**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CƠ KHÍ CHẾ TẠO MÁY**

****

**ĐỒ ÁN TỐT NGHIỆP**

***Đề tài*: “ĐIỀU KHIỂN TAY MÁY ỨNG DỤNG TRONG XẾP SẢN PHẨM”**

**GVHD: TS. NGUYỄN XUÂN QUANG**

**SVTH: TRẦN NGỌC HIỂU**

**MSSV: 20146127**

**SVTH: NGUYỄN BÁ VŨ THẠCH**

**MSSV: 20146530**

**SVTH: LÊ TẤN LỘC**

**MSSV: 20146121**

**Tp. Hồ Chí Minh, tháng 6/2024**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CƠ KHÍ CHẾ TẠO MÁY**

**BỘ MÔN CƠ ĐIỆN TỬ**



**ĐỒ ÁN TỐT NGHIỆP**

***Đề tài*: “ĐIỀU KHIỂN TAY MÁY ỨNG DỤNG TRONG XẾP SẢN PHẨM”**

**GVHD: TS. NGUYỄN XUÂN QUANG**

**SVTH: TRẦN NGỌC HIỂU**

**MSSV: 20146127**

**SVTH: NGUYỄN BÁ VŨ THẠCH**

**MSSV: 20146530**

**SVTH: LÊ TẤN LỘC**

**MSSV: 20146121**

**Tp. Hồ Chí Minh, tháng 6/2024**

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

**KHOA CƠ KHÍ CHẾ TẠO MÁY**

**Bộ môn: Cơ điện tử**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

***Độc lập – Tự do – Hạnh phúc***

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

**Học kỳ I/ Năm học 2023 – 2024**

Giảng viên hướng dẫn: TS. Nguyễn Xuân Quang

Sinh viên thực hiện:

1. Trần Ngọc Hiểu. MSSV: 20146127. Điện thoại: 0328743450.
2. Nguyễn Bá Vũ Thạch. MSSV: 20146530. Điện thoại: 0985075354
3. Lê Tấn Lộc. MSSV: 20146121. Điện thoại: 0877708465
4. ***Mã số đề tài: …***

* Tên đề tài: Điều khiển tay máy ứng dụng trong xếp sản phẩm.

1. ***Các số liệu, tài liệu ban đầu***

Nguyên lý hoạt động: Tay máy 5 bậc tự do được sử dụng để bốc các sản phẩm hình hộp chữ nhật với nhiều kích thước khác nhau chuyển từ hệ thống băng tải này sang băng tải khác, quỹ đạo chuyển động của tay máy được điều khiển từ máy tính.

Chức năng: Gắp và xếp có chọn lọc các loại sản phẩm.

Yêu cầu kỹ thuật:

* Tay máy có thể gắp các sản phẩm hình hộp chữ nhật với đa dạng kích thước.
* Có thể hoạt động dựa trên chương trình được lập trình từ trước bởi người dùng.

1. ***Nội dung chính của đồ án***

Hoàn thiện bản vẽ cơ khí, bản vẽ điện, tính toán động học, động lực học, viết chương trình điều khiển và giám sát trên máy tính.

Thiết kế lại tay gắp của robot.

1. ***Các sản phẩm dự kiến***

Tay máy 5 bậc tự do và găng tay điều khiển.

1. ***Ngày giao đồ án: 19/02/2024***
2. ***Ngày nộp đồ án: 01/07/2024***
3. ***Ngôn ngữ trình bày:*** ***Bản báo cáo:*** ***Tiếng Anh***  ***Tiếng Việt***

***Trình bày bảo vệ:*** ***Tiếng Anh***  ***Tiếng Việt***

**TRƯỞNG KHOA TRƯỞNG BỘ MÔN GIẢNG VIÊN HƯỚNG DẪN**

*(Ký, ghi rõ họ tên) (Ký, ghi rõ họ tên) (Ký, ghi rõ họ tên)*

Được phép bảo vệ: ............................................

*(GVHD ký, ghi rõ họ tên)*

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

**KHOA CƠ KHÍ CHẾ TẠO MÁY**

**Bộ môn: Cơ điện tử**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

***Độc lập – Tự do – Hạnh phúc***

PHIẾU NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

**(Dành cho giảng viên hướng dẫn)**

**Họ tên sinh viên:**

Trần Ngọc Hiểu :20146127

Nguyễn Bá Vũ Thạch :20146530

Lê Tấn Lộc :20146121

* **Tên đề tài:** Điều khiển tay máy ứng dụng trong xếp sản phẩm.

**Ngành đào tạo:** Công nghệ kỹ thuật cơ điện tử.

**Giảng viên hướng dẫn:** Nguyễn Xuân Quang

**Ý KIẾN NHẬN XÉT**

1. ***Nhận xét về tinh thần, thái độ làm việc của sinh viên.***

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

1. ***Nhận xét về kết quả thực hiện của đồ án tốt nghiệp***
   1. *Kết cấu, cách thức trình bày đồ án tốt nghiệp*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....................................

* 1. *Nội dung đồ án tốt nghiệp*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....................................

* 1. *Kết quả đạt được*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....................................

* 1. *Những tồn tại (nếu có)*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....................................

1. ***Đánh giá***

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mục đánh giá** | **Điểm tối đa** | **Điểm đạt được** |
| **1.** | **Hình thức và kết cấu ĐACĐT** | **30** |  |
| *Đúng format với đầy đủ cả hình thức và nội dung của các mục* | *10* |  |
| *Mục tiêu, nhiệm vụ, tổng quan của đề tài* | *10* |  |
| *Tính cấp thiết của đề tài* | *10* |  |
| **2.** | **Nội dung ĐACĐT** | **50** |  |
| *Khả năng ứng dụng kiến thức toán*  *học, khoa học và kỹ thuật, khoa học xã hội…* | *5* |  |
| *Khả năng thực hiện/phân tích/tổng hợp/đánh giá* | *10* |  |
| *Khả năng thiết kế chế tạo một hệ*  *thống, thành phần, hoặc quy trình đáp ứng yêu cầu đưa ra với những ràng buộc thực tế.* | *15* |  |
| *Khả năng cải tiến và phát triển* | *5* |  |
| *Khả năng sử dụng công cụ kỹ thuật, phần mềm chuyên ngành…* | *5* |  |
| **3.** | **Đánh giá về khả năng ứng dụng của đề tài.** | **10** |  |
| **4.** | **Sản phẩm cụ thể của ĐACĐT** | **10** |  |
|  | **Tổng** | **100** |  |

1. ***Kết luận:***

Được phép bảo vệ

Không được phép bảo vệ

TP.HCM, ngày.....tháng.....năm

Giảng viên hướng dẫn

*(Ký, ghi rõ họ tên)*

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

**KHOA CƠ KHÍ CHẾ TẠO MÁY**

**Bộ môn: Cơ điện tử**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

***Độc lập – Tự do – Hạnh phúc***

PHIẾU NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP

**(Dành cho giảng viên phản biện)**

**Họ tên sinh viên:**

Trần Ngọc Hiểu :20146127

Nguyễn Bá Vũ Thạch :20146530

Lê Tấn Lộc :20146121

**Tên đề tài:** Điều khiển tay máy ứng dụng trong xếp sản phẩm.

**Ngành đào tạo:** Công nghệ kỹ thuật cơ điện tử.

**Giảng viên hướng dẫn:** Nguyễn Xuân Quang

**Ý KIẾN NHẬN XÉT**

1. ***Nhận xét về tinh thần, thái độ làm việc của sinh viên.***

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

1. ***Nhận xét về kết quả thực hiện của đồ án tốt nghiệp***
   1. *Kết cấu, cách thức trình bày đồ án tốt nghiệp*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....................................

* 1. *Nội dung đồ án tốt nghiệp*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....................................

* 1. *Kết quả đạt được*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....................................

* 1. *Những tồn tại (nếu có)*

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....................................

1. ***Đánh giá***

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mục đánh giá** | **Điểm tối đa** | **Điểm đạt được** |
| **1.** | **Hình thức và kết cấu ĐACĐT** | **30** |  |
| *Đúng format với đầy đủ cả hình thức và nội dung của các mục* | *10* |  |
| *Mục tiêu, nhiệm vụ, tổng quan của đề tài* | *10* |  |
| *Tính cấp thiết của đề tài* | *10* |  |
| **2.** | **Nội dung ĐACĐT** | **50** |  |
| *Khả năng ứng dụng kiến thức toán*  *học, khoa học và kỹ thuật, khoa học xã hội…* | *5* |  |
| *Khả năng thực hiện/phân tích/tổng hợp/đánh giá* | *10* |  |
| *Khả năng thiết kế chế tạo một hệ*  *thống, thành phần, hoặc quy trình đáp ứng yêu cầu đưa ra với những ràng buộc thực tế.* | *15* |  |
| *Khả năng cải tiến và phát triển* | *5* |  |
| *Khả năng sử dụng công cụ kỹ thuật, phần mềm chuyên ngành…* | *5* |  |
| **3.** | **Đánh giá về khả năng ứng dụng của đề tài.** | **10** |  |
| **4.** | **Sản phẩm cụ thể của ĐACĐT** | **10** |  |
|  | **Tổng** | **100** |  |

1. ***Kết luận:***

Được phép bảo vệ

Không được phép bảo vệ

TP.HCM, ngày.....tháng.....năm

Giảng viên phản biện

*(Ký, ghi rõ họ tên)*

LỜI CẢM ƠN

Lời đầu tiên nhóm xin gửi lời cảm ơn sâu sắc đối với thầy hướng dẫn đề tài là thầy Nguyễn Xuân Quang đã luôn đồng hành với nhóm trong quãng thời gian một học kỳ vừa qua. Thầy đã kịp thời có những giúp đỡ, sự định hướng và giải đáp những thắc mắc của nhóm trong lúc khó khăn khi thực hiện đồ án "Điều khiển tay máy ứng dụng trong xếp sản phẩm ".

Bên cạnh đó nhóm cũng xin cảm ơn các thầy cô thuộc khoa Cơ khí Chế tạo máy đã tận tình chỉ dẫn, trang bị những kiến thức cần thiết cho chúng em ngay từ khi bước vào giảng đường đại học đến nay. Những kiến thức đó là nền tảng rất quý báu và hữu ích, được sử dụng trong đồ án này.

Tuy nhiên vì kiến thức chuyên môn và mỗi cá nhân trong nhóm còn thiếu kinh nghiệm thực tế nên đề tài không tránh khỏi những thiếu sót. Nhóm rất mong được sự góp ý, chỉ bảo thêm của quý thầy cô để bài báo cáo này hoàn thiện hơn.

Cuối cùng, nhóm xin một lần nữa gửi lời cảm ơn chân thành đến Quý thầy cô.

*TP. Hồ Chí Minh, tháng 06 năm 2024*

Sinh viên thực hiện

Trần Ngọc Hiểu

Nguyễn Bá Vũ Thạch

Lê Tấn Lộc

TÓM TẮT ĐỀ TÀI

Sự bùng nổ của công nghệ robot trong những năm gần đây đang tạo nên một cuộc cách mạng lớn trong đời sống con người cũng như các ngành công nghiệp. Trong khi robot được biết đến rộng rãi là những thiết bị giúp việc vặt, chăm sóc sức khỏe hoặc hỗ trợ bán hàng, thì trên thực tế chúng đang đóng một vai trò quan trọng hơn nhiều trong sản xuất công nghiệp.

Các nhà máy, xí nghiệp trên khắp thế giới ngày càng trang bị nhiều robot hơn để thực hiện các công đoạn lặp đi lặp lại, nhờ đó tiết kiệm chi phí nhân công và nâng cao năng suất, chất lượng. Một trong những loại robot phổ biến nhất là cánh tay robot, thường được lập trình để thực hiện các công việc chế tạo, lắp ráp hay xếp hộp sản phẩm.

Tại Việt Nam, chúng ta mới chỉ lắp ráp được cánh tay robot nhập khẩu, việc thiết kế và sản xuất trong nước vẫn còn nhiều hạn chế. Do đó, đề tài nghiên cứu của nhóm sinh viên chúng em nhằm giải quyết bài toán thiết kế và chế tạo một cánh tay robot có khả năng điều khiển bằng cử động tay, ứng dụng trong xếp hộp sản phẩm.

**MỤC LỤC**

[**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 1**](#_Toc170723203)

[**PHIẾU NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP 3**](#_Toc170723204)

[**PHIẾU NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP 5**](#_Toc170723205)

[**LỜI CẢM ƠN 7**](#_Toc170723206)

[**TÓM TẮT ĐỀ TÀI 8**](#_Toc170723207)

[**CHƯƠNG 1: GIỚI THIỆU CHUNG 12**](#_Toc170723208)

[**1.1. Tính cấp thiết của đề tài 12**](#_Toc170723209)

[**1.2. Ý nghĩa khoa học và thực tiễn của đề tài 12**](#_Toc170723210)

[**1.2.1. Ý nghĩa khoa học 12**](#_Toc170723211)

[**1.2.2. Ý nghĩa thực tiễn 12**](#_Toc170723212)

[**1.3. Mục tiêu nghiên cứu của đề tài 13**](#_Toc170723213)

[**1.4. Đối tượng và phạm vi nghiên cứu 13**](#_Toc170723214)

[**1.4.1. Đối tượng nghiên cứu 13**](#_Toc170723215)

[**1.4.2. Phạm vi nghiên cứu 13**](#_Toc170723216)

[**1.5. Phương pháp nghiên cứu 13**](#_Toc170723217)

[**1.6. Kết cấu của đồ án tốt nghiệp 14**](#_Toc170723218)

[**CHƯƠNG 2: TỔNG QUAN NGHIÊN CỨU ĐỀ TÀI 15**](#_Toc170723219)

[**2.1. Giới thiệu về đối tượng nghiên cứu và các vấn đề liên quan dến đề tài nghiên cứu 15**](#_Toc170723220)

[**2.1.1. Cánh tay robot 5 bậc tự do 15**](#_Toc170723221)

[**2.1.2. Thiết bị theo dõi chuyển động 15**](#_Toc170723222)

[**2.1.3. Thuật toán nhận dạng vật thể 15**](#_Toc170723223)

[**2.2. Đặc tính của hệ thống 15**](#_Toc170723224)

[**2.3. Kết cấu của hệ thống 15**](#_Toc170723225)

[**2.4. Các nghiên cứu liên quan đến đề tài 15**](#_Toc170723226)

[**2.4.1. Nghiên cứu ngoài nước: 15**](#_Toc170723227)

[**2.4.2. Nghiên cứu trong nước 16**](#_Toc170723228)

[**2.5. Các tồn tại của hệ thống 16**](#_Toc170723229)

[**2.5.1. Các tồn đọng của hệ thống 16**](#_Toc170723230)

[**2.5.2. Phương hướng giải quyết 16**](#_Toc170723231)

[**CHƯƠNG 3: CƠ SỞ LÝ THUYẾT 17**](#_Toc170723232)

[**3.1. Cơ sở lý thuyết về điều khiển 17**](#_Toc170723233)

[**3.1.1. Lý thuyết về điều khiển robot 17**](#_Toc170723234)

[**3.1.1.1. Tính toán động học cho cánh tay robot 5 DOF 17**](#_Toc170723235)

[**3.1.1.1.1. Động học thuận 17**](#_Toc170723236)

[**3.1.1.1.2. Động học nghịch 19**](#_Toc170723237)

[**3.1.1.2. Tính toán động lực học cho cánh tay robot 5DOF 20**](#_Toc170723238)

[**3.1.2. Lý thuyết áp dụng trong thiết bị theo dõi chuyển động 29**](#_Toc170723239)

[**3.1.2.1. Biểu diễn định hướng 29**](#_Toc170723240)

[**3.1.2.2. Chuyển đổi từ Quaternion sang 3 góc Euler 30**](#_Toc170723241)

[**3.1.2.3. Chuyển đổi từ Quaternion sang ma trận xoay 30**](#_Toc170723242)

[**3.1.2.4. Kiến thức về vi điều khiển 31**](#_Toc170723243)

[**3.1.2.4.1. Giới thiệu về vi điều khiển ARM Cotex M 31**](#_Toc170723244)

[**3.1.2.5. Cảm biến gygroscope, cảm biến gia tốc và cảm biến từ trường 31**](#_Toc170723245)

[**3.1.2.5.1. Cảm biến gygroscope 31**](#_Toc170723246)

[**3.1.2.5.2. Cảm biến gia tốc 32**](#_Toc170723247)

[**3.1.2.5.3. Cảm biến từ trường 32**](#_Toc170723248)

[**3.1.2.6. Một số các thuật toán lọc nhiễu cho cảm biến gia tốc, gygroscope và từ trường 33**](#_Toc170723249)

[**3.2. Cơ sở lý thuyết về lập trình 34**](#_Toc170723250)

[**3.2.1. Lập trình WPF 34**](#_Toc170723251)

[**3.2.1.1. Cấu trúc cơ bản của WPF 34**](#_Toc170723252)

[**3.2.1.1.1. XAML 34**](#_Toc170723253)

[**3.2.1.1.2. Code-behind 34**](#_Toc170723254)

[**3.2.1.2. Giới thiệu về Helixtookit với WPF 35**](#_Toc170723255)

[**3.2.1.2.1. Khái niệm 35**](#_Toc170723256)

[**3.2.1.2.2. Tính năng chính của HelixToolkit 35**](#_Toc170723257)

[**3.2.2. Lập trình C 35**](#_Toc170723258)

[**3.2.2.1. Cấu trúc cơ bản của chương trình C 35**](#_Toc170723259)

[**3.2.2.2. Các khác niệm cơ bản trong chương trình C 36**](#_Toc170723260)

[**3.3. Cơ sở lý thuyết về xử lí ánh 37**](#_Toc170723261)

[**3.4. Cơ sở lý thuyết về điện – điện tử 37**](#_Toc170723262)

[**3.4.1. Giao tiếp UART 37**](#_Toc170723263)

[**3.4.2. Giao tiếp I2C 38**](#_Toc170723264)

[**3.4.3. Giao tiếp TCP/IP 38**](#_Toc170723265)

[**3.4.3.1. Định nghĩa 38**](#_Toc170723266)

[**3.4.3.2. Nguyên lý hoạt động 39**](#_Toc170723267)

[**CHƯƠNG 4: THIẾT KẾ PHẦN CỨNG HỆ THỐNG 42**](#_Toc170723268)

[**4.1. Thiết kế phần cứng cho cánh tay robot 42**](#_Toc170723269)

[**4.2. Thiết kế phần cứng cho băng tải 42**](#_Toc170723270)

[**4.3. Thiết kế phần cứng cho thiết bị theo dõi chuyển động 42**](#_Toc170723271)

[**CHƯƠNG 5: XÂY DỰNG PHẦN MỀM CHO HỆ THỐNG 42**](#_Toc170723272)

[**5.1. Xây dựng phần mềm cho cánh tay robot 42**](#_Toc170723273)

[**5.2. Xây dựng phần mềm cho băng tải 42**](#_Toc170723274)

[**5.3. Xây dựng phần mềm cho thiết bị theo dõi chuyển động 42**](#_Toc170723275)

[**5.4. Xây dựng giao diện giám sát, điều khiển cho hệ thống bằng WPF 42**](#_Toc170723276)

[**CHƯƠNG 6: KẾT QUẢ - THỰC NGHIỆM – HƯỚNG PHÁT TRIỂN 42**](#_Toc170723277)

