

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT

BÁO CÁO TỔNG KẾT
ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ CẤP TRƯỜNG

NGHIÊN CỨU CHẾ TẠO MÔ HÌNH ĐIỀU KHIỂN VỊ
TRÍ BÀN TRƯỢT
Mã số: T2023-06-16

Chủ nhiệm đề tài: ThS. Nguyễn Văn Nam
Đơn vị: Khoa Điện – Điện tử
Chương trình đào tạo: Kỹ thuật điều khiển và tự động hóa

Đà Nẵng, tháng 3 năm 2025

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT

BÁO CÁO TỔNG KẾT
ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ CẤP TRƯỜNG

NGHIÊN CỨU CHẾ TẠO MÔ HÌNH ĐIỀU KHIỂN VỊ
TRÍ BÀN TRƯỢT
Mã số: T2023-06-16

Xác nhận của cơ quan chủ trì đề tài
KT. HIỆU TRƯỞNG
PHÓ HIỆU TRƯỞNG

Chủ nhiệm đề tài

PGS.TS. Võ Trung Hùng

ThS. Nguyễn Văn Nam

DANH SÁCH THÀNH VIÊN THAM GIA NGHIÊN CỨU ĐỀ TÀI

STT	Họ và tên	Đơn vị
1	TS. Phạm Duy Dưởng	Khoa Điện –Điện tử Trường Đại học Sư phạm Kỹ thuật
2	ThS. Phan Thị Thanh Vân	Khoa Điện –Điện tử Trường Đại học Sư phạm Kỹ thuật

MỤC LỤC

MỤC LỤC	i
DANH MỤC BẢNG	iv
DANH MỤC HÌNH	v
INFORMATION ON RESEARCH RESULTS	x
MỞ ĐẦU	1
1. Tính cấp thiết của đề tài	1
2. Đối tượng, phạm vi nghiên cứu	2
2.1. Đối tượng nghiên cứu	2
2.2. Phạm vi nghiên cứu	2
3. Cách tiếp cận, phương pháp nghiên cứu	2
3.1. Cách tiếp cận.....	2
3.2. Phương pháp nghiên cứu	2
4. Nội dung nghiên cứu	3
CHƯƠNG 1: TỔNG QUAN	4
1.1. Tổng quan về mô hình.....	4
1.2. Tổng quan về động cơ servo	5
1.2.1. Định nghĩa về động cơ servo	5
1.2.2. Cấu tạo của động cơ servo	5
1.2.3. Nguyên lý làm việc của động cơ servo.....	6
1.2.4. Phân loại động cơ servo.....	6
1.2.5. Ứng dụng của động cơ servo trong thực tế.....	8
1.2.6. Ưu điểm và nhược điểm của động cơ servo	8
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	10
2.1. Cơ sở lý thuyết về PLC S7-1200	10
2.1.1. Giới thiệu về PLC S7-1200	10
2.1.2. Cấu trúc phần cứng	10
2.1.3. Cấu trúc bộ nhớ.....	14
2.2. Cơ sở lý thuyết về lập trình PLC S7-1200	22
2.2.1. Phương pháp lập trình.....	22

2.2.2. Phần mềm lập trình	22
CHƯƠNG 3: THIẾT KẾ VÀ CHẾ TẠO MÔ HÌNH	26
3.1. Thiết kế hệ thống chính của mô hình	26
3.2. Thiết kế bản đầu nối và lắp đặt bàn trượt.....	29
3.3. Thiết kế sơ đồ nguyên lý nối dây thiết bị mô hình.....	29
3.4. Hoàn thiện lắp ráp mô hình.....	30
CHƯƠNG 4: XÂY DỰNG CÁC BÀI TẬP THỰC HÀNH.....	31
4.1. Bài thực hành 1	31
4.1.1. Nội dung bài thực hành.....	31
4.1.2. Mục tiêu kiến thức đạt được	31
4.1.3. Hướng dẫn thực hiện	31
4.2. Bài thực hành 2	45
4.2.1. Nội dung bài thực hành.....	45
4.2.2. Mục tiêu kiến thức đạt được	45
4.2.3. Hướng dẫn thực hiện	46
4.3. Bài thực hành 3	48
4.3.1. Nội dung bài thực hành.....	48
4.3.2. Mục tiêu kiến thức đạt được	48
4.3.3. Hướng dẫn thực hiện	48
CHƯƠNG 5: KIỂM TRA THỰC NGHIỆM VÀ ĐÁNH GIÁ MÔ HÌNH	51
5.1. Kiểm tra thực nghiệm mô hình	51
5.1.1. Bài thực hành 1	51
5.1.2. Bài thực hành 2	55
5.1.3. Bài thực hành 3	56
PHẦN KẾT LUẬN VÀ KIẾN NGHỊ.....	57
1. Kết luận	57
2. Kiến nghị	57
TÀI LIỆU THAM KHẢO	58
THUYẾT MINH ĐỀ TÀI KHOA HỌC CÔNG NGHỆ VÀ CÔNG NGHỆ CẤP	
TRƯỜNG	
HỢP ĐỒNG TRIỂN KHAI ĐỀ TÀI	

BẢNG MỤC LỤC MINH CHỨNG SẢN PHẨM CỦA ĐỀ TÀI
BỘ MINH CHỨNG SẢN PHẨM CỦA ĐỀ TÀI

DANH MỤC BẢNG

Bảng 2.1: Bảng thông số kỹ thuật của dòng PLC S7-1200.....	11
Bảng 2.2: Một số module tín hiệu của PLC S7-1200.....	12
Bảng 2.3: Module bảng tín hiệu của PLC S7-1200.....	13
Bảng 2.4: Bảng các kiểu dữ liệu trong PLC S7-1200	18
Bảng 2.5: Bảng chuyển đổi định dạng số BCD.....	18
Bảng 2.6: Ví dụ định dạng kiểu dữ liệu chuỗi.....	20
Bảng 2.7: Ví dụ các mảng dữ liệu	20
Bảng 2.8: Kiểu dữ liệu định dạng DTL.....	21
Bảng 2.9: Kiểu dữ liệu trong DTL	21
Bảng 3.1: Thông số kỹ thuật của bộ trượt điện	26
Bảng 3.2: Các thông số chính của động cơ servo.....	27
Bảng 3.3: Bảng thông số kỹ thuật driver servo	27
Bảng 3.4: Thông số kỹ thuật của bộ mã hóa vòng quay	28
Bảng 3.5: Bảng thông số kỹ thuật bộ nguồn DC 24V	28

DANH MỤC HÌNH

Hình 1.1: Sơ đồ nguyên lý của hệ thống điều khiển vị trí.....	4
Hình 1.2: Mô hình phần cứng của hệ thống điều khiển vị trí.....	4
Hình 1.3: Cấu tạo của động cơ servo.....	5
Hình 1.4: Sơ đồ nguyên lý làm việc của động cơ servo	6
Hình 1.5: Phân loại động cơ servo	7
Hình 1.6: Cấu tạo động cơ DC Servo	7
Hình 1.7: Cấu tạo động cơ AC Servo	8
Hình 2.1: Hình dáng bên ngoài của PLC S7-1200	11
Hình 2.2: Module tín hiệu (SM) của PLC S7-1200	12
Hình 2.3: Module tín hiệu (SB) của PLC S7-1200	13
Hình 2.4: Module truyền thông (CM) của PLC S7-1200	14
Hình 2.5: Vô hiệu hóa vùng nhớ chương trình.....	14
Hình 2.6: Cấu hình Retain Memory để PLC lưu trữ giá trị tag khi mất điện.....	15
Hình 2.7: Thao tác cấu hình thẻ nhớ cho PLC S7-1200.....	16
Hình 2.8: Kiểm tra hiện trạng các vùng nhớ của PLC S7-1200.....	17
Hình 2.9: Các nhóm lệnh cơ bản (Basic instructions).....	24
Hình 2.10: Các nhóm lệnh mở rộng (Extended instructions).....	25
Hình 2.11: Các nhóm lệnh ứng dụng (Technology).....	25
Hình 2.12: Các nhóm lệnh xử lý truyền thông	25
Hình 3.1: Hệ thống các khâu chính của mô hình	26
Hình 3.2: Kích thước tổng thể mô hình.....	29
Hình 3.3: Sơ đồ nguyên lý nối dây của hệ thống mô hình	29
Hình 3.4: Mô hình hoàn thiện.....	30
Hình 4.1: Giao diện phần mềm khi được khởi động	32
Hình 4.2: Các bước tạo một bộ lập trình S7-1200.....	32
Hình 4.3: Khởi tạo bộ điều khiển vị trí.....	33
Hình 4.4: Cài đặt cấu hình cho bộ điều khiển	34
Hình 4.5: Cấu hình tại mục Drive của bộ điều khiển	34

Hình 4.6: Cấu hình tại mục Mechanics của bộ điều khiển.....	35
Hình 4.7: Cấu hình tại mục Position limits	35
Hình 4.8: Cấu hình tại mục Dynamics- General	36
Hình 4.9: Cấu hình tại mục Dynamics- Emergency stop	36
Hình 4.10: Cấu hình tại mục Homing	37
Hình 4.11: Khởi tạo khối MC_Power	38
Hình 4.12: Khởi tạo bộ đếm tốc độ cao (HSC1)	40
Hình 4.13: Khởi tạo bộ SCADA	42
Hình 4.14: Gắn cổng card IE	43
Hình 4.15: Kết nối bộ lập trình PLC S7-1200 với PC-System	43
Hình 4.16: Giao diện điều khiển và giám sát	44
Hình 4.17: Các bước thực hiện liên kết tag	44
Hình 4.18: Thực hiện liên kết tag nút KHỎI ĐỘNG hoàn thiện	45
Hình 4.19: Giao diện điều khiển và giám sát Bài thực hành 2	48
Hình 4.20: Giao diện điều khiển và giám sát Bài thực hành 3	50
Hình 5.1: Kết nối các thiết bị mô hình và nguồn cấp	51
Hình 5.2: Thực hiện tải chương trình xuống bộ lập trình.....	51
Hình 5.3: Hoàn thiện quá trình tải chương trình	52
Hình 5.4: Chạy RT màn hình điều khiển và giám sát mô hình	52
Hình 5.5: Giao diện màn hình chạy thực nghiệm.....	53
Hình 5.6: Bàn trượt tại vị trí 130 trên mô hình.....	53
Hình 5.7: Kết quả thực nghiệm di chuyển bàn trượt chạy tới	53
Hình 5.8: Kết quả thực nghiệm di chuyển bàn trượt chạy lùi	54
Hình 5.9: Thực hiện lấy vị trí góc bàn trượt.....	54
Hình 5.10: Kết quả di chuyển bàn trượt chạy tới và lùi từ vị trí gốc	54
Hình 5.11: Xác nhận vị trí gốc mô hình	55
Hình 5.12: Kết quả thực nghiệm di chuyển bàn trượt đến vị trí 1	55
Hình 5.13: Kết quả thực nghiệm di chuyển bàn trượt đến vị trí 2	55
Hình 5.14: Kết quả thực nghiệm di chuyển bàn trượt đến vị trí 3	56
Hình 5.15: Đưa mô hình về vị trí gốc.....	56
Hình 5.16: Thực nghiệm di chuyển bàn trượt đến một vị trí bất kỳ (70 mm).....	56

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: Nghiên cứu chế tạo mô hình điều khiển vị trí bàn trượt
- Mã số: T2023-06-16
- Chủ nhiệm: ThS. Nguyễn Văn Nam
- Thành viên tham gia: TS. Phạm Duy Dường
ThS. Phan Thị Thanh Vân
- Cơ quan chủ trì: Trường Đại học Sư Phạm Kỹ thuật – Đại học Đà Nẵng
- Thời gian thực hiện: Tháng 3/2024 đến tháng 2/2025

2. Mục tiêu:

- Tạo ra một thiết bị giảng dạy thực hành cho sinh viên ngành Tự động hóa với kiến thức về điều khiển nâng cao, đáp ứng đào tạo theo chương trình tiên tiến.
- Ứng dụng các giải pháp về điều khiển vị trí nhằm cung cấp kiến thức thực tế, trực quan giúp sinh viên tiếp thu nhanh kiến thức và vận dụng trong thực tế.
- Cung cấp một thiết bị ứng dụng cho đào tạo thực hành học phần Thực hành điều khiển logic tại Khoa Điện – Điện tử trường Đại học Sư phạm Kỹ thuật.
- Xây dựng 3 bài thực hành phục vụ cho giảng dạy kiến thức điều khiển nâng cao cho các học phần Thực hành điều khiển logic gồm:
 1. Đấu dây kết nối encoder với PLC S7-1200, đọc xung từ encoder và tính toán ra tốc độ vòng quay.
 2. Đấu dây kết nối PLC S7-1200 với động cơ servo, xuất xung tốc độ cao điều khiển động cơ servo
 3. Đấu dây kết nối PLC S7-1200 với động cơ servo, lập trình điều khiển vị trí bàn

trượt di chuyển trong khoảng 0 đến 200mm.

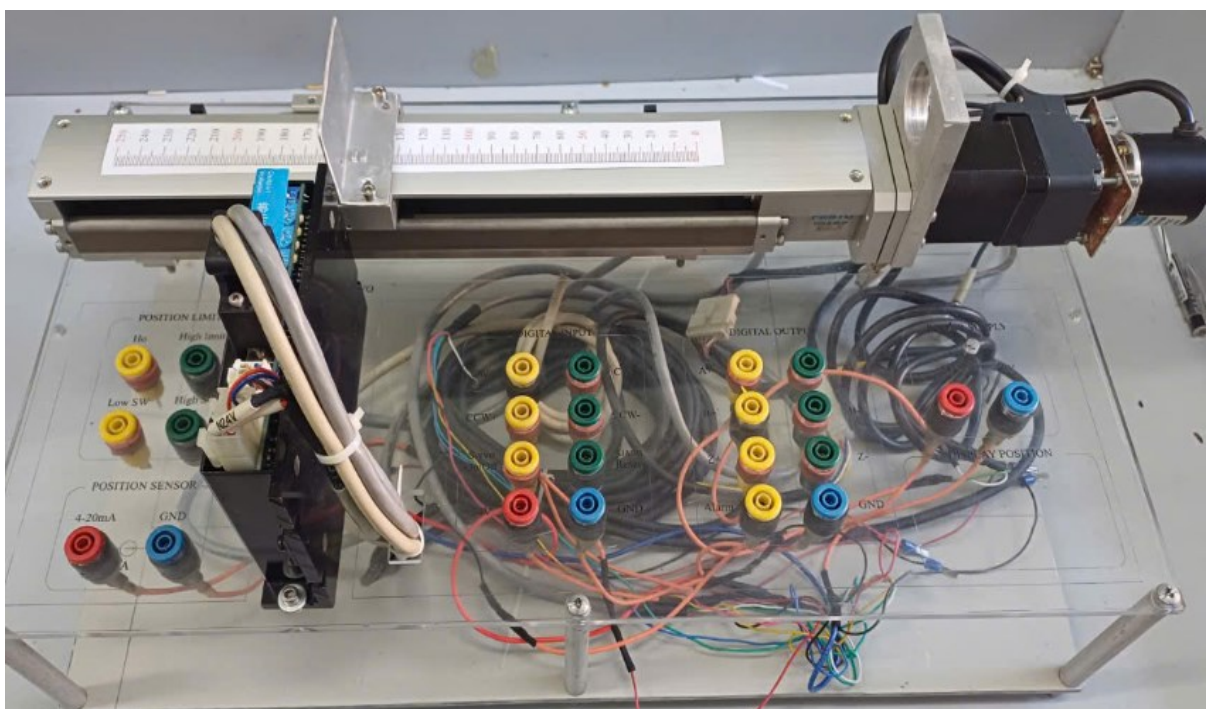
3. Tính mới và sáng tạo: Mô hình phục vụ cho giảng dạy thực hành nhằm cung cấp cho sinh viên các kiến thức thực tế về điều khiển vị trí. Tạo ra mô hình ứng dụng bộ điều khiển vị trí bàn trượt nhằm cung cấp cho sinh viên kiến thức thực tế đang được ứng dụng rất phổ biến trong điều khiển hiện nay.

4. Tóm tắt kết quả nghiên cứu: Sản phẩm nghiên cứu đã hoàn thiện gồm thiết bị đào tạo thực hành “Mô hình thực hành điều khiển vị trí bàn trượt”, với 3 bài tập đã được xây dựng vận hành trên mô hình phục vụ giảng dạy thực hành cho sinh viên.

5. Tên sản phẩm: Mô hình thực hành điều khiển vị trí bàn trượt.

6. Hiệu quả, phương thức chuyển giao kết quả nghiên cứu và khả năng áp dụng: Mô hình thực hành điều khiển vị trí bàn trượt sẽ được giới thiệu và chuyển giao đến các giảng viên và cán bộ giảng dạy thực hành chuyên ngành Tự động hóa trường Đại học Sư phạm Kỹ thuật – Đại học Đà Nẵng để giảng dạy thực hành cho sinh viên.

7. Hình ảnh, sơ đồ minh họa chính



Ngày 22 tháng 03 năm 2025

TM. Hội đồng Khoa
Chủ tịch
(ký, họ và tên)

Chủ nhiệm đề tài
(ký, họ và tên)

TS. Trần Hoàng Vũ

Nguyễn Văn Nam

XÁC NHẬN CỦA TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
KT. HIỆU TRƯỞNG
PHÓ HIỆU TRƯỞNG

PGS. TS. Võ Trung Hùng

INFORMATION ON RESEARCH RESULTS

1. General information:

Project title: Research on Developing a Position Control Model for Sliding Table

Code number: T2023-06-16

Coordinator: Nguyen Van Nam

Implementing institution: The University of Da Nang - University of Technology and Education

Duration: March 2024 – February 2025

2. Objective(s):

1. Develop a practical teaching device for students in the field of Automation, focusing on advanced control techniques to meet the standards of an advanced curriculum.

2. Implement position control solutions to deliver practical and intuitive knowledge, enabling students to quickly comprehend and apply concepts in real-world scenarios.

3. Provide a practical application device to support training in the *Logic Control* course within the Department of Electrical and Electronics, University of Technology and Education.

4. Design three practical exercises to enhance the teaching of advanced control knowledge in the *Logic Control* course, including:

- Connecting an encoder to a PLC S7-1200, reading pulses from the encoder, and calculating rotational speed.
- Connecting a PLC S7-1200 to a servo motor and outputting high-speed pulses for servo motor control.
- Programming position control for a sliding table connected to a PLC S7-1200, enabling movement within the range of 0 to 200 mm.

3. Creativity and innovation: The model is designed to facilitate practical teaching by providing students with comprehensive and hands-on knowledge of position control systems. It incorporates a position control system for a sliding table, allowing students to gain practical experience with technologies widely utilized in modern control systems.

4. Research results: The research has resulted in the successful development of a practical training device, titled "Position Control Sliding Table Practice Model." This device includes three fully developed and operational exercises to support hands-on teaching for students.

5. Products: Position Control Sliding Table Practice Model.

6. Effects, transfer alternatives of research results and applicability: The Position Control Sliding Table Practice Model will be introduced and transferred to lecturers and practical teaching staff specializing in Automation at the University of Technology and Education - The University of Da Nang. This model aims to enhance practical training and improve the teaching and learning experience for students.

MỞ ĐẦU

Hiện nay, trong lĩnh vực kỹ thuật điều khiển tự động, điều khiển vị trí là một kỹ thuật phổ biến, được sử dụng rộng rãi trong nhiều lĩnh vực kỹ thuật [1] – [6]. Rất nhiều máy móc công nghiệp như máy phay CNC, máy tiện CNC, máy cắt laser, máy in 3D và nhiều thiết bị khác đều phụ thuộc vào các giải pháp điều khiển vị trí. Do đó, nhiều trường đào tạo kỹ thuật đã thể hiện sự quan tâm trong việc tích hợp kiến thức cơ bản về điều khiển vị trí vào chương trình giảng dạy.

1. Tính cấp thiết của đề tài

Để đáp ứng đào tạo theo định hướng ứng dụng, chương trình đào tạo đã rà soát, cập nhật tăng thời lượng thực hành, thực tập theo định hướng chung của nhà trường. Đi kèm với sự phát triển số lượng và chất lượng đội ngũ, cán bộ giảng dạy và phục vụ giảng dạy thì vấn đề đầu tư phát triển trang thiết bị phục vụ trong đào tạo thực hành, thí nghiệm cũng là vấn đề rất cần thiết.

Để đào tạo đạt theo yêu cầu chương trình đào tạo đổi mới, Ngành công nghệ kỹ thuật điều khiển và tự động hóa cũng đã đưa ra lộ trình phát triển về đội ngũ giảng dạy, về cơ sở vật chất và đặc biệt là trang thiết bị phục vụ trong đào tạo thực hành, thí nghiệm. Tuy nhiên, với nguồn kinh phí hạn hẹp hiện nay thì việc trang bị đầy đủ trang thiết bị để đáp ứng cho tất cả các môn thực hành, thí nghiệm là rất khó khăn. Do vậy, cần huy động thêm các nguồn kinh phí khác nhằm hỗ trợ tạo ra các trang thiết bị phục vụ cho thực hành, thí nghiệm là một giải pháp cần được chú trọng. Vì thế, ứng dụng nghiên cứu khoa học có sản phẩm là tạo ra các trang thiết bị nhằm phục vụ cho đào tạo thực hành, thí nghiệm là một hướng cần phát huy nhằm giải quyết bớt một phần khó khăn hiện nay của nhà trường về đầu tư thiết bị mới.

Ngành công nghệ kỹ thuật điều khiển và tự động hóa đã nghiên cứu, rà soát và cập nhật tăng thời lượng thực hành, thực tập cho một số học phần mũi nhọn của ngành. Trong đó học phần “Thực hành điều khiển logic” là một trong những học phần trọng tâm của sinh viên tự động hóa. Những năm gần đây trang thiết bị thực hành học phần này vẫn được chú trọng đầu tư nhưng vẫn còn hạn chế, kiến thức cơ bản thì hiện tại trang thiết bị đào tạo đáp ứng được, ở kiến thức nâng cao của học phần này thì hiện đang thiếu các mô hình thực hành trực quan. Vì vậy, việc nghiên cứu tạo ra sản phẩm

ứng dụng phục vụ cho đào tạo các kiến thức nâng cao cho học phần “Thực hành điều khiển logic” là rất cần thiết.

Ngoài ra, việc học thực hành ứng dụng cung cấp cho các bạn sinh viên các kiến thức thực tế cơ bản, kết hợp giữa lý thuyết cơ sở và việc thực hành trực tiếp. Do đó, chương trình đào tạo được thiết kế theo mô hình này cũng thực tế hơn, chú trọng vào việc nâng cao năng lực và kỹ năng làm việc cho các em sinh viên. Từ đó, giúp các em rút ngắn quá trình làm quen với công việc sau khi ra trường, tăng cơ hội việc làm cho sinh viên tốt nghiệp.

2. Đối tượng, phạm vi nghiên cứu

2.1. Đối tượng nghiên cứu

- Thiết bị phục vụ trong đào tạo thực hành
- Cơ sở lý thuyết về bộ lập trình PLC
- Cơ sở lý thuyết về bộ phát xung PTO, PWM

2.2. Phạm vi nghiên cứu

- Nghiên cứu, chế tạo thiết bị phục vụ cho đào tạo thực hành.

3. Cách tiếp cận, phương pháp nghiên cứu

3.1. Cách tiếp cận

- Tìm hiểu các tài liệu về các bộ lập khả trình PLC (PLC - Programmable Logic Controller)
- Tìm hiểu các mô hình tương tự đã có trong thực tế, các ứng dụng về bộ phát xung PTO, PWM trong điều khiển hiện nay.

3.2. Phương pháp nghiên cứu

Phương pháp lý thuyết:

- Nghiên cứu thiết kế bản vẽ chi tiết của mô hình
- Nghiên cứu ứng dụng về bộ phát xung PTO, PWM trên PLC

Phương pháp thực nghiệm:

- Gia công chế tạo mô hình điều khiển vị trí bàn trượt

- Lập trình điều khiển và kiểm tra thực nghiệm.

4. Nội dung nghiên cứu

Đề cương chi tiết của đề tài được chia thành 6 phần như sau:

Mở đầu

Chương 1: Tổng quan

Chương 2: Cơ sở lý thuyết

Chương 3: Thiết kế và chế tạo mô hình

Chương 4: Kiểm tra thực nghiệm và đánh giá mô hình

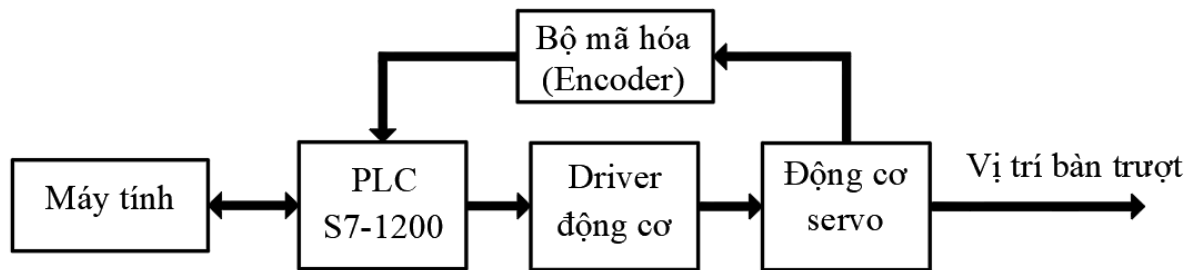
Kết luận

CHƯƠNG 1: TỔNG QUAN

1.1. Tổng quan về mô hình

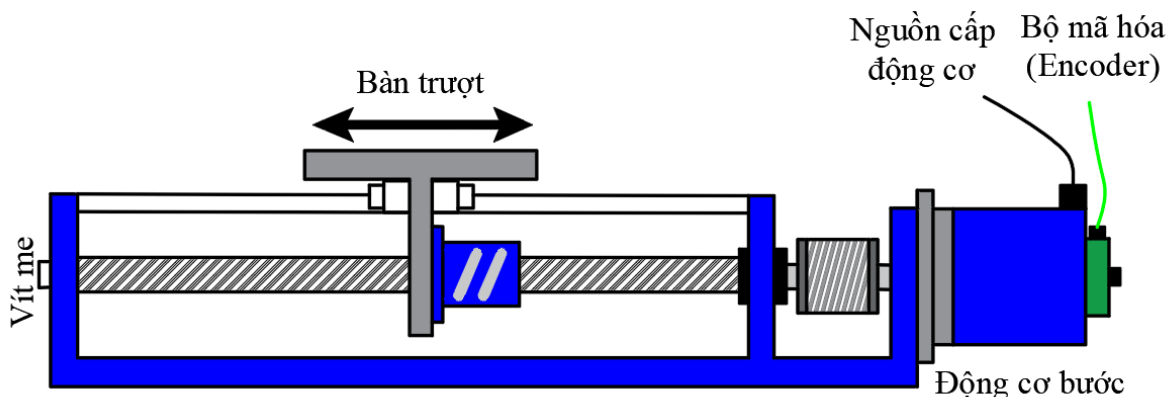
Mô hình điều khiển vị trí là một mô hình thực nghiệm cần thiết cho sinh viên học tập và nghiên cứu, với cấu tạo đơn giản nhưng kiến thức nền tảng về điều khiển vị trí được vận dụng nhằm giúp sinh viên có cái nhìn trực quan.

