài đặt Android Studio,” 2023. [Trực tuyến]. Available: https://fptshop.com.vn/tin-tuc/thu-thuat/android-studio-la-gi-150412. [Đã truy cập 29 07 2025]. |