CHƯƠNG 1: GIỚI THIỆU CHUNG

* 1. Tính cấp thiết của đề tài
* ****Tăng năng suất và hiệu quả sản xuất****: Tay máy có thể làm việc liên tục và không mệt mỏi, giúp tăng năng suất và giảm thời gian sản xuất so với lao động thủ công
* ****Giảm chi phí lao động****: Sử dụng tay máy có thể giảm chi phí lao động, đặc biệt trong các công việc lặp đi lặp lại và đòi hỏi độ chính xác cao. Việc này cũng giúp giảm bớt sự phụ thuộc vào nguồn lao động thủ công, vốn có thể khan hiếm hoặc không ổn định
* ****Đảm bảo an toàn lao động****: Trong các ngành công nghiệp có môi trường làm việc nguy hiểm, việc sử dụng tay máy giúp bảo vệ sức khỏe và an toàn cho người lao động, giảm thiểu nguy cơ tai nạn lao động
* ****Nâng cao chất lượng sản phẩm****: Tay máy có khả năng làm việc với độ chính xác cao, giúp giảm thiểu sai sót và nâng cao chất lượng sản phẩm.
* ****Ứng dụng công nghệ cao****: Đề tài ứng dụng xử lý ảnh để nhận dạng vật thể nhằm phân loại sản phẩm trong công nghiệp, thiết kế và phát triển thiết bị theo dõi chuyển động kết hợp giữa chuyển động của người - máy nhằm nâng cao hiệu quả sản xuất và tạo ra các cơ hội nghiên cứu và phát triển mới.
  1. Ý nghĩa khoa học và thực tiễn của đề tài
     1. Ý nghĩa khoa học
* ****Mở rộng ứng dụng của tự động hóa trong sản xuất****: Kết quả nghiên cứu có thể áp dụng vào nhiều lĩnh vực sản xuất khác nhau, từ đó mở rộng phạm vi ứng dụng của công nghệ tự động hóa, đặc biệt là trong việc xếp sản phẩm
* ****Phát triển các thuật toán và mô hình mới****: Đề tài này có thể dẫn đến sự phát triển của các thuật toán điều khiển và mô hình mô phỏng mới, góp phần nâng cao hiệu suất và độ chính xác của các hệ thống tay máy
* ****Nghiên cứu và phát triển công nghệ điều khiển mới****: Đề tài này tập trung vào việc phát triển các phương pháp điều khiển tiên tiến cho tay máy, góp phần nâng cao hiểu biết về cơ chế hoạt động và khả năng ứng dụng của các hệ thống robot trong công nghiệp
  + 1. ****Ý nghĩa thực tiễn****
* ****Tối ưu hóa quy trình xếp sản phẩm****: Đề tài giúp cải thiện các quy trình xếp sản phẩm hiện tại, làm cho chúng trở nên hiệu quả và tiết kiệm thời gian hơn, đồng thời giảm thiểu sự cố và lỗi trong quá trình sản xuất
* ****Thúc đẩy sự đổi mới và sáng tạo trong sản xuất****: Việc nghiên cứu và ứng dụng các phương pháp điều khiển tay máy tiên tiến sẽ thúc đẩy sự đổi mới và sáng tạo trong sản xuất, giúp các doanh nghiệp duy trì và nâng cao sức cạnh tranh trên thị trường
* ****Tạo điều kiện thuận lợi cho việc triển khai công nghệ mới****: Đề tài cung cấp cơ sở lý thuyết và thực tiễn cho việc triển khai các công nghệ điều khiển tay máy mới, giúp các doanh nghiệp dễ dàng áp dụng các giải pháp công nghệ hiện đại vào quy trình sản xuất của mình
  1. Mục tiêu nghiên cứu của đề tài
* Thiết kế giao diện sử dụng WPF nhằm giám sát, điều khiển cũng như mô phỏng robot một cách tùy biến và linh hoạt.
* Tính toán, thiết kế và chế tạo thiết bị theo dõi chuyển động, kết hợp chuyển động từ cơ thể người để thực hiện điều khiển cánh tay robot.
* Xây dựng thuật toán nhận diện và phân loại vật thể sử dụng camera công nghiệp.
* Xây dựng hệ thống tự động hóa trong bốc xếp và phân loại sản phẩm.
* Tạo quỹ đạo từ chuyển động của cơ thể người áp dụng cho việc điều khiển cánh tay robot.
  1. Đối tượng và phạm vi nghiên cứu
     1. Đối tượng nghiên cứu
* Cánh tay robot 5 bậc tự do
* Thiết bị theo dõi chuyển động
* Thuật toán nhận dạng vật thể
  + 1. Phạm vi nghiên cứu
* Nghiên cứu, thiết kế thiết bị theo dõi chuyển động.
* Điều khiển cánh tay robot 5 bậc tự do sử dụng PLC Mitsubishi.
* Thiết kế và lập trình thuật toán nhận dạng vật thể sử dụng camera công nghiệp.
* Điều khiển băng tải và lấy dữ liệu từ cảm biến dùng PLC Siemens.
* Thiết kế, lập trình giao diện hệ thống với WPF.
  1. Phương pháp nghiên cứu
* Tra cứu tài liệu các tài liệu của nhà sản xuất cũng như bên ngoài để hỗ trợ trong việc điều khiển robot và thuật toán nhận dạng vật thể.
* Kiểm tra các cơ sở lý thuyết bằng cách tính toán, thực nghiệm điều khiển robot nhằm so sánh và đối chiếu.
  1. Kết cấu của đồ án tốt nghiệp

Báo cáo đồ án tốt nghiệp được chia làm 6 chương, với các nội dung:

CHƯƠNG 1: GIỚI THIỆU CHUNG: Trình bày rõ tính cấp thiết và mục đích chọn đề tài, mục tiêu nghiên cứu đề tài, đối tượng nghiên cứu và phạm vi nghiên cứu và đưa ra được những phương pháp được áp dụng để nghiên cứu đề tài.

CHƯƠNG 2: TỔNG QUAN NGHIÊN CỨU ĐỀ TÀI: Đưa ra được những đặc tính và kết cấu của hệ thống điều khiển, liên hệ đến những nghiên cứu liên quan đến đề tài và nêu ra những gì còn tồn đọng của hệ thống điều khiển từ đó đặt ra những giải pháp và cải tiến phù hợp.

CHƯƠNG 3: CƠ SỞ LÝ THUYẾT: Trình bày các lý thuyết được sử dụng để giải quyết vấn đề trong đề tài và hệ thống hóa các công thức cần sử dụng.

CHƯƠNG 4: THIẾT KẾ PHẦN CỨNG HỆ THỐNG: Thiết kế phần cứng cho cánh tay robot, cho băng tải và thiết bị theo dõi chuyển động. Bao gồm: Thiết kế cơ khí và thiết kế điện.

CHƯƠNG 5: XÂY DỰNG PHẦN MỀM CHO HỆ THỐNG: Xây dựng phần mềm cho cánh tay robot, băng tải kết hợp cảm biến và thiết bị theo dõi chuyển động.

CHƯƠNG 6: KẾT LUẬN, HƯỚNG PHÁT TRIỂN: Kiểm nghiệm lại kết quả và đúc kết lại quá trình thực hiện đồ án tốt nghiệp.

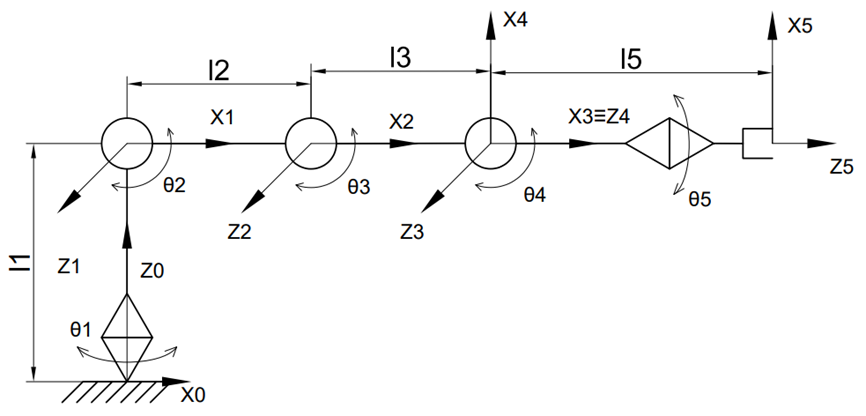
CHƯƠNG 2: TỔNG QUAN NGHIÊN CỨU ĐỀ TÀI

* 1. Giới thiệu về đối tượng nghiên cứu và các vấn đề liên quan dến đề tài nghiên cứu
     1. Cánh tay robot 5 bậc tự do
* Tính toán và điều khiển vị trí cho robot.
* Tính toán và điều khiển vận tốc cho robot.
* Tính toán, thiết kế và điều khiển quỹ đạo mong muốn cho robot.
* Tính toán và điều khiển robot ứng dụng trong bốc xếp hàng hóa theo một dây chuyền tự động.
  + 1. Thiết bị theo dõi chuyển động
* Tương tác giữa người và máy nhằm tối ưu hóa quá trình tạo quỹ đạo cho robot
  + 1. Thuật toán nhận dạng vật thể
* Nhận dạng các loại hàng hóa khác nhau nhằm tự động hóa quá trình sản xuất và phân loại sản phẩm.
  1. Đặc tính của hệ thống
* Khả năng điều khiển và tùy chỉnh linh hoạt
* Độ an toàn cao trong quá trình vận hành
* Tốc độ nhận dạng vật thể và xử lí cao
  1. Kết cấu của hệ thống
* Chương trình điều khiển chính
* Bộ điều khiển PLC và cánh tay robot công nghiệp
* Cụm camera và băng tải kết hợp với cảm biến công nghiệp
* Thiết bị tương tác giữa người và máy
  1. Các nghiên cứu liên quan đến đề tài
     1. Nghiên cứu ngoài nước:
* Robot bán tự động vận hành từ xa với khả năng tránh chướng ngại vật thông qua điều khiển dự đoán mô hình: https://www.researchgate.net/publication/333335847\_Semi-Autonomous\_Robot\_Teleoperation\_With\_Obstacle\_Avoidance\_via\_Model\_Predictive\_Control
* Hệ thống theo dõi chuyển động cánh tay người không dây 7DOF: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7309037/
* Thuật toán điều khiển IMU và AHRS: https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/
* Thuật toán lọc nhiễu Kanman: <https://kdientu.duytan.edu.vn/media/133474/bo-loc-kalman.pdf>
  + 1. Nghiên cứu trong nước
* Thiết kế hệ thống điều khiển mô phỏng chuyển động cánh tay: https://www.slideshare.net/man2017/thit-k-h-thng-iu-khin-m-phng-chuyn-ng-cnh-taypdf
* TAY MÁY 5 BẬC TỰ DO VÀ ĐIỀU KHIỂN CHUYỂN ĐỘNG BẰNG CỬ ĐỘNG TAY ỨNG DỤNG TRONG XẾP SẢN PHẨM
  1. Các tồn tại của hệ thống
     1. Các tồn đọng của hệ thống
* Giao diện còn đơn giản, chưa thân thiện với người dùng, chưa có khả năng giám sát, mô phỏng robot.
* Cách điều khiển bằng găng tay chưa ổn định, khó có thể áp dụng vào thực tế.
* Hệ thống nhận diện vật thể còn phụ thuộc vào tool của hãng, nên chưa thể áp dụng linh hoạt vào thực tế.
  + 1. Phương hướng giải quyết
* Sử dụng WPF thay cho Winform để tăng khả năng tùy biến của ứng dụng, viết thêm tính năng giám sát robot trên tool.
* Thiết kế lại thiết bị điều khiển robot dựa trên cử động của cánh tay thay vì dùng ngón tay
* Thực hiện lấy ảnh trực tiếp từ camera và viết lại thuật toán nhận dạng vật thể sử dụng C#

# CHƯƠNG 3: CƠ SỞ LÝ THUYẾT

* 1. Cơ sở lý thuyết về điều khiển
     1. Lý thuyết về điều khiển robot
        1. Tính toán động học cho cánh tay robot 5 DOF
           1. Động học thuận

Sơ đồ động Robot:

****

*Sơ đồ động Robot*

**Biểu diễn Denavit – Hartenberg:**

Mô hình hóa Denavit-Hartenberg (Viết tắt là phương pháp D - H) là cách biểu diễn đơn giản mô hình các khâu và khớp của robot và có thể sử dụng cho bất cứ cấu hình robot nào, kể cả bài toán phức tạp hay đơn giản.

Áp dụng phương pháp D – H, ta gắn một hệ trục tham chiếu tới mỗi khớp và sau đó xác định sự chuyển vị từ khớp này đến khớp kế tiếp, với mỗi khớp chúng ta sẽ gắn trục z và trục x lên chúng (hình 3.1).

Áp dụng phương pháp D - H cho khớp 1, tương tự với các khớp còn lại. Ta có:

* Đánh số trục z của khớp 1 là 0. Như vậy, trục z biểu diễn khớp 1 là z0. Trục x sẽ song song với đường vuông góc chung giữa các trục khớp của khâu, tương tự với các khớp còn lại.
* Với:
* l: biểu diễn khoảng cách dường vuông góc chung giữa trục z0­ và trục z1.
* θ1: góc quay từ trục x0 tới x1 xung quanh trục z­0.
* α1: góc quay của trục z0 tới z1 xung quanh trục x1.
* d1: khoảng cách từ gốc tọa độ thứ 0 tới giao điểm của của trục z0 và x1 dọc theo trục z0.

🡪 Từ đó, ta có thể lập ra bảng D – H với 5 khớp một cách tương ứng dựa theo hình 3.1:

Bảng D-H:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Khâu** | **θ** | **l** | **α** | **d** | **Biến** |
| **1** |  | **0** |  | **l1** |  |
| **2** |  |  | **0** | **0** |  |
| **3** |  |  | **0** | **0** |  |
| **4** |  | **0** |  | **0** |  |
| **5** |  | **0** | **0** | **l5** |  |

**Bảng Denavit–Hartenberg của robot**

Ở hệ trục thứ i sẽ mô tả ma trận vị trí và hướng so với hệ trục thứ i-1 như sau:

Ma trận chuyển trục Ai:

** (1)**

Với **.**

Từ đó, ta có ma trận chuyển trục của 5 khớp từ A1 đến A5 như sau:

** **

** **

****

Với kết quả trên, ta có ma trận chuyển vị tổng giữa nền Robot và cánh tay là:

****

**=**

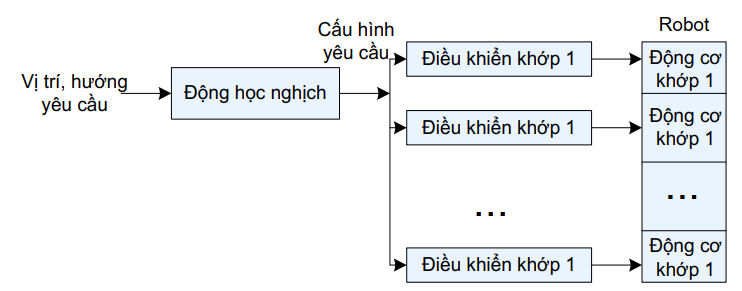
** (2)**

Theo động học thuận, tọa độ điểm P là:



* + - * 1. Động học nghịch

Bài toán động học thuận gồm việc giải quyết vị trí và hướng của tay gắp hay cơ cấu chấp hành cuối khi biết tất cả các biến khớp. Bài toán động học nghịch yêu cầu tìm tập hay nhiều tập nghiệm của các khớp khi biết vị trí và hướng tay gắp.



*Điều khiển vị trí của cánh tay robot*

Với các giá trị đầu vào là Px, Py, P­z, φ, γ, Ta có:





(4)



Ta đặt:



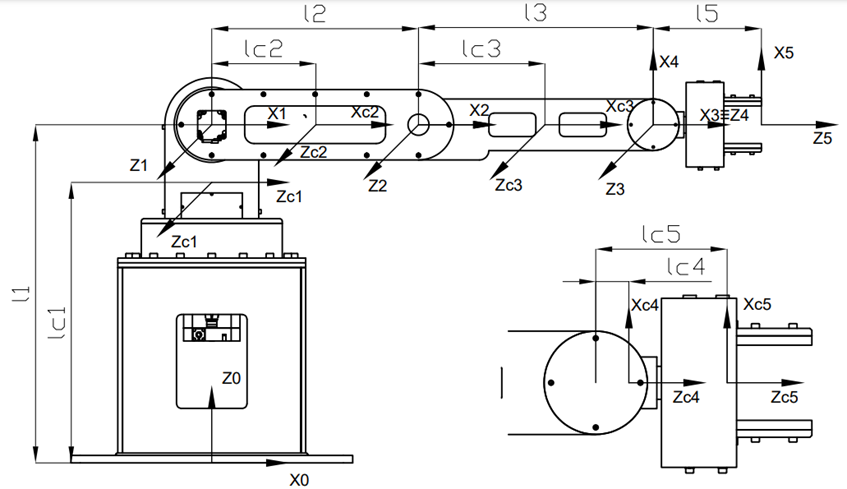




(6)



* + - 1. Tính toán động lực học cho cánh tay robot 5DOF

****

*Sơ đồ tính động học robot*

**Phương trình động lực học tổng quát:**



Hay



Với







là ma trận quán tính

là vector hướng tâm

là vector trọng lực

**Tính ma trận quán tính :**

Ta có:



Ma trận chỉ hướng Ri là ma trận lấy các phần tử của 3 hàng và 3 cột đầu của ma trận chuyển vị Ti (đã tính ở phần động học thuận)

Ma trận moment quán tính Ii (là ma trận đặt trưng cho moment quán tính của các khâu)

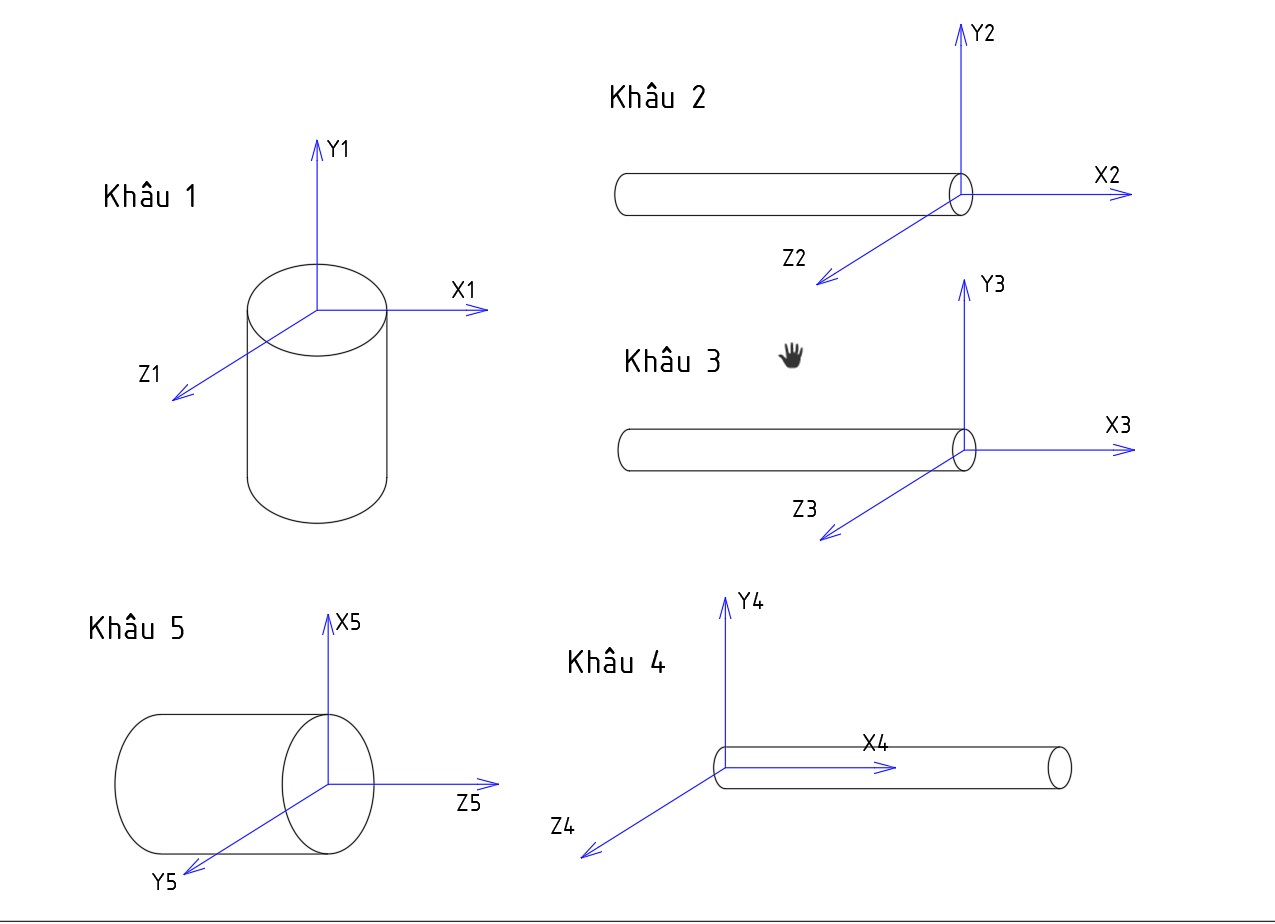
Để đơn giản hóa việc tính toán, giả sử các khâu đều đối xứng qua các trục xx, yy, zz

Ta có

Xét hình dáng của các khâu 2,3,4 là thanh thẳng có tiết diện ngnang không đáng kể.