Nguyên lý hệ thống điều khiển vị trí bằng PLC được trình bày như Hình 1.1. Việc kiểm soát vị trí bàn trượt bằng một động cơ servo. Ở đó, bộ điều khiển trung tâm (PLC) nhận giá trị phản hồi vị trí từ bộ mã hóa vòng quay (encoder) từ động cơ servo (Hình 1.2).



Hình 1.1: Sơ đồ nguyên lý của hệ thống điều khiển vị trí

Phần cứng hệ thống mô hình gồm bốn khối chức năng quan trọng: (i) Bộ điều khiển lập khả trình S7-1200, (ii) bộ động cơ servo, (iii) bộ cảm biến đo khoảng cách và (iv) bộ trượt điện (Hình 1.2)



Hình 1.2: Mô hình phần cứng của hệ thống điều khiển vị trí

Việc kiểm soát vị trí cũng như tốc độ của các bộ truyền động điện là cần thiết cho rất nhiều quy trình công nghiệp khác nhau [1]– [3]. Hệ thống định vị có thể đạt được

độ chính xác cao là rất cần thiết và đóng vai trò quan trọng trong các thiết bị công nghiệp như máy công cụ, hệ thống robot.

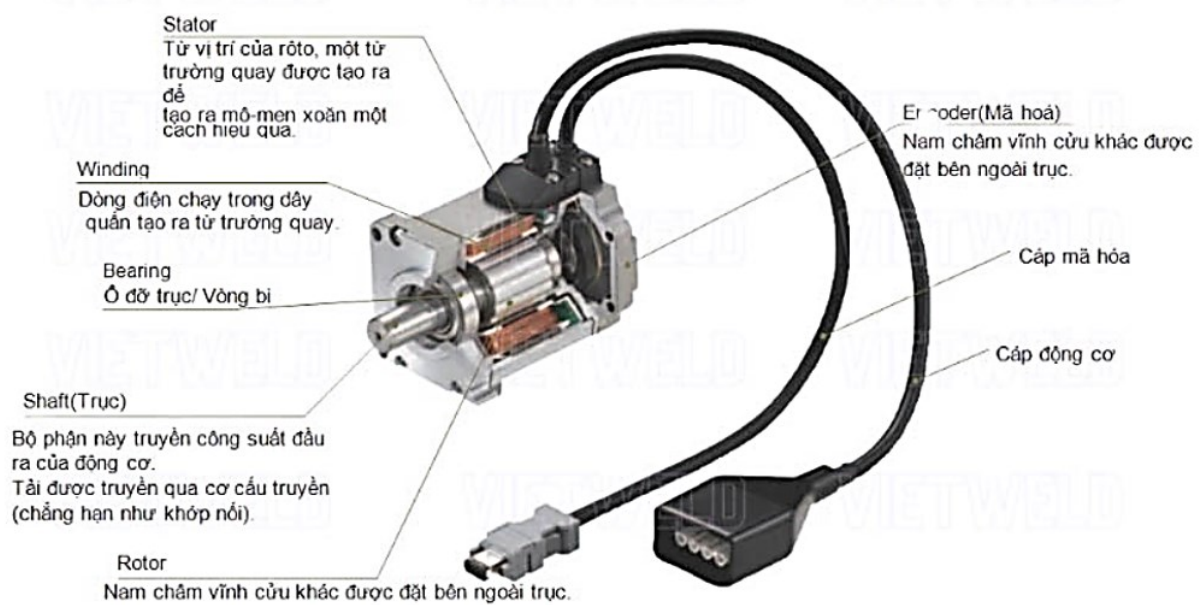
1.2. Tổng quan về động cơ servo

Động cơ servo, một công nghệ không thể thiếu trong thế giới tự động hóa hiện đại, đã mang lại nhiều lợi ích cho ngành công nghiệp sản xuất, robot và các lĩnh vực khác yêu cầu độ chính xác và hiệu suất cao.

1.2.1. Định nghĩa về động cơ servo

Động cơ servo là một loại động cơ được điều khiển vị trí hoặc tốc độ thông qua bộ điều khiển (Servo controller) trong hệ thống điều khiển có phản hồi kín. Chúng được sử dụng để tạo ra chuyển động theo yêu cầu. Bằng cách sử dụng cảm biến vị trí, thường là bộ mã hóa (encoder), động cơ servo có khả năng phát hiện vị trí hiện tại và điều chỉnh chính xác đến khi đạt được vị trí mong muốn.

1.2.2. Cấu tạo của động cơ servo



Hình 1.3: Cấu tạo của động cơ servo

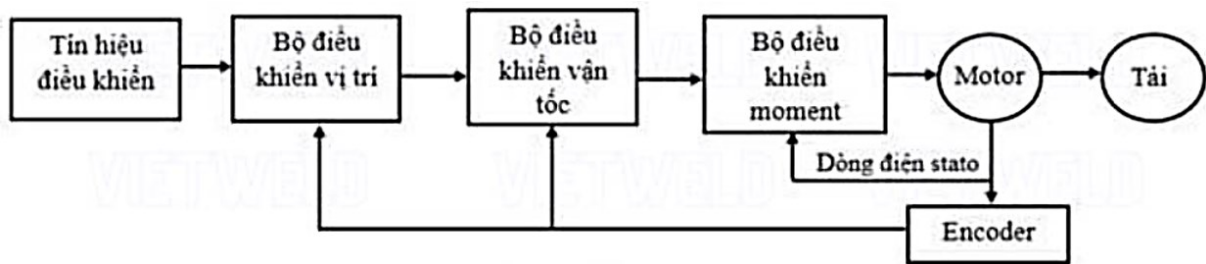
Động cơ bao gồm hai cuộn dây là cuộn dây stato và rotor. Động cơ bao gồm hai ổ trục ở mặt trước và mặt sau để trục chuyển động tự do.

Stato đóng vai trò là thỏi nam châm hút từ trường, hỗ trợ động cơ hoạt động. Khi dòng điện được dẫn qua các cuộn dây quấn đặt bên trong lõi thép, nó sẽ tạo ra một lực cảm ứng điện từ để chuyển đổi từ năng lượng điện thành năng lượng cơ học.

Dây quấn rotor được quấn trên phần quay của động cơ và dây quấn này còn được gọi là dây quấn phản ứng của động cơ. Rotor là 1 nam châm vĩnh cửu, là phần chuyển động trong hệ thống điện từ của động cơ điện. Lực tương tác giữa các cuộn dây và các từ trường điện tích tạo ra một mô men xoắn xoay quanh trục quay của Rotor.

Bộ mã hóa có nhiệm vụ giám sát để xác định tốc độ quay và vị trí quay của động cơ sau đó phản hồi để điều khiển chính xác vị trí.

1.2.3. Nguyên lý làm việc của động cơ servo



Hình 1.4: Sơ đồ nguyên lý làm việc của động cơ servo

- Thiết lập mục tiêu: Trước hết, ta phải cung cấp cho động cơ một mục tiêu vị trí hoặc tốc độ. Mục tiêu này có thể được đặt thông qua phần mềm, hoặc thông qua tín hiệu từ một nguồn bên ngoài, như một bộ điều khiển tay cầm, bàn phím, hoặc tín hiệu từ một hệ thống điều khiển tự động.
- Đọc giá trị cảm biến: Hệ thống servo sau đó đọc giá trị từ cảm biến vị trí, thường là một bộ mã hóa, để xác định vị trí hiện tại của đầu ra động cơ.
- So sánh với mục tiêu: Bộ điều khiển servo sau đó so sánh vị trí hiện tại với mục tiêu đã đặt. Sự khác biệt giữa hai giá trị này được gọi là “độ lệch tín hiệu”.
- Điều chỉnh động cơ: Dựa vào độ lệch tín hiệu, bộ điều khiển servo sẽ quyết định cách điều chỉnh động cơ. Nếu vị trí hiện tại thấp hơn mục tiêu, động cơ sẽ được đưa vào chế độ chạy để tăng vị trí. Nếu vị trí hiện tại cao hơn mục tiêu, động cơ có thể được đưa vào chế độ chạy ngược để giảm vị trí. Mức độ điều chỉnh động cơ phụ thuộc vào cả sai lệch hiện tại và thời gian mà sai lệch này tồn tại – đây chính là nguyên lý của việc điều khiển PID (Proportional, Integral, Derivative).
- Lặp lại quy trình: Quá trình này sau đó được lặp lại liên tục, với bộ điều khiển servo đọc giá trị cảm biến, so sánh nó với mục tiêu, và điều chỉnh động cơ theo cần thiết. Điều này cho phép hệ thống servo duy trì vị trí hoặc tốc độ mục tiêu một cách chính xác, thậm chí khi có các yếu tố ngoại vi như tải trọng thay đổi.

1.2.4. Phân loại động cơ servo

Người ta phân loại động cơ servo theo dòng điện, . Dòng DC và AC như sau:

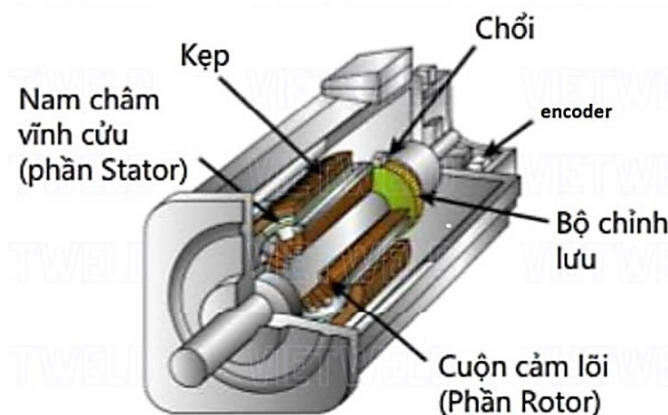


Hình 1.5: Phân loại động cơ servo

1.2.4.1. Động cơ servo DC

Loại động servo DC được thiết kế để ứng dụng vào những hệ thống sử dụng dòng điện nhỏ. Nó được phân chia thành 2 loại đó là: động cơ DC Servo 1 chiều có chổi than (Brushed Servo Motors) và loại động cơ DC Servo không có chổi than (brushless servo motor).

- Cấu tạo:



Hình 1.6: Cấu tạo động cơ DC Servo

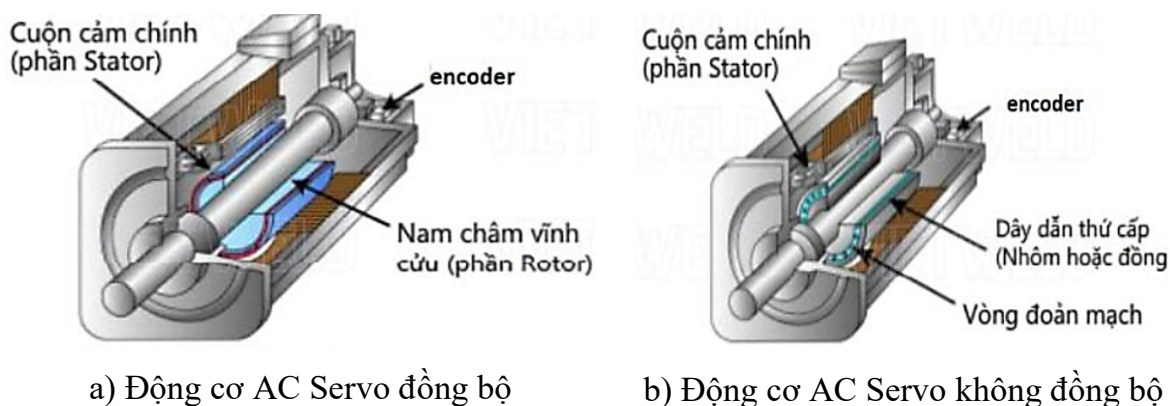
Bộ chỉnh lưu, kẹp, bộ mã hóa (encoder), chổi và cuộn cảm lõi, nam châm vĩnh cửu là những thành phần cấu tạo cơ bản của 1 một Servo DC. Loại động cơ DC Servo có chổi than có cấu tạo gồm: chổi than, stato, cuộn lõi và roto. Loại động cơ DC Servo không có chổi than đặc biệt hơn khi cuộn pha lắp ở rotor chính là động cơ vĩnh cửu.

1.2.4.2. Động cơ servo AC

Động cơ Servo AC được xem giống với động cơ bước vì có nhiều nét tương đồng với động cơ bước. Stato là cuộn dây quấn riêng biệt và rotor là nam châm vĩnh cửu. Trong động cơ Servo AC chia thành 2 loại nhỏ:

- Động cơ AC Servo không đồng bộ cấu tạo gồm: cuộn cảm chính, dây dẫn thứ cấp, vòng đoản mạch, bộ mã hóa (encoder).

- Động cơ AC Servo đồng bộ cấu tạo đơn giản hơn gồm: bộ mã hóa (encoder), cuộn cảm chính, và một nam châm vĩnh cửu.



Hình 1.7: Cấu tạo động cơ AC Servo

1.2.5. Ứng dụng của động cơ servo trong thực tế

Động cơ Servo thường được sử dụng trong các ứng dụng yêu cầu độ chính xác cao: máy tiện cnc, máy phay cnc, trung tâm gia công cnc, robot, máy móc công nghiệp, hệ thống điều khiển dân dụng và quân sự, máy móc sản xuất, hàng không vũ trụ, y tế, điện tử tiêu dùng và nhiều ứng dụng khác. Chính nhờ tính linh hoạt và khả năng kiểm soát chính xác, chúng đã trở thành một thành phần quan trọng để đảm bảo hiệu suất cao và đáng tin cậy trong các hệ thống và quy trình công nghiệp.

1.2.6. Ưu điểm và nhược điểm của động cơ servo

1.2.6.1. Ưu điểm

- Độ chính xác cao: Có thể kiểm soát vị trí và tốc độ một cách chính xác.
- Hiệu suất tốt ở mọi tốc độ: Có thể cung cấp mô-men xoắn đầu ra ổn định ở cả tốc độ thấp và tốc độ cao. Mô-men không đổi trong khoảng tốc độ từ 0 đến tốc độ định mức. Do đó động cơ servo thường được sử dụng trong một số ứng dụng cần mô-men cao ở tốc độ thấp.
- Điều chỉnh tốc độ dễ dàng: Tốc độ có thể được điều chỉnh một cách dễ dàng và linh hoạt để phù hợp với nhu cầu của ứng dụng.
- Tính linh hoạt: Động cơ servo thường có nhiều kích cỡ và loại khác nhau để phù hợp với nhiều loại ứng dụng khác nhau. Kích thước và trọng lượng nhỏ hơn, ít bị hư hỏng.
- Đáp ứng nhanh: Động cơ servo có thể đáp ứng nhanh chóng khi có sự thay đổi

trong tín hiệu đầu vào. Điều này cho phép hệ thống điều chỉnh nhanh chóng để phù hợp với các yêu cầu thay đổi.

- Hoạt động êm ái: Do không có chổi than như trong động cơ DC, động cơ servo hoạt động êm ái và không gây tiếng ồn. Tiết kiệm điện năng (động cơ servo có thể tiết kiệm 5-20% điện năng so với động cơ thường).

1.2.6.2. Nhược điểm

- Chi phí cao: So với các loại động cơ khác như động cơ DC hay AC thông thường, động cơ servo thường có giá thành cao hơn do cấu trúc và hệ thống điều khiển phức tạp.

- Yêu cầu điều khiển phức tạp: Động cơ servo cần một bộ điều khiển (servo controller) để hoạt động. Điều khiển động cơ này đòi hỏi kiến thức và kỹ thuật phức tạp hơn so với các động cơ thông thường.

- Bảo dưỡng và sửa chữa khó khăn: Việc bảo dưỡng và sửa chữa động cơ servo có thể khá khó khăn do cấu trúc phức tạp. Ngoài ra, việc tìm kiếm và thay thế các bộ phận hỏng cũng có thể tốn kém và mất thời gian.

- Cần nhiều dây kết nối: Động cơ servo thường cần nhiều dây kết nối hơn so với các loại động cơ khác. Điều này có thể gây khó khăn trong việc lắp đặt và bảo dưỡng.

- Nhiệt độ hoạt động: Động cơ có thể tạo ra nhiệt khi hoạt động, đặc biệt khi hoạt động ở công suất cao, đòi hỏi hệ thống làm mát hiệu quả.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Cơ sở lý thuyết về PLC S7-1200

2.1.1. Giới thiệu về PLC S7-1200

Bộ điều khiển logic lập khả trình (PLC) S7-1200 mang lại tính linh hoạt và sức mạnh để điều khiển nhiều thiết bị đa dạng hỗ trợ các yêu cầu về điều khiển tự động. Sự kết hợp giữa thiết kế thu gọn, cấu hình linh hoạt và tập lệnh mạnh mẽ đã khiến cho S7-1200 trở thành một giải pháp hoàn hảo dành cho việc điều khiển nhiều ứng dụng đa dạng khác nhau.

Kết hợp một bộ vi xử lý trung tâm (CPU), một bộ nguồn tích hợp, các mạch ngõ vào và mạch ngõ ra trong một kết cấu thu gọn, CPU trong S7-1200 đã tạo ra một PLC mạnh mẽ. Sau khi người dùng tải xuống một chương trình, CPU sẽ chứa mạch logic được yêu cầu để giám sát và điều khiển các thiết bị nằm trong ứng dụng. CPU giám sát các ngõ vào và làm thay đổi ngõ ra theo logic của chương trình người dùng, có thể bao gồm các hoạt động như logic Boolean, việc đếm, định thời, các phép toán phức hợp và việc truyền thông với các thiết bị thông minh khác.

Một số tính năng bảo mật giúp bảo vệ việc truy xuất đến cả CPU và chương trình điều khiển:

- Mỗi CPU cung cấp một sự bảo vệ bằng mật khẩu cho phép người dùng cấu hình việc truy xuất đến các chức năng của CPU.
- Người dùng có thể sử dụng chức năng “know-how protection” để ẩn mã nằm trong một khối xác định.

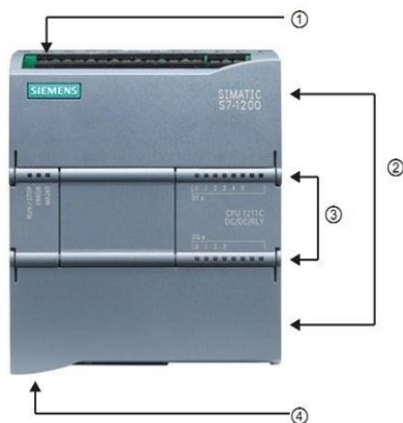
CPU cung cấp một cổng PROFINET để giao tiếp qua một mạng PROFINET. Các module truyền thông là có sẵn dành cho việc giao tiếp qua các mạng RS232 hay RS485.

Các kiểu CPU khác nhau cung cấp một sự đa dạng các tính năng và dung lượng giúp cho người dùng tạo ra các giải pháp có hiệu quả cho nhiều ứng dụng khác nhau.

2.1.2. Cấu trúc phần cứng

2.1.2.1. Hình dáng và cấu trúc bên ngoài

Hình dáng bên ngoài của PLC-S7-1200 như hình bên dưới.



- ① Bộ phận kết nối nguồn
- ② Các bộ phận kết nối dây của người dùng có thể tháo được (phía sau các nắp che)
- ③ Khe cắm thẻ nhớ nằm dưới cửa phía trên
- ④ Các LED trạng thái dành cho I/O tích hợp
- ⑤ Bộ phận kết nối PROFINET (phía trên của CPU)

Hình 2.1: Hình dáng bên ngoài của PLC S7-1200

Một số dòng PLC S7-1200 như bảng sau:

Bảng 2.1: Bảng thông số kỹ thuật của dòng PLC S7-1200

Chức năng	CPU 1211C	CPU 1212C	CPU 1214C
Kích thước vật lý (mm)	90 x 100 x 75		110 x 100 x 75
Bộ nhớ người dùng: * Bộ nhớ làm việc * Bộ nhớ nạp	* 25 kB * 1 MB		* 50 kB * 2 MB
I/O tích hợp cục bộ * Kiểu số * Kiểu tương tự	* 6 ngõ vào / 4 ngõ ra	* 8 ngõ vào / 6 ngõ ra	* 14 ngõ vào / 10 ngõ ra
Kích thước ảnh tiến trình	1024-byte ngõ vào (I) và 1024-byte ngõ ra (Q)		
Bộ nhớ bit (M)	4096-byte		8192-byte
Module mở rộng vào/ra	Không	2	8
Bảng tín hiệu	1		
Các module truyền thông	3 (mở rộng về bên trái)		
Các bộ đếm tốc độ cao * Đơn pha * Vuông pha	3 * 3 tại 100 kHz * 3 tại 80 kHz	4 * 3 tại 100 kHz * 1 tại 30 kHz * 3 tại 80 kHz	6 * 3 tại 100 kHz * 3 tại 30 kHz * 3 tại 80 kHz
Các ngõ ra xung	4		

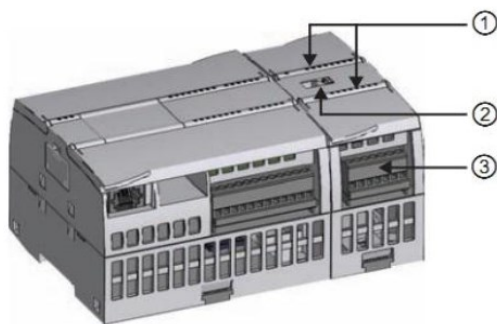
Thẻ nhớ	Thẻ nhớ SIMATIC (tùy chọn)
Thời gian lưu giữ đồng hồ	Thông thường 10 ngày / ít nhất 6 ngày tại 40°C
PROFINET	1 cổng truyền thông Ethernet
Tốc độ thực thi tính toán	18 μ s/lệnh
Tốc độ thực thi Boolean	0,1 μ s/lệnh

2.1.2.2. Các module mở rộng

Họ S7-1200 cung cấp một số lượng lớn các module tín hiệu và bảng tín hiệu để mở rộng dung lượng của CPU. Người dùng còn có thể lắp đặt thêm các module truyền thông để hỗ trợ các giao thức truyền thông khác.

- Module tín hiệu

Người dùng có thể sử dụng các module tín hiệu để thêm vào CPU các chức năng. Các module tín hiệu kết nối vào phía bên phải của CPU.



- ① Các LED trạng thái dành cho I/O của module tín hiệu
- ② Bộ phận kết nối đường dẫn
- ③ Bộ phận kết nối nối dây của người dùng có thể tháo rời

Hình 2.2: Module tín hiệu (SM) của PLC S7-1200

Thông số của module tín hiệu được thể hiện như bảng sau:

Bảng 2.2: Một số module tín hiệu của PLC S7-1200

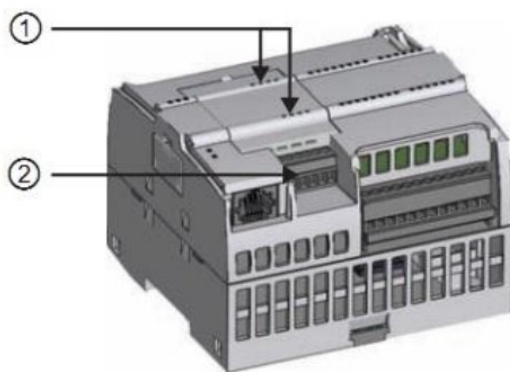
Module		Chỉ ngõ vào	Chỉ ngõ ra	Kết hợp In/Out
Module tín hiệu (SM)	Kiểu số	8 x DC In	8 x DC Out 8 x Relay Out	8 x DC In / 8 x DC Out 8 x DC In / 8 x Relay Out
		16 x DC In	16 x DC Out 16 x Relay Out	16 x DC In / 16 x DC Out 16 x DC In / 16 x Relay Out
	Kiểu	4 x Analog In	2 x Analog In	4 x Analog In / 2 x

	tương tự	8 x Analog In	4 x Analog In	Analog Out
Bảng tín hiệu (SB)	Kiểu số	-	-	2 x DC In / 2 x DC Out
	Kiểu tương tự	-	1 x Analog In	-

- **Module bảng tín hiệu**

Một bảng tín hiệu (SB) cho phép người dùng thêm vào I/O cho CPU. Người dùng có thể thêm một SB với cả I/O kiểu số hay kiểu tương tự. SB kết nối vào phía trước của CPU.

- SB với 4 I/O kiểu số (ngõ vào 2 x DC và ngõ ra 2 x DC).
- SB với 1 ngõ ra kiểu tương tự.



① Các LED trạng thái trên SB

② Bộ phận kết nối dây của người dùng có thể tháo ra

Hình 2.3: Module tín hiệu (SB) của PLC S7-1200

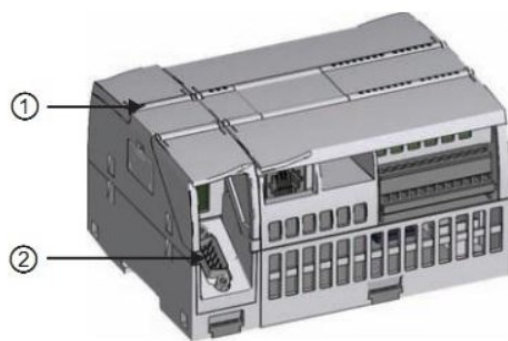
Bảng 2.3: Module bảng tín hiệu của PLC S7-1200

Module		Chỉ ngõ vào	Chỉ ngõ ra	Kết hợp In/Out
Bảng tín hiệu (SB)	Kiểu số	-	-	2 x DC In / 2 x DC Out
	Kiểu tương tự	-	1 x Analog In	-

- **Module truyền thông**

Họ S7-1200 cung cấp các module truyền thông (CM) dành cho các tính năng bổ sung vào hệ thống. Có 2 module truyền thông: RS232 và RS485.

- CPU hỗ trợ tối đa 3 module truyền thông
- Mỗi CM kết nối vào phía bên trái của CPU (hay về phía bên trái của một CM khác).