Xét hình dáng của các khâu 1,5 là hình trụ tròn.

Ta có hình dáng tổng quát cùng vị trí đặt trục tọa độ như sau:



*Hình 3.16: Hình dáng tổng quát để tính Moment quán tính của các khớp*

Ta có:

I1: ,

I2: ,

I3: ,

I4: ,

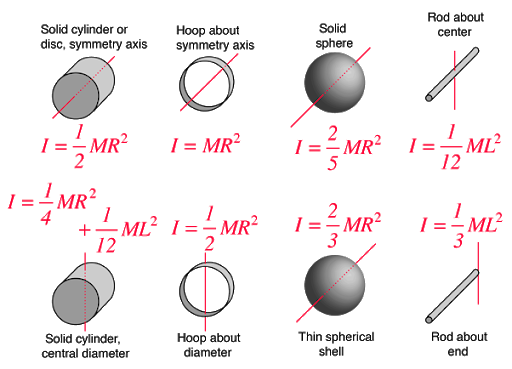
I5: ,

Tổng hợp lại ta có:

; ; ;

;

Sử dụng công thức tính moment quán tính cho các khối cơ bản ta xác định được giá trị của các phần tử trong ma trận I.



*Các công thức tính moment quán tính cho các khối cơ bản*

Ma trận biểu diễn vi trí trọng tâm khâu so với hệ tọa độ gốc O0 (Các giá trị được tính bằng phương pháp hình học)

; ; ;

**Ma trận Jacobi khâu**

Ma trận Jacobi vận tốc gốc :

0 với

Ta có:

; ; ; ;

0; 0; 0; 0;

0;

**Ma trận Jacobi vận tốc dài :**

***Kết quả:***

(22)

(23)

(24)



Với giá trị i chạy từ 1 đến 5, thay các giá trị vừa tính được vào biểu thức ban đầu, ta tìm được ma trận quán tính D(q).

Tính Vector hướng tâm

với k chạy từ 1 đến 5 và i và j thể hiện vị trí phần tử của ma trận quán tính

- Tính Vector trọng lực G(q):

với

*P* là thế năng tổng cộng của cánh tay được tính bằng

🡪 với

Mà chính là cột thứ k trong ma trận

Tiến thành tính toán số liệu cụ thể

Ta có các thông số đã biết:

* m1=27,5 [kg]; m2=21 [kg]; m3=25,1 [kg]; m4=3 [kg]; m5=5,2 [kg];
* l1=690 [mm]; l2=440 [mm]; l3=500 [mm]; l5=230 [mm];
* lc1=660 [mm]; lc2=255 [mm]; lc3=143 [mm]; lc4=6 [mm]; lc5=143 [mm];
* g=9,81 [m/s2]
* Ixx1=461147 N.mm2; Iyy1=299920 N.mm2; Iyy2=803322 N.mm2; Iyy3=1604504 N.mm2 Iyy4=6328 N.mm2; Iyy5=24861 N.mm2; Izz5=14388 N.mm2

Vì khối lượng tính toán rất phức tạp nên ta sẽ sử dụng MATLAB để tăng độ chính xác và tin cậy cho quá trình tính toán. Ta sẽ thế giá trị trực tiếp và áp dụng thêm điều kiện ràng buộc để kết quả tính toán ngắn gọn nhất có thể.

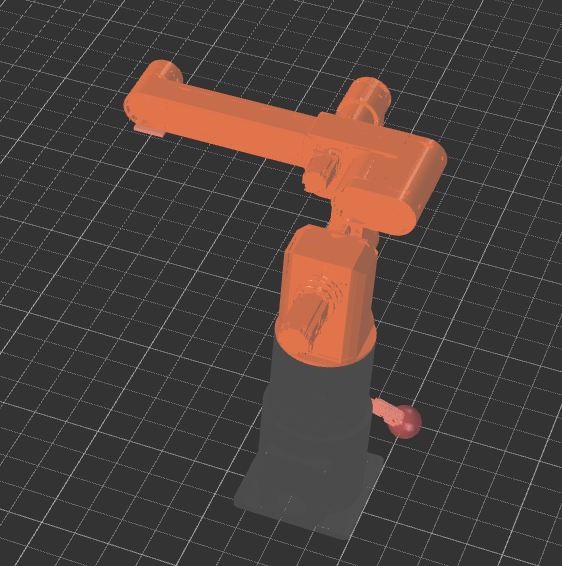
Xác định tải tĩnh lớn nhất mà cánh tay robot chịu được:

Xét trường hợp cánh tay robot ở trạng thái tĩnh, khi đó các giá trị vận tốc và gia tốc sẽ bằng 0, từ đó dẫn đến . Phương trình động lực học khi đó chỉ còn phụ thuộc vào vector trọng lực :

= (49\*m4\*(l3\*cos(t2 + t3) + l2\*cos(t2)))/5 + (49\*m5\*(l3\*cos(t2 + t3) + l2\*cos(t2)))/5 + (49\*m3\*(lc3\*cos(t2 + t3) + l2\*cos(t2)))/5 + (49\*lc2\*m2\*cos(t2))/5

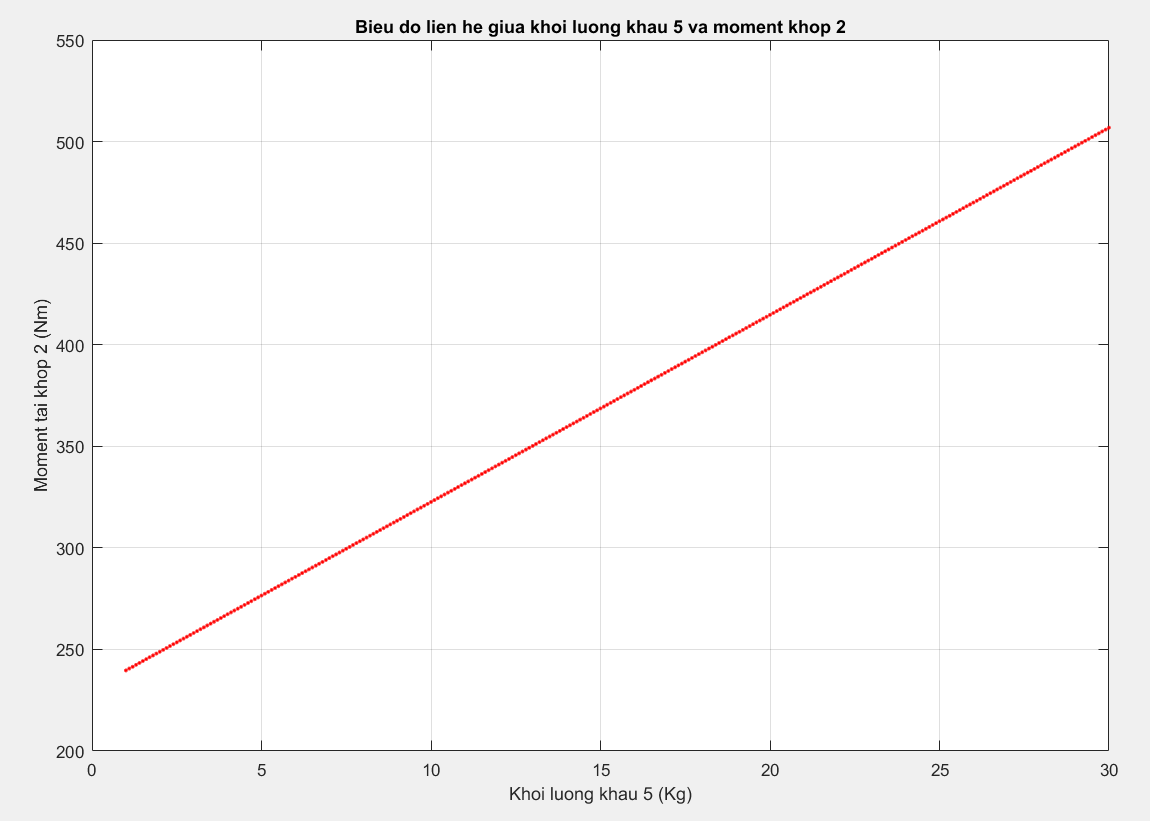
= (49\*cos(t2 + t3)\*(l3\*m4 + l3\*m5 + lc3\*m3))/5

Khi đó, ta xác định vị trí mà cánh tay robot chịu moment lớn nhất. Về nguyên tắc, moment được xác định bằng tích của lực và độ dài cánh tay đòn. Lực ở đây chính là trọng lực của vật thể cần gắp, cánh tay đòn là khoảng cách từ vật gắp đến gốc tọa độ ban đầu của robot. Ta xác định được cấu hình mà cánh tay chịu lực lớn nhất đó chính là khi cánh tay bị duỗi thẳng như hình:

****

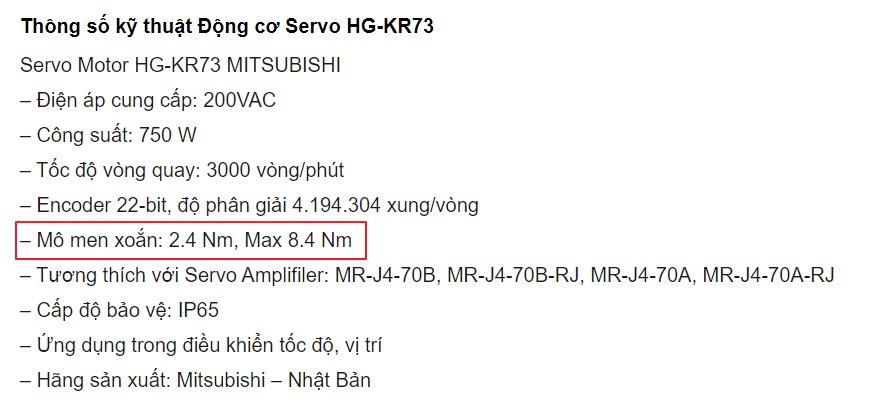
*Vị trí của robot mà động cơ chịu moment lớn nhất thõa mãn đầu tay gắp vuông góc với mặt đất*

Với khâu 5 gắn với vật cần gắp nên ta đặt là 1 biến số. Cho biến m5 tăng dần, ta thu được biểu đồ liên hệ giữa khối lượng vật càn gắp và moment phải chịu của các động cơ:

****

Mối liên hệ giữa khối lượng khâu 5 và moment phải chịu của khớp 2

Tra thông số động cơ tại khớp 2 của robot (HG-KR73) ta có:

****

*Thông số động cơ tại khớp 2*

Bộ truyền Hamornic có tỉ số truyền 1:180, cho hiệu suất bộ truyền, do đó moment mà khớp 2 chịu được sẽ là:

T2max = 2.4 \* 180 = 432 Nm

Suy ra khối lượng lớn nhất của khâu số 5 mà động cơ vẫn chịu được là 19Kg

Từ đó, vật lớn nhất mà robot gắp được sẽ khoảng 17kg.

**Ma trận Jacobi nghịch sử dụng phương pháp The Pseudo-Inverse:**

Từ ma trận Jacobi thuận Jv5 . Vì ma trận cho robot 5 bậc tự do là một ma trận không vuông, ta không thể tính Jacobi nghịch theo cách thông thường. Ta sẽ áp dụng công thức Pseudo-Inverse:



Với  và là vận tốc dài của robot trong tọa độ Oxyz, từ dó ta sẽ có được ma trận chứa vận tốc cho từng khớp của robot.

Từ những kiến thức nêu trên, ta có thể áp dụng đưa vào hệ thống điều khiển để tính toán cũng như đưa ra những giải thuật điều khiển robot về vị trí cũng như vận tốc, đảm bảo tính linh hoạt và hiệu quả trong quy trình hoạt động.

* + 1. Lý thuyết áp dụng trong thiết bị theo dõi chuyển động
       1. Biểu diễn định hướng

Định hướng trong không gian có thể được biểu diễn bởi ba phương pháp khác nhau là dùng góc Euler (pitch, roll, yaw), Ma trận xoay hoặc Quaternion.

* Quaternion được định nghĩa như là một số phức có ba thành phần ảo:

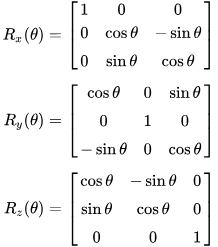
\boldsymbol{q} = w + x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k} 

Các thành phần ảo i, j, k  có thể được coi như ba vector đơn vị của trục tọa độ x, y, z. Và một quaternion có thể được viết lại như sau:

\boldsymbol{q} = w + \boldsymbol{v} 

Với \boldsymbol{v} là một vector trong không gian 3 chiều, và w là một đại lượng vô hướng chỉ độ lớn (nói chung là một con số).

* Biểu diễn góc Euler bao gồm 3 phép xoay:
* Một phép xoay ϕ quanh trục x (góc roll)
* Một phép xoay θ quanh trục y (góc pitch)
* Một phép xoay ψ quanh trục z (góc yaw)
* Ma trận xoay là một ma trận 3x3 biểu diễn 3 vector đơn vị tạo ra 9 tham số được sử dụng cho việc định hướng trong không gian. Ta có các giá trị ma trận khi xoay quanh 3 trục x, y, z một góc θ như sau:



* + - 1. Chuyển đổi từ Quaternion sang 3 góc Euler

Góc pitch (góc xoay quanh trục X):

pitch = atan2(2 \* (yz + wx), ww - xx - yy + zz)

Góc roll (góc xoay quanh trục Y):

roll = asin(2 \* (xz - wy))

Góc yaw (góc xoay quanh trục Z):

yaw = atan2(2 \* (xy + wz), ww + xx - yy - zz)

Trong các công thức trên, hàm atan2(x, y) tính arctan(x/y) với đầu ra nằm trong khoảng [-π, π], và hàm asin(x) tính arcsin(x) với đầu ra nằm trong khoảng [-π/2, π/2].

* + - 1. Chuyển đổi từ Quaternion sang ma trận xoay

Chuẩn hóa các giá trị w, x, y, z về range [0, 1]:

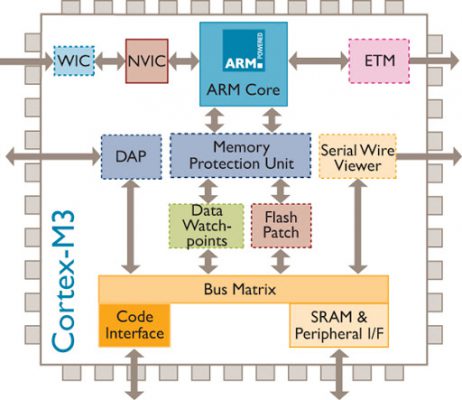


Tính giá trị ma trận xoay:

 với

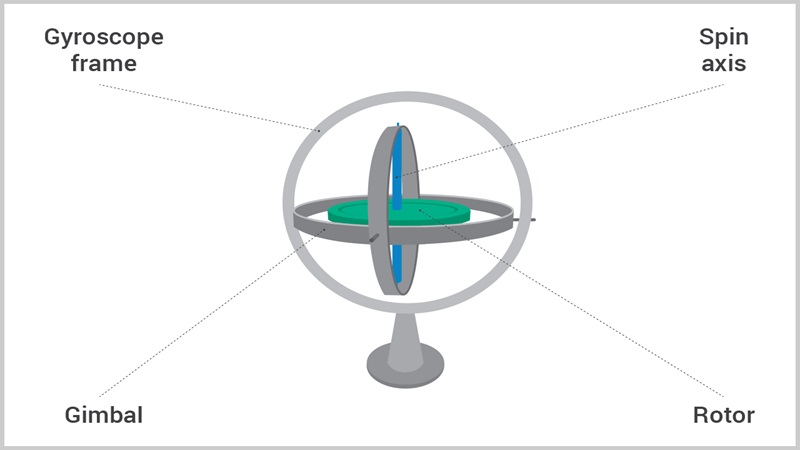


* + - 1. Kiến thức về vi điều khiển
         1. Giới thiệu về vi điều khiển ARM Cotex M



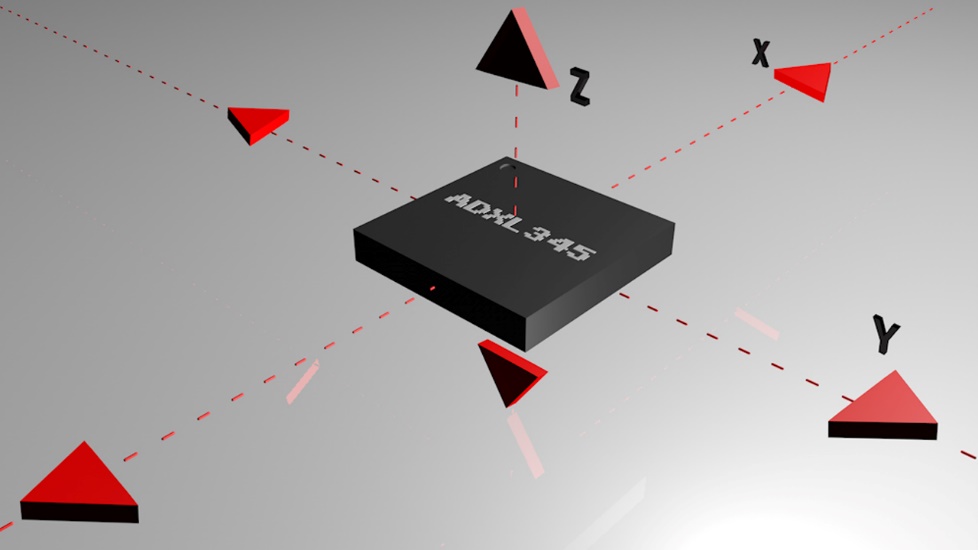
*Kiến trúc của ARM Cortex M3 – Một trong những dòng vi điều khiển sử dụng rất phổ biến hiện nay*

* + - 1. Cảm biến gygroscope, cảm biến gia tốc và cảm biến từ trường
         1. Cảm biến gygroscope

Cảm biến con quay hồi chuyển, được biết đến như là "gyroscope sensor" trong tiếng Anh, là một thiết bị dùng để đo và xác định góc quay hoặc sự thay đổi góc quay. Nó được thiết kế đặc biệt để cung cấp thông tin về ba trục không gian: trục x, trục y và trục z. Điều này cho phép cảm biến có khả năng chính xác xác định hướng và chuyển động của một thiết bị trong không gian.

* + - * 1. **Cảm biến gia tốc**

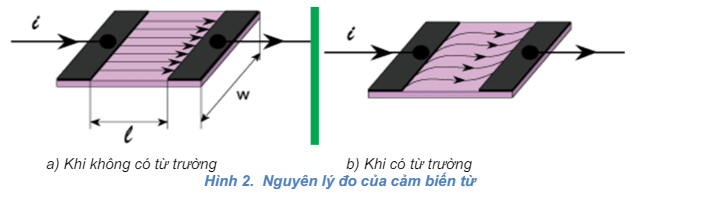
Cảm biến gia tốc, còn được gọi là cảm biến gia tốc kế, là một thiết bị điện tử được sử dụng để đo gia tốc, tức là thay đổi về tốc độ hoặc hướng chuyển động. Chức năng chính của cảm biến này là phát hiện và ghi lại thông tin về tăng tốc, giảm tốc và thay đổi hướng di chuyển của vật thể.

Cảm biến gia tốc kế cho phép thu thập dữ liệu về các biến đổi gia tốc này, giúp các thiết bị điện tử hiểu và phản ứng với các chuyển động xung quanh. Ứng dụng của cảm biến này rất đa dạng và được áp dụng trong nhiều lĩnh vực, từ công nghiệp ô tô để giám sát và kiểm soát động cơ, đến điện thoại thông minh và thiết bị y tế để theo dõi hoạt động vận động và sức khỏe của người dùng.

* + - * 1. Cảm biến từ trường

La bàn số sử dụng hai cảm biến từ trường nằm vuông góc nhau trên mặt phẳng nằm ngang để đo cường độ từ trường của trái đất. Tỉ số hai giá trị đo này cho phép tính được góc giữa các trục của cảm biến với đường sức từ của trái đất, đó chính là hướng (số chỉ) của la bàn từ. Trên thị trường hiện nay có nhiều loại cảm biến từ. Thông thường các cảm biến từ này có gắn 3 cảm biến dọc theo ba trục của hệ tọa độ Decac và có thể đo với độ nhạy rất cao. Cảm biến được chế tạo bằng hợp chất InSb (hợp chất của In – Indium và Sb – Antimony)[4] . Hợp chất này có tính chất như sau:

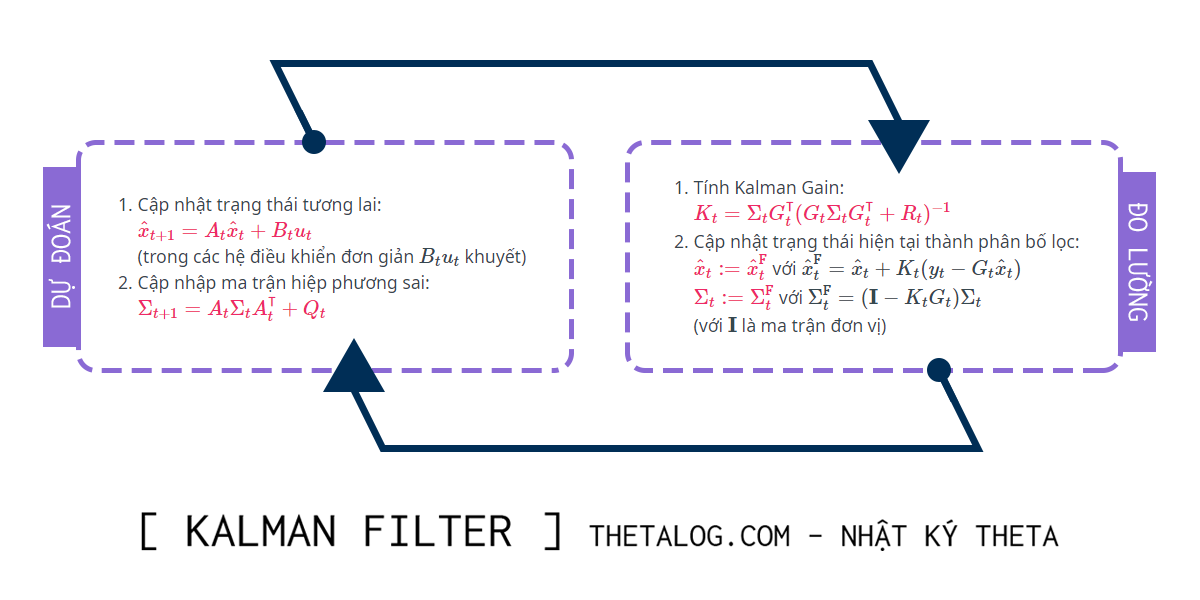
Khi cho dòng điện chạy qua hợp chất, nếu không có từ trường các điện tử sẽ chuyển động theo đường thẳng. Khi đặt hợp chất trong từ trường, dưới tác động của từ trường, các điện tử sẽ chuyển động trên các đường chéo, quãng đường chuyển động này dài hơn làm cho điện trở của vật liệu tăng lên. Người ta sử dụng hiện tượng này để chế tạo cảm biến đo cường độ từ trường.



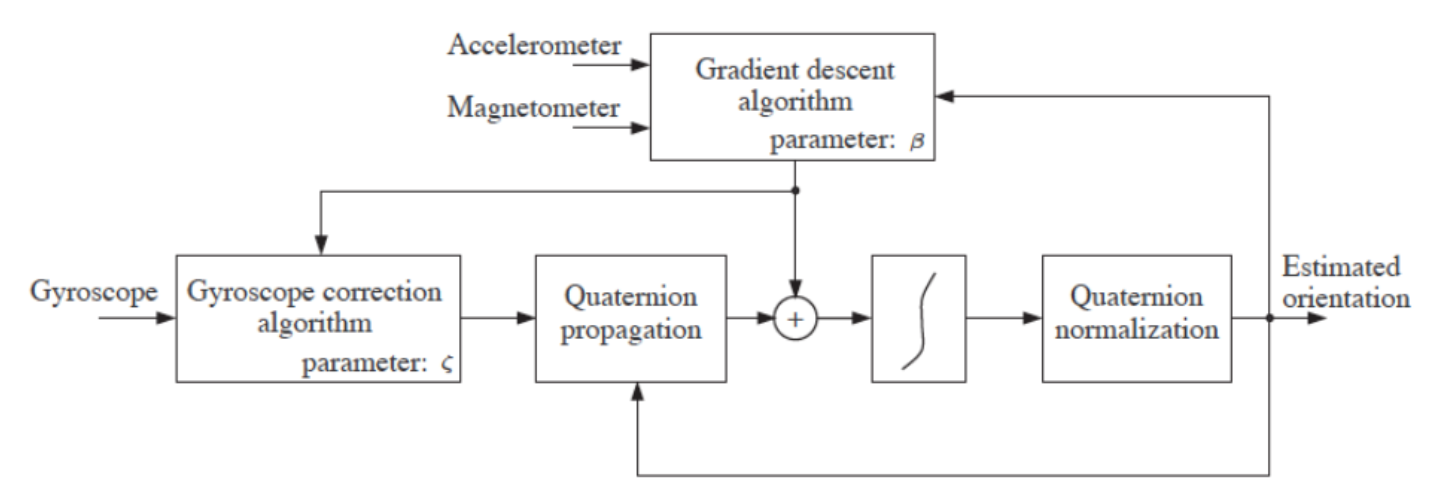
* + - 1. **Một số các thuật toán lọc nhiễu cho cảm biến gia tốc, gygroscope và từ trường**

Dưới đây là một số thuật toán lọc nhiễu được sử dụng cho cảm biến gyroscope, gia tốc và từ trường:

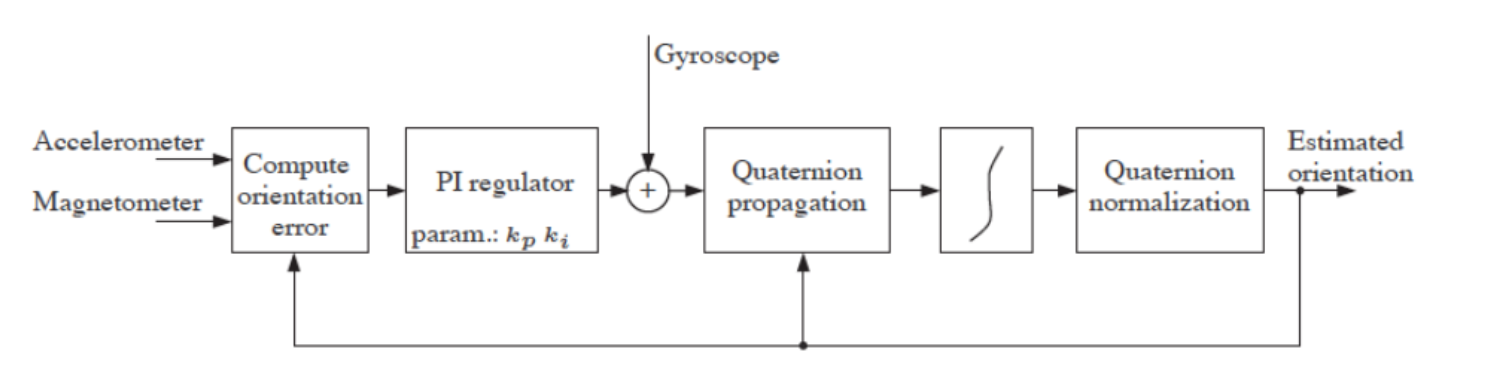
**Thuật toán Kalman (Kalman filter):** Thuật toán lọc Kalman là một phương pháp ước tính tối ưu dựa trên việc kết hợp dữ liệu đo và mô hình hệ thống. Nó sử dụng một bộ lọc tuyến tính và phân phối xác suất để ước tính trạng thái và hiệu suất của hệ thống.



**Thuật toán Madgwick:** Thuật toán Madgwick là một phương pháp lọc bù dựa trên các đặc điểm của cảm biến quay (gyroscope) và gia tốc (accelerometer). Nó sử dụng một bộ lọc không tuyến tính để ước tính trạng thái và hướng của hệ thống dựa trên dữ liệu đo.



**Thuật toán Mahony:** Thuật toán Mahony cũng là một phương pháp lọc bù sử dụng cảm biến gyroscope và accelerometer. Nó tập trung vào ước tính hướng của hệ thống và sử dụng một bộ lọc không tuyến tính để giảm nhiễu và lỗi ước tính.



* 1. Cơ sở lý thuyết về lập trình
     1. Lập trình WPF

Windows Presentation Foundation (WPF) là một công nghệ của Microsoft dùng để xây dựng các ứng dụng giao diện đồ họa trên Windows. WPF cung cấp một hệ thống mạnh mẽ và linh hoạt để phát triển ứng dụng, sử dụng XAML (Extensible Application Markup Language) kết hợp với C# để thiết kế giao diện người dùng và xử lý logic.

* + - 1. Cấu trúc cơ bản của WPF
         1. XAML

Là ngôn ngữ đánh dấu XML-based dùng để định nghĩa và khởi tạo các đối tượng WPF.

<Window x:Class="WpfApp.MainWindow"

        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

        Title="MainWindow" Height="350" Width="525">

    <Grid>

        <Button Content="Click Me" HorizontalAlignment="Left" VerticalAlignment="Top" Width="100" Height="50" Click="Button\_Click"/>

    </Grid>

</Window>

* + - * 1. Code-behind

Là file C# chứa logic xử lý cho giao diện người dùng được định nghĩa trong XAML. Code-behind và XAML được liên kết với nhau thông qua thuộc tính ‘x:Class’

using System.Windows;

namespace WpfApp

{

    public partial class MainWindow : Window

    {

        public MainWindow()

        {

            InitializeComponent();

        }

        private void Button\_Click(object sender, RoutedEventArgs e)

        {

            MessageBox.Show("Button clicked!");

        }

    }

}

* + - 1. Giới thiệu về Helixtookit với WPF
         1. Khái niệm

Là một thư viện mã nguồn mở bổ sung cho WPF, cung cấp các công cụ và thành phần để làm việc với đồ họa 3D một cách dễ dàng và mạnh mẽ hơn. HelixToolkit mở rộng khả năng của WPF trong việc xử lý và hiển thị các mô hình 3D.

* + - * 1. Tính năng chính của HelixToolkit
* **Dễ dàng hiển thị mô hình 3D:** Cung cấp các điều khiển và tiện ích để hiển thị các mô hình 3D với ít mã cần viết.
* **Hỗ trợ nhiều định dạng mô hình 3D:** Hỗ trợ đọc và hiển thị các định dạng mô hình phổ biến như .obj, .3ds, .stl, .lwo.
* **Các công cụ tương tác:** Bao gồm các công cụ để di chuyển, xoay, phóng to, thu nhỏ các mô hình 3D.
* **Hiệu ứng và ánh sáng:** Cung cấp các hiệu ứng và ánh sáng để cải thiện việc hiển thị mô hình 3D.
  + 1. Lập trình C

Lập trình C là một trong những ngôn ngữ lập trình phổ biến và cơ bản nhất, được phát triển vào những năm 1970 bởi Dennis Ritchie tại Bell Labs. Ngôn ngữ C nổi bật với tính linh hoạt, hiệu suất cao và khả năng kiểm soát phần cứng, khiến nó trở thành một lựa chọn phổ biến trong nhiều lĩnh vực, từ phát triển hệ điều hành đến lập trình nhúng.

* + - 1. Cấu trúc cơ bản của chương trình C

Một chương trình C cơ bản gồm các thành phần sau:

* ****Tiêu đề thư viện****: Các thư viện chuẩn được đưa vào để cung cấp các hàm và macro hữu ích.
* ****Hàm chính (main)****: Điểm bắt đầu của chương trình, nơi thực hiện các lệnh và xử lý logic.

#include <stdio.h>

int main() {

    printf("Hello, World!\n");

    return 0;

}

* + - 1. Các khác niệm cơ bản trong chương trình C

Một số khái niệm cơ bản cần được nắm trong ngôn ngữ lập trình C:

* Kiểu dữ liệu: Các kiểu dữ liệu cơ bản như kiểu số thực, kiểu số nguyên, kiểu ký tự và kiểu con trỏ.
* Toán tử và biểu thức: C hỗ trợ nhiều loại toán tử để thực hiện các phép toán số học, logic và so sánh.
* Cấu trúc điều khiển: Bao gồm các câu lệnh điều khiển như: If, Else if, Else, Switch… và vòng lặp như: For, While, Do-while…
* Hàm: là một phần của mã nguồn được định nghĩa để thực hiện một nhiệm vụ cụ thể. Hàm giúp phân chia chương trình thành các phần nhỏ, dễ quản lý và tái sử dụng
* Mảng và chuỗi: Mảng là một tập hợp các phần tử cùng kiểu dữ liệu, được lưu trữ liên tiếp trong bộ nhớ, Chuỗi là một mảng các ký tự kết thúc bằng ký tự null (‘\0’).
* Con trỏ: là một biến lưu trữ địa chỉ của một biến khác. Con trỏ rất mạnh mẽ và linh hoạt, cho phép thao tác trực tiếp với bộ nhớ.
* Cấu trúc (Struct): cho phép nhóm các biến khác nhau lại với nhau dưới một tên duy nhất, giúp quản lý dữ liệu phức tạp hơn.
* Quản lý bộ nhớ: các hàm để quản lý bộ nhớ động như *malloc*, *calloc*, *realloc*, và *free.*
* Tiền xử lý (Preprocessor): Trình tiền xử lý của C thực hiện các thao tác như bao gồm thư viện (*#include*), định nghĩa macro (*#define*), và các điều kiện tiền xử lý (*#if*, *#ifdef*, *#ifndef*)
  + 1. Ngôn ngữ lập trình ladder/SFC trong PLC
       1. Ngôn ngữ lập trình ladder

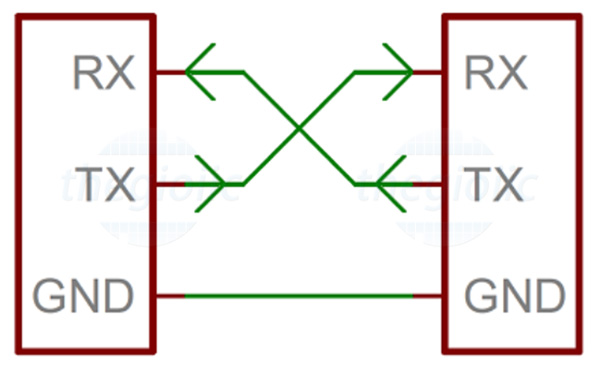
Ladder Logic (hay còn gọi là Ladder Diagram - LD) là một ngôn ngữ lập trình được phát triển dựa trên sơ đồ mạch điện và sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp. Nó được thiết kế để mô phỏng các mạch relay và contactor điện tử.

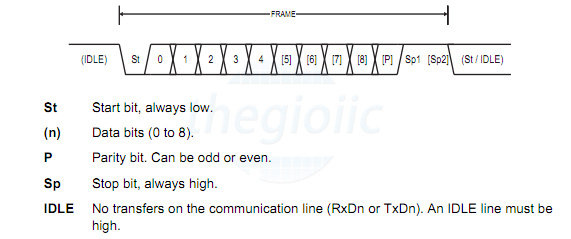
* + - * 1. Cấu trúc cơ bản của ladder
* ****Rungs****: Mỗi chương trình Ladder được chia thành các rungs (bậc thang), mỗi rung tương đương với một mạch điều khiển độc lập.
* ****Contacts****: Đại diện cho các đầu vào và các điều kiện logic (NO - Normally Open, NC - Normally Closed).
* ****Coils****: Đại diện cho các đầu ra, thường là các thiết bị hoặc tác vụ được điều khiển
* ****Timers and Counters****: Được sử dụng để thực hiện các tác vụ liên quan đến thời gian và đếm.
  + - 1. Ngôn ngữ lập trình SFC

Sequential Function Chart (SFC) là một ngôn ngữ lập trình cấp cao được sử dụng để thiết kế các quy trình điều khiển tuần tự phức tạp. SFC chia chương trình điều khiển thành các bước (steps) và chuyển tiếp (transitions).

* + - * 1. Cấu trúc cơ bản của SFC
* ****Steps****: Đại diện cho các trạng thái của quy trình, mỗi bước thường liên kết với một hành động cụ thể.
* ****Transitions****: Điều kiện để chuyển từ bước này sang bước khác.
* ****Actions****: Các hành động thực hiện tại mỗi bước.
  1. Cơ sở lý thuyết về xử lí ánh
  2. Cơ sở lý thuyết về điện – điện tử
     1. Giao tiếp UART

UART (Universal Asynchronous Receiver-Transmitter) bộ truyền nhận dữ liệu nối tiếp bất đồng bộ, đây là một trong những giao thức truyền thông giữa thiết bị với thiết bị được sử dụng nhiều nhất. Giao tiếp UART được sử dụng nhiều trong các ứng dụng để giao tiếp với các module như: Wifi, Bluetooth, Xbee, module đầu đọc thẻ RFID với Raspberry Pi, Arduino hoặc vi điều khiển khác. Đây cũng là chuẩn giao tiếp thông dụng và phổ biến trong công nghiệp

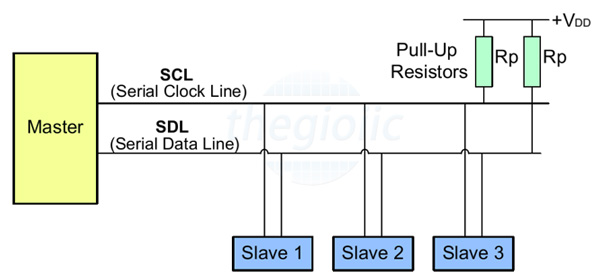




* + 1. Giao tiếp I2C

I2C hay IIC (Inter – Integrated Circuit) là 1 giao thức giao tiếp nối tiếp đồng bộ được phát triển bởi Philips Semiconductors, sử dụng để truyền nhận dữ liệu giữa các IC với nhau chỉ sử dụng hai đường truyền tín hiệu.

I2C kết hợp các tính năng tốt nhất của SPI và UART. I2C có thể kết nối nhiều slave với một master duy nhất (như SPI) và có thể có nhiều master điều khiển một hoặc nhiều slave. Điều này thực sự cần thiết khi muốn có nhiều hơn một vi điều khiển ghi dữ liệu vào một thẻ nhớ duy nhất hoặc hiển thị văn bản trên một màn hình LCD.