- ① Các LED trạng thái dành cho module truyền thông
- ② Bộ phận kết nối truyền thông

Hình 2.4: Module truyền thông (CM) của PLC S7-1200

2.1.3. Cấu trúc bộ nhớ

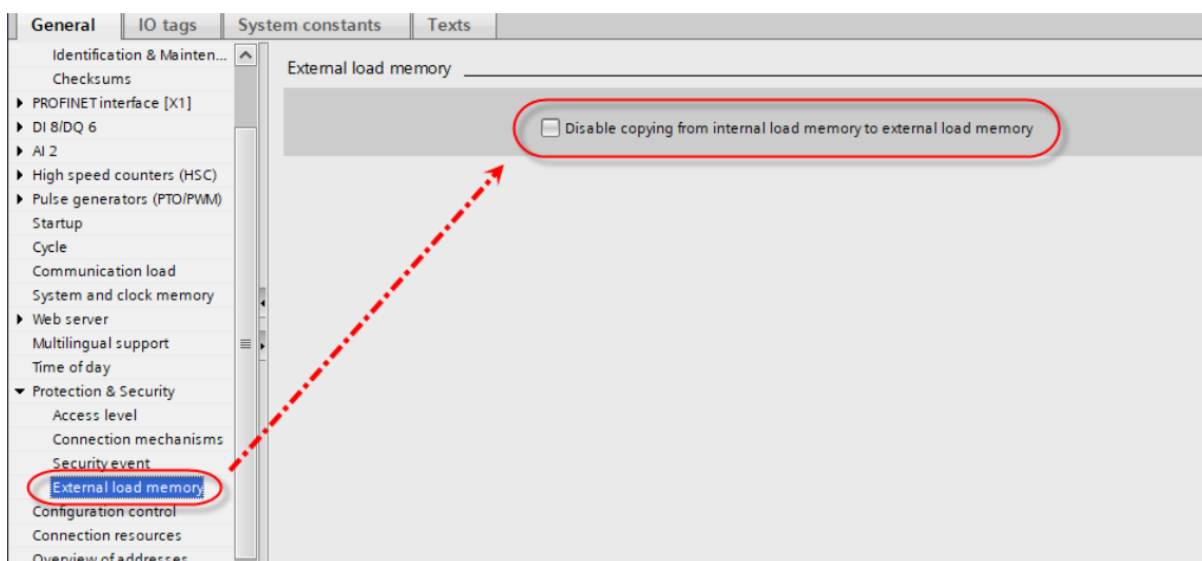
Tất cả các bộ điều khiển PLC đều có thành phần chính là: Một bộ nhớ chương trình RAM bên trong (có thể mở rộng thêm một số bộ nhớ ngoài EPROM). Một bộ vi xử lý có cổng giao tiếp dùng cho việc ghép nối với PLC, Các Modul vào/ra.

2.1.3.1. Phân chia bộ nhớ

PLC sử dụng vùng nhớ để lưu trữ chương trình người dùng, dữ liệu và cấu hình. Có 3 loại vùng nhớ được sử dụng trong PLC S7-1200 như sau:

- Vùng nhớ chương trình (Load Memory)
- Vùng nhớ làm việc (Work Memory)
- Vùng nhớ lưu trữ (Retain Memory)

2.1.3.2. Vùng nhớ chương trình (Load Memory)



Hình 2.5: Vô hiệu hóa vùng nhớ chương trình

Load Memory có thể so sánh nó như ổ cứng trên máy tính, dùng chứa dữ liệu chương trình mà ta download xuống. Load memory có thể bị thay thế bằng thẻ nhớ

ngoài. Để vô hiệu hóa chức năng đó, ta vào mục Hardware và chọn 'Disable' tính năng này.

Nếu không nhấn chọn “Disable copying from internal load memory to external load memory” thì khi gắn thẻ nhớ với chế độ Program, vùng nhớ Load Memory của PLC sẽ chuyển sang lưu trữ trên thẻ nhớ.

Load Memory lưu trữ các dữ liệu sau:

- Khối hàm chương trình: OB, FC, FB.
- Data Blocks (DB).
- Cấu hình phần cứng (Hardware Config).
- Technology object.

2.1.3.3. Vùng nhớ làm việc (Work Memory)

Work Memory có thể so sánh như là RAM trên máy tính. Work memory dùng lưu trữ các thành phần của chương trình đang được gọi ra để thực thi. Và tất nhiên là vùng nhớ không tự lưu lại.

Work memory chứa code thực thi như: FC, FB, OB và Data thực thi được gọi ra từ các khối data DB, Technology object. Thực thi lệnh nào thì lệnh đó sẽ được lòi ra Work memory để làm việc.

Mỗi vị trí bộ nhớ khác nhau có một địa chỉ đơn nhất. Chương trình người dùng sử dụng các địa chỉ này để truy xuất thông tin trong vị trí bộ nhớ.

2.1.3.4. Vùng nhớ lưu trữ (Retain Memory)

Khi PLC đang hoạt động, nếu muốn thông số của tag được giữ lại khi PLC restart thì chúng ta sẽ dùng đến bộ nhớ Retain Memory.

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	C
1	Input_1	Bool	%I0.0	☐	☒	☒	☒	
2	Status_Data	Word	%QW100	☐	☒	☒	☒	
3	Tag_3	DWord	%MD10	☒	☒	☒	☒	
4	Tag_4	DWord	%MD14	☐	☒	☒	☒	
5	Tag_1	Bool	%M0.0	☒	☒	☒	☒	
6	<Add new>			☐	☒	☒	☒	

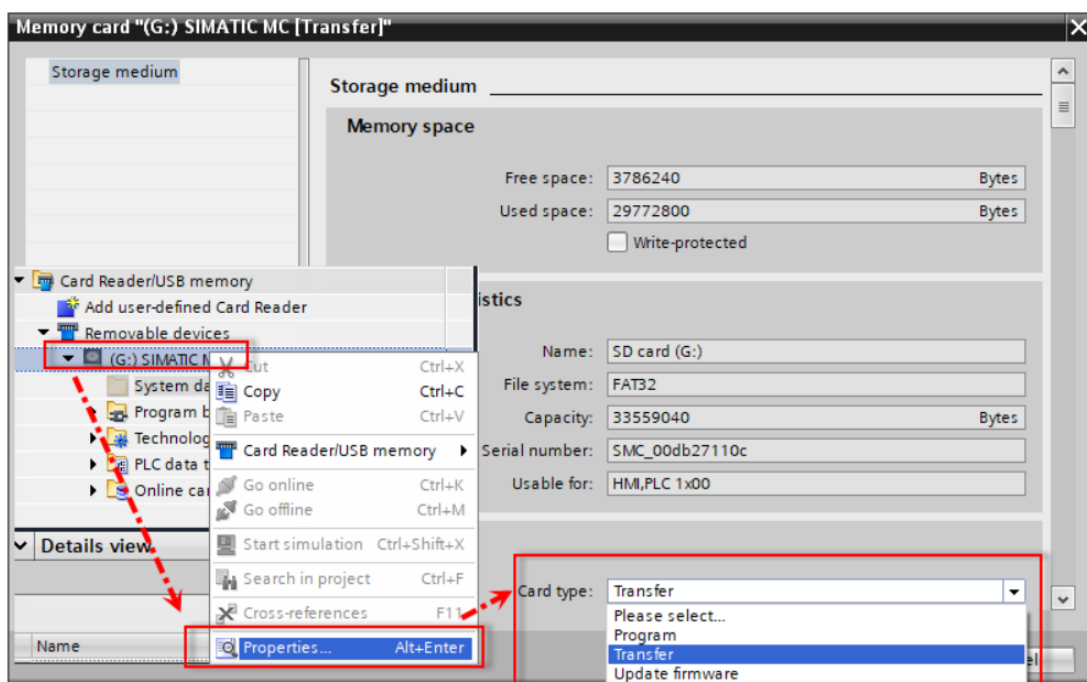
Hình 2.6: Cấu hình Retain Memory để PLC lưu trữ giá trị tag khi mất điện

Theo mặc định thì thông số của tag sẽ không được Retain memory lưu lại. Tuy nhiên, nếu chúng ta muốn lưu thông số của tag nào thì sẽ cấu hình cho phép lưu trong Tia portal.

2.1.3.5. Thẻ nhớ

Thẻ nhớ cho PLC S7-1200 là một thẻ SD được định dạng theo chuẩn Siemens để tương thích với PLC S7-1200. Tính năng chung của thẻ nhớ S7 1200 là:

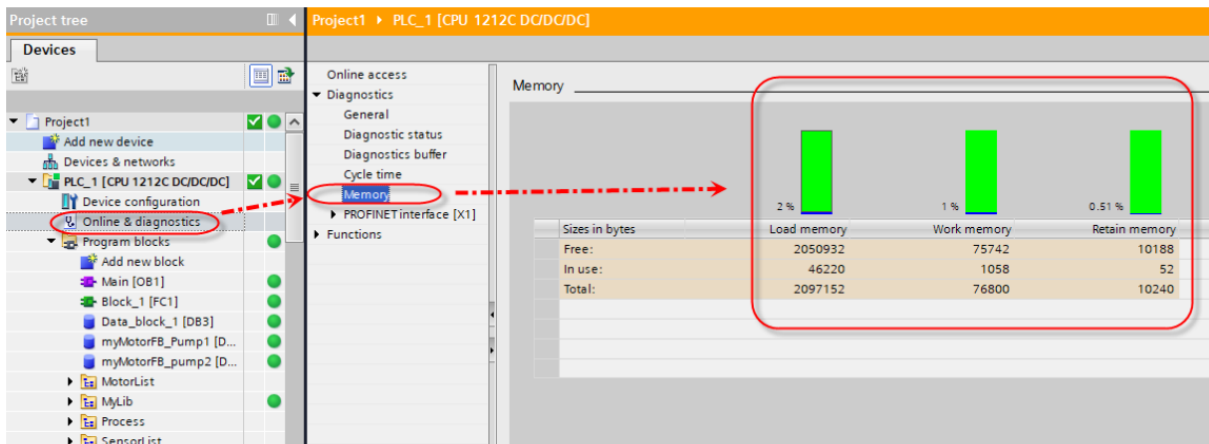
- Lưu trữ dữ liệu dạng: file datalog, file báo cáo, report, txt..... từ PLC
- Dùng để tải chương trình vào PLC (Transfer): ví dụ bạn không có cách nào load chương trình plc vào, hay khi plc bị lỗi load hoài không được, bạn có thể load chương trình vào thẻ nhớ rồi để thẻ nhớ ở chế độ Transfer, gắn vào PLC, khởi động lại là chương trình sẽ load vào PLC.
- Tính năng này hay dùng khi PLC bị lỗi không kết nối được, hoặc do password cài full không read/write chương trình PLC được, ta sử dụng thẻ nhớ với chức năng này để xóa hết chương trình PLC và trả về mặc định.
- Chế độ Program (chế độ thẻ nhớ SD thay thế Load Memory của PLC): cài đặt chế độ cho thẻ nhớ là Program, gắn vào PLC là bạn có thể dùng thẻ nhớ để lưu trữ chương trình PLC, thẻ nhớ lúc này sẽ thay thế vùng nhớ load Memory trong PLC, nó sẽ chứa chương trình chạy và mất thẻ nhớ là chương trình không chạy.
- Update Firmware: có thể dùng thẻ nhớ để update firmware cho CPU.



Hình 2.7: Thao tác cấu hình thẻ nhớ cho PLC S7-1200

2.1.3.6. Kiểm tra vùng nhớ PLC đã dùng

Để xem vùng nhớ của PLC chúng ta có thể vào phần mềm kết nối với PLC và kiểm tra hiện trạng của bộ nhớ như hình sau:



Hình 2.8: Kiểm tra hiện trạng các vùng nhớ của PLC S7-1200

2.1.3.7. Kiểu dữ liệu trong PLC S7-1200

Các kiểu dữ liệu được sử dụng để xác định cả kích thước của một phần tử dữ liệu cũng như cách thức mà dữ liệu được diễn dịch. Mỗi thông số lệnh hỗ trợ ít nhất một kiểu dữ liệu, và một số thông số còn hỗ trợ nhiều kiểu dữ liệu. Ta giữ con trỏ qua trường thông số của một lệnh để xem kiểu dữ liệu nào được hỗ trợ đối với một thông số đã cho.

Một thông số chính thức là bộ định danh trên một lệnh đánh dấu vị trí của dữ liệu được sử dụng bởi lệnh đó (ví dụ: ngõ vào IN1 của một lệnh ADD). Thông số thực tế là vị trí bộ nhớ hay hằng số chứa dữ liệu dùng cho lệnh (ví dụ %MD400 “Number_of_Widgets”). Kiểu dữ liệu của thông số thực tế được chỉ định bởi người dùng phải phù hợp với một trong các kiểu dữ liệu được hỗ trợ của thông số chính thức được chỉ định bởi lệnh.

Khi chỉ định một thông số thực tế, ta phải chỉ định cả một thẻ ghi (ký hiệu) hay một địa chỉ nhớ tuyệt đối. Các thẻ ghi có liên quan đến một tên ký hiệu (tên thẻ ghi) với một kiểu dữ liệu, một vùng nhớ, độ dịch chỉnh bộ nhớ, và dòng chú giải, và có thể được tạo ra cả trong trình soạn thảo thẻ ghi PLC hay trong trình soạn thảo Interface cho một khối (OB, FC, FB hay DB). Nếu nhập vào một địa chỉ tuyệt đối mà không có thẻ ghi liên quan, ta phải sử dụng một kích thước thích hợp phù hợp với kiểu dữ liệu được hỗ trợ, và một thẻ ghi mặc định sẽ được tạo ra dưới mục nhập vào.

Ta còn có thể nhập vào giá trị hằng số cho nhiều thông số ngõ vào. Bảng sau đây miêu tả các kiểu dữ liệu cơ bản được hỗ trợ, gồm các ví dụ của mục nhập cố định. Tất cả các kiểu ngoại trừ kiểu String đều có sẵn trong trình soạn thảo các thẻ ghi PLC và trong trình soạn thảo Interface của khối. Kiểu String chỉ có sẵn trong trình soạn thảo Interface của khối. Bảng sau đây xác định các kiểu dữ liệu cơ bản.

Bảng 2.4: Bảng các kiểu dữ liệu trong PLC S7-1200

Kiểu dữ liệu	Kích thước (bit)	Phạm vi	Các ví dụ mục nhập vào cố định
Bool	1	0 đến 1	TRUE, FALSE, 0, 1
Byte	8	16#00 đến 16#FF	16#12, 16#AB
Word	16	16#0000 đến 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 đến 16#FFFFFFFF	16#02468ACE
Char	8	16#00 đến 16#FF	“A”, “t”, “@”
SInt	16	-128 đến 127	123, -123
Int	16	-32768 đến 32767	123, -123
DInt	32	-2147483648 đến 2147483647	123, -123
USInt	8	0 đến 255	123
UInt	16	0 đến 65535	123
UDInt	32	0 đến 4294967295	123
Real	32	+/- 1.18 x 10 ⁻³⁸ đến +/- 3.40 x 10 ³⁸	123456, -3.4, -1.2E+12, 3.4E-3
LReal	32	+/- 2.23 x 10 ⁻³⁰⁸ đến +/- 1.79 x 10 ³⁰⁸	12345.123456789, -1.2E+40
Time	32	T#-24d_20h_31m_23s_648ms đến T#24d_20h_31m_23s_647ms Được lưu trữ dưới dạng : -2147483648 ms đến +2147483647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30s_45ms
String	Thay đổi	Các ký tự có kích thước 0 đến 254 <i>byte</i>	‘ABC’

Mặc dù không có sẵn như một kiểu dữ liệu, định dạng số BCD sau đây được hỗ trợ bởi các lệnh chuyển đổi.

Bảng 2.5: Bảng chuyển đổi định dạng số BCD

Định dạng	Kích thước (bit)	Phạm vi số	Các ví dụ mục nhập cố định
BCD16	16	-999 đến 999	123, -123
BCD32	32	-9999999 đến 9999999	1234567, -1234567

- Định dạng cho các số thực

Các số thực (hay số có dấu chấm động) được hiển thị dưới dạng số với độ chính xác đơn 32-bit (Real) hay số độ chính xác kép 64-bit (LReal) như được miêu tả trong tiêu chuẩn ANSI/IEEE 754-1985. Các số dấu chấm động với độ chính xác đơn là chính xác đến tối đa 6 chữ số có nghĩa và các số dấu chấm động với độ chính xác kép là chính xác đến tối đa 15 chữ số có nghĩa. Ta có thể chỉ định tối đa 6 chữ số có nghĩa (kiểu Real) hay 15 chữ số có nghĩa (LReal) khi nhập vào một hằng số dấu chấm động để duy trì độ chính xác.

Các tính toán có liên quan đến một chuỗi dài các giá trị gồm các số rất lớn hay rất nhỏ có thể tạo ra các kết quả không chính xác. Điều này có thể xuất hiện nếu các số hơn kém nhau đến 10 lũy thừa x , trong đó $x > 6$ (kiểu Real) hay $x > 15$ (kiểu LReal). Ví dụ đối với kiểu Real: $100\,000\,000 + 1 = 100\,000\,000$.

- Định dạng cho kiểu dữ liệu chuỗi

CPU hỗ trợ kiểu dữ liệu String dành để lưu trữ một tuần tự các ký tự byte đơn lẻ. Kiểu dữ liệu String chứa một bộ đếm ký tự tổng (số lượng các ký tự trong chuỗi) và bộ đếm ký tự hiện thời. Kiểu String cung cấp tối đa 256-byte cho việc lưu trữ kết quả đếm ký tự tổng tối đa (chiếm 1 byte), kết quả đếm ký tự hiện thời (chiếm 1 byte) và tối đa 254 ký tự, trong đó mỗi ký tự được lưu trữ trong 1 byte.

Ta có thể sử dụng các chuỗi trực kiện (hằng số) cho các thông số lệnh của kiểu IN bằng cách sử dụng các dấu nháy đơn. Ví dụ, một chuỗi gồm 3 ký tự „ABC“ có thể được sử dụng như là ngõ vào cho thông số IN của lệnh S_CONV. Ta còn có thể tạo ra các biến chuỗi bằng cách lựa chọn kiểu dữ liệu “String” trong trình soạn thảo giao diện khối cho OB, FC, FB và DB. Ta không thể tạo ra một chuỗi trong trình soạn thảo thẻ ghi của PLC. Ta có thể xác định kích thước chuỗi tối đa tính theo byte khi khai báo chuỗi, ví dụ, “MyString[10]” sẽ xác định kích thước tối đa là 10 byte cho MyString. Nếu không bao gồm dấu ngoặc vuông với bộ chỉ thị kích thước lớn nhất, giá trị 254 sẽ được cho.

Ví dụ sau đây xác định một chuỗi với giá trị đếm ký tự tối đa bằng 10 và giá trị đếm ký tự hiện thời bằng 3. Điều này có nghĩa là hiện thời chuỗi chứa 3 ký tự 1 byte, nhưng có thể được mở rộng để chứa tối đa 10 ký tự 1 byte.

Bảng 2.6: Ví dụ định dạng kiểu dữ liệu chuỗi

Bộ đếm ký tự tổng	Bộ đếm ký tự hiện thời	Ký tự 1	Ký tự 2	Ký tự 3	...	Ký tự 10
10	3	“C” (16#43)	“A” (16#41)	“T” (16#54)	...	-
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	...	Byte 11

- Các mảng dữ liệu

Ta có thể tạo ra một mảng chứa nhiều phần tử của một kiểu cơ bản. Các mảng có thể được tạo ra trong trình soạn thảo giao diện khối dành cho OB, FC, FB và DB. Ta không thể tạo ra một mảng trong trình soạn thảo thẻ ghi PLC.

Để tạo ra một mảng từ trình soạn thảo giao diện khối, ta chọn kiểu dữ liệu “Array [lo...hi] of type”, sau đó chỉnh sửa “lo”, “hi” và “type” như sau:

- lo – chỉ số bắt đầu (thấp nhất) của mảng
- hi – chỉ số kết thúc (cao nhất) của mảng
- type – một trong các kiểu dữ liệu cơ bản, như là Bool, SInt, UDInt

Các chỉ số âm đều được hỗ trợ. Ta có thể đặt tên mảng trong cột Name của trình soạn thảo giao diện khối. Bảng sau đây cho thấy các ví dụ của các mảng như chúng có thể xuất hiện trong trình soạn thảo giao diện khối.

Bảng 2.7: Ví dụ các mảng dữ liệu

Tên	Kiểu dữ liệu	Chú giải
My_Bits	Array [1 .. 10] đối với kiểu Bool	Mảng này chứa 10 giá trị Boolean
My_Data	Array [-5 .. 5] đối với kiểu SInt	Mảng này chứa 11 giá trị SInt, bao gồm chỉ mục 0

Ta tham chiếu các phần tử của mảng trong chương trình bằng cách sử dụng cú pháp sau:

- Array_name[i], trong đó i là chỉ mục mong muốn.

Các ví dụ có thể xuất hiện trong trình soạn thảo chương trình như một ngõ vào thông số:

- #My_Bits[3] – tham chiếu bit thứ 3 của mảng “My_Bits”
- #My_Data[-2] – tham chiếu bit thứ 4 của mảng “My_Data”

Ký hiệu # được chèn vào một cách tự động bởi trình soạn thảo chương trình.

- **Kiểu dữ liệu DTL (Data and Time Long)**

Kiểu dữ liệu DTL là một cấu trúc gồm 12-byte lưu trữ thông tin trên ngày và giờ trong một cấu trúc được xác định trước. Ta xác có thể xác định một DTL cả trong bộ nhớ tạm Temp của khối hay trong một DB.

Bảng 2.8: Kiểu dữ liệu định dạng DTL

Độ dài (byte)	Định dạng	Phạm vi giá trị	Ví dụ về ngõ vào giá trị
12	Đếm thời gian và lịch (Year:Month:Hour:Minute:Second.Nanosecond)	Tối thiểu: DTL#1970-01-01:00:00:00.0 Tối đa: DTL#2554-12-31:23:59:59.999999999	DTL#2008-12-6 20:30:20.250

Mỗi phần của DTL chứa một kiểu dữ liệu khác nhau và phạm vi của các giá trị. Kiểu dữ liệu của một giá trị được định trước phải phù hợp với kiểu dữ liệu của các thành phần tương ứng.

Bảng 2.9: Kiểu dữ liệu trong DTL

Byte	Thành phần	Kiểu dữ liệu	Phạm vi giá trị
0	Year	UInt	1970 đến 2554
1			
2	Month	UInt	1 đến 12
3	Day	UInt	1 đến 31
4	Day of week	UInt	1 (Chủ nhật) đến 2 (thứ bảy) Ngày trong tuần không cần chú ý trong mục nhập giá trị
5	Hour	UInt	0 đến 23
6	Minute	UInt	0 đến 59
7	Second	UInt	0 đến 59
8	Nanosecond	UDInt	0 đến 999 999 99
9			
10			

11			
----	--	--	--

2.2. Cơ sở lý thuyết về lập trình PLC S7-1200

2.2.1. Phương pháp lập trình

Lập trình cho PLC Siemens S7-1200 người dùng có thể sử dụng các ngôn ngữ lập trình như sau:

Phương pháp lập trình hình thang (LAD – Ladder Logic).

Phương pháp khối hàm (FBD – Function Block Diagram).

Phương pháp theo cấu trúc (SCL – Structured Control Language)

Phương pháp liệt kê câu lệnh (STL – Statement List).

Phương pháp lập trình hướng đối tượng (Graph).

- **LAD:** là một trong những ngôn ngữ lập trình được sử dụng phổ biến nhất cho PLC. Nó được sử dụng để biểu diễn các tín hiệu đầu vào và đầu ra của PLC dưới dạng các mạch điện tử dạng thanh ngang. LAD là một ngôn ngữ lập trình đồ họa, dễ hiểu và thân thiện với người dùng.

- **FBD:** được sử dụng để biểu diễn các hàm logic hoặc toán học. Nó được tạo ra từ các khối chức năng, được kết nối với nhau bằng các đường dẫn, tạo thành các chương trình điều khiển tương đối phức tạp.

- **SCL:** được sử dụng để viết các chương trình phức tạp hơn, với các công cụ lập trình phức tạp hơn so với LAD và FBD. Nó cho phép các nhà phát triển viết các hàm toán học, điều khiển bộ nhớ, chuyển đổi dữ liệu và xử lý các chuỗi dữ liệu.

- **STL:** là ngôn ngữ lập trình bậc thấp, thường được sử dụng để tối ưu hóa hiệu suất chương trình, đặc biệt là trong các ứng dụng yêu cầu thời gian xử lý nhanh.

- **Graph:** là ngôn ngữ lập trình hướng đối tượng được sử dụng để thiết kế các chương trình phức tạp, cung cấp khả năng tái sử dụng mã và tăng tính linh hoạt trong quản lý các chương trình.

2.2.2. Phần mềm lập trình

2.2.2.1. Giới thiệu về phần mềm lập trình cho PLC S7-1200

TIA Portal (Totally Integrated Automation Portal) là một phần mềm tổng hợp của nhiều phần mềm điều hành quản lý tự động hóa, vận hành điện của hệ thống. Có

thể hiệu, TIA Portal là phần mềm tự động hóa đầu tiên, có sử dụng chung 1 môi trường, nền tảng để thực hiện các tác vụ, điều khiển hệ thống.

TIA Portal được phát triển vào năm 1996 bởi các kỹ sư của Siemens, nó cho phép người dùng phát triển và viết các phần mềm quản lý riêng lẻ một cách nhanh chóng, trên 1 nền tảng thống nhất. Giải pháp giảm thiểu thời gian tích hợp các ứng dụng riêng biệt để thống nhất tạo hệ thống.

TIA Portal tích hợp tự động toàn diện là phần mềm cơ sở cho tất cả các phần mềm khác phát triển: Lập trình, tích hợp cấu hình thiết bị trong dải sản phẩm. Đặc điểm TIA Portal cho phép các phần mềm chia sẻ cùng 1 cơ sở dữ liệu, tạo nên tính thống nhất, toàn vẹn cho hệ thống ứng dụng quản lý, vận hành.

TIA Portal tạo môi trường dễ dàng để lập trình thực hiện các thao tác:

- Thiết kế giao diện kéo thả thông tin dễ dàng, với ngôn ngữ hỗ trợ đa dạng.
- Quản lý phân quyền User, Code, Project tổng quát.
- Thực hiện go online và Diagnostic cho tất cả các thiết bị trong project để xác định bệnh, lỗi hệ thống.
- Tích hợp mô phỏng hệ thống.
- Dễ dàng thiết lập cấu hình và liên kết giữa các thiết bị Siemens.

Hiện tại phần mềm TIA Portal có nhiều phiên bản như TIA Portal V14, TIA Portal V15, TIA Portal V16 và mới nhất là TIA Portal V17. Tùy theo nhu cầu sử dụng mà người dùng sẽ lựa chọn cài đặt TIA portal phiên bản tương ứng.

2.2.2.2. Các thành phần trong bộ cài TIA Portal

Phần mềm TIA Portal được Siemens phát triển với nhiều thành phần giúp người dùng quản lý, lập trình PLC, HMI (Human Machine Interface) hiệu quả. Các thành phần có trong bộ TIA Portal:

- Simatic Step 7 professional và Simatic step 7 PLCSIM: Giải pháp lập trình và mô phỏng PLC S7-300, S7-400, Simatic S7-1200, Simatic S7-1500...
- Simatic WinCC Professional: Được dùng để lập trình màn hình HMI và giao diện SCADA (Supervisory Control and Data Acquisition).
- Simatic Start Driver: Được lập trình cấu hình Siemens.
- Sirius và Simocode: Thiết lập cấu hình và chuẩn đoán lỗi linh hoạt.

- Điều khiển chuyển động đơn trục và đa trục với hỗ trợ Scout TIA. Thư viện Simatic Robot đầy đủ dữ liệu cho phép người dùng thiết lập cấu hình và hệ thống nhanh chóng.

2.2.2.3. Bảo mật lập trình PLC với TIA Portal

Phần mềm TIA Portal cung cấp cho người dùng nhiều sự lựa chọn để bảo mật cho dự án của mình:

- **Full access:** Ứng với khối bảo mật mà ai cũng có thể đọc và viết mà không cần mật mã (password).
- **Read Access:** Bảo mật phần viết cho PLC, cần có password. HMI và SCADA hay user đọc được chương trình không cần password.
- **HMI access:** Bảo mật phần đọc và ghi của PLC cần có password. HMI và SCADA đọc không cần password.
- **No Access:** Tất cả các ứng dụng truy xuất vào PLC đều cần password.

2.2.2.4. Tập lệnh trong TIA Portal

- Các nhóm lệnh cơ bản (Basic instructions)

▼ Basic instructions			
Name	Description	Version	
▶ General			
▶ Bit logic operations		V1.0	
▶ Timer operations		V1.0	
▶ Counter operations		V1.0	
▶ Comparator operations			
▶ Math functions		V1.0	
▶ Move operations		V2.5	
▶ Conversion operations			
▶ Program control operati...		V1.1	
▶ Word logic operations		V1.4	
▶ Shift and rotate			

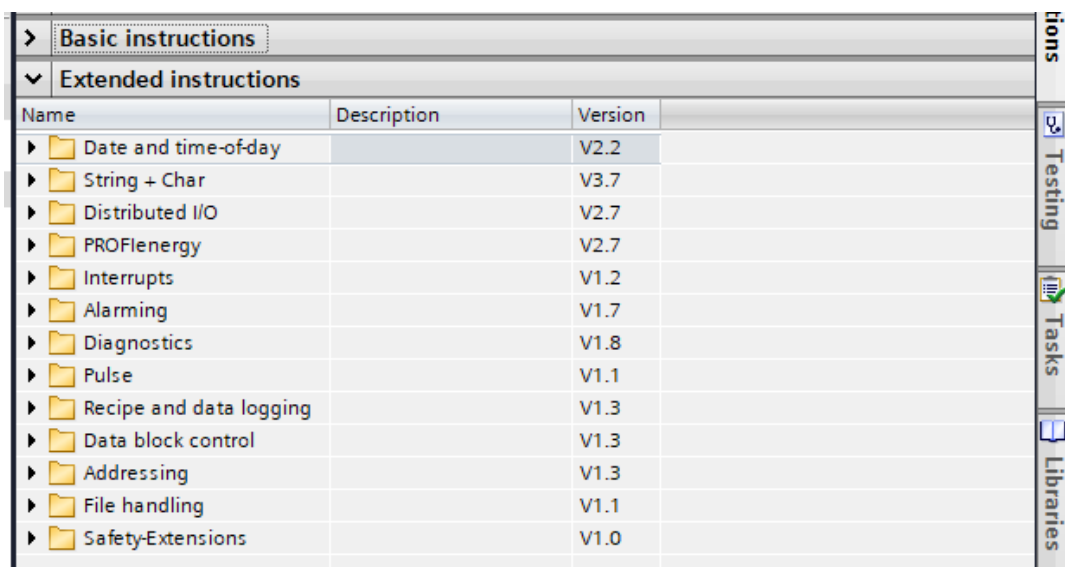
Hình 2.9: Các nhóm lệnh cơ bản (Basic instructions)

Đây là nhóm lệnh cơ bản nhất nhưng lại được dùng nhiều nhất, có thể thấy các lệnh về Logic, Counter, lệnh so sánh, lệnh tính toán, ...

- Tập lệnh mở rộng (Extended instructions)

Đây là nhóm lệnh chuyên sâu hơn nhưng ít sử dụng hơn như lệnh xử lý ngắt, thời

gian thực, xử lý chuỗi và kí tự,

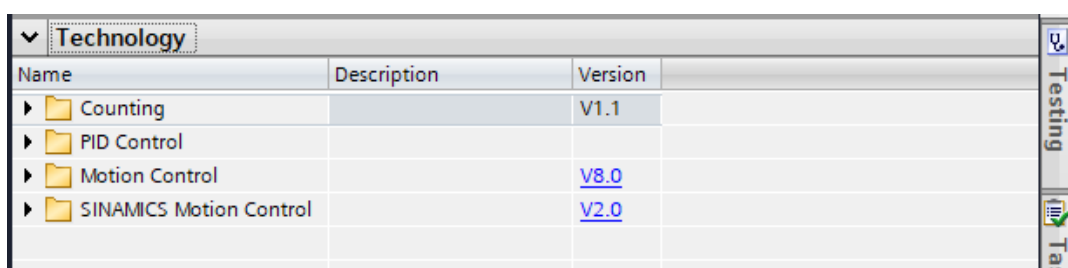


Name	Description	Version
▶ Date and time-of-day		V2.2
▶ String + Char		V3.7
▶ Distributed I/O		V2.7
▶ PROFenergy		V2.7
▶ Interrupts		V1.2
▶ Alarming		V1.7
▶ Diagnostics		V1.8
▶ Pulse		V1.1
▶ Recipe and data logging		V1.3
▶ Data block control		V1.3
▶ Addressing		V1.3
▶ File handling		V1.1
▶ Safety-Extensions		V1.0

Hình 2.10: Các nhóm lệnh mở rộng (Extended instructions)

- Tập lệnh ứng dụng (Technology)

Đây là nhóm lệnh sử dụng cho các ứng dụng đặc biệt của PLC, bao gồm bộ đếm tốc độ cao (SHC), bộ điều khiển PID, phát xung điều khiển vị trí Servo, ...

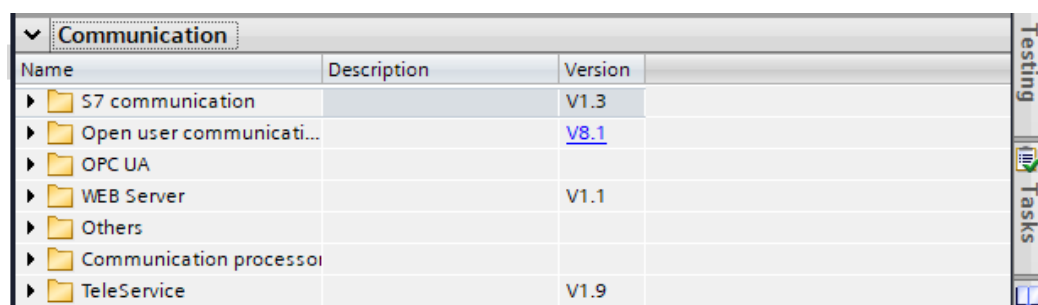


Name	Description	Version
▶ Counting		V1.1
▶ PID Control		
▶ Motion Control		V8.0
▶ SINAMICS Motion Control		V2.0

Hình 2.11: Các nhóm lệnh ứng dụng (Technology)

- Tập lệnh xử lý truyền thông (Communication)

Đây là nhóm lệnh sử dụng lập trình cho ứng dụng truyền thông giữa PLC với PLC, với biến tần, máy tính, ... ví dụ lệnh truyền thông Ethernet, Modbus, Profibus, ... trong mục S7 Communication, sử dụng Web Server, ...



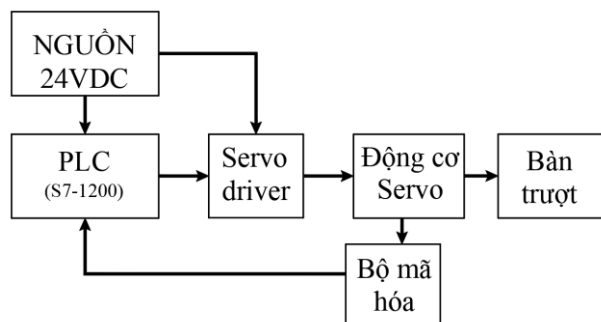
Name	Description	Version
▶ S7 communication		V1.3
▶ Open user communicati...		V8.1
▶ OPC UA		
▶ WEB Server		V1.1
▶ Others		
▶ Communication processor		
▶ TeleService		V1.9

Hình 2.12: Các nhóm lệnh xử lý truyền thông

CHƯƠNG 3: THIẾT KẾ VÀ CHẾ TẠO MÔ HÌNH

3.1. Thiết kế hệ thống chính của mô hình

Mô hình được thiết kế theo tiêu chí chú trọng đơn giản nhưng vẫn đảm bảo về mặt kỹ thuật nhằm giảm chi phí. Hệ thống mô hình hoàn thiện gồm các khâu như Hình 4.1 sau:



Hình 3.1: Hệ thống các khâu chính của mô hình

- Bàn trượt (Sliding table): Thực hiện chuyển đổi vòng quay sang chuyển động thẳng, một bộ trượt điện được thực hiện nhiệm vụ này. Thông số của bộ chuyển động trượt điện được thể hiện như tại Bảng 3.1.

Bảng 3.1: Thông số kỹ thuật của bộ trượt điện

Thông số kỹ thuật	Chi tiết
Hành trình làm việc	200 mm
Kích thước	26
Độ rơ lý thuyết	$\leq 20 \mu\text{m}$
Đường kính trục vít	8 mm
Bước ren trục vít	6 mm/U
Hằng số cấp liệu	6 mm/U
Vị trí lắp đặt	Tùy chọn
Hướng dẫn	Ồ bi tuần hoàn
Thiết kế	Trục tuyến tính điện cơ với vít me bi
Tính năng tham chiếu	Công tắc tham chiếu
Loại trục vít	Vít me bi

- Động cơ servo (Servo motor): Kiểm soát vị trí bàn trượt bởi động cơ servo của hãng Fastech, thông số động cơ servo như tại Bảng 3.2.

Bảng 3.2: Các thông số chính của động cơ servo

Thông số kỹ thuật	Chi tiết
Model Động cơ	EzM-42XL-A-D
Model Driver	EzS-PD-42XL-A-D
Điện áp đầu vào	24VDC $\pm$ 10%
Phương pháp điều khiển	Điều khiển vòng kín với MCU 32-bit
Dòng điện tiêu thụ	Tối đa 500mA (ngoại trừ dòng điện động cơ)
Nhiệt độ môi trường	Khi sử dụng: 0~50°C
Khả năng chịu rung	0.5g
Tốc độ quay	0~3,000 rpm
Tần số xung đầu vào tối đa	500kHz (Duty 50%)
Tín hiệu đầu vào	Xung lệnh vị trí, Servo On/Off, Reset cảnh báo (Photocoupler Input)
Tín hiệu đầu ra	Vị trí, Cảnh báo (Photocoupler Output), Tín hiệu encoder (A+, A-, B+, B-, Z+, Z-, Line Driver Output, 26C31 hoặc tương đương)

▪ Driver động cơ servo (Driver servo): Bộ điều khiển đi cùng động cơ servo, thông số kỹ thuật được thể hiện như tại Bảng 3.3.

Bảng 3.3: Bảng thông số kỹ thuật driver servo

Thông số kỹ thuật	Giá trị
Model Driver	EzS-PD-42XL-A-D
Điện áp đầu vào	24VDC $\pm$ 10%
Phương pháp điều khiển	Điều khiển vòng kín với DSP 32-bit
Dòng điện đầu ra tối đa	3.5A
Tần số xung đầu vào tối đa	500 kHz
Độ phân giải encoder	10,000 xung/vòng (có thể tùy chọn độ phân giải khác)
Kích thước	120mm x 75mm x 30mm
Trọng lượng	300g
Nhiệt độ hoạt động	0°C đến 50°C
Độ ẩm hoạt động	35% đến 85% RH (không ngưng tụ)
Chức năng bảo vệ	Bảo vệ quá dòng, quá áp, quá nhiệt, và lỗi encoder
Giao diện điều khiển	Tín hiệu xung và hướng, hoặc giao thức truyền thông RS-485 (tùy chọn)

▪ Bộ mã hóa vòng quay (Encoder): Khâu mã hóa vòng quay, sử dụng encoder để chuyển hóa chuyển động quay sang vị trí nhằm phản hồi và kiểm tra vị trí của bàn trượt. Thông số bộ encoder như tại Bảng 3.4.

Bảng 3.4: Thông số kỹ thuật của bộ mã hóa vòng quay

Thông số kỹ thuật	Chi tiết
Mã sản phẩm	LPD3806-600BM
Loại encoder	Rotary Encoder
Số xung	600 pulses per revolution (PPR)
Loại đầu ra	NPN (Open Collector)
Nguồn cung cấp	5V - 24V DC
Dòng điện tiêu thụ	$\leq 80\text{mA}$
Tín hiệu đầu ra	A, B, Z (Z là tín hiệu zero)
Tần số tối đa	300kHz
Khoảng cách dây nối	$\leq 30\text{m}$
Đường kính trục	6mm
Tốc độ quay tối đa	5000 RPM
Chất liệu vỏ	Nhôm
Nhiệt độ hoạt động	-10°C đến +70°C
Độ ẩm hoạt động	35% - 85% RH
Khối lượng	Khoảng 200g
Cấp bảo vệ	IP54
Ứng dụng	Điều khiển công nghiệp, tự động hóa, đo lường chuyển động

▪ Nguồn cấp 24VDC: Nguồn cấp cho hệ thống mô hình, sử dụng bộ nguồn DC 24V

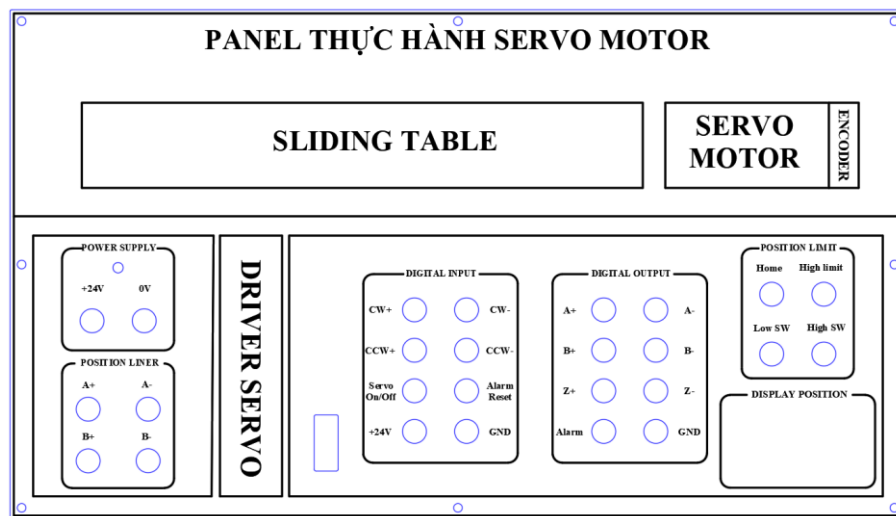
Bảng 3.5: Bảng thông số kỹ thuật bộ nguồn DC 24V

Thông số	Giá trị
Model	S8VK-C12024
Điện áp đầu ra	24V DC
Dòng điện đầu ra tối đa	5A
Công suất tối đa	120W

Điện áp đầu vào	100-240V AC hoặc 90-350V DC
Tần số đầu vào	50/60 Hz
Hiệu suất	Khoảng 85-90%
Phạm vi nhiệt độ hoạt động	-25°C đến +60°C
Bảo vệ	Quá tải, ngắn mạch, quá nhiệt
Loại lắp đặt	Lắp trên thanh DIN
Chuẩn an toàn	UL508, EN50178, EN61204-3, EN55011

3.2. Thiết kế bản đầu nối và lắp đặt bàn trượt

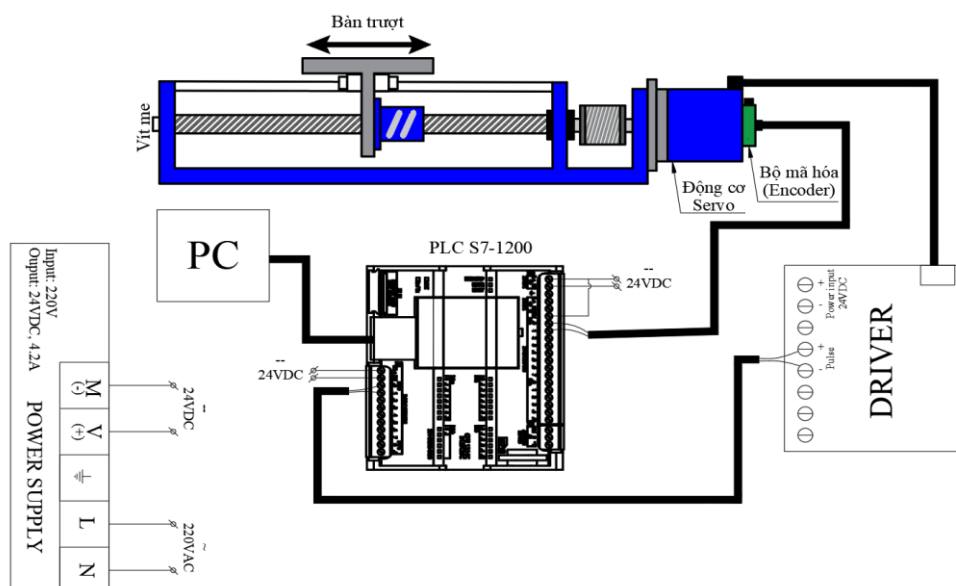
- Kích thước tổng của mô hình (kích thước tính theo mm)



Hình 3.2: Kích thước tổng thể mô hình

3.3. Thiết kế sơ đồ nguyên lý nối dây thiết bị mô hình

- Sơ đồ nguyên lý nối dây tổng thể của mô hình được thể hiện như tại Hình 3.3.



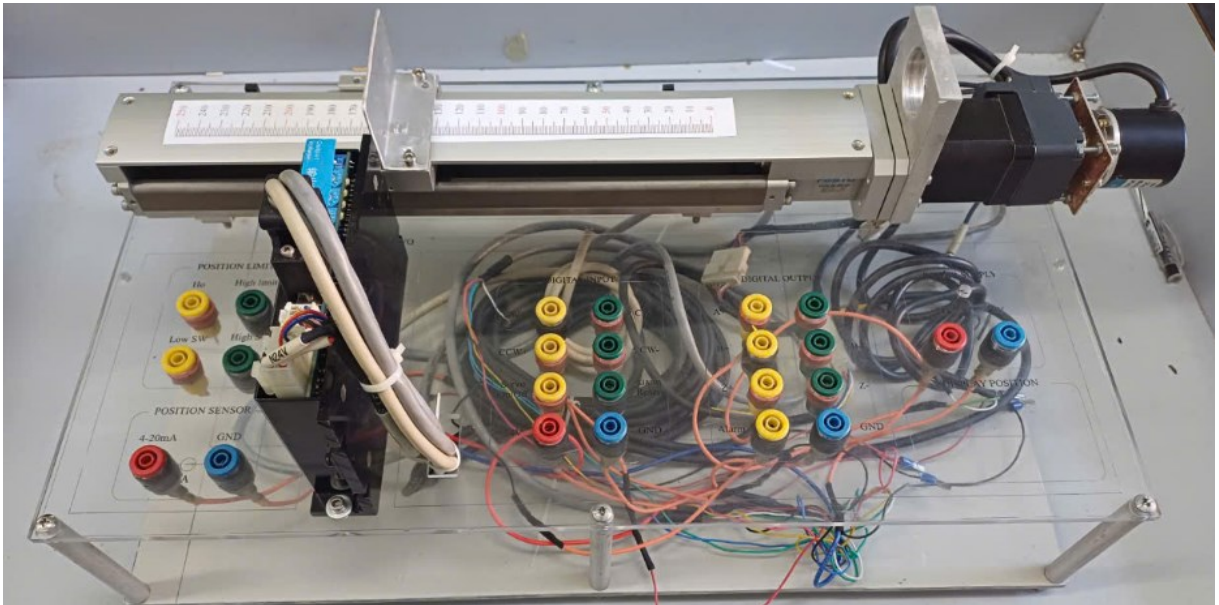
Hình 3.3: Sơ đồ nguyên lý nối dây của hệ thống mô hình

3.4. Hoàn thiện lắp ráp mô hình

Hoàn thiện lắp ráp mô hình với hai nội dung chính sau:

- Thực hiện gia công và gắn kết các chi tiết cơ khí bằng đai ốc
- Lắp đặt các thiết bị điện, các jack cắm theo sơ đồ

Hình ảnh mô hình lắp ráp hoàn thiện như tại Hình 3.4.



Hình 3.4: Mô hình hoàn thiện

CHƯƠNG 4: XÂY DỰNG CÁC BÀI TẬP THỰC HÀNH

Mô hình có thiết kế đơn giản nhằm mục đích trang bị một số kiến thức về điều khiển vị trí cơ bản cho sinh viên thông qua một số bài thực hành. Giúp sinh viên, học viên tiếp cận, làm quen và thực nghiệm về điều khiển vị trí bàn trượt.

4.1. Bài thực hành 1

4.1.1. Nội dung bài thực hành

Xây dựng chương trình điều khiển trên PLC S7-1200 để điều khiển mô hình điều khiển vị trí bàn trượt với yêu cầu như sau:

- Một nút nhấn Khởi động, một nút nhấn Tạm dừng
- Một nút nhấn để xác nhận vị trí gốc của bàn trượt
- Một nút nhấn chạy tiến và một nút nhấn chạy lùi bàn trượt.
- Hiện thị giá trị vị trí hiện tại, thiết đặt giá trị tốc độ trên giao diện WINCC phần mềm TIA Portal.

4.1.2. Mục tiêu kiến thức đạt được

Người học xây dựng chương trình phát xung điều khiển động cơ servo

Thiết lập các giá trị để điều khiển vị trí bàn trượt.

Thao tác kết nối dây phần cứng, cài đặt phần mềm điều khiển, hiển thị giá trị vị trí trên phần mềm điều khiển.

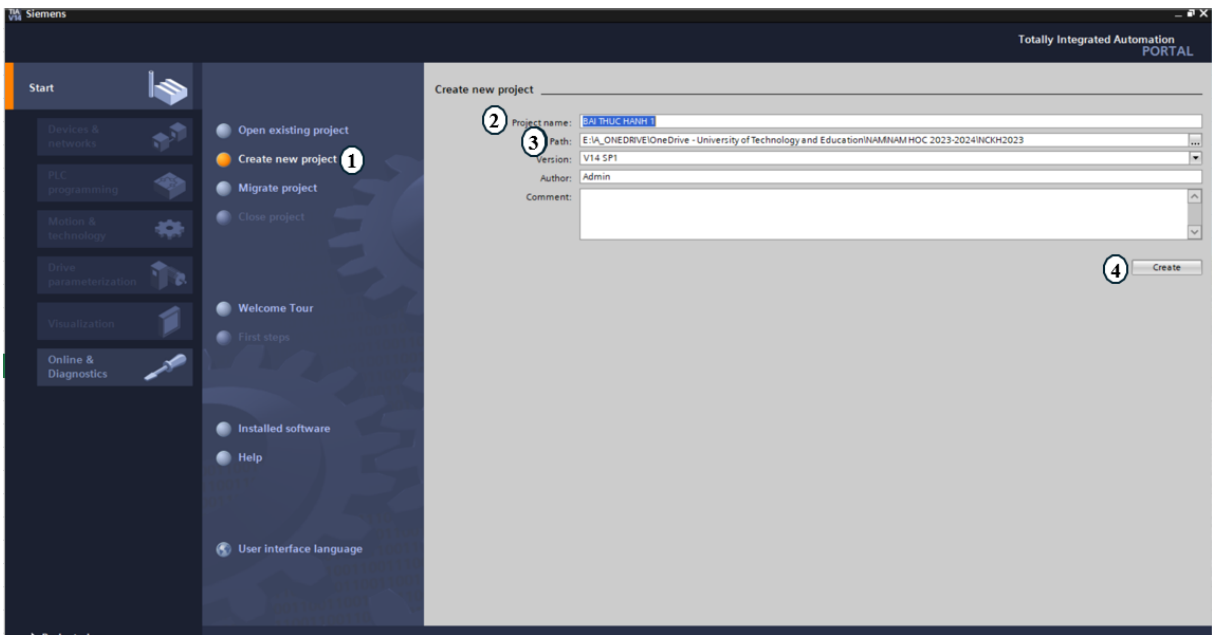
4.1.3. Hướng dẫn thực hiện

Bước 1. Kết nối dây cho hệ thống (xem Hình 3.3)

- 1) Cấp nguồn 220V cho mô đun nguồn 24VDC
- 2) Cấp nguồn cho mô đun PLC S7-1200
- 3) Kết nối dây giữa mô đun nguồn 24VDC và mô đun PLC với mô hình.