Giống như giao tiếp UART, I2C chỉ sử dụng hai dây để truyền dữ liệu giữa các thiết bị:

**SDA (Serial Data)** - đường truyền cho master và slave để gửi và nhận dữ liệu.

**SCL (Serial Clock)** - đường mang tín hiệu xung nhịp.

Các bit dữ liệu sẽ được truyền từng bit một dọc theo một đường duy nhất (SDA) theo các khoảng thời gian đều đặn được thiết lập bởi 1 tín hiệu đồng hồ (SCL).

* + 1. Giao tiếp TCP/IP
       1. Định nghĩa

TCP/IP là cụm từ viết tắt của Transmission Control Protocol/Internet Protocol hay còn gọi là giao thức điều khiển truyền nhận/ Giao thức liên mạng. Đây là một bộ các giao thức truyền thông được sử dụng để kết nối các thiết bị mạng với nhau trên internet. TCP/IP cũng có thể được sử dụng như một giao thức truyền thông trong mạng máy tính riêng (mạng nội bộ). Trong đó, bộ Giao thức internet – một tập hợp các quy tắc và thủ tục – thường gọi là TCP/IP (TCP/IP Protocol) TCP và IP là hai giao thức chính bên cạnh những giao thức khác trong bộ. Bộ giao thức TCP/IP hoạt động như một lớp trừu tượng giữa các ứng dụng internet và hạ tầng router/switch.TCP/IP chỉ định cách dữ liệu được trao đổi qua internet. Nó thực hiện bằng cách cung cấp thông tin liên lạc đầu cuối. Từ đó xác định cách nó được chia thành các packet, xác định địa chỉ, truyền dẫn, định tuyến và nhận dữ liệu. TCP/IP được thiết kế để đảm bảo độ tin cậy, nó có khả năng khôi phục tự động khi gặp sự cố trong quá trình truyền dữ liệu.

* + - 1. Nguyên lý hoạt động

Giao thức TCP/IP, IP đóng góp một vai trò cực kỳ quan trọng. Như tên gọi đã nói lên tất cả, TCP/TP là sự kết hợp giữa 2 giao thức. IP cho phép máy tính chuyển tiếp gói tin tới một máy tính khác. Thông qua một hoặc nhiều khoảng (chuyển tiếp) gần với người nhận gói tin. Còn TCP sẽ giúp kiểm tra các gói dữ liệu xem có lỗi không sau đó gửi yêu cầu truyền lại nếu có lỗi được tìm thấy.

Như vậy, để trả lời cho câu hỏi về quy cách hoạt động của TCP/IP là gì thật ra rất đơn giản. Bạn có thể hình dung việc truyền tin trên Internet tựa như một dây chuyền sản xuất. Các công nhân sẽ lần lượt chuyền các bán thành phẩm qua những giai đoạn khác nhau để bổ sung hoàn thiện sản phẩm. Khi đó, IP giống như là quy cách hoạt động của nhà máy, còn TCP lại đóng vai trò là một người giám sát dây chuyền, đảm bảo cho dây chuyền liên tục nếu có lỗi xảy ra.

* + - 1. Liên kết truyền thông qua thư viện Socket

Về giao thức TCP/IP, cần quan tâm đến địa chỉ và cổng của server/client muốn kết nối. Quy trình kết nối và ra lệnh để camera chụp ảnh:

A diagram of a network

Description automatically generated

*Sơ đồ thực hiện liên kết truyền nhận dữ liệu.*

* + 1. Điều khiển AC Servo
       1. **Định nghĩa**

AC Servo Motor là một loại động cơ xoay chạy bằng điện 3 pha, hoạt động dựa trên nguyên lý nam châm vĩnh cửu, giống như các động cơ thông thường được sử dụng trong máy bơm hoặc máy quạt. Điểm đặc biệt của AC Servo Motor nằm ở chỗ sự tích hợp của nhiều công cụ điện tử và cảm biến, bao gồm bộ khuếch đại, bộ điều khiển, bộ mã hóa và màn hình. Điều này giúp tăng cường khả năng chính xác và dễ dàng điều khiển. Đây là một yếu tố quan trọng đáp ứng nhu cầu ngày càng cao trong sản xuất công nghiệp, đặc biệt là trong thời đại Cách mạng Công nghiệp 4.0 hiện nay.

* + - 1. **Nguyên lí hoạt động**

AC Servo Motor được kết hợp với encoder để cung cấp phản hồi vị trí và tốc độ. Nói một cách đơn giản, chúng ta chỉ đo vị trí. Sau đó, vị trí đo của đầu ra được so sánh với vị trí lệnh, đầu vào bên ngoài để điều khiển.  Nếu vị trí đầu ra khác với vị trí đầu ra dự kiến, tín hiệu lỗi sẽ tạo ra. Điều này làm cho động cơ quay theo một trong hai hướng, vì cần phải đưa trục đầu ra đến vị trí thích hợp. Khi vị trí đến gần, tín hiệu lỗi giảm xuống không. Cuối cùng động cơ dừng lại.

Có 3 chế độ điều khiển động cơ: Tốc độ, Vị trí và Moment, cần cài đặt tùy theo mục đích sử dụng động cơ và từng loại driver của từng hãng thì mới hoạt động được.

* + 1. Điều khiển PLC
       1. **Định nghĩa**

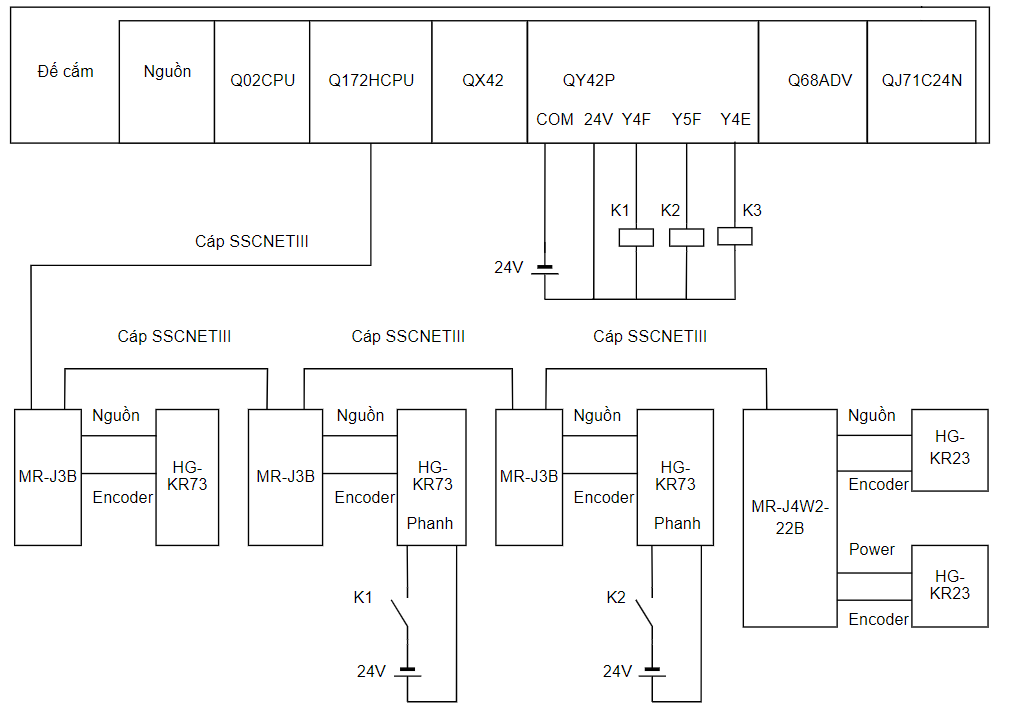
PLC là tên viết tắt của dòng chữ Programmable Logic Controller. Trong quá khứ các bộ điều khiển chỉ được sản xuất ra để phục vụ riêng cho một mục đích điều khiển và không thể thay đổi (Hay còn gọi là điều khiển kết nối cứng), điều này đã tạo ra những hạn chế và nhược điểm vô cùng lớn trong việc lập trình điều khiển. Thông qua bộ điều khiển PLC, người dùng hoàn toàn có thể thay đổi thuật toán điều khiển thông qua việc lập trình PLC (Viết bằng ngôn ngữ lập trình).

Hiện nay trên thế giới có một số hãng sản xuất PLC rất nổi tiếng và được nhiều công ty trên thế giới sử dụng: Siemens (Đức), Omron và Mitsubishi (Nhật Bản), Delta (Đài Loan). Tại Việt Nam dòng PLC của Siemens và Mitsubishi là phổ biến nhất và được đưa vào chương trình đào tạo của các trường kỹ thuật.

* + - 1. **Cách thức điều khiển**
* **Điều khiển phần cứng:** Điều khiển kết nối cứng là loại điều khiển mà các chức năng của nó được đặt cố định(nối dây). Nếu muốn thay đổi chức năng điều đó có nghĩa là thay đổi kết nối dây. Điều khiển kết nối cứng có thể thực hiện với các tiếp điểm…
* **Điều khiển logic:** Điều khiển logic khả trình là loại điều khiển mà chức năng của nó được đặt cố định thông qua một chương trình còn gọi là bộ nhớ chương trình. Các phần tử nhập tín hiệu được nối ở ngõ vào của bộ điều khiển, các phần tử này khởi động các cuộn dây đặt ở ngõ ra. Quá trình điều khiển ở đây được thực hiện bằng một chương trình đã soạn thảo theo mục đích, yêu cầu của việc điều khiển thiết bị.
  + - 1. **Cấu trúc PLC**
* **Module CPU (Central Processing Unit):** đơn vị xử lý trung tâm (bao gồm: bộ vi xử lý và bộ nhớ).
* **Module I/O:** là các module đảm nhận là cổng giao tiếp giữa các tín hiệu điều khiển bên ngoài với PLC để điều khiển chương trình.
* **Hệ thống đường bus tín hiệu giữa các PLC với nhau:**
* Address bus: Địa chỉ đường bus.
* Data bus: Mạng dữ liệu từ khối này tới khối khác.
* Control bus: truyền tín hiệu để đồng bộ hóa các hoạt động trong PLC.

# CHƯƠNG 4: THIẾT KẾ PHẦN CỨNG HỆ THỐNG

* 1. Thiết kế phần cứng cho cánh tay robot
     1. Sơ lược về hệ thống điều khiển



Ta có danh sách của các bộ phận thuộc hệ thống điều khiển như sau:

**- Base Q35B:**

Module nguồn cấp cho họ PLC Q gồm 5 chỗ cho các module PLC và 1 chỗ cho module nguồn.



**- PLC Q02HCPU**:

* **Vai trò:**

PLC này hỗ trợ đa dạng các kiểu module chức năng (truyền thông, Analog, phát xung, I/O,…). Với khả năng hỗ trợ số lượng đầu vào I/O tối đa ở mức khá lớn và chức năng ngắt thì Q02HCPU đáp ứng hầu hết các ứng dụng điều khiển phân tán quy mô vừa cần sử dụng ngắt ngoài. Ngoài ra, việc mở rộng thêm các Remote I/O cũng như kết nối mạng truyền thông công nghiệp rất thuận tiện với các module chức năng có sẵn.

* **Chức năng:**

Trong hệ thống điều khiển, module Q02HCPU có vai trò như một kênh giao tiếp giữa các module với nhau để đưa ra những tín hiệu điều khiển cho động cơ để thực thi. Ngoài ra, module Q02HCPU còn có khả năng liên kết với C# để có thể giúp người dùng tương tác với giao diện một cách linh hoạt.

**- Motion Controller Q172HCPU:**

* **Vai trò:**

Motion Controller Q172HCPU của Mitsubishi là bộ điều khiển đa trục, lý tưởng cho các ứng dụng điều khiển chuyển động phức tạp. Với khả năng điều khiển chính xác nhiều động cơ servo, hỗ trợ đa dạng tín hiệu và giao diện kết nối như Ethernet và CC-Link, Q172HCPU đảm bảo hiệu suất cao và ổn định. Thiết bị này là lựa chọn tối ưu cho các hệ thống tự động hóa và điều khiển robot trong công nghiệp.

* **Chức năng:**

Trong hệ thống điều khiển, Q172HCPU có vai trò quan trọng trong việc đưa ra tín hiệu điều khiển đến cho động cơ AC servo, là bộ não quyết định việc thực thi cho động cơ.

**- Module input QX42:**

* **Vai trò:**

Module input QX42 của Mitsubishi thuộc dòng MELSEC-Q SERIES, là module đầu vào số DC với 64 kênh và điện áp định mức 24VDC. Module này chiếm 64 điểm I/O trên PLC và sử dụng kết nối kiểu connector. Với dòng tiêu thụ 0,075A và kích thước nhỏ gọn (27,4mm x 98mm x 90mm), QX42 dễ dàng tích hợp vào các hệ thống tự động hóa. Trọng lượng chỉ 0,18kg, QX42 là lựa chọn lý tưởng cho các ứng dụng yêu cầu thu thập tín hiệu nhanh và chính xác trong các môi trường công nghiệp.

* **Chức năng:**

Trong hệ thống điều khiển, module này được sử dụng để đọc những tín hiệu từ cảm biến tiệm cận cũng như công tắc được gắn trên băng tải để từ đó làm dữ kiện để những module khác thực thi những chức năng phù hợp.



**- Module output QY42:**

* **Vai trò:**

Module output QY42P của Mitsubishi thuộc dòng MELSEC-Q SERIES, là module đầu ra transistor (sink) với 64 kênh. Thiết bị này hoạt động với điện áp nguồn DC từ 12-24V và có logic đầu ra loại sink. QY42P chiếm 64 điểm I/O trên PLC, sử dụng kết nối dây kiểu screw, và có dòng tiêu thụ 0,15A. Kích thước nhỏ gọn (27,4mm x 98mm x 90mm) và trọng lượng nhẹ (0,17kg) giúp QY42P dễ dàng tích hợp vào các hệ thống tự động hóa, đảm bảo điều khiển các thiết bị ngoại vi một cách hiệu quả và chính xác.

* **Chức năng:**

Trong hệ thống điều khiển, module này dùng để xuất tín hiệu ngõ ra để điều khiển băng tải và kết nối đến servo module để bật tắt thắng điện từ cho động cơ AC trong những trường hợp khẩn cấp.



**- Servo amplifier MR-J3-70B:**

* **Vai trò:**

Servo amplifier MR-J3-70B thuộc dòng MELSERVO J3 series của Mitsubishi Electric, là một bộ điều khiển servo AC mạnh mẽ và linh hoạt. Với công suất định mức 0.75kW và dòng điện 5.8A, MR-J3-70B là lựa chọn lý tưởng cho các ứng dụng đòi hỏi độ chính xác và độ tin cậy cao. Bộ điều khiển này được thiết kế với nhiều tính năng bảo vệ như ngắn mạch, quá tải, và quá tốc độ, đảm bảo an toàn cho cả thiết bị và hệ thống. MR-J3-70B có kích thước nhỏ gọn (168mm x 60mm x 185mm) và tiêu thụ điện năng thấp (3.8A), dễ dàng tích hợp và vận hành trong các môi trường công nghiệp khắc nghiệt. Đặc biệt, với khả năng giao tiếp nhanh chóng qua chuẩn SSCNET III/H, MR-J3-70B giúp tối ưu hóa hiệu suất và linh hoạt trong các ứng dụng điều khiển chuyển động.

* **Chức năng:**

Trong hệ thống điều khiển, module servo này sẽ dùng để truyền tín hiệu điều khiển cho động cơ AC bằng việc giao tiếp với **module controller** với những quyết định từ phía người dùng.



**- Servo amplifier MR-J4W2-22B:**

* **Vai trò:**

Servo amplifier MR-J4W2-22B thuộc dòng MELSERVO J4 series của Mitsubishi Electric, là một bộ điều khiển servo AC đa chức năng và tiên tiến. Với công suất định mức 0.2kW và dòng điện 1.5A, MR-J4W2-22B đáp ứng nhu cầu điều khiển chuyển động chính xác và linh hoạt trong các ứng dụng công nghiệp. Bộ điều khiển này tích hợp nhiều tính năng bảo vệ và điều khiển như chức năng an toàn Safe Torque Off (STO), giao tiếp nhanh chóng qua chuẩn SSCNET III/H, và các chức năng tự động điều chỉnh như tự động điều chỉnh và tự động điều chỉnh bằng một nút chạm. Với kích thước nhỏ gọn (168mm x 60mm x 195mm), MR-J4W2-22B dễ dàng tích hợp và vận hành trong các môi trường công nghiệp đòi hỏi hiệu suất và độ tin cậy cao.

* **Chức năng:**

Trong hệ thống điều khiển, Servo là bộ phận quyết định cho việc di chuyển của robot, nhận tín hiệu điều khiển từ các module PLC để thực thi một cách chính xác những hành động, câu lệnh đã được xử lí một cách nhanh chóng, chính xác nhất.

|  |  |
| --- | --- |
| **Bộ phận** | **Link tham khảo** |
| Base Q35B | [Q35B - Main Base Mitsubishi - Codienhaiau.com](https://codienhaiau.com/product/mitsubishi-q35b/) |
| Q02HCPU | [Q02HCPU - PLC Mitsubishi - Nhà Phân phối Tự Động Hóa Toàn Cầu](https://plcmitsubishi.com/q02hcpu) |
| Q172HCPU |  |
| QX42 | [QX42 - Module ngõ vào Mitsubishi - Codienhaiau.com](https://codienhaiau.com/product/mitsubishi-qx42/) |
| QY42P | [QY42P - Module Output Mitsubishi - Codienhaiau.com](https://codienhaiau.com/product/mitsubishi-qy42p/) |
| MR-J3-70B | [MR-J3-70B - Bộ điều khiển Servo Mitsubishi 0.75kW (codienhaiau.com)](https://codienhaiau.com/product/servo-amplifier-mitsubishi-mr-j3-70b/) |
| MR-J4W2-22B | [MR-J4W2-22B - Bộ điều khiển Servo Misubishi 0.2kW (codienhaiau.com)](https://codienhaiau.com/product/servo-amplifier-mitsubishi-mr-j4w2-22b/) |

* + 1. Cách lắp đặt hệ thống điều khiển
       1. Lắp các module lên Base Q35B

Ta sẽ tiến hành lắp nguồn vào trước trên Base Q35B, tiếp theo sẽ là Q02HCPU. Đối với những module còn lại, sẽ không có quy tắc nào quyết định về thứ tự. Phần mềm sẽ đọc được những parameters từ Base Q35B và nhận diện được những module thành phần tại từng vị trí với khai báo trước đó.

Nguồn sẽ nhận điện áp từ 100 – 240VAC, nên ta có thể cấp trực tiếp điện 220V vào cục nguồn.



Khi Q02HCPU đã được cấp nguồn, module này sẽ hiện error. Sẽ có 2 trường hợp mà CPU báo lỗi: CPU chưa có chương trình hoặc thứ tự cắm của các module khác với những gì đã khai báo trước đó trong CPU.

* + - 1. Q02HCPU và máy tính thông qua GX Work2
         1. **Q02HCPU**

**Kết nối:**

Để kết nối module PLC Q02HCPU với máy tính, sẽ có nhiều lựa chọn để kết nối: RS-232, Ethernet và USB.

Trong hệ thống điều khiển hiện tại, chúng ta sử dụng giao thức RS-232 để truyền nhận dữ liệu từ máy tính với Q02HCPU.

* + - * 1. **Máy tính (GX Works2)**

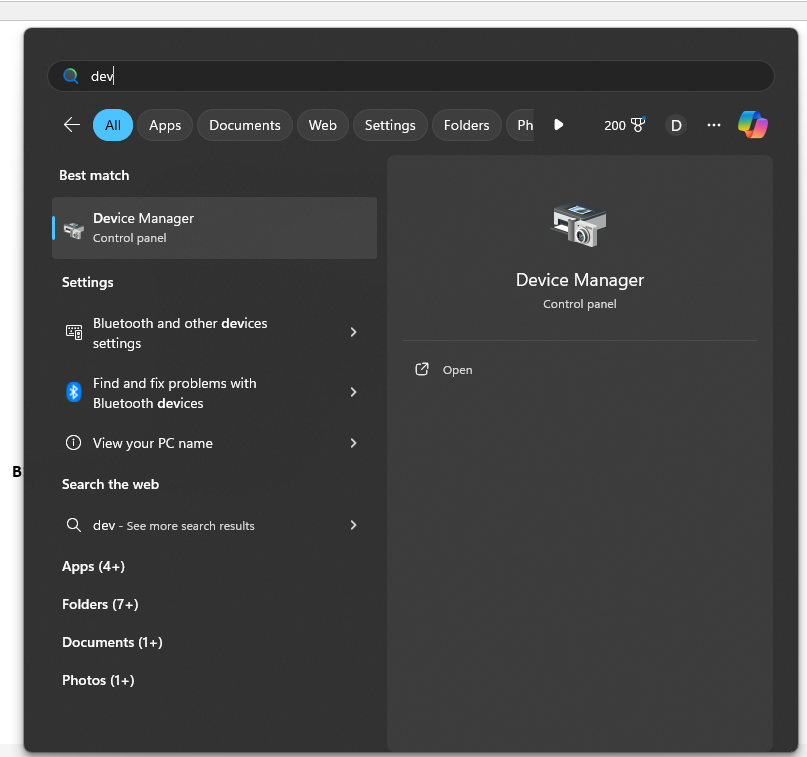
**Kiểm tra kết nối:**

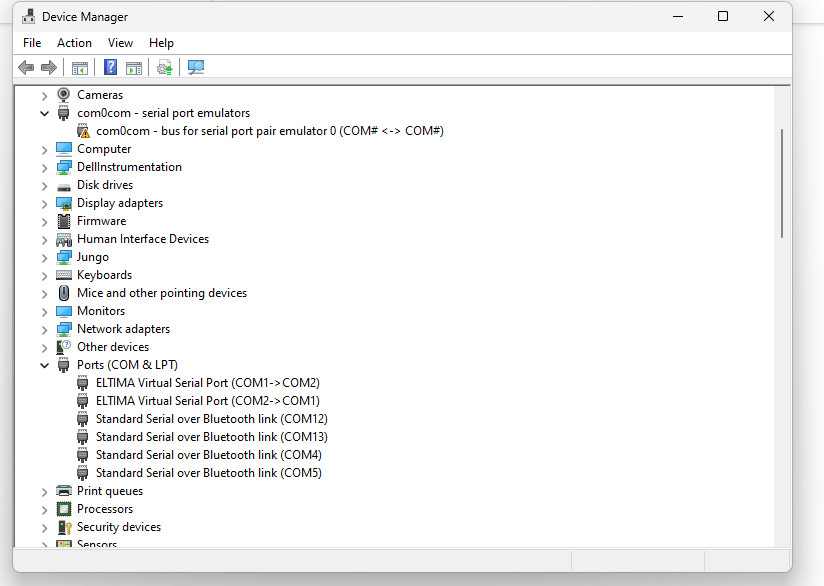
Để kiểm tra kết nối, ta tiến hành thực hiện những bước sau:

**B1: Mở device manager trên máy tính**

Nhấn vào biểu tượng Window trên bàn phím 🡪 Nhập Device manager.

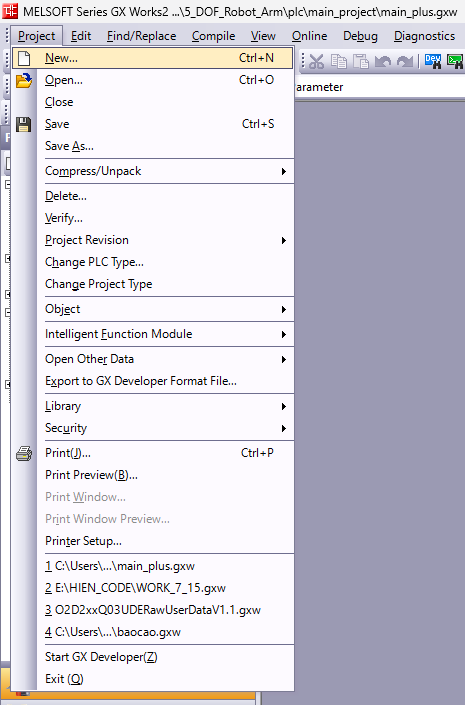
Sau khi mở Device manager 🡪 Đi đến tab Ports 🡪 Hãy thử rút dây kết nối ra và cắm lại. ta sẽ thấy được port sử dụng để kết nối giữa PLC và máy tính sẽ bật/tắt ở trong thư mục port.

****

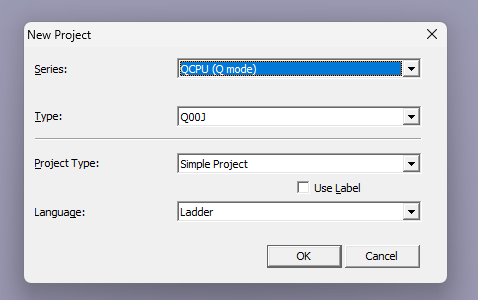


**B2: Kết nối với phần mềm GX Works2**

Sau khi cài đặt xong phần mềm GX Works2 trên máy tính 🡪 Ta có thể sử dụng chương trình có sẵn hoặc tạo mới một project 🡪 Ở trên tab Project trên thanh công cụ 🡪 Chọn New hoặc dùng tổ hợp phím Ctrl + N.



**B3:** **Thiết đặt cho new project:**

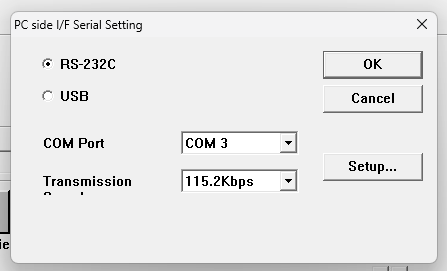


Sau khi có được một new project 🡪 Ta sẽ thử kết nối với Q02HCPU

**B4: Kiểm tra kết nối**

Ở bên tab Navigation 🡪 Ở mục All Connections 🡪 Chọn Connection hiện có.

Click vào icon  🡪 Chọn RS-232C 🡪 Chọn COM Port đang sử dụng theo Device Manager 🡪 Chọn Transmission Baud: 115.2Kbps 🡪 Nhấn OK 🡪 Nhấn Connection Test 🡪 Sẽ có cửa sổ thông báo install successfully nếu kết nối giữa PLC và máy tính được tạo.

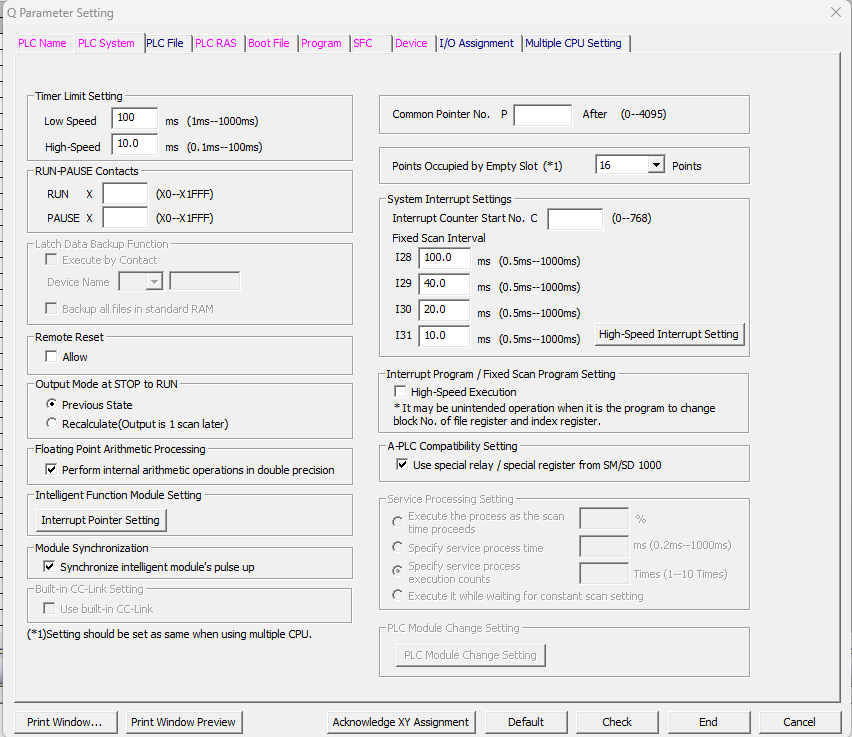


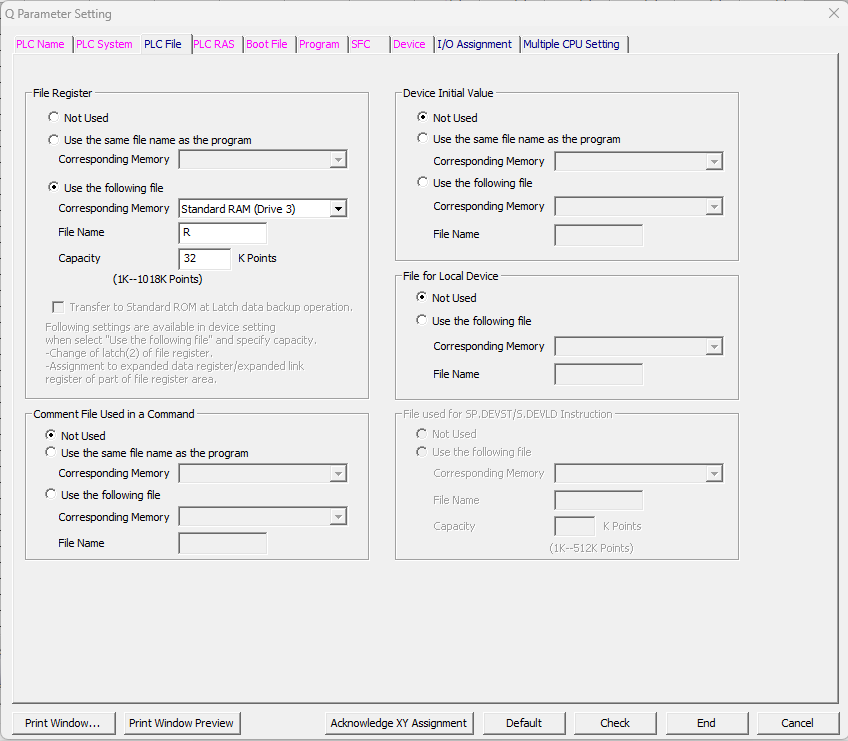


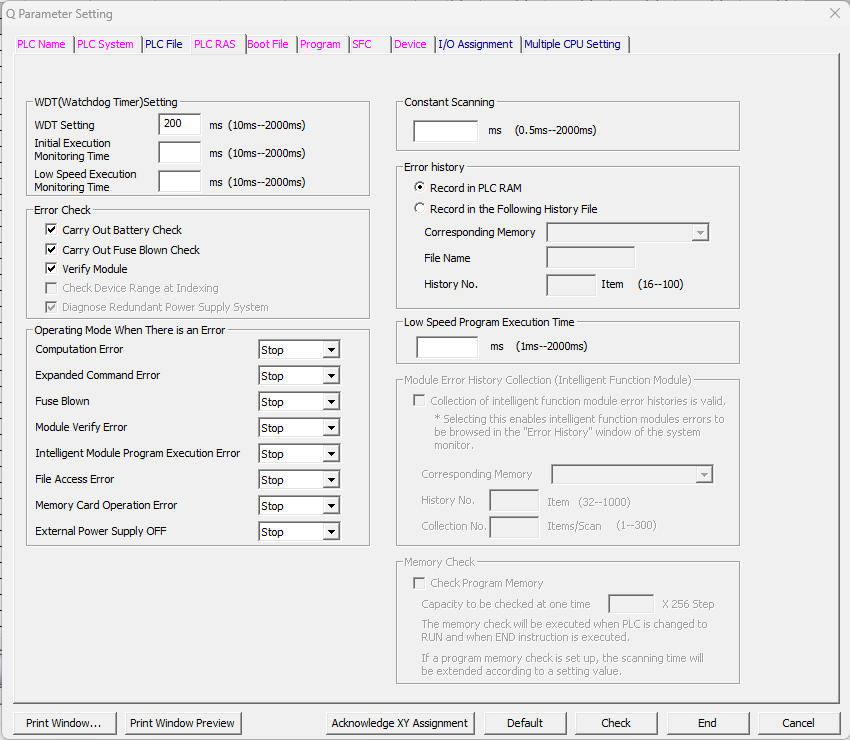
*Lưu ý: Sau khi kết nối thành công, phải nhấn nút OK thì chương trình mới lưu lại kết nối. không thì khi write chương trình xuống PLC sẽ thông báo là không kết nối được.*

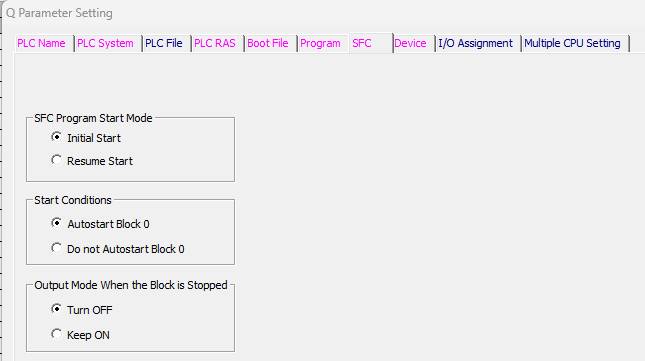
* + - * 1. **Thiết đặt thông số cho các module trong hệ thống điều khiển trên GX Works2**

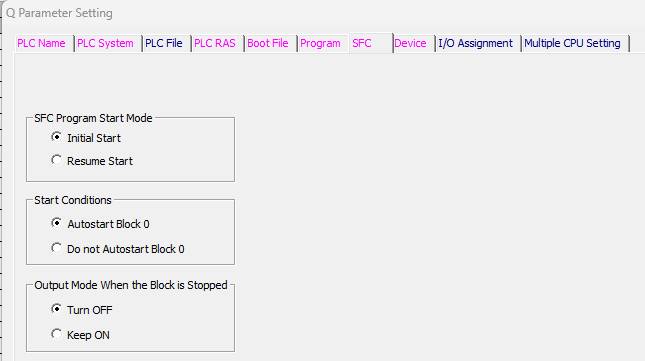
Để thiết đặt các thông số cho những module được lắp đặt trong hệ thống điều khiển, ta có thể tham khảo qua những datasheet do nhà sản xuất cung cấp. Đối với hệ thống điều khiển hiện tại, ta sẽ có những thông số cần setup như sau:

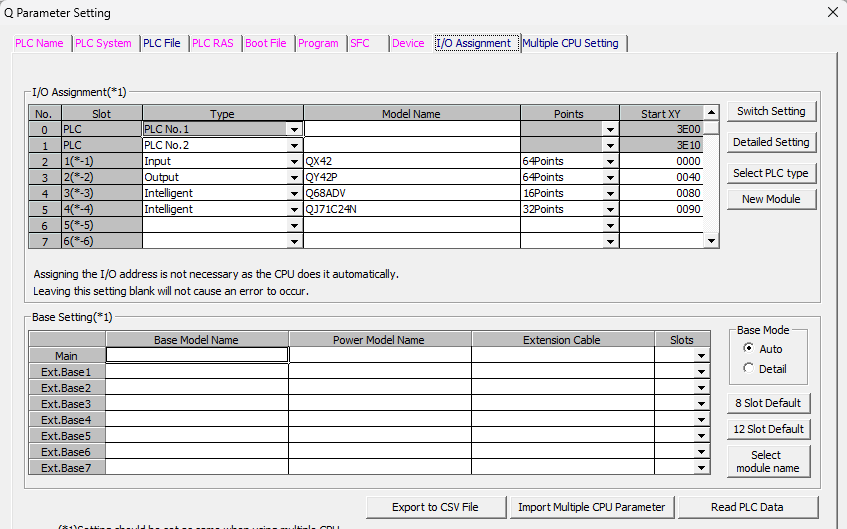


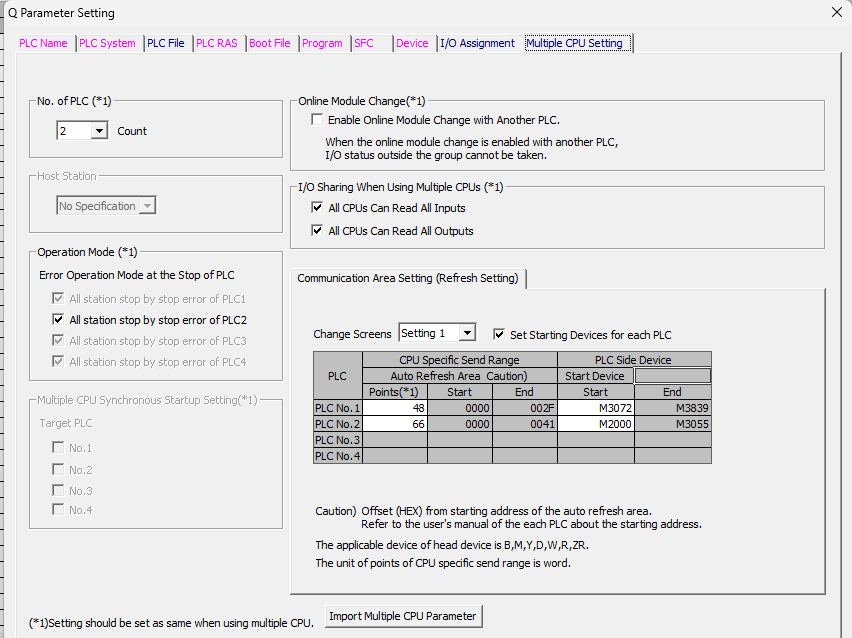












* + - 1. Q02HCPU và QY42P

Thông qua module nguồn Base Q35B cũng như setting trên GX Works2, ta dễ dàng kết nối giữa 2 module lại với nhau từ đó có thể điều khiển được bật/tắt relays để kích hoạt thắng điện từ giúp đảm bảo tính an toàn cho robot.

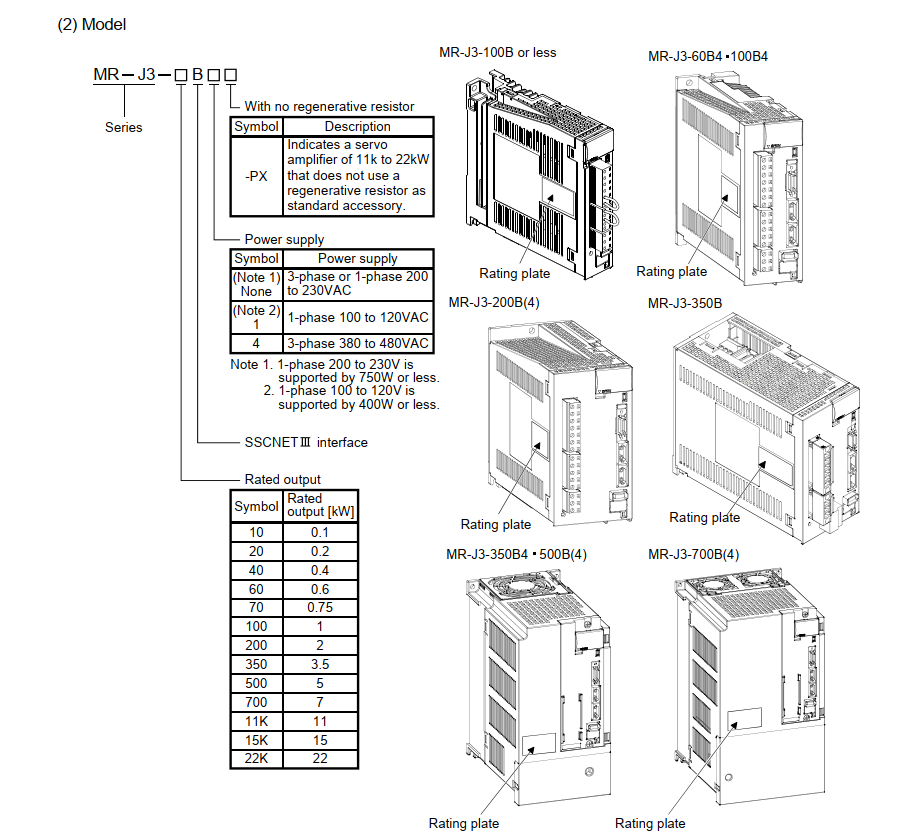
* + - 1. Q172HCPU và những thành phần liên quan
         1. Q172HCPU và Q02HCPU

Thông qua module nguồn Base Q35B cũng như setting trên GX Works2, ta dễ dàng kết nối giữa 2 module lại với nhau từ đó Q02HCPU có thể thông qua phần mềm GX Works2 để đưa ra được những tín hiệu điều khiển thông qua việc chia sẻ vùng nhớ cũng như những hàm đặc biệt như SFC SFCS… cho việc điều khiển robot.