Bước 2. Thực hiện khởi tạo một dự án trên phần mềm TIA Portal

Mở phần mềm TIA Portal sau khi cài xong ta được giao diện như Hình 4.1 sau:

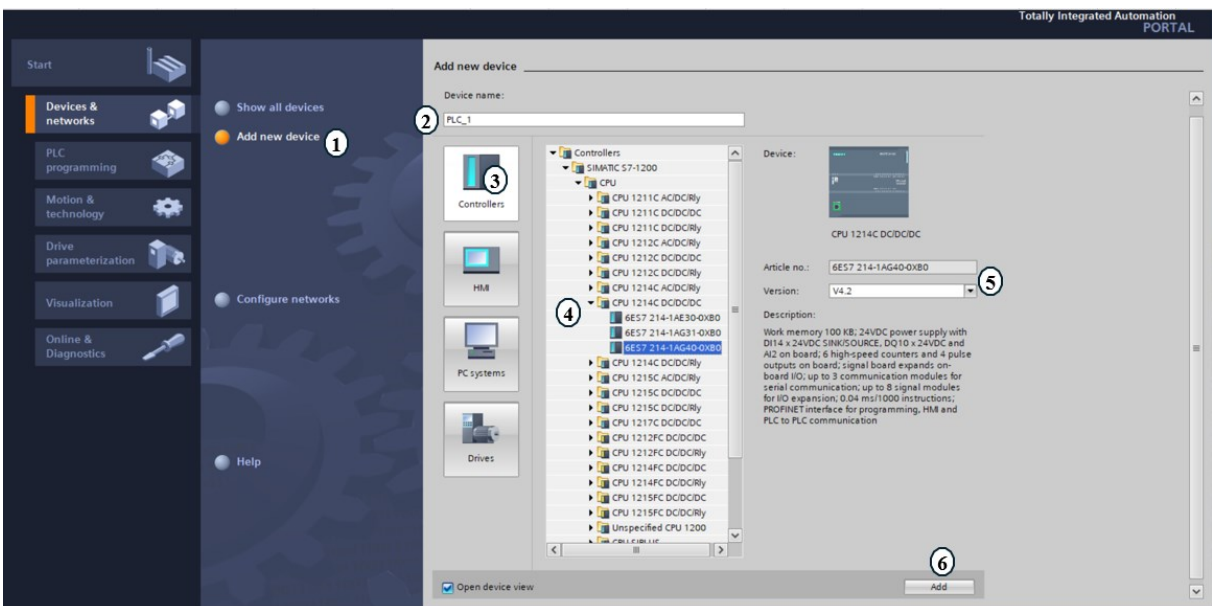


Hình 4.1: Giao diện phần mềm khi được khởi động

- ① Tạo một dự án mới
- ② Đặt tên cho dự án
- ③ Vị trí lưu dự án
- ④ Tiến hành tạo dự án

Bước 3. Thực hiện khởi tạo một bộ lập trình mới

Thực hiện các bước tạo một bộ lập trình mới như Hình 4.2 sau:



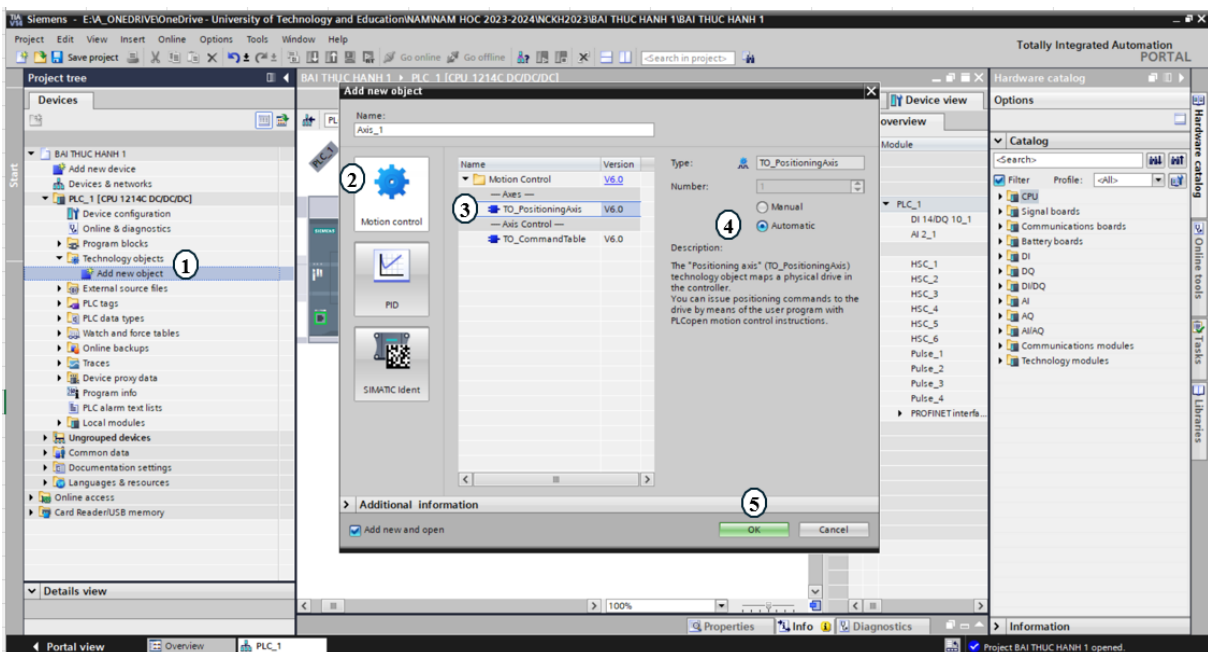
Hình 4.2: Các bước tạo một bộ lập trình S7-1200

- ① Tạo một thiết bị mới

- ② Đặt tên cho bộ lập trình
- ③ Chọn bộ lập trình
- ④ Chọn bộ lập trình CPU1214C DCDCDC
- ⑤ Chọn version cho bộ lập trình
- ⑥ Add bộ lập trình

Bước 4. Thực hiện khởi tạo khởi tạo bộ điều khiển cho động cơ servo

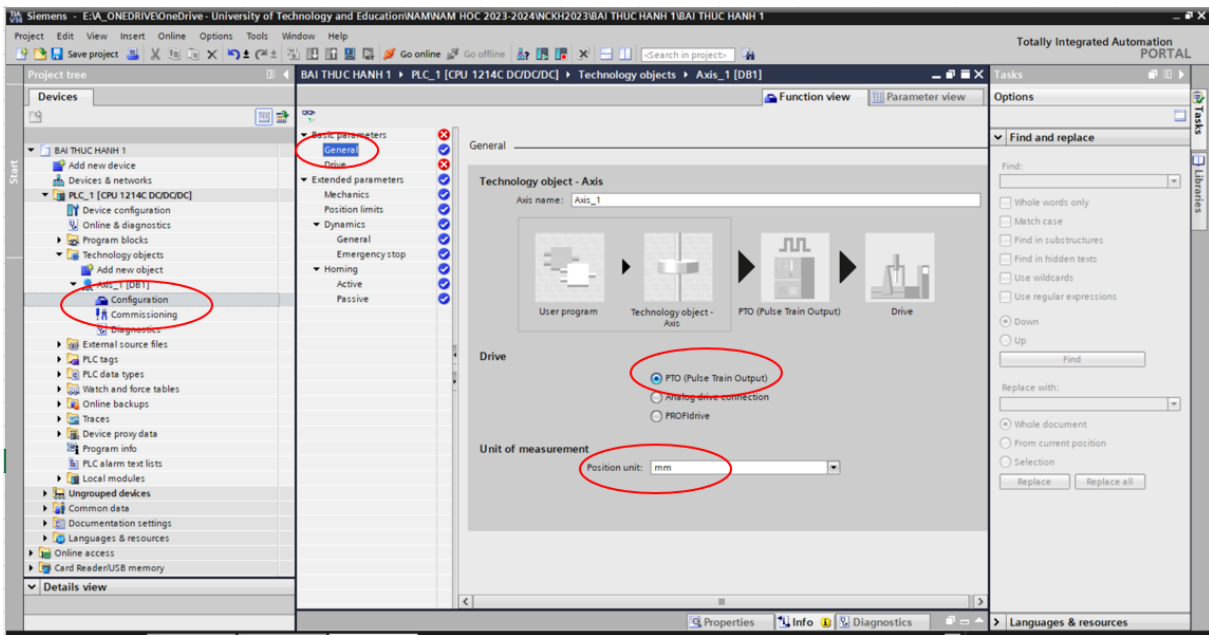
- Sau khi các bạn tạo 1 dự án mới (project) mới sau đó add cpu và chọn đúng version để cấu hình phần cứng cho chính xác. Tiếp theo bấm vào mục: Technology object rồi chọn Add new object như Hình 4.3 sau:



Hình 4.3: Khởi tạo bộ điều khiển vị trí

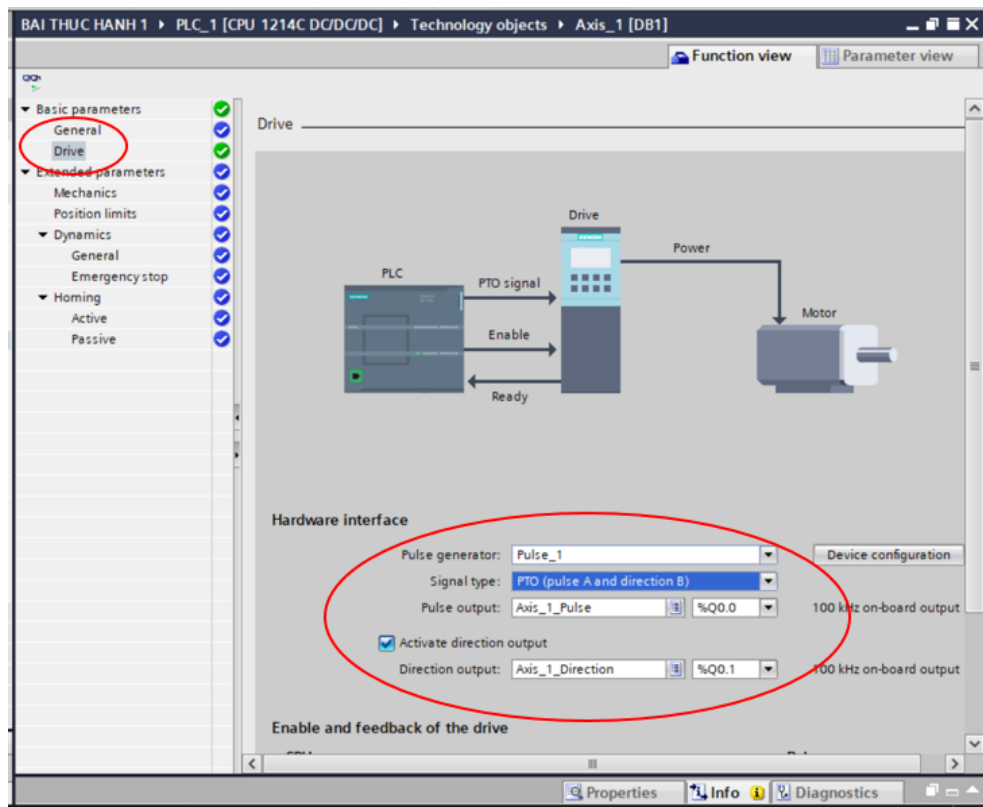
- ① Chọn mục Technology object rồi chọn Add new object.
- ② Ở cửa sổ mới mở ra chọn Motion control.
- ③ Chọn mục TO_PositioningAxis
- ④ Kích chọn Automatic
- ⑤ Bấm OK

- Sau đó thì ở phần Technology Objects đã xuất hiện thêm mục Axis_1[DB1], thực hiện bấm vào Configuration rồi chọn General. Tiếp tục chọn PTO(Pulse Train Output) và đơn vị đo milimet ở phần Positon Unit như tại Hình 4.4 sau:



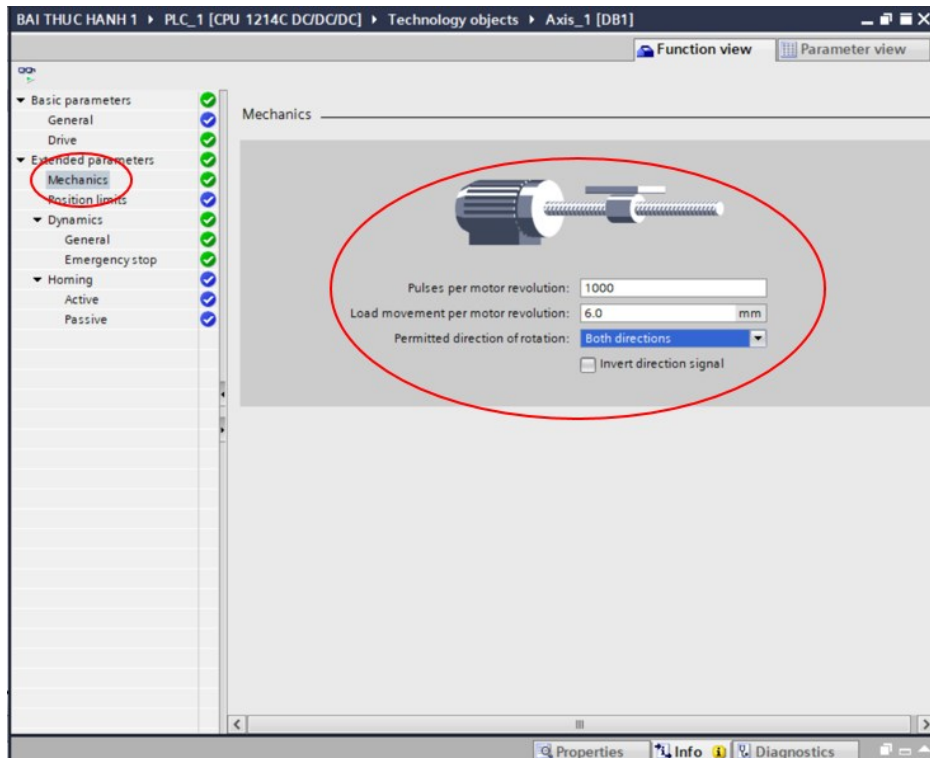
Hình 4.4: Cài đặt cấu hình cho bộ điều khiển

▪ Tiếp tục chuyển vào mục Drive để cấu hình chân phát xung và hướng chạy của động cơ servo, cấu hình như Hình 4.5 bên dưới thì Q0.0 sẽ là ngõ phát xung, còn chọn chiều là chân Q0.1.



Hình 4.5: Cấu hình tại mục Drive của bộ điều khiển

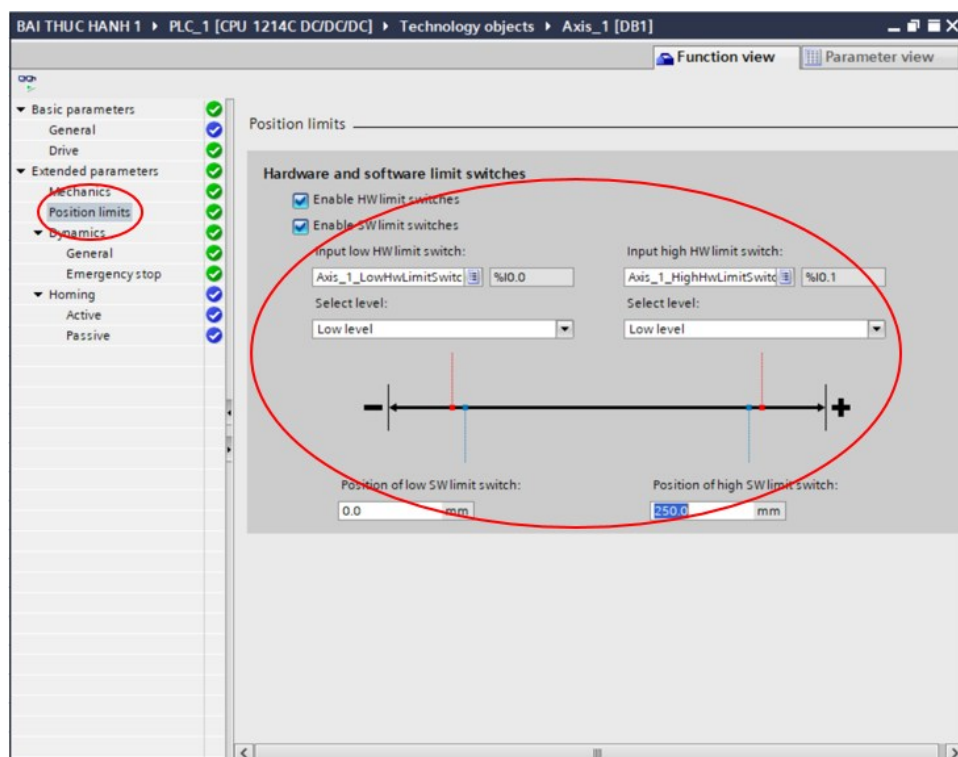
▪ Tiếp theo bấm vào mục Mechanics để cấu hình độ phân giải của servo và vít me. Thông số như Hình 4.6, có nghĩa là phát 1000 xung thì động cơ quay 1 vòng, 1 vòng thì vít me sẽ đi được 6mm.



Hình 4.6: Cấu hình tại mục Mechanics của bộ điều khiển

Sau khi cấu hình 4 phần nêu trên về cơ bản đã hoàn thành, chúng ta có thể cấu hình thêm cho những phần còn lại như sau:

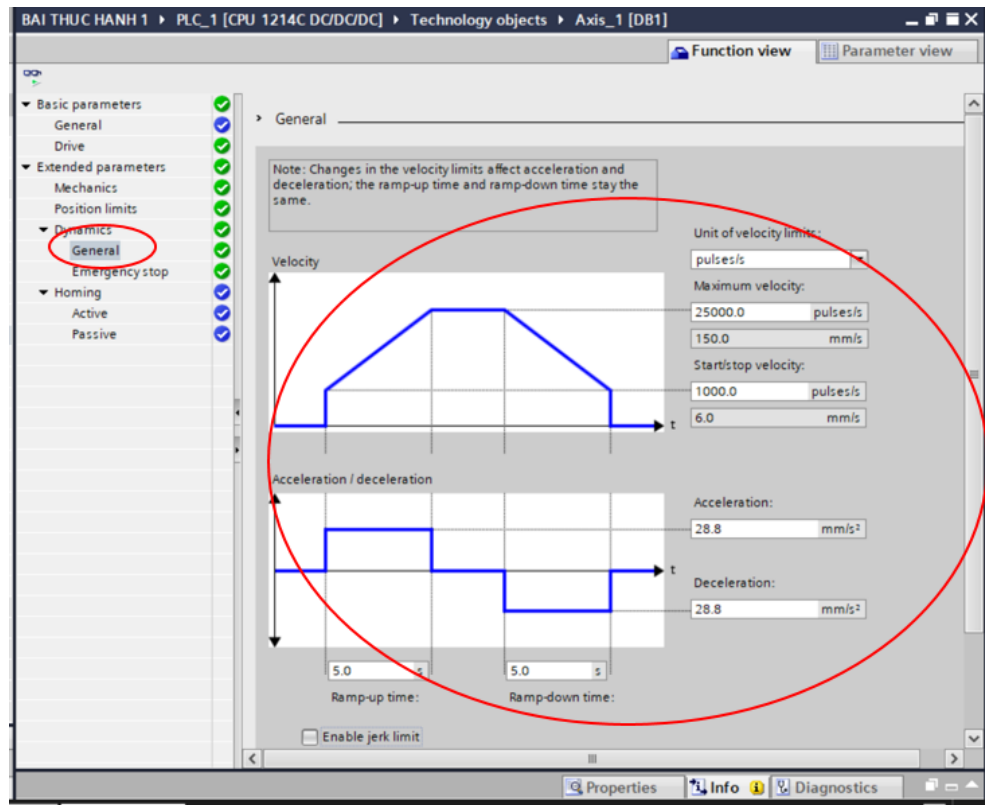
- Position limits: phân khai báo giới hạn hành trình như tại Hình 4.7 sau:



Hình 4.7: Cấu hình tại mục Position limits

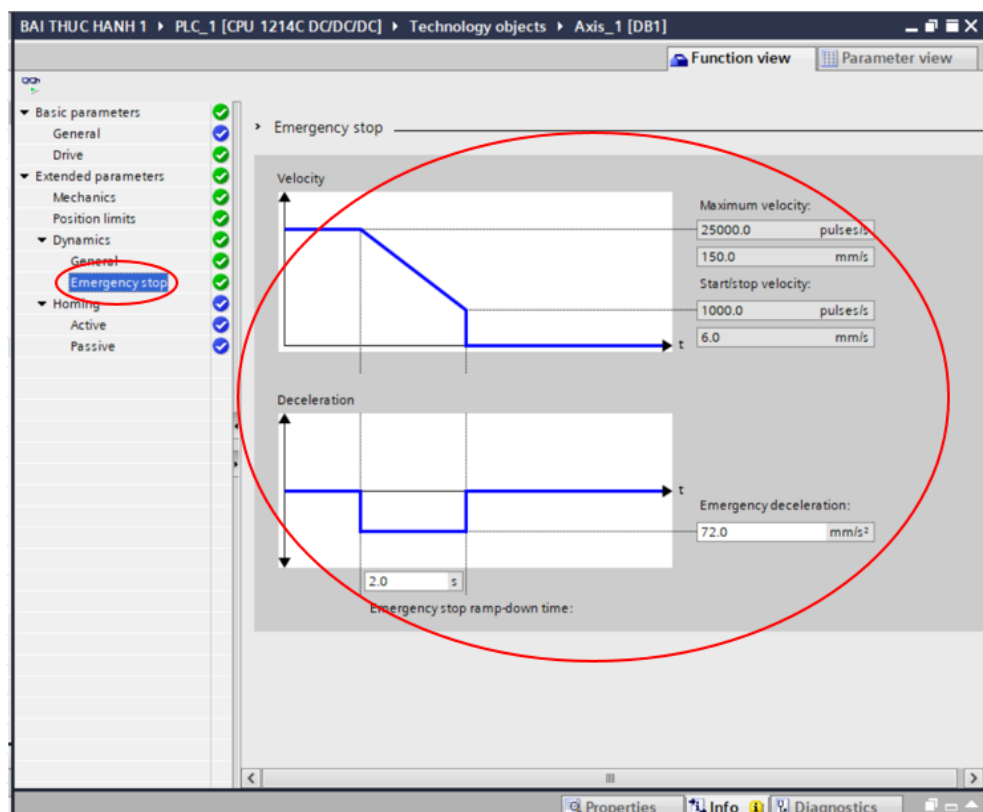
- Dynamics- General: cấu hình phần tốc độ cũng như thời gian tăng tốc và giảm

tốc như tại Hình 4.8.



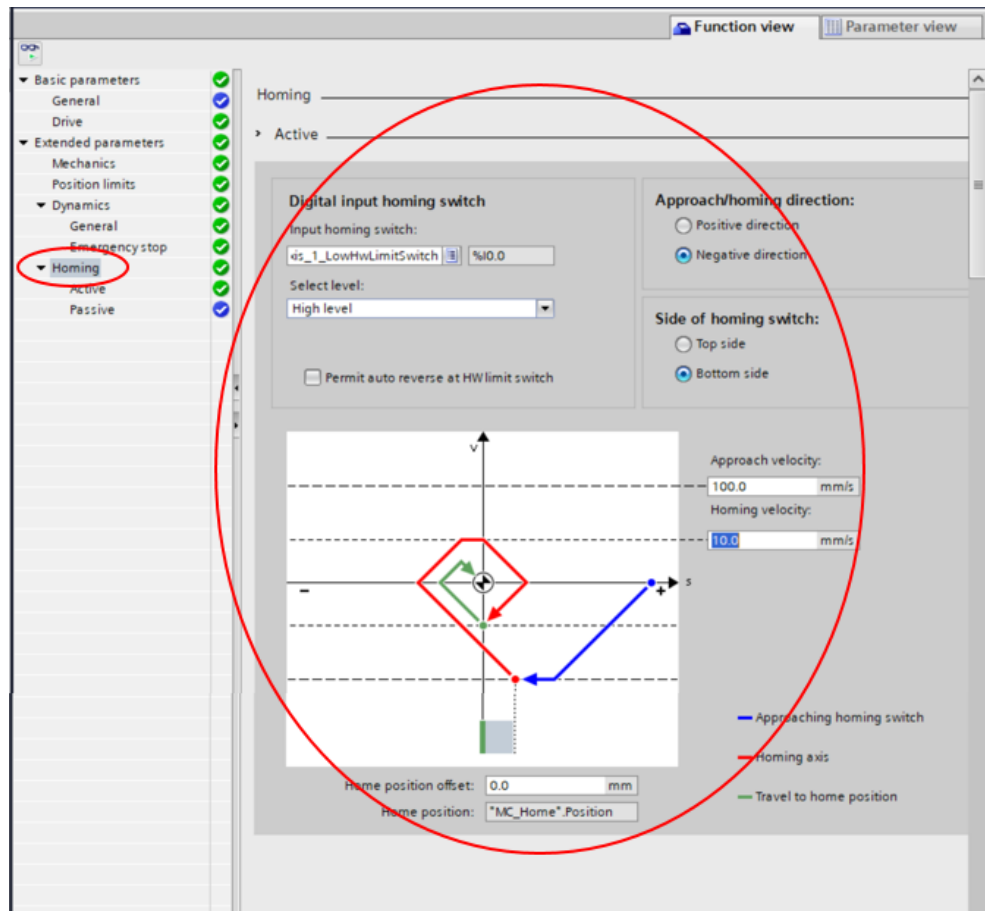
Hình 4.8: Cấu hình tại mục Dynamics- General

- Dynamics- Emergency stop: khai báo phần dừng khẩn cấp như tại Hình 4.9.



Hình 4.9: Cấu hình tại mục Dynamics- Emergency stop

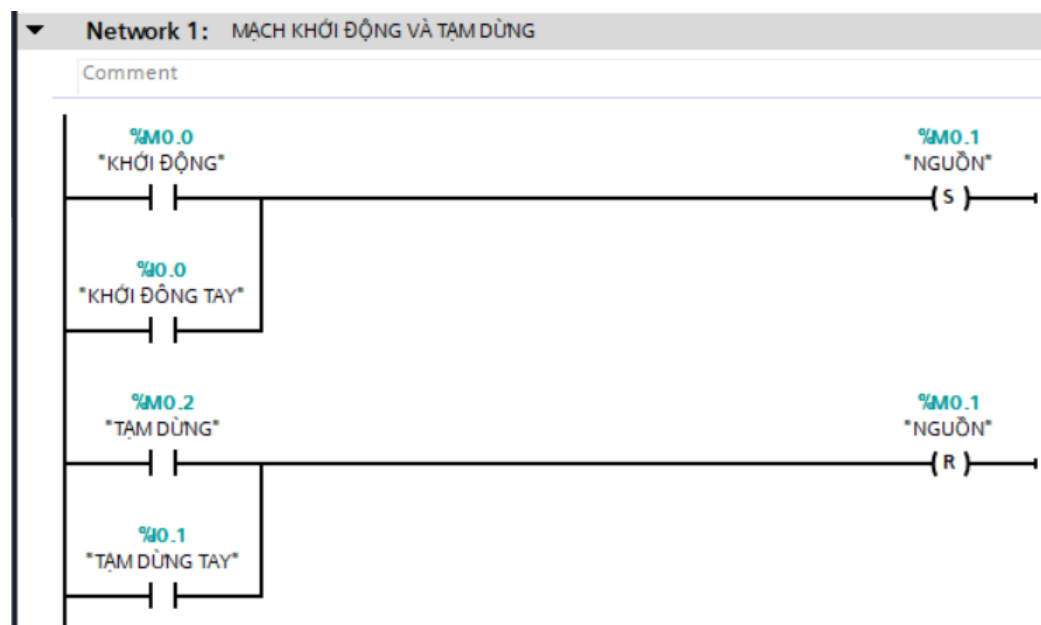
- Homing: kích hoạt tính năng home



Hình 4.10: Cấu hình tại mục Homing

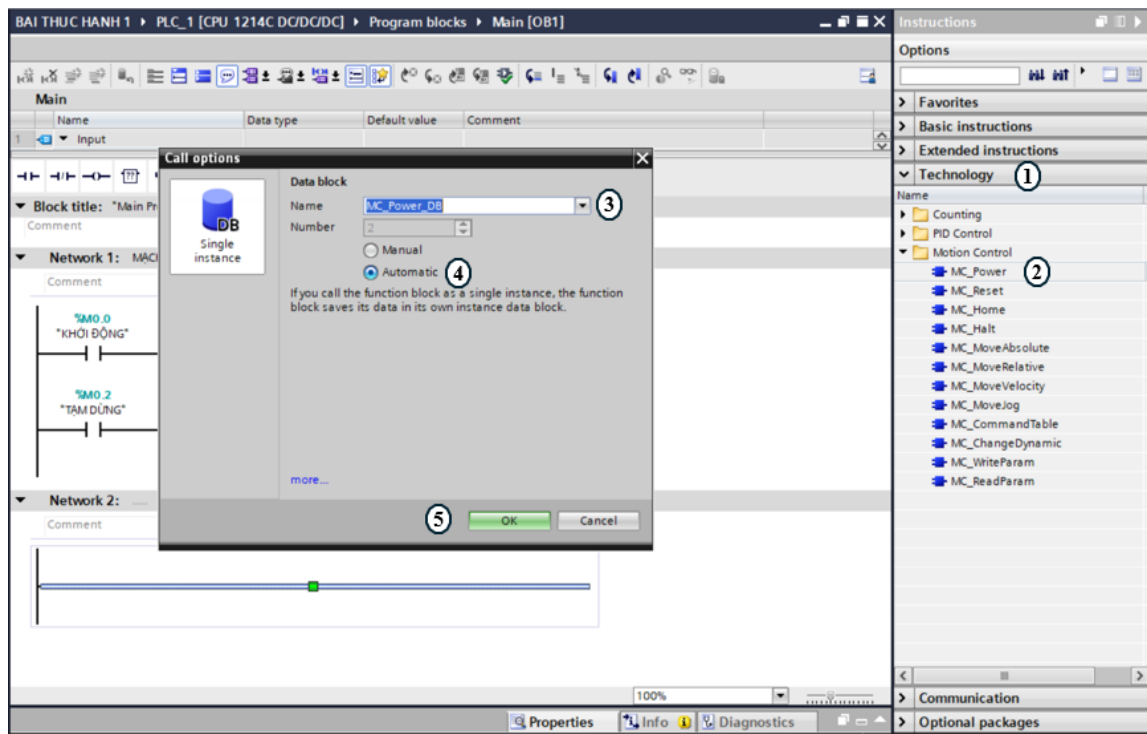
Bước 5. Viết chương trình điều khiển

- Đầu tiên ta tạo mạch khởi động gồm nút KHỞI ĐỘNG và TẠM DỪNG tại Network 1.



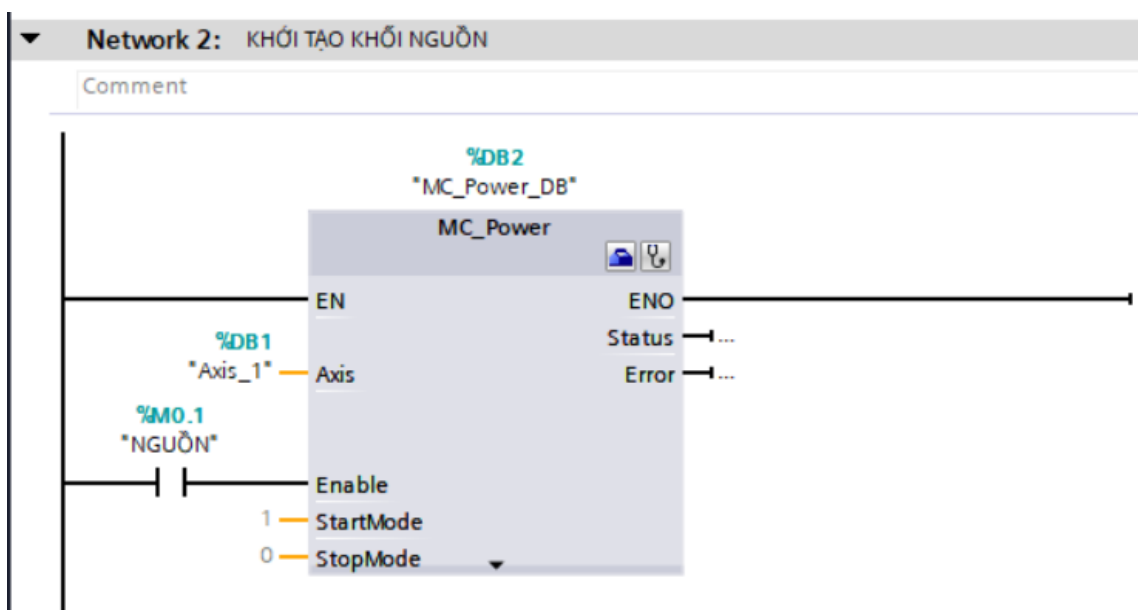
- Khởi tạo khối tạo nguồn (MC_Power) tại Network 2 như Hình 4.11.

- ① Chọn tab Technology
- ② Ở cửa sổ Motion control chọn MC_Power xuất hiện hộp thoại Call option
- ③ Đặt tên cho Data block
- ④ Kích chọn Automatic
- ⑤ Bấm OK hoàn thành



Hình 4.11: Khởi tạo khối MC_Power

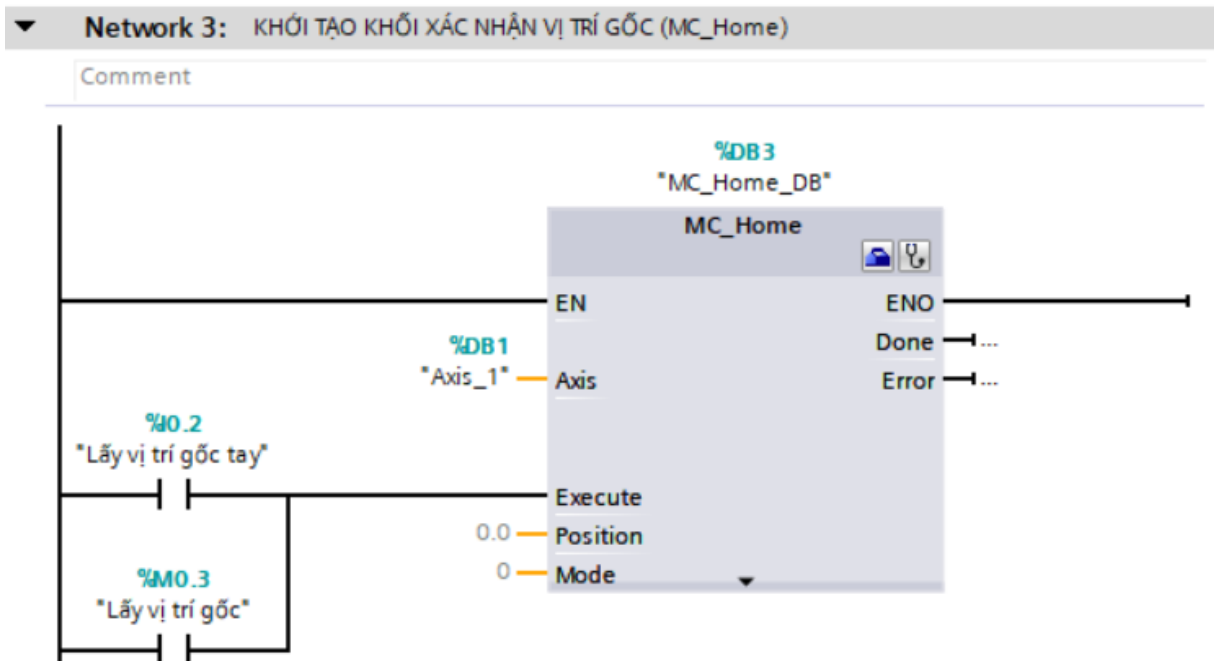
Sau đó thiết đặt các chân của khối MC_Power như tại Network 2.



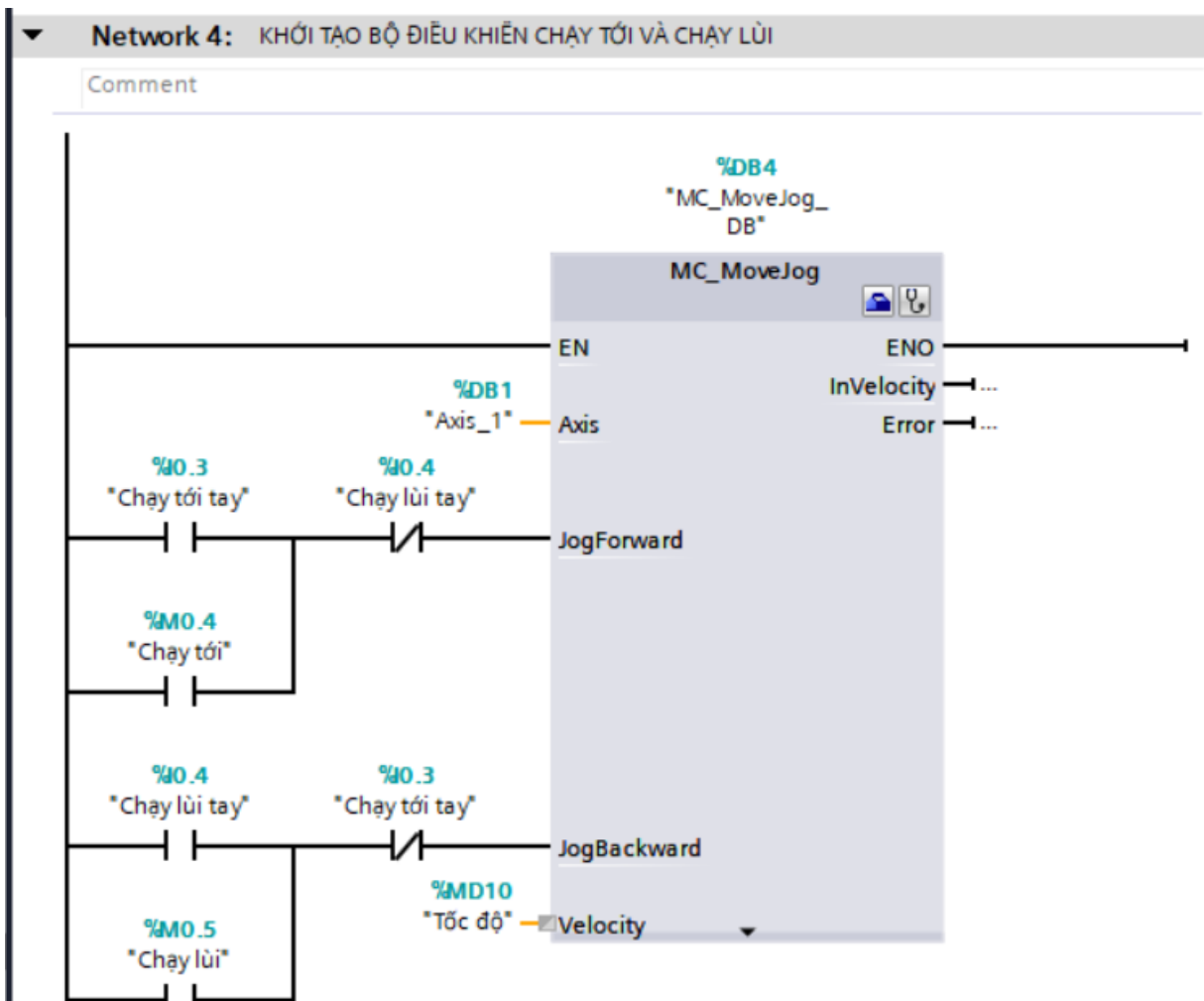
- Axis: vị trí bàn trượt

- Enable: bộ điều khiển vị trí được kích hoạt

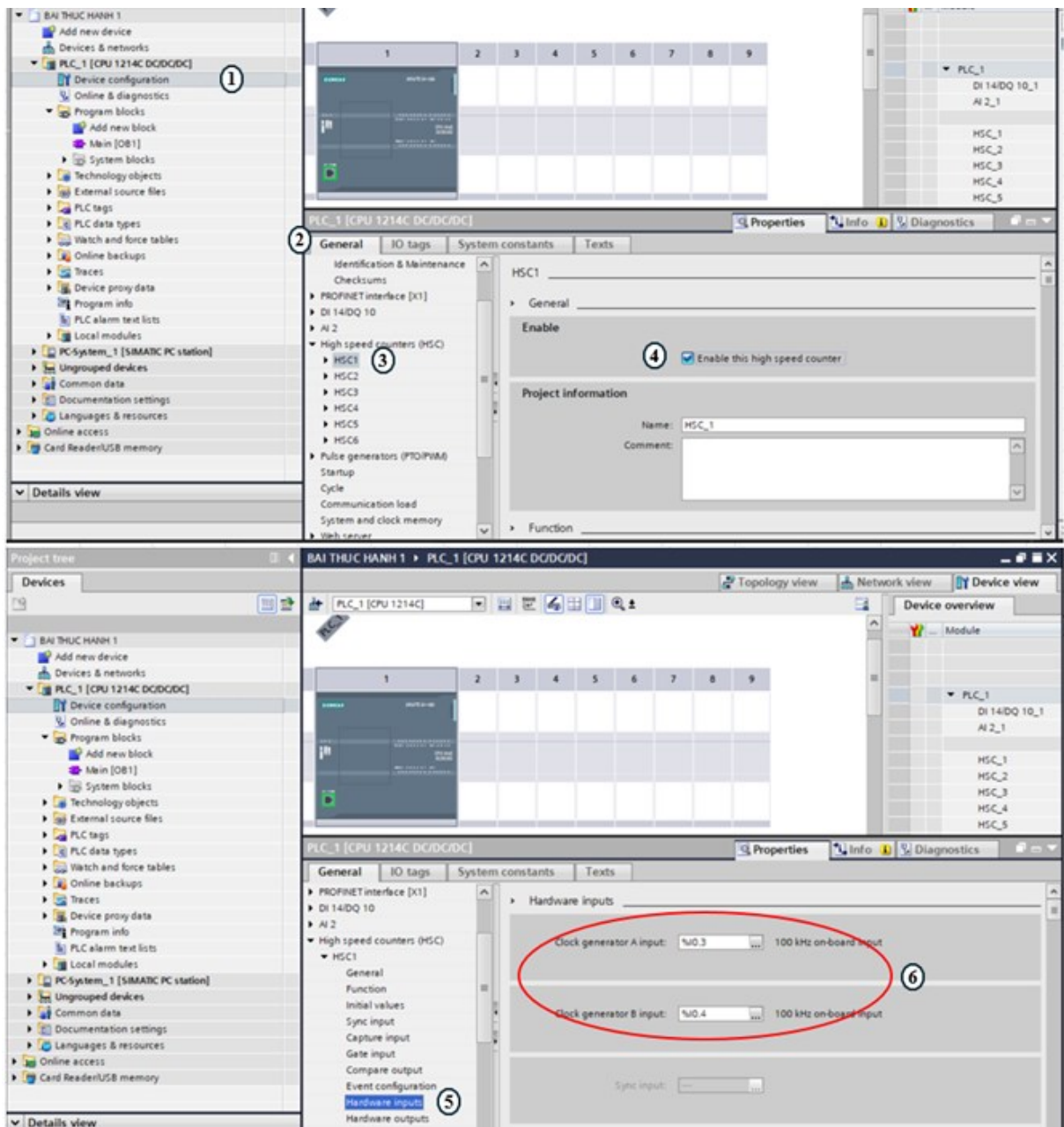
- Tương tự, khởi tạo khối xác định tọa độ gốc như tại Network 3.



- Khởi tạo bộ chạy tới và chạy lùi (MC_MoveJog) tại Network 4.



- JogForward: chạy tiến
- JogBackward: chạy lùi
- Velocity: Tốc độ
- Tạo bộ nhớ lưu vị trí phản hồi bàn trượt thông qua bộ mã hóa vòng quay (encoder) và các bước được thực hiện như sau:
- Đầu tiên ta khởi tạo bộ HSC1 để thực hiện đọc xung từ bộ encoder như Hình 4.12.

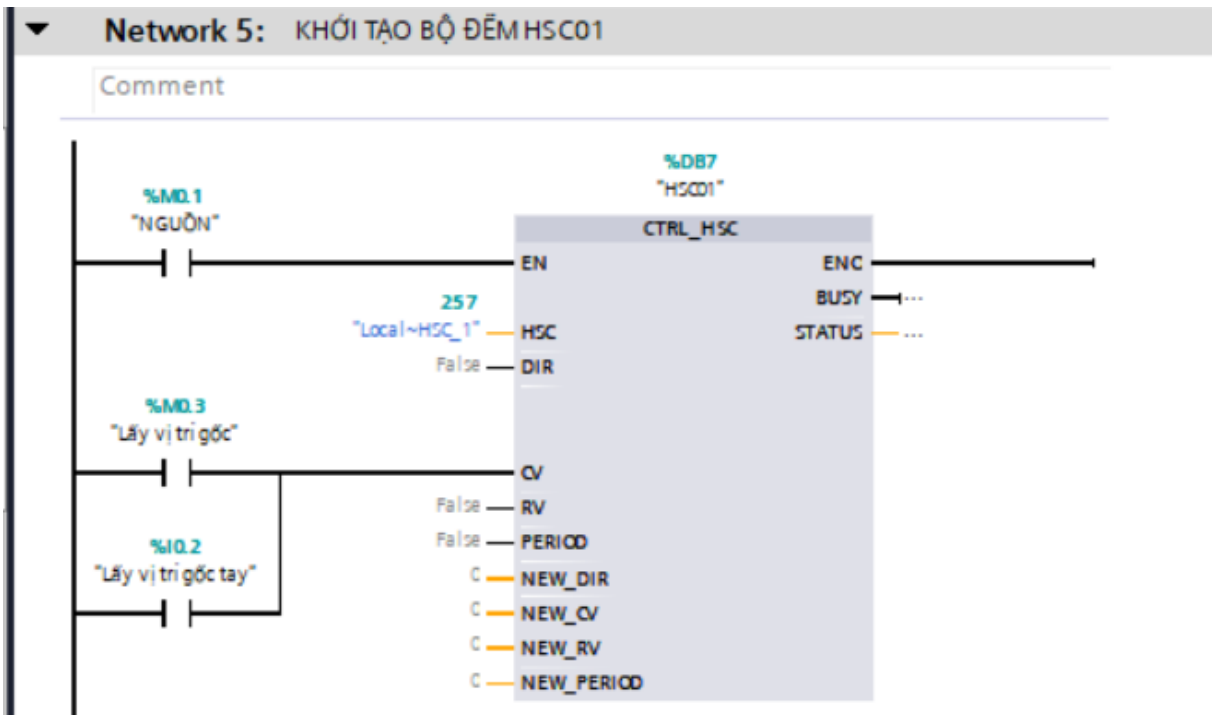


Hình 4.12: Khởi tạo bộ đếm tốc độ cao (HSC1)

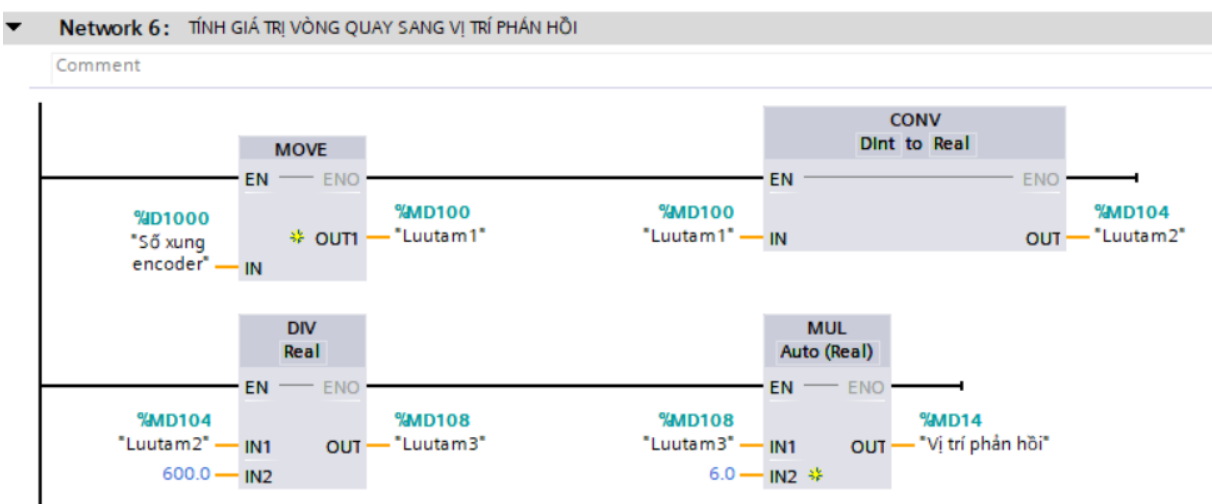
- ① Chọn mục cấu hình phần cứng cho thiết bị (Device configuration)

- ② Chọn tab General
- ③ Chọn bộ đếm tốc độ cao HSC1
- ④ Bấm chọn kích hoạt bộ đếm tốc độ cao
- ⑤ Chọn cấu hình đầu vào cho bộ HSC1
- ⑥ Thiết lập chân đếm cho bộ đếm HSC1

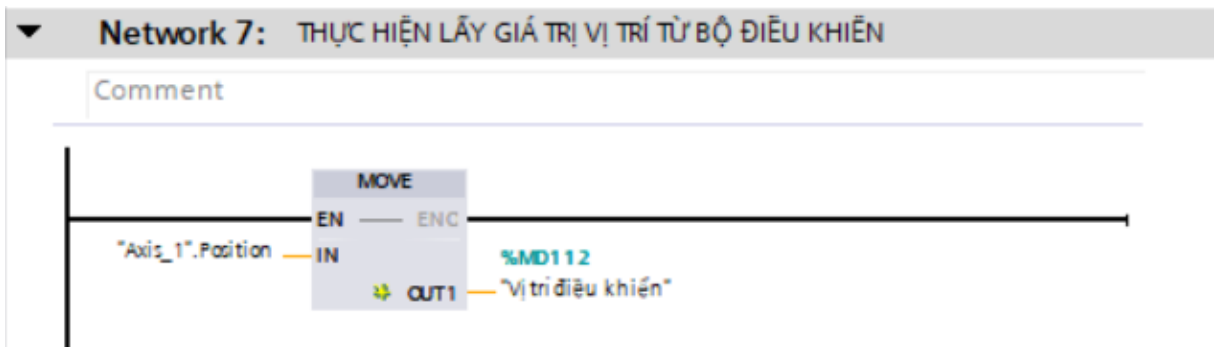
- Thực hiện khởi tạo bộ HSC01 như tại Network 5.



- Thực hiện việc tính toán đọc từ giá trị bộ mã hóa vòng quay sang vị trí được thực hiện như tại Network 6.



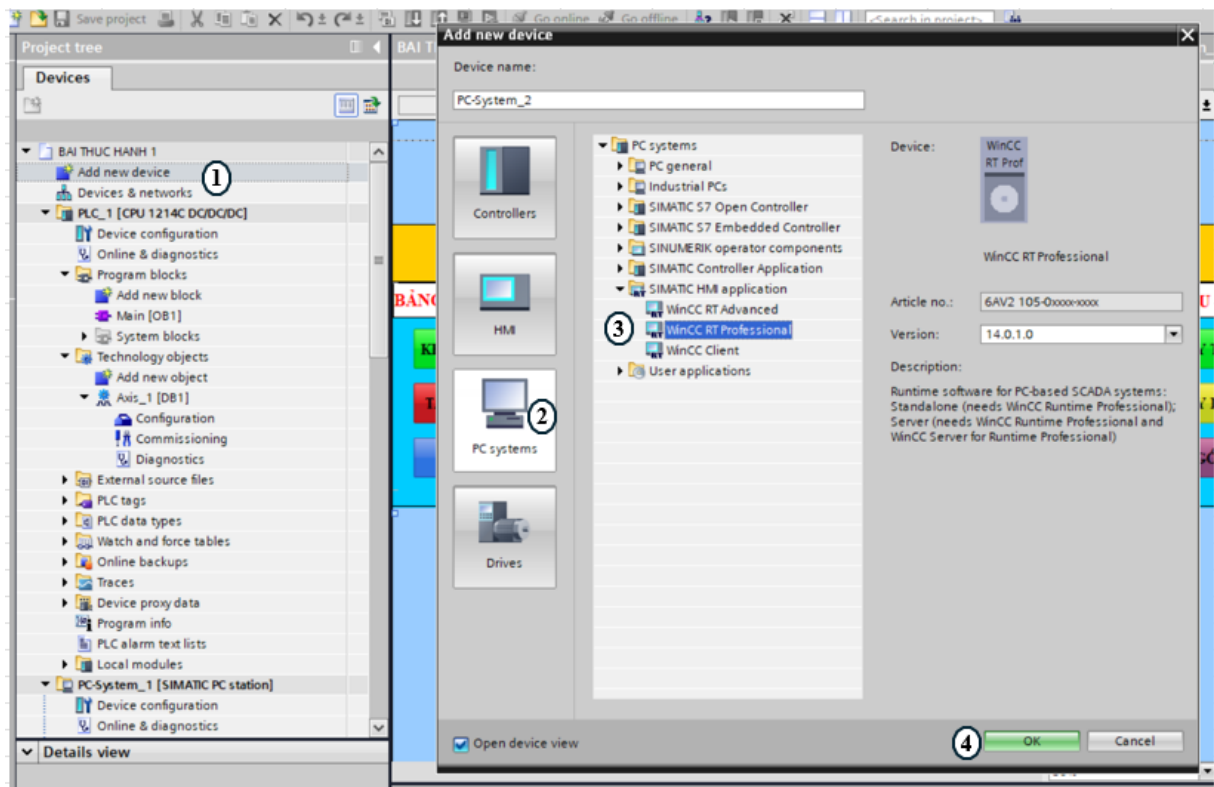
- Thực hiện đọc giá trị vị trí từ bộ điều khiển như tại Network 7.