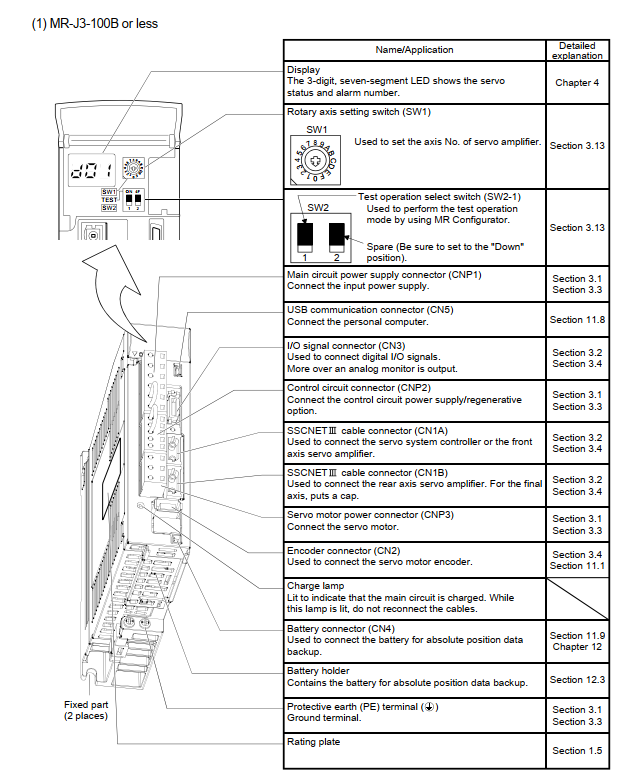
* + - * 1. Q172HCPU và Servo motor

**Lắp đặt Servo Amplifier**

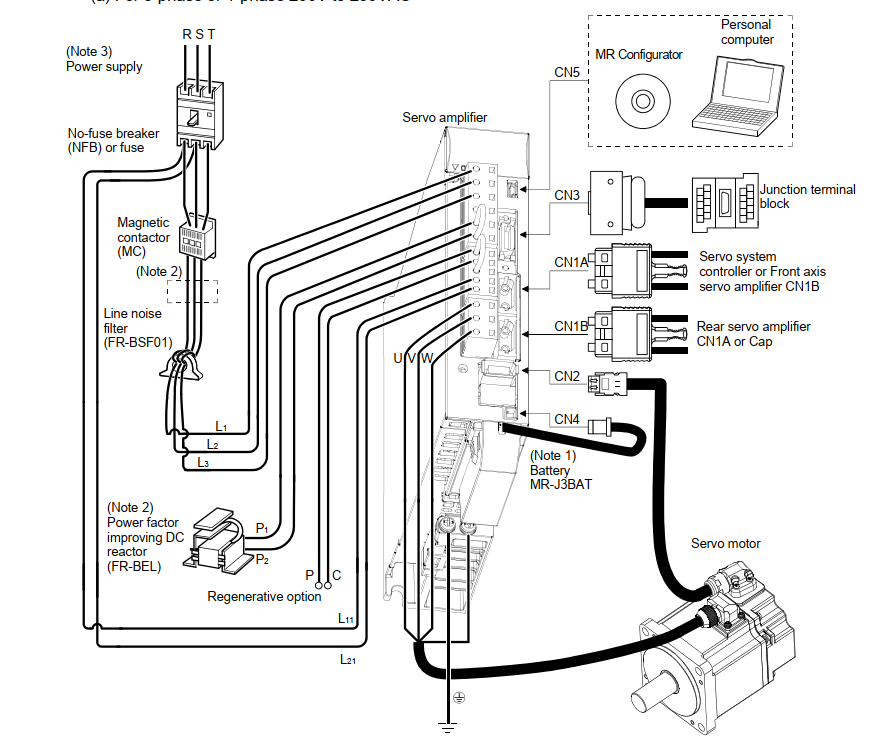
x: Rated input tương ứng với động cơ đang sử dụng



Đối với loại Servo chúng ta đang sử dụng là MR-J[]-[]B, ta có những thông số sau:



*Lưu ý: cần có nguồn điện 3 pha hoặc 1 pha từ 200VAC đến 230 VAC.*



*Nguồn điện từ pin được sử dụng cho hệ thống phát hiện vị trí tuyệt đối ở chế độ điều khiển vị trí*

*Reactor AC đang được được sử dụng. Trong trường hợp này, không thể sử dụng reactor DC. Khi không sử dụng cuộn kháng DC, ngắn mạch P1 và P2.*

*Có thể sử dụng nguồn điện 1 pha 200V đến 230VAC với bộ khuếch đại servo MR-J3-70B trở xuống*

*Đối với 1 pha 200V đến 230VAC, kết nối nguồn điện với L1 L2 và để L3 mở.*

**Lắp đặt 2 hoặc nhiều servo amplifier**

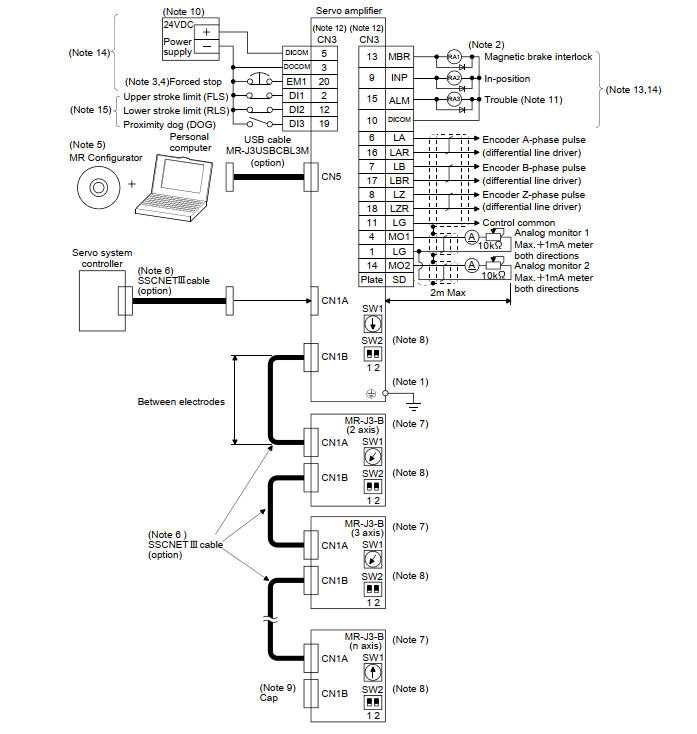
Chừa một khoảng trống lớn giữa mặt trên của bộ khuếch đại servo và bề mặt bên trong của hộp điều khiển, đồng thời lắp quạt làm mát để ngăn nhiệt độ bên trong hộp điều khiển vượt quá điều kiện môi trường.

Khi lắp đặt các bộ khuếch đại servo gần nhau, hãy chừa khoảng cách 1mm giữa các bộ khuếch đại servo liền kề để cân nhắc dung sai lắp đặt.

Trong trường hợp này, hãy điều chỉnh nhiệt độ môi trường trong khoảng từ 0o đến 45o (32o đến 113o) hoặc sử dụng ở tỷ lệ tải hiệu quả 75% hoặc nhỏ hơn.



**Cách kết nối tín hiệu I/O giữa 2 hoặc nhiều servo amplifier**





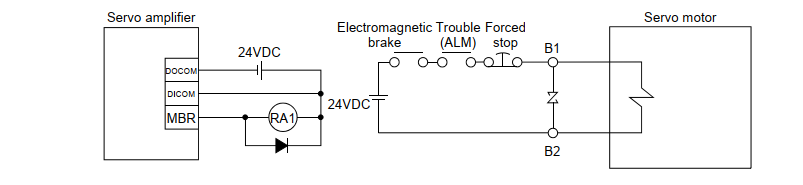
*Lưu ý:*

* *Để tránh bị điện giật, luôn kết nối đầu nối đất bảo vệ (PE) (đầu nối được đánh dấu ) của bộ khuếch đại servo với nối đất bảo vệ (PE) của hộp điều khiển*
* *Kết nối diode theo đúng hướng. Nếu được kết nối ngược lại, bộ khuếch đại servo sẽ bị lỗi và không phát ra tín hiệu, làm mất khả năng dừng cưỡng bức (EM1) và các mạch bảo vệ khác.*

**Servo motor với thắng điện từ**

Không dùng chung nguồn điện giao diện 24VDC giữa giao diện và phanh điện từ. Luôn sử dụng nguồn điện được thiết kế dành riêng cho phanh điện từ. Phanh sẽ hoạt động khi nguồn điện (24VDC) tắt. Tắt lệnh bật servo sau khi motor servo đã dừng.

**Sơ đồ kết nối:**



* 1. Thiết kế phần cứng cho băng tải
  2. Thiết kế phần cứng cho thiết bị theo dõi chuyển động

# CHƯƠNG 5: XÂY DỰNG PHẦN MỀM CHO HỆ THỐNG

5.1. Xây dựng phần mềm cho cánh tay robot

5.1.1. Q172HCPU và máy tính thông qua MT Developer2

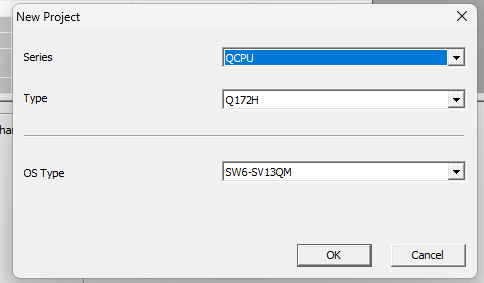
5.1.1.1. Cách kiểm tra kết nối với PLC

Giao thức truyền dữ liệu giữa Q172HCPU và máy tính sẽ thông qua USB, nên cơ bản về việc cài đặt phần mềm MT Developer2 sẽ có sẵn driver đã được thiết lập sẵn.

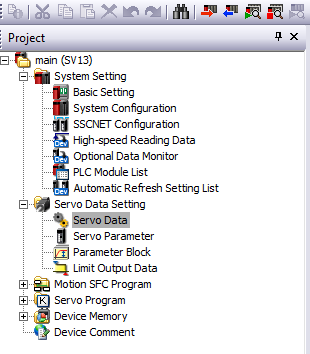
#### **5.1.1.2. Cài đặt thông số thông qua phần mềm MT Developer2**

Để thiết đặt các thông số cho những module được lắp đặt trong hệ thống điều khiển, ta có thể tham khảo qua những datasheet do nhà sản xuất cung cấp. Đối với hệ thống điều khiển hiện tại, ta sẽ có những thông số cần setup như sau:

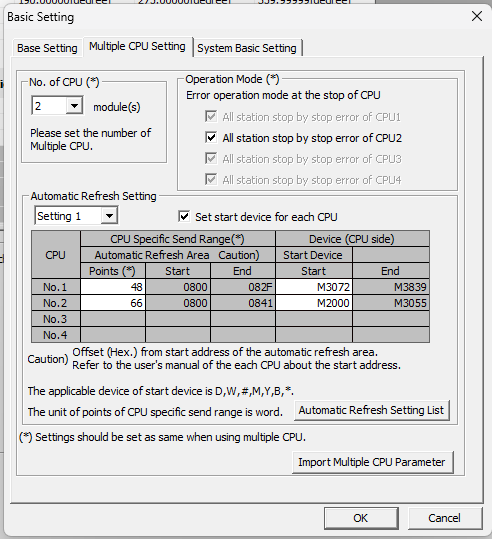
Tạo mới một project:

****

Khi tạo xong project, ta sẽ thấy bên cột **Project** bên trái ngoài cùng, tiến hành thiết đặt lần lượt các thông số sau:



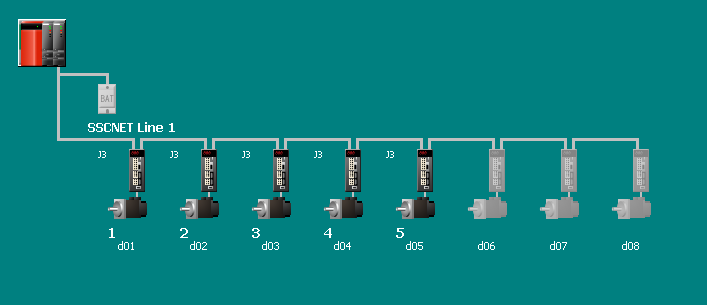
**Basic setting:**



**System configuration:**



**SSCNET Configuration:**

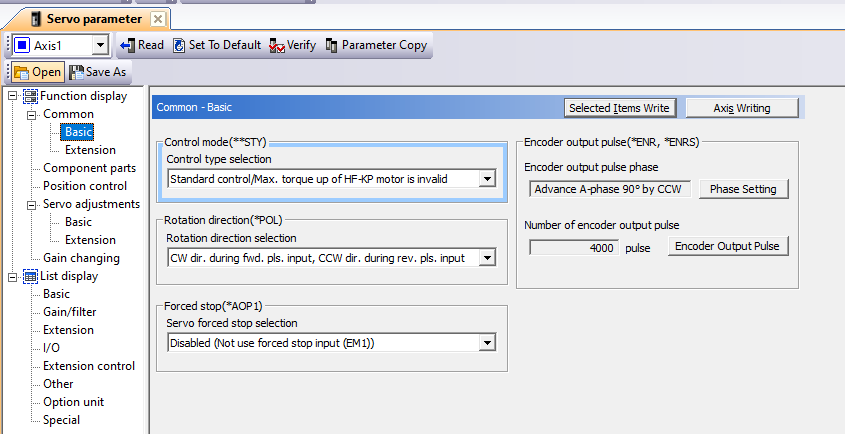


Nháy đúp chuột vào icon động cơ, cửa sổ Amplifier Setting sẽ hiện lên, tiến hành cài đặt thông số như hình:

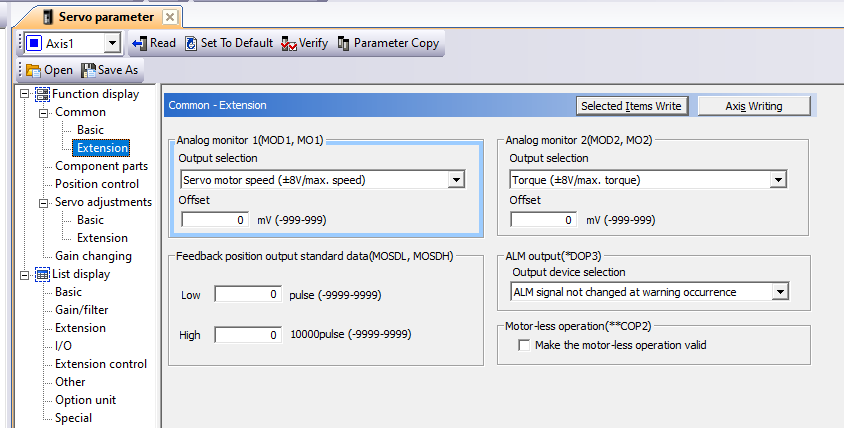


Tiếp tục nhấn vào *Servo Paramenter Setting,* 1 tab Servo parameter sẽ xuất hiện, tiến hành cài đặt với các thông số sau:

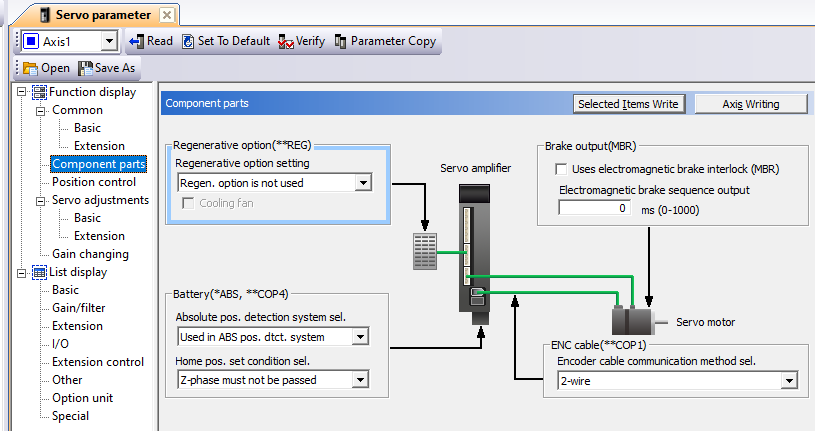
**Function display\Common\Basic:**



**Function display\Common\Extension:**

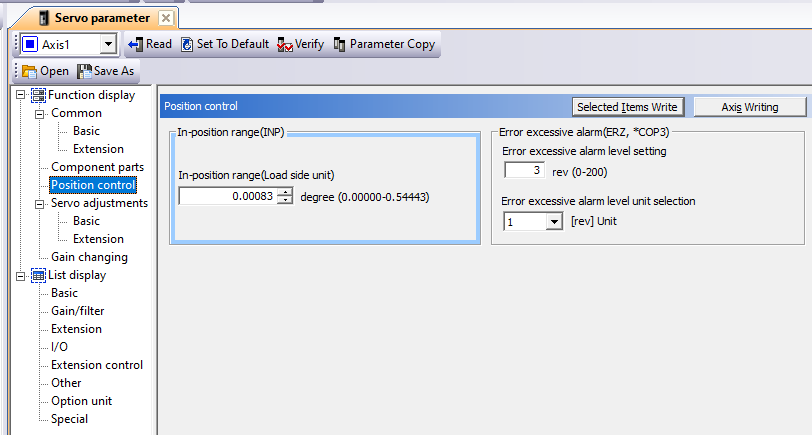


**Function display\Component parts:**

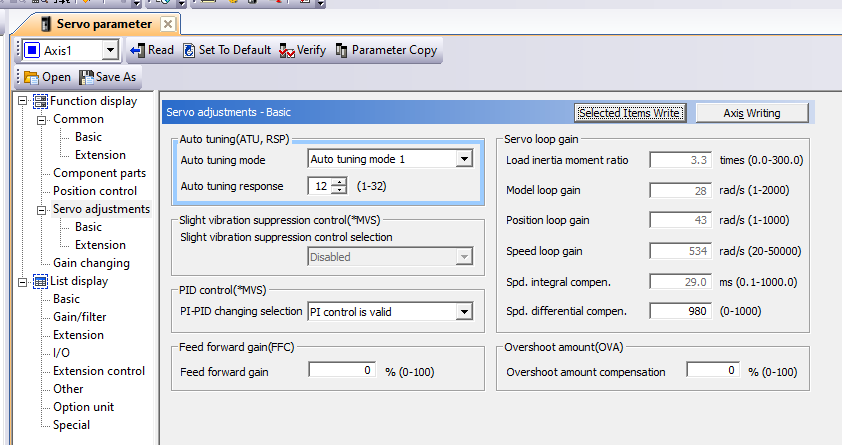


**Function display\Position control:**

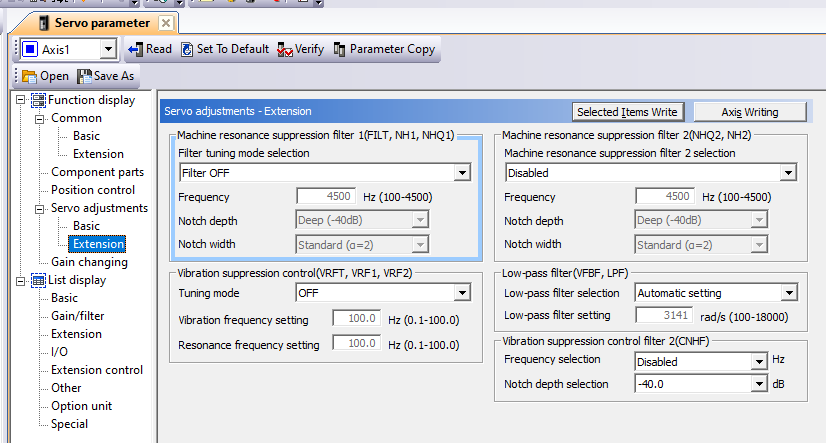
*Đây là phần thiết lập cho Axis 1, tương tự với 4 Axis còn lại, ta có thể dựa vào file tham khảo để thiết đặt theo cho đúng.*