Bước 6. Xây dựng bản điều khiển và giám sát

- Khởi tạo bộ giám sát PC-System như Hình 4.13 sau:

- ① Thêm một thiết bị mới
- ② Chọn mục PC-System
- ③ Chọn WINCC RTProfessional
- ④ Bấm OK

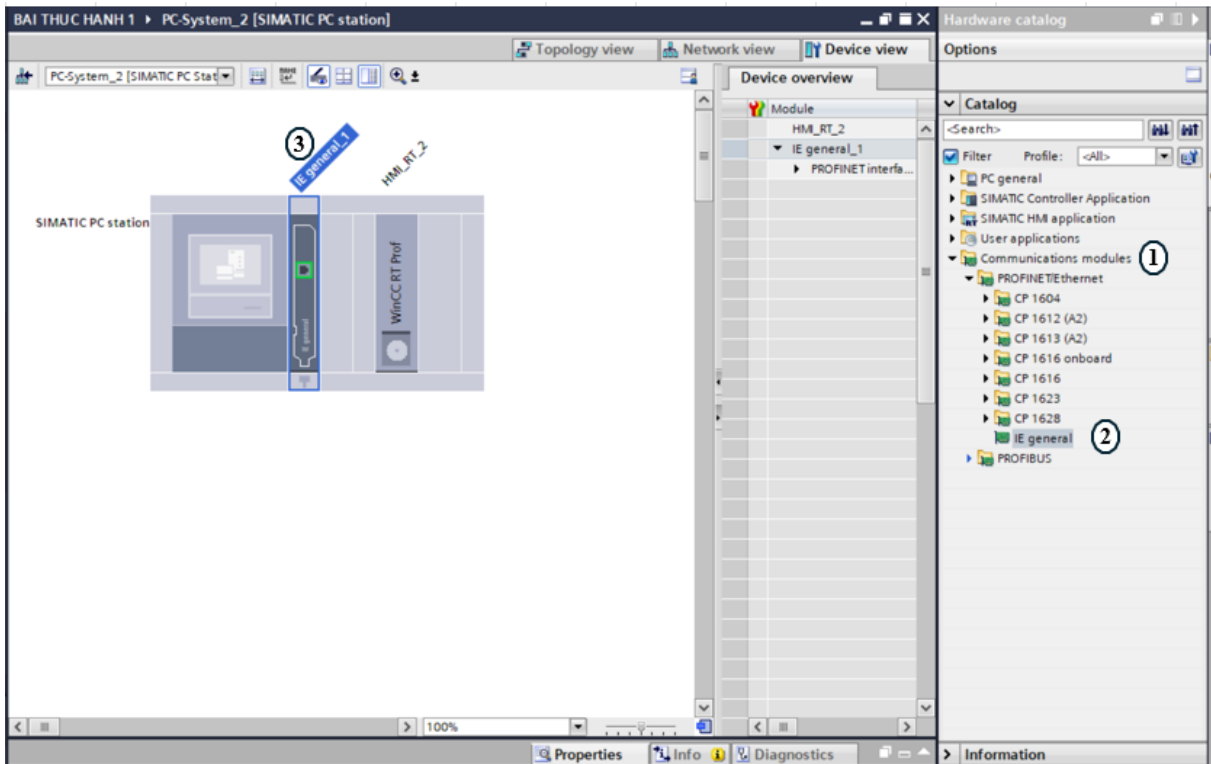


Hình 4.13: Khởi tạo bộ SCADA

- Gắn bộ IE General như tại Hình 4.14.

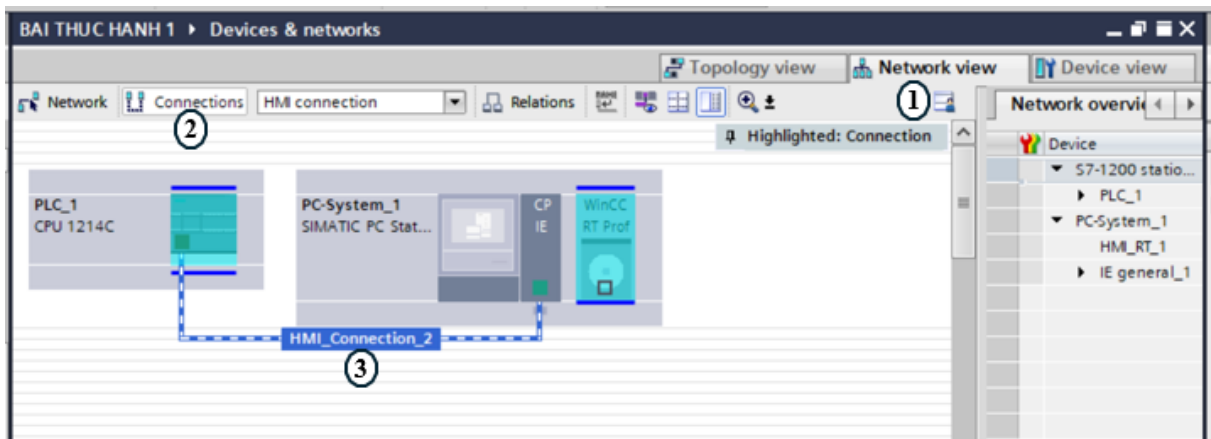
- ① Chọn Tab Communications modules
- ② Trong cửa sổ PROFINET/Ethernet chọn IE General

③ Hoàn thiện gắn modules truyền thông



Hình 4.14: Gắn cổng card IE

- Kết nối bộ lập trình PLC S7-1200 với PC-System như Hình 4.15 sau:



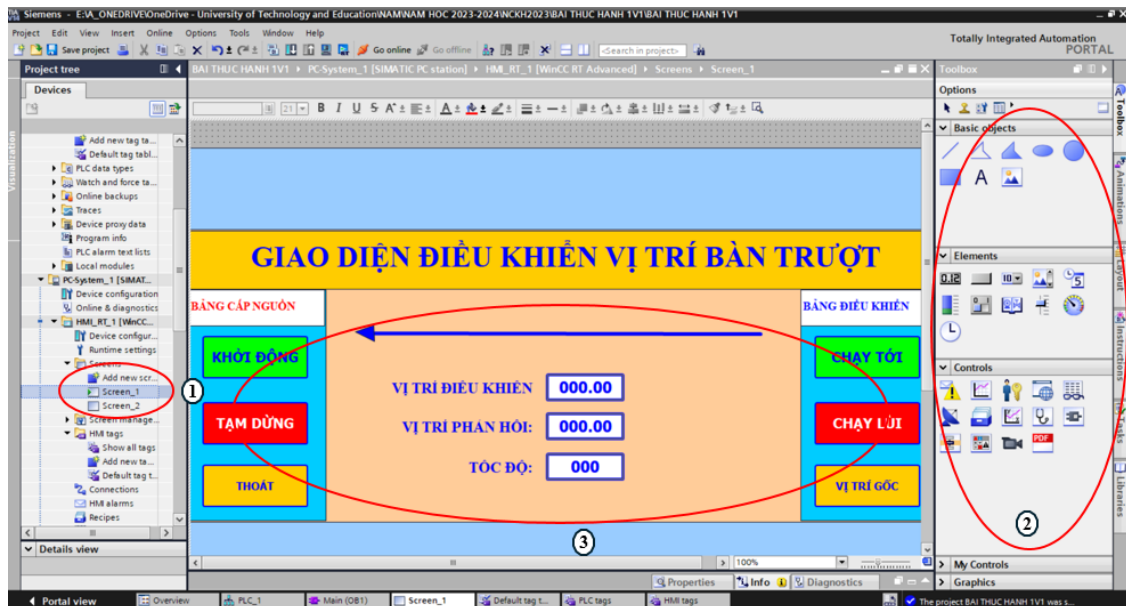
Hình 4.15: Kết nối bộ lập trình PLC S7-1200 với PC-System

- ① Chọn Tab Network view
- ② Chọn Tab Connections
- ③ Liên kết PLC và PC-System

- Tạo một giao diện điều khiển và giám sát mô hình điều khiển vị trí như Hình 4.16.

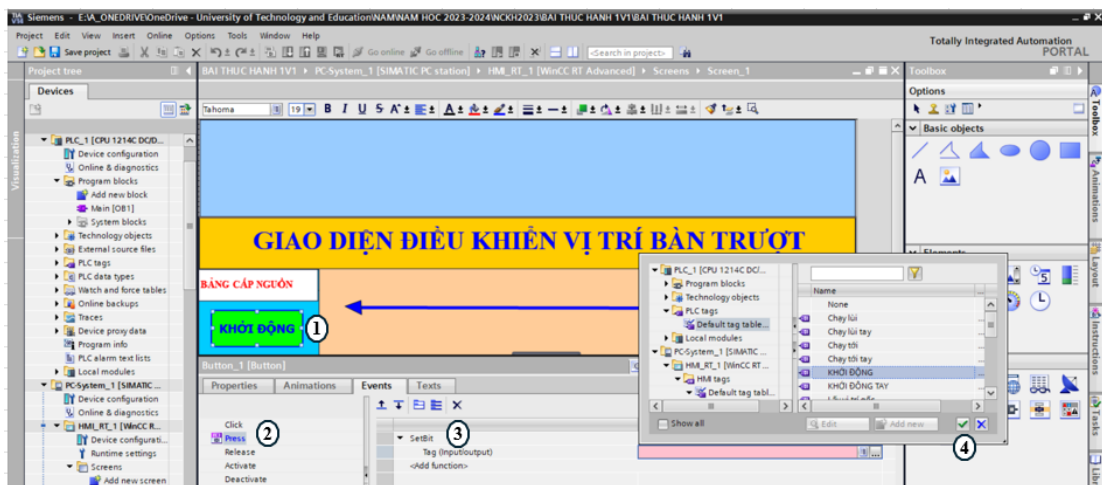
- ① Tạo một màn hình mới

- ② Thanh công cụ để thực hiện tạo các chức năng trên giao diện màn hình
- ③ Giao diện màn hình hoàn thiện



Hình 4.16: Giao diện điều khiển và giám sát

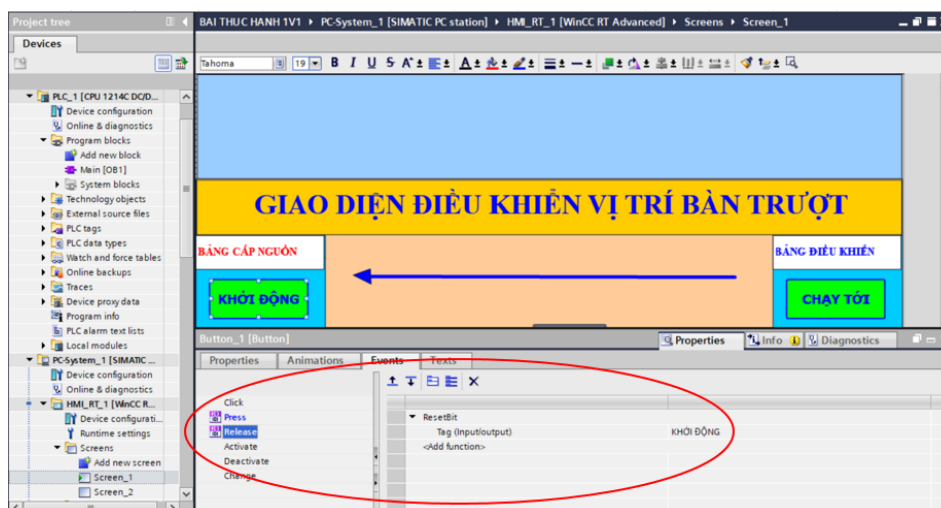
- Liên kết các Tag trên PLC với giao diện điều khiển (Hình 4.17)



Hình 4.17: Các bước thực hiện liên kết tag

- ① Chọn thuộc tính Properties nút KHỞI ĐỘNG
- ② Chọn mục Press
- ③ Chọn SetBit
- ④ Tại ô chọn mục 3 chấm, hộp thoại hiện ra chọn mục PLC tag và chọn tag KHỞI ĐỘNG ở bên trái và bấm ok.

Tương tự, tại mục *Release* ta cũng thực hiện *ResetBit*, kết quả sau thực hiện như tại Hình 4.18 sau:



Hình 4.18: Thực hiện liên kết tag nút KHỞI ĐỘNG hoàn thiện

Tương tự, tiến hành thực hiện liên kết tag cho tất cả các nút điều khiển, ô hiển thị vị trí và nhập tốc độ.

4.2. Bài thực hành 2

4.2.1. Nội dung bài thực hành

Xây dựng chương trình điều khiển trên PLC S7-1200 để điều khiển mô hình điều khiển vị trí bàn trượt với yêu cầu như sau:

- Một nút nhấn Khởi động, một nút nhấn Tạm dừng
- Một nút nhấn để xác nhận vị trí gốc và một nút nhấn Về gốc (Home) để đưa bàn trượt về vị trí gốc.
- Một nút nhấn chạy tiến và một nút nhấn chạy lùi bàn trượt. Khi nhấn nút Về gốc thì bàn trượt di chuyển về vị trí gốc.
- Ba nút nhấn vị trí 1, 2 và 3 với các giá trị vị trí tương ứng là 100 mm, 150 mm và 200 mm. Khi nhấn nút Home thì bàn trượt di chuyển về vị trí gốc
- Hiển thị giá trị vị trí hiện tại, thiết đặt giá trị tốc độ trên giao diện WINCC phần mềm TIA Portal.

4.2.2. Mục tiêu kiến thức đạt được

Người học xây dựng chương trình phát xung điều khiển động cơ servo

Thiết lập các giá trị để điều khiển vị trí bàn trượt đến các vị trí cụ thể

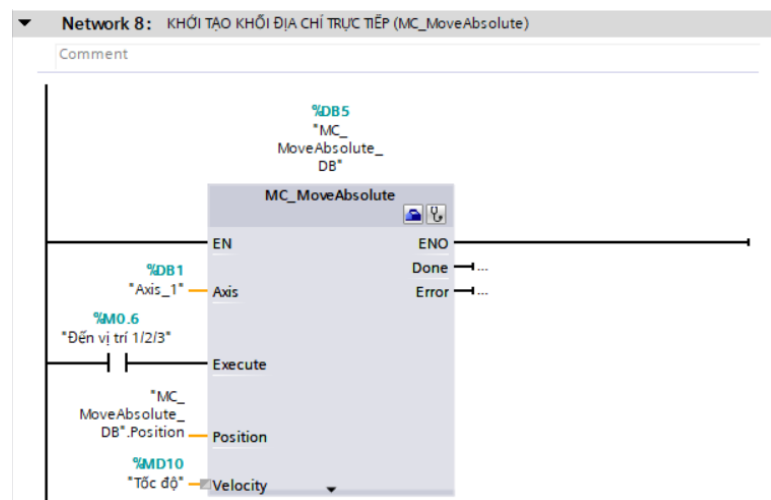
Thao tác kết nối dây phần cứng, cài đặt phần mềm điều khiển, hiển thị giá trị vị trí trên phần mềm điều khiển.

4.2.3. Hướng dẫn thực hiện

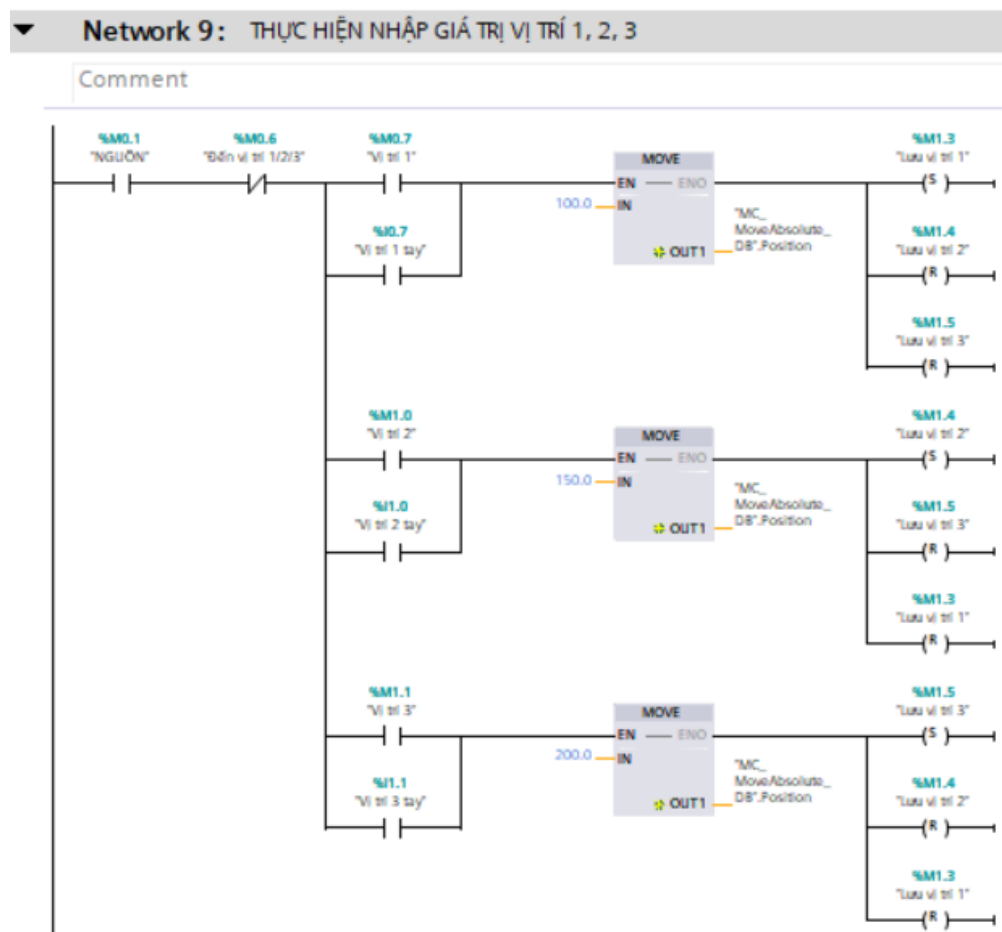
Bước 1. Thực hiện tương tự như Bước 1, 2, 3, 4 như tại Bài thực hành 1.

Bước 2. Xây dựng chương trình điều khiển

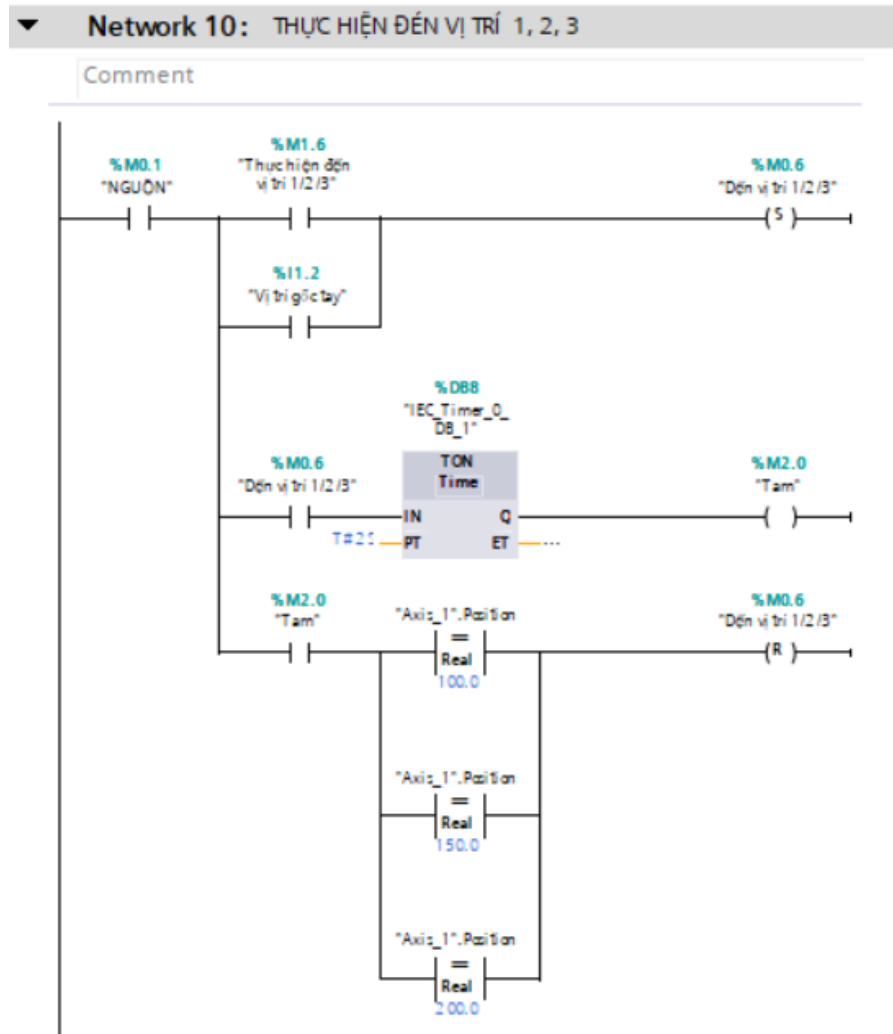
- Tại Network 1, 2, 3, 4, 5, 6, 7 xây dựng giống như tại Bài thực hành 1
- Tại Network 8 ta khởi tạo khối địa chỉ trực tiếp như Hình sau:



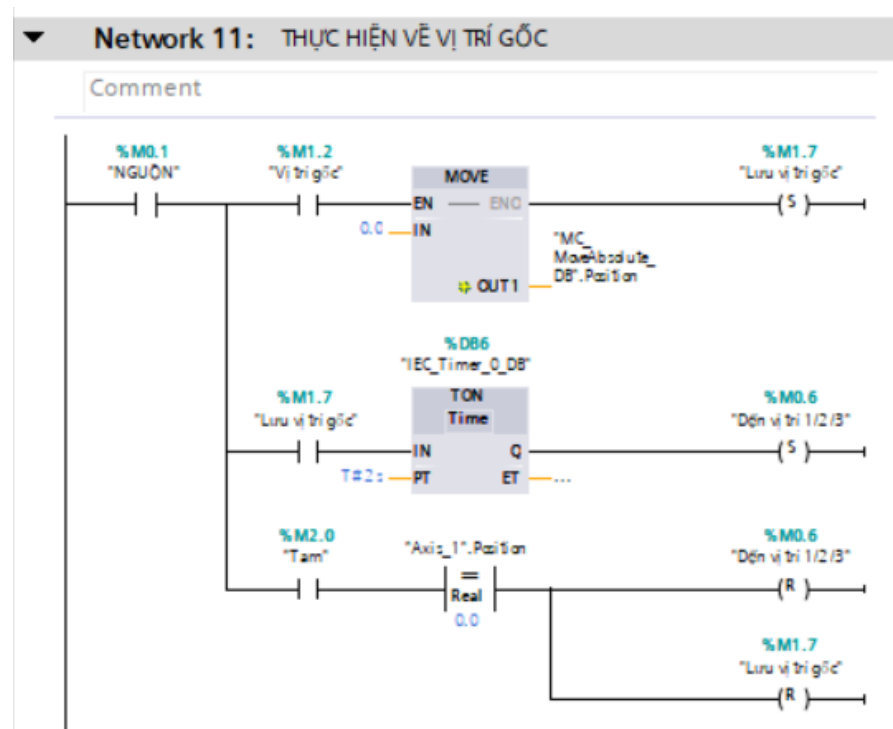
- Thực hiện việc nhập vị trí 100 mm, 150 mm, 200 mm như tại Network 9 sau:



- Thực hiện di chuyển đến các vị trí 1, 2, 3 được thực hiện như tại Network 10



- Thực hiện về vị trí gốc như tại Network 11



Bước 3. Xây dựng bản điều khiển và giám sát

- Thực hiện tương tự như Bài thực hành 1 với giao diện như tại Hình sau:

GIAO DIỆN ĐIỀU KHIỂN VỊ TRÍ BÀN TRƯỢT

BẢNG CẤP NGUỒN

KHỞI ĐỘNG

TẠM DỪNG

THOÁT

BẢNG ĐIỀU KHIỂN

CHẠY TIỀN

CHẠY LÙI

VỀ GỐC

VỊ TRÍ GỐC

VỊ TRÍ ĐIỀU KHIỂN: 000.00 (mm)

VỊ TRÍ PHẢN HỒI: 000.00 (mm)

TỐC ĐỘ: 000

VỊ TRÍ 1 (100mm)

VỊ TRÍ 2 (150mm)

VỊ TRÍ 3 (200mm)

THỰC HIỆN

Hình 4.19: Giao diện điều khiển và giám sát Bài thực hành 2

4.3. Bài thực hành 3

4.3.1. Nội dung bài thực hành

Xây dựng chương trình điều khiển trên PLC S7-1200 để điều khiển mô hình điều khiển vị trí bàn trượt với yêu cầu như sau:

- Một nút nhấn Khởi động, một nút nhấn Tạm dừng
- Một nút nhấn để xác nhận vị trí gốc và một nút nhấn Về gốc (Home) để đưa bàn trượt về vị trí gốc.
- Một nút nhấn chạy tiến và một nút nhấn chạy lùi bàn trượt. Khi nhấn nút Về gốc thì bàn trượt di chuyển về vị trí gốc.
- Một ô để đặt giá trị vị trí bàn trượt dịch chuyển đến, một nút nhấn thực hiện di chuyển bàn trượt.
- Hiển thị giá trị vị trí hiện tại, một ô thiết đặt giá trị tốc độ trên giao diện WINCC phần mềm TIA Portal.

4.3.2. Mục tiêu kiến thức đạt được

Người học xây dựng chương trình phát xung điều khiển động cơ servo.

Thiết lập các giá trị để điều khiển vị trí bàn trượt di chuyển đến một vị trí bất kỳ.

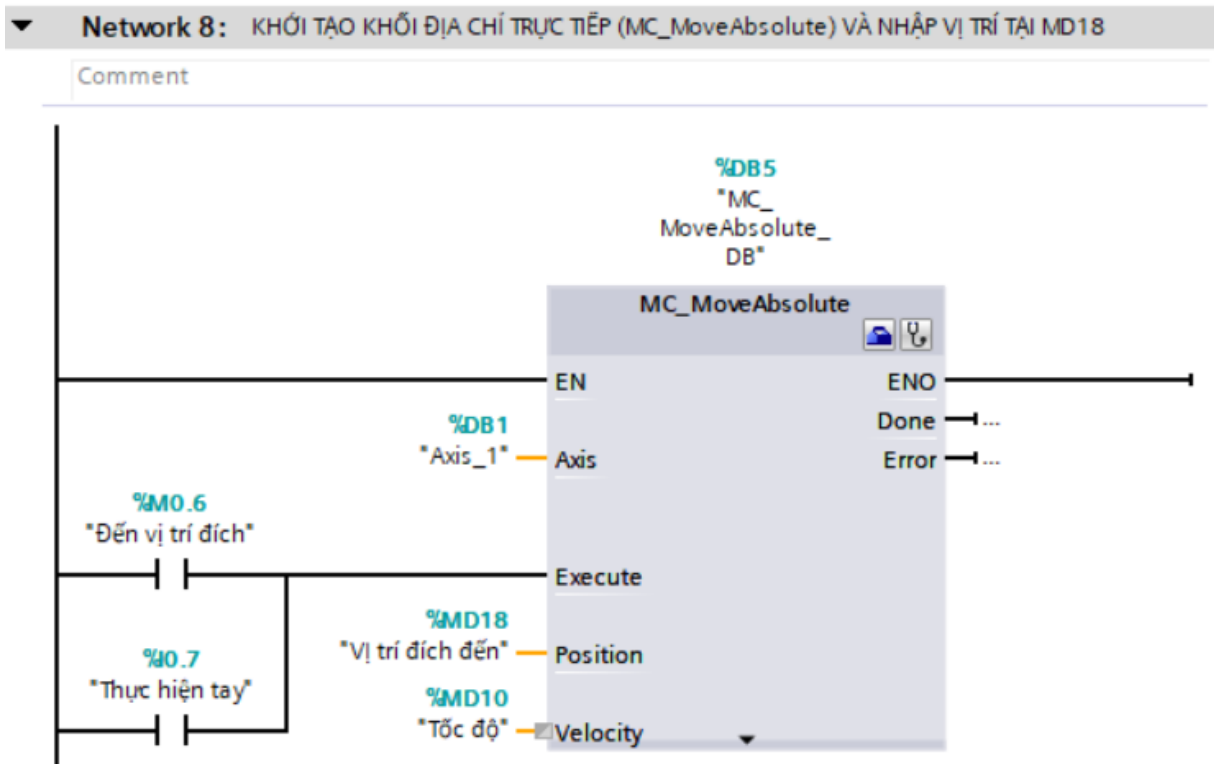
Thao tác kết nối dây phần cứng, cài đặt phần mềm điều khiển, hiển thị giá trị vị trí bàn trượt trên phần mềm điều khiển.

4.3.3. Hướng dẫn thực hiện

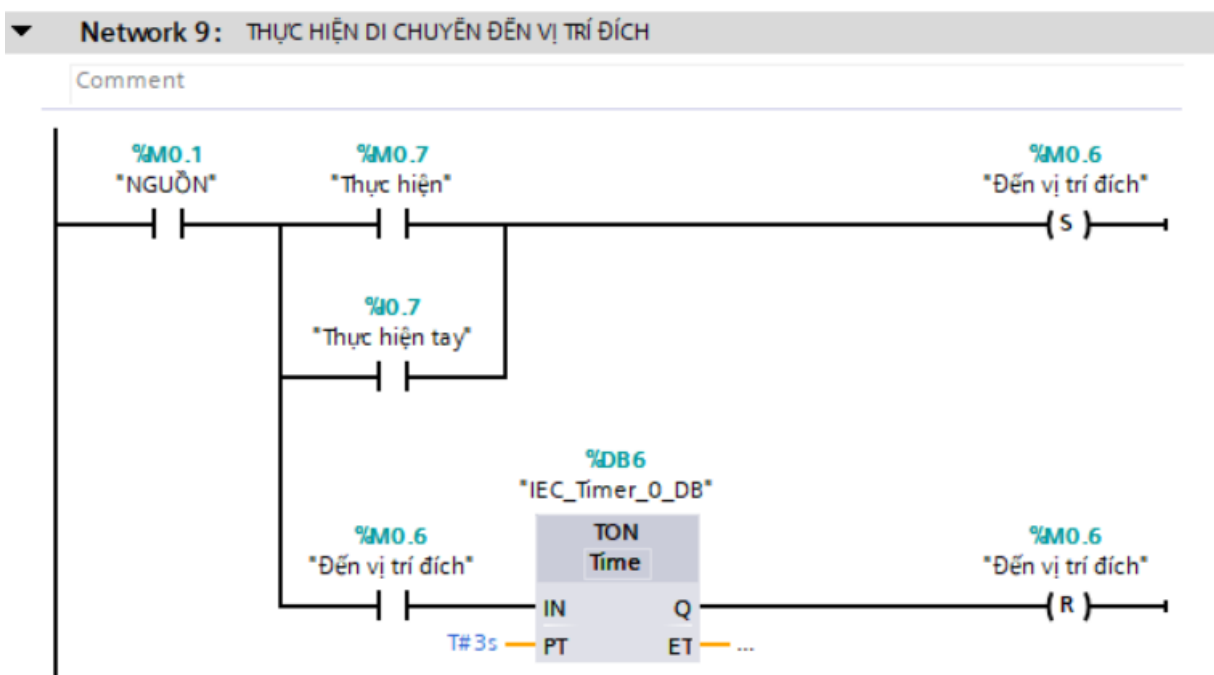
Bước 1. Thực hiện tương tự như Bước 1, 2, 3, 4 như tại Bài thực hành 1.

Bước 2. Xây dựng chương trình điều khiển

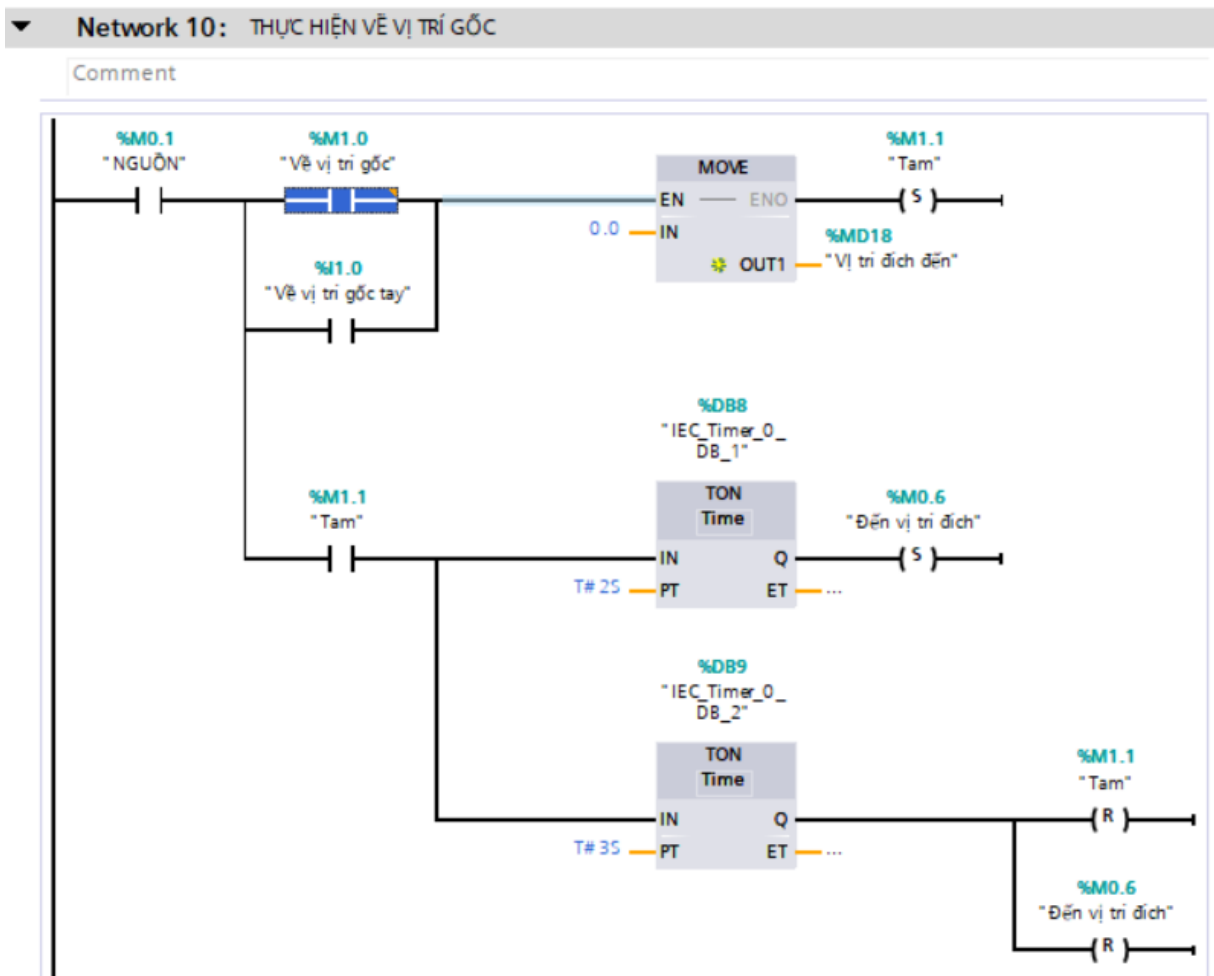
- Tại Network 1, 2, 3, 4, 5, 6, 7 xây dựng giống như tại Bài thực hành 1
- Tại Network 8 ta khởi tạo khối địa chỉ trực tiếp (MC_MoveAbsolute) VÀ NHẬP VỊ TRÍ TẠI MD18



- Tại Network 9 thực hiện việc di chuyển bàn trượt đến đích sau khi cài đặt giá trị đích đến cho bàn trượt.



- Thực hiện về gốc như tại Network 10 sau:



Bước 3. Xây dựng bản điều khiển và giám sát

- Thực hiện tương tự như Bài thực hành 1 với giao diện như tại Hình sau:

GIAO DIỆN ĐIỀU KHIỂN VỊ TRÍ BÀN TRƯỢT			
BẢNG CẤP NGUỒN			BẢNG ĐIỀU KHIỂN
KHỞI ĐỘNG TẠM DỪNG THOÁT	VỊ TRÍ ĐIỀU KHIỂN 000.00 (mm) THỰC HIỆN VỊ TRÍ PHẢN HỒI: 000.00 (mm) VỊ TRÍ GỐC TỐC ĐỘ: 000 VỊ TRÍ ĐÍCH: 000.00 (mm)		CHẠY TỚI CHẠY LÙI VỀ GỐC

Hình 4.20: Giao diện điều khiển và giám sát Bài thực hành 3

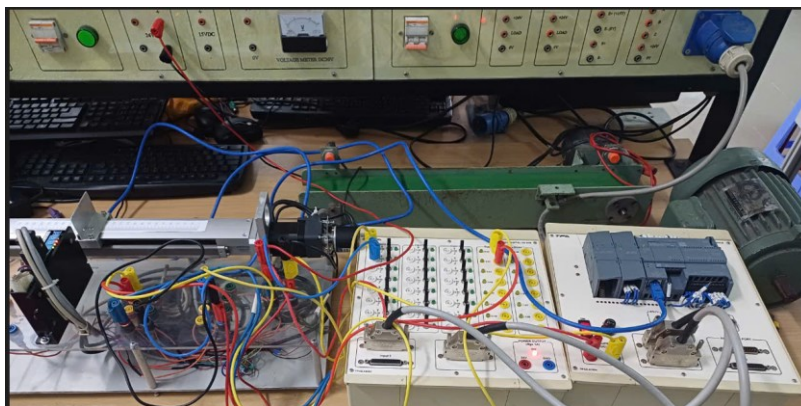
CHƯƠNG 5: KIỂM TRA THỰC NGHIỆM VÀ ĐÁNH GIÁ MÔ HÌNH

5.1. Kiểm tra thực nghiệm mô hình

Thực nghiệm mô hình dựa trên các bài tập được xây dựng ở Chương 5, để thực nghiệm mô hình trước tiên chúng ta liên kết các thiết bị và cấp nguồn theo sơ đồ Hình 3.3.

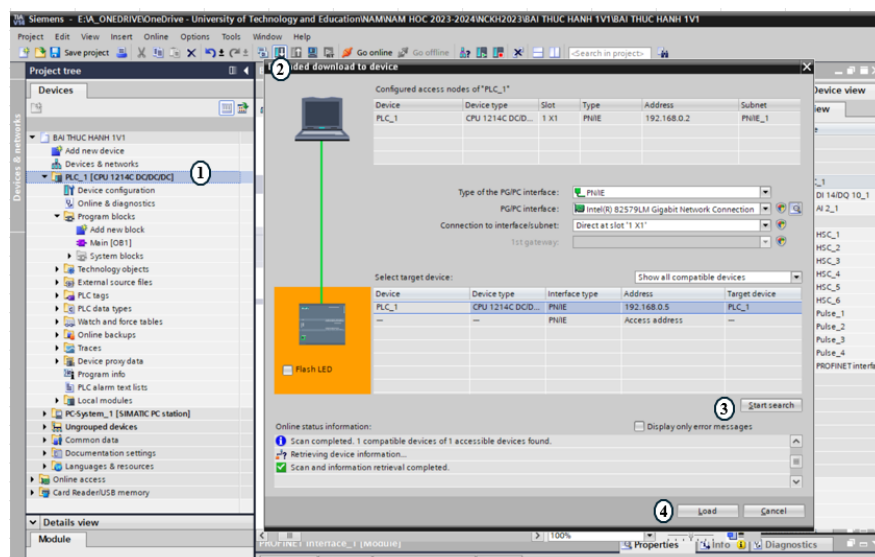
5.1.1. Bài thực hành 1

Bước 1: Hoàn thiện kết nối các thiết bị và nguồn như tại Hình 5.1.



Hình 5.1: Kết nối các thiết bị mô hình và nguồn cấp

Bước 2: Thực hiện tải chương trình xuống bộ lập trình và thực hiện chạy thực nghiệm mô hình như Hình 5.2.

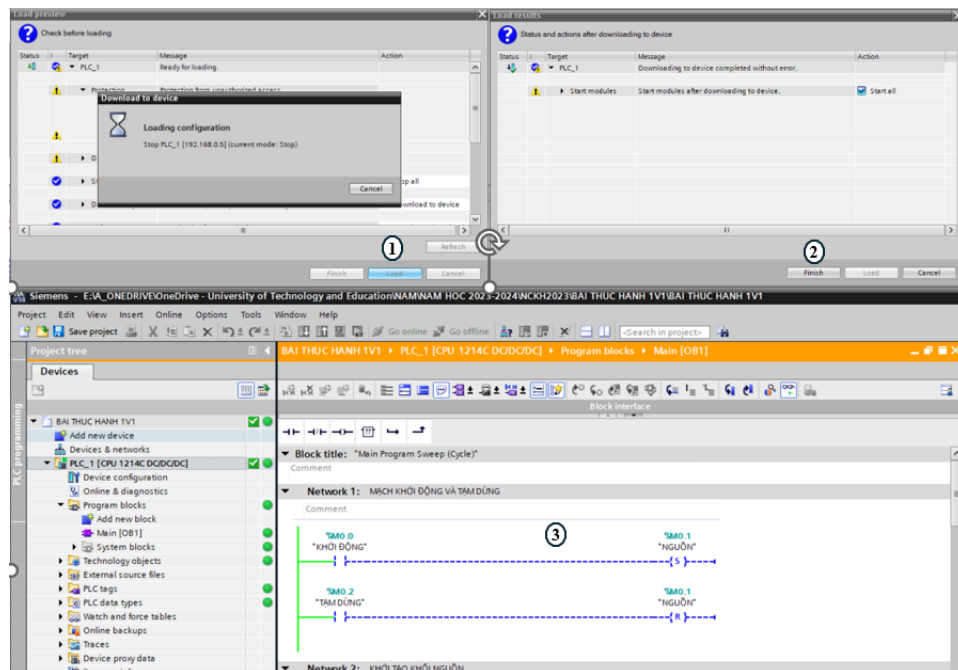


Hình 5.2: Thực hiện tải chương trình xuống bộ lập trình

① Chọn bộ lập trình

- ② Bấm vào nút tải chương trình
- ③ Tìm thiết bị lập trình
- ④ Thực hiện tải chương trình

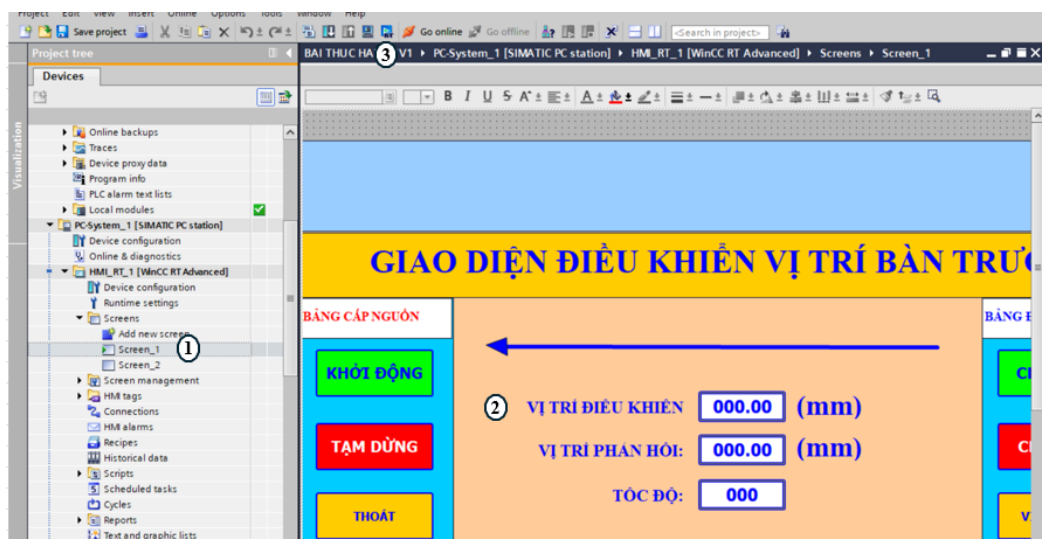
Hoàn thiện quá trình tải chương trình điều khiển xuống bộ lập trình (Hình 5.3).



Hình 5.3: Hoàn thiện quá trình tải chương trình

- ① Quá trình tải chương trình.
- ② Quá trình tải chương trình hoàn thành .
- ③ Chạy online chương trình.

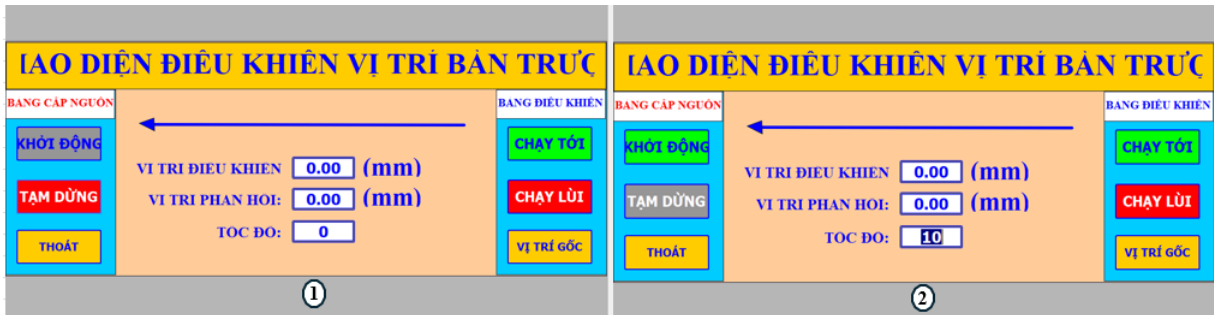
Thực hiện chạy màn hình giám sát và điều khiển mô hình như Hình 5.4.



Hình 5.4: Chạy RT màn hình điều khiển và giám sát mô hình

- ① Thực hiện mở giao diện màn hình.
- ② Giao diện màn hình giám sát và điều khiển.
- ③ Bấm vào nút RT để thực hiện chạy Real time.

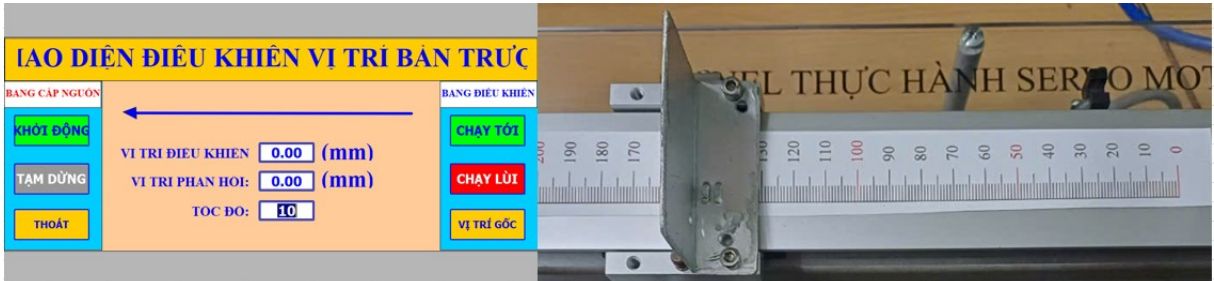
Tiến hành thực nghiệm mô hình như Hình 5.5.



Hình 5.5: Giao diện màn hình chạy thực nghiệm

- ① Giao diện màn hình.
- ② Giao diện màn hình đã khởi động hệ thống và nhập tốc độ cho bàn trượt.

Hiện tại vị trí bàn trượt hiện tại đang ở tại 130 như Hình 5.6, thực hiện bấm nút chạy tới và chạy lùi để thực nghiệm mô hình.



Hình 5.6: Bàn trượt tại vị trí 130 trên mô hình

Tại vị trí 130 thì màn hình điều khiển cho giá trị vị trí hiện tại và vị trí phản hồi đều là 0. Ta tiến hành thiết đặt giá trị tốc độ là 10 và thực nghiệm cho mô hình chạy tới và kết quả như tại Hình 5.7.



Hình 5.7: Kết quả thực nghiệm di chuyển bàn trượt chạy tới

Hình 5.7, Giá trị vị trí được di chuyển là 67.72 và trên mô hình là giá trị 187.7

(tính tương đối $130 + 57.7 = 187.7$).

Tại vị trí 187.7 bấm nút chạy lùi và kết quả như tại Hình 5.8 sau:



Hình 5.8: Kết quả thực nghiệm di chuyển bàn trượt chạy lùi

Hình 5.8, vị trí điều khiển và vị trí bàn trượt hiển thị -69.68 nghĩa là lùi âm so với vị trí 130 từ mô hình và hiện tại vị trí tương đối bàn trượt trên mô hình cho thấy tại 60 ($130 - 69.68 = 60.32$).

Tiếp theo ta di chuyển bàn trượt về vị trí 0 trên mô hình và thực hiện lấy vị trí gốc cho bàn trượt như hình 5.9, khi nhấn vị trí gốc thì vị trí điều khiển và vị trí phản hồi đều được đặt về 0.



Hình 5.9: Thực hiện lấy vị trí gốc bàn trượt

Tại vị trí gốc thực hiện di chuyển bàn trượt chạy tới và chạy lùi và kết quả như Hình 5.10.



Hình 5.10: Kết quả di chuyển bàn trượt chạy tới và lùi từ vị trí gốc

Vậy thực nghiệm mô hình điều khiển vị trí bàn trượt chạy tới và chạy lùi với kết quả đạt yêu cầu.

5.1.2. Bài thực hành 2

Tương tự như Bài thực hành 1, thực hiện download chương trình và chạy thực nghiệm mô hình.

Thực hiện xác nhận vị trí gốc của mô hình như tại Hình 5.11 sau:



Hình 5.11: Xác nhận vị trí gốc mô hình

- Thực nghiệm di chuyển đến vị trí 1 cách vị trí gốc 100 mm, kết quả như Hình 5.12.



Hình 5.12: Kết quả thực nghiệm di chuyển bàn trượt đến vị trí 1

Để thực nghiệm đến vị trí 1 ta nhấn chọn vị trí 1 trên bản điều khiển sau đó nhấn chọn thực hiện và kết quả đạt được như hình trên. Vị trí điều khiển, vị trí phản hồi và vị trí thực tế trên mô hình tại 100 mm.

- Thực nghiệm di chuyển đến vị trí 2 cách vị trí gốc 150 mm, kết quả như Hình 5.13.



Hình 5.13: Kết quả thực nghiệm di chuyển bàn trượt đến vị trí 2

Kết quả thực nghiệm cho thấy vị trí điều khiển, vị trí phản hồi và vị trí thực tế trên mô hình tại 150 mm.

- Thực nghiệm di chuyển đến vị trí 3 cách vị trí gốc 200 mm, kết quả như Hình 5.14.



Hình 5.14: Kết quả thực nghiệm di chuyển bàn trượt đến vị trí 3

- Thực nghiệm di chuyển đến vị trí gốc kết quả như hình sau



5.1.3. Bài thực hành 3

Tương tự, trước tiên đưa mô hình về vị trí gốc và xác nhận gốc như Hình 5.15.



Hình 5.15: Đưa mô hình về vị trí gốc

- Thực hiện di chuyển bàn trượt đến vị trí 70 mm, tại ô vị trí đích ta nhập vào giá trị vị trí 70 sau đó nhấn nút thực hiện và kết quả như Hình 5.16.



Hình 5.16: Thực nghiệm di chuyển bàn trượt đến một vị trí bất kỳ (70 mm)

Kết quả thể hiện tại ô vị trí điều khiển, ô vị trí phản hồi và trên thước đo của mô hình điều hiển thị giá trị 70 mm đúng với giá trị vị trí cần di chuyển đến tại ô vị trí đích.

PHẦN KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Đề tài được hoàn thành với các phần

- Mô hình điều khiển vị trí bàn trượt
- Kết quả chạy thực nghiệm, kiểm tra và đánh giá
- Một báo cáo tổng kết đề tài
- Ba (03) bài thực hành mẫu

Thiết bị chế tạo hoạt động ổn định, các bài thực hành cung cấp cho sinh viên các kiến thức về điều khiển vị trí. Thông qua mô hình người học có cái nhìn trực quan việc ứng dụng các kiến thức lý thuyết đã học vào thực tiễn.

2. Kiến nghị

- Kết quả của đề tài cần được quan tâm đầu tư nhân rộng để triển khai áp dụng vào công tác đào tạo của nhà trường trong thời gian đến.
- Từ kết quả này có thể nghiên cứu bổ sung phù hợp để đào tạo sinh viên ở nhiều cấp độ khác nhau.
- Cần quan tâm đầu tư biên soạn tài liệu chuẩn để đào tạo cho sinh viên phù hợp với ngành công nghiệp hiện nay và đào tạo với nhiều cấp độ khác nhau.

TÀI LIỆU THAM KHẢO

- [1] T. Vũ Hoàng Trung, Nghiên cứu ứng dụng PLC FX3U-40MT điều khiển động cơ Servo máy CNC trong công nghiệp, Luận án Thạc sĩ, Trường Đại học Sao Đỏ, (2018).
- [2] J. S. Chauhan, R. K. Singh, and M. K. Soni, "Design and Simulation of Position Control System for DC Servo Motor Using PID Controller," *International Journal of Engineering and Advanced Technology*, vol. 3, no. 6, pp. 99-103, Aug. 2014.
- [3] J. Liu, P. Zhang, and F. Wang, "Real-Time DC Servo Motor Position Control by PID Controllers Using Labview," 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, Aug. 2009, pp. 206-209, doi: 10.1109/IHMSC.2009.59.
- [4] T. A. Dũng, Đ. A. Tuấn, and Đ. Q. Khanh, “Nghiên cứu điều khiển vị trí cho robot Scara sử dụng bộ điều khiển mờ,” *J. Sci. Technol. - HaUI*, vol. 59, no. 4, pp. 36–40, 2023.
- [5] S. Jee and Y. Koren, “Adaptive fuzzy logic controller for feed drives of a CNC machine tool,” *Mechatronics*, vol. 14, no. 3, pp. 299–326, 2004.
- [6] E. Verstappe, Design and Implementation of a Standalone CNC Controller, Grado en Ingenierías Específicas de Telecomunicación, Mención en Telemática, Universidad de Valladolid, 2019