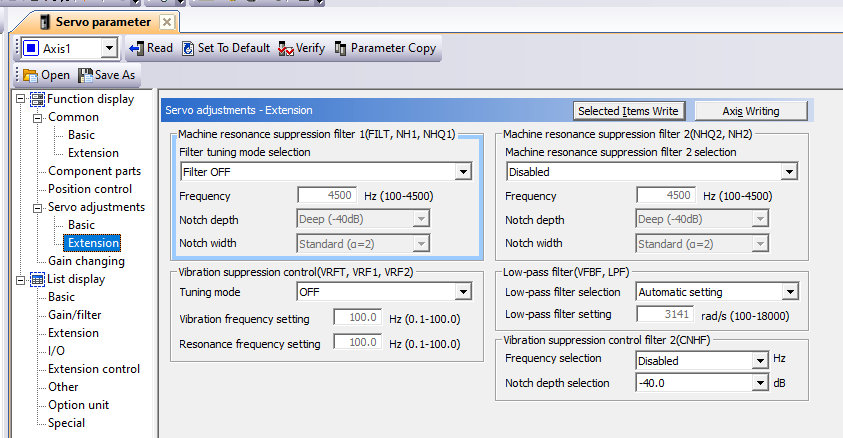
**Function display\Servo adjustments\Basic:**



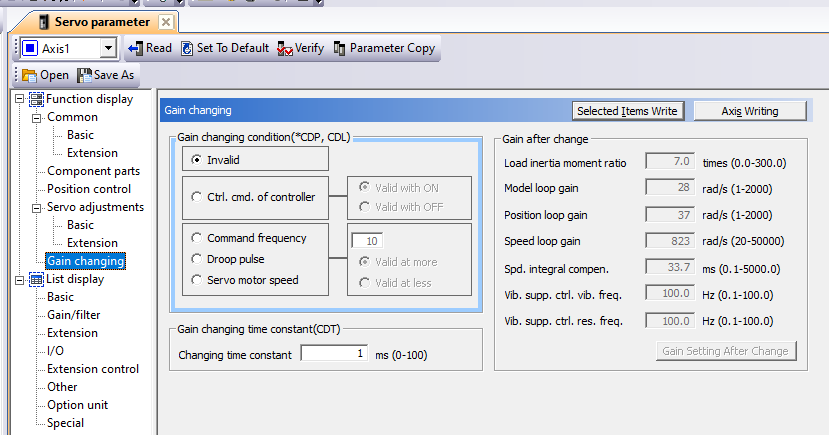
**Function display\Servo adjustments\Basic:**

****

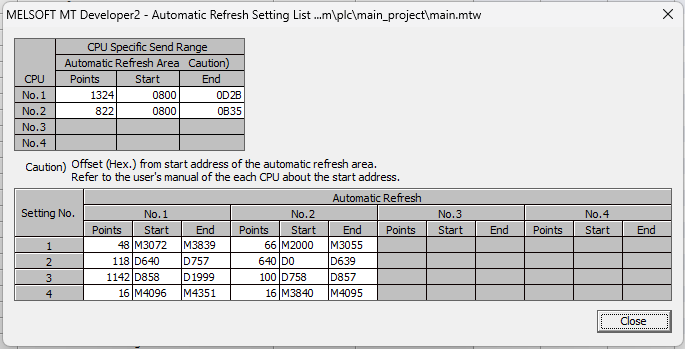
**Function display\Servo adjustments\Extension:**

****

**Function display\Gain changing:**

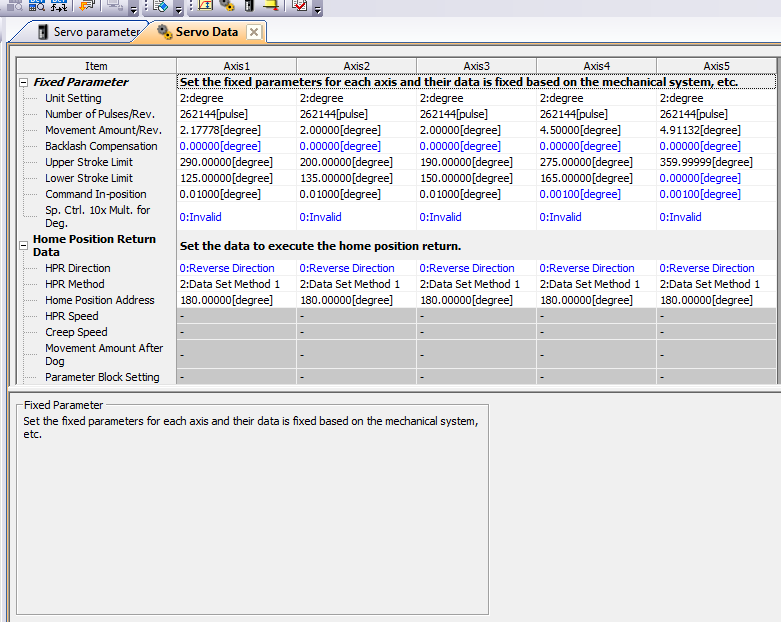
****

Sau khi cài đặt hết thông số, ta vô tab *Automatic Refresh Setting List* bên cây thư mục Project, thiết đặt vùng nhớ được chia sẻ với Q02HCPU như cấu hình sau:



Tất cả những bước nêu trên đã hoàn thành cho việt thiết đặt cấu hình cho *System Setting,* tiến hành cài đặt cấu hình cho phần *Servo Data Setting*

**Servo Data:**



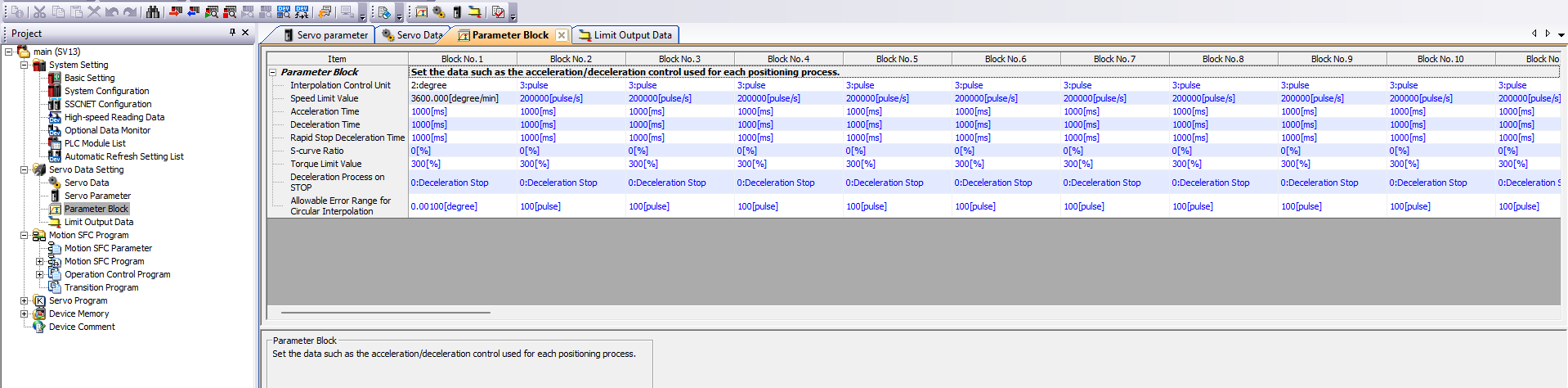
*Lưu ý:*

*Home Position Address* *đang được để là 180.00000 [degree] sẽ là vị trí ban đầu cho các khớp.*

*Unit Setting đang được thiết đặt là degree quyết định đơn vị mà ta sẽ điều khiển cho robot.*

**Parameter Block:**

Thiết đặt các thông số liên quan đến tốc độ giới hạn và thời gian tăng tốc giảm tốc cho từng block, tiến hành cài đặt các thông số như hình bên dưới.

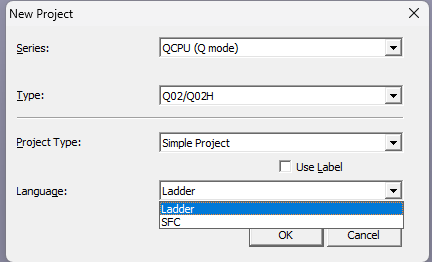


### **5.1.2. Lập trình điều khiển cho robot thông qua phần mềm MT Developer2 và GX Works2**

**5.1.2.1. Lập trình điều khiển robot trên GX Works2**

Tương tự như Tia Portal cho những dòng PLC Siemen, với mục đích chính để lập trình cho PLC hoạt động đúng với chức năng mong muốn từ phía người dùng. Dòng PLC Mitsubishi cũng được hãng đưa ra 1 phần mềm GX Work2 để có thể tiện lợi cho người sử dụng thiết lập các chương trình tải xuống PLC để thực thi những câu lệnh của người lập trình để điều khiển các thiết bị ngoại vi.

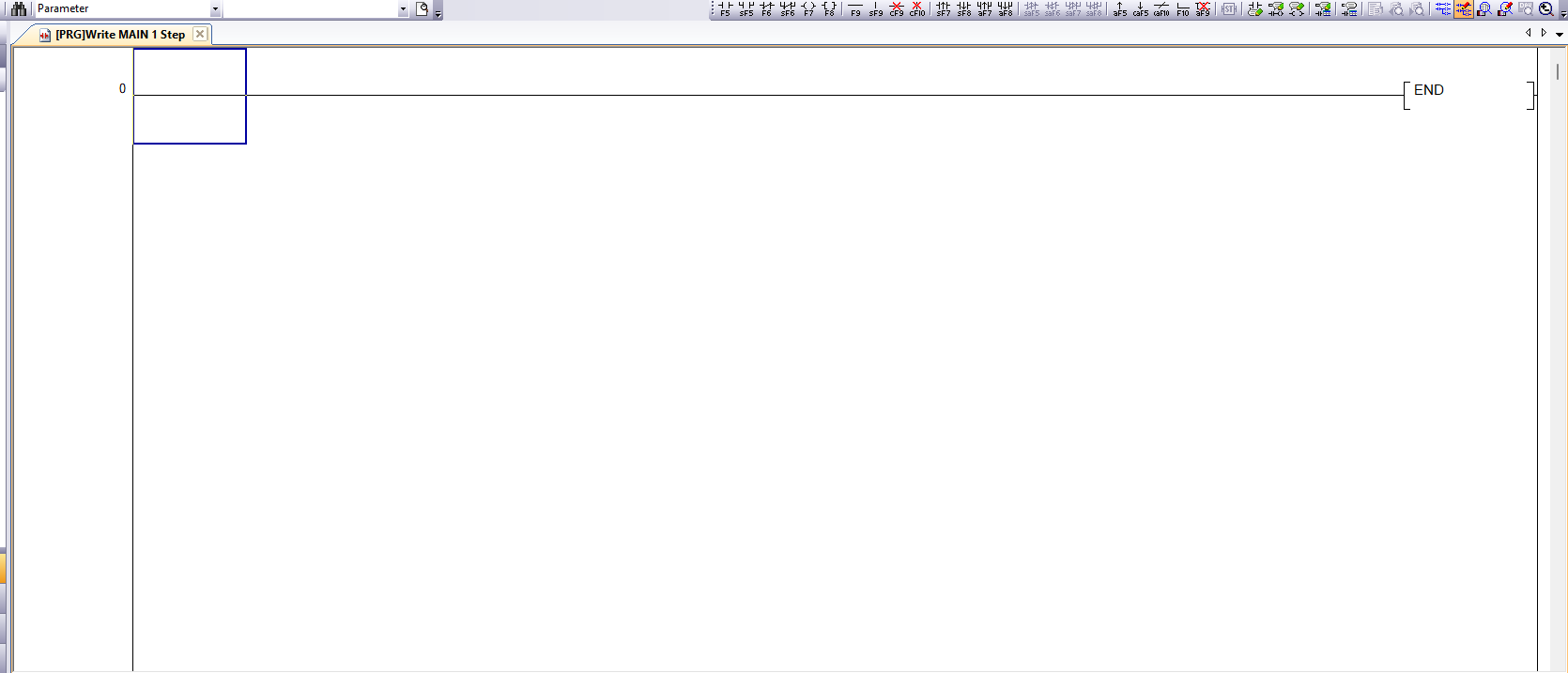
Ở lúc tạo new project, sẽ có 2 lựa chọn cho việc lập trình trong GX Work2: Ladder và SFC



**Ladder:** Đây là một ngôn ngữ lập trình đồ họa nhằm thể hiện các hoạt động logic thông qua các ký hiệu tượng trưng. Ladder Logic được cấu tạo từ các nấc thang logic, tạo thành một cấu trúc trông giống như cái thang, do đó có tên gọi là “Ladder”.

**SFC:** Viết tắt của cụm từ tiếng anh “Sequential Function Charts” tạm dịch là “Biểu đồ chức năng tuần tự”, bạn sẽ sử dụng các bước và quá trình chuyển đổi để đạt được kết quả cuối cùng của mình. SFC không phải là một ngôn ngữ, mà là một phương tiện đồ họa nhằm phân vùng mã và hiển thị trạng thái hoặc chế độ hoạt động một cách trực quan.

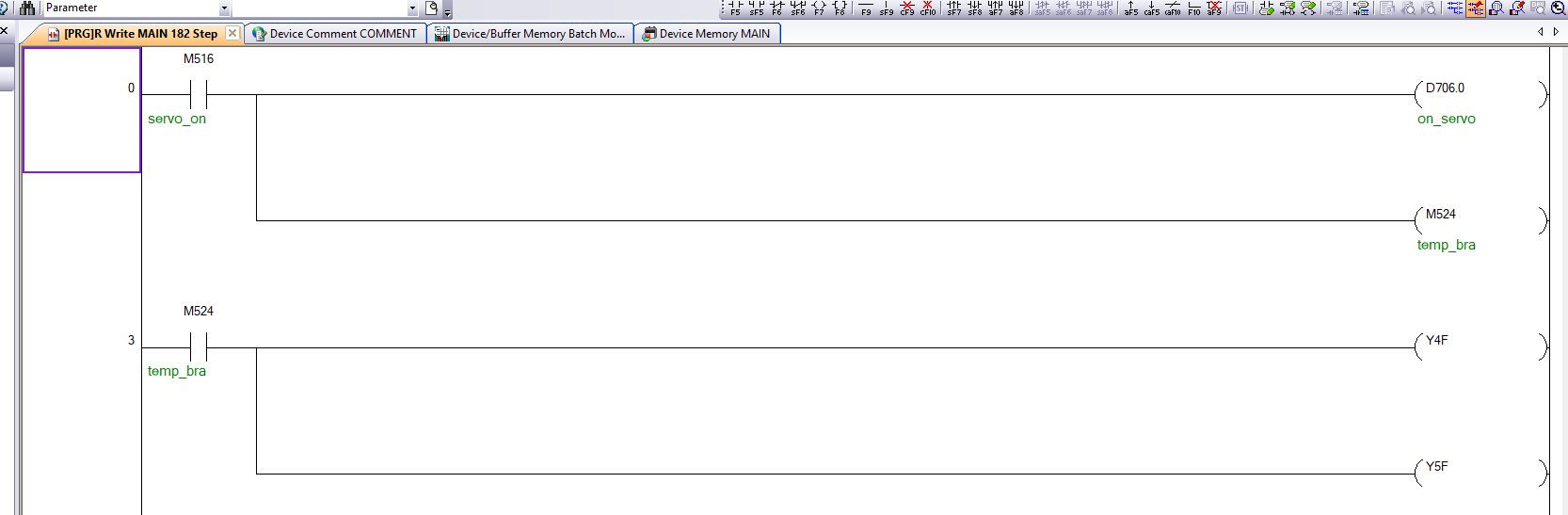
Trong chương trình hiện tại, ta chọn ngôn ngữ lập trình **Ladder** như là một phương thức quen thuộc để lập trình cho PLC.



**5.1.2.1.1. Chương trình điều khiển robot**

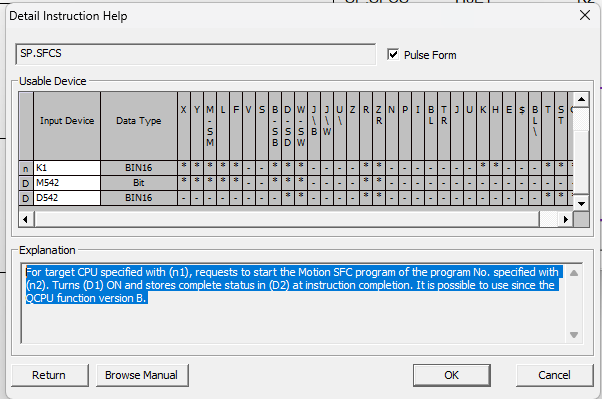
**Bật tắt Servo:**

Ta sẽ có những vùng nhớ được chia sẻ giữa Q02HCPU và Q172HCPU. Vùng nhớ **D706.0** được thiết đặt cho việc bật tắt servo. Và khi đã bật Servo, đồng nghĩa với việc ta phải nhả phanh điện từ ra khỏi động cơ với 2 ngõ ra **Y4F** và **Y5F**.



**Các chương trình điều khiển cơ bản:**

Khi đã hoàn tất cấu hình cho module Q02HCPU và Q172HCPU, ta có thể kết nối chúng bằng việc dùng chung các vùng dữ liệu với nhau. Đồng thời, GX Work2 cũng hỗ trợ hàm **SFCS** để có thể gọi chương trình bên MT Developer để thực hiện. Để rõ hơn thì ta có thể tra cứu trong datasheet do nhà sản xuất cung cấp



Cấu trúc của một hàm SP.SFCS được thể hiện như sau:



**H3E1:** Là vị trí của Q172HCPU được thể hiện dưới dạng mã hex.

**K[x]:** Là chương trình x được tạo ra trong phần mềm MT Developer2.

**M[x]:** Là relay ảo được bật lên lưu status vào vùng nhớ **D[x]** được cấu hình từ phía người dùng.

Tương ứng với từng chương trình, ta sẽ dùng lệnh này để gọi lên các chương trình SFC được tạo ở bên MT Developer2 để phục vụ cho từng mục đích khác nhau:

|  |  |
| --- | --- |
| **Định nghĩa** | **Tên chương trình** |
| Set home | K0 |
| Go home | K1 |
| MoveJ | K2 |
| Movepat | K3 |
| Adapt Control | K6 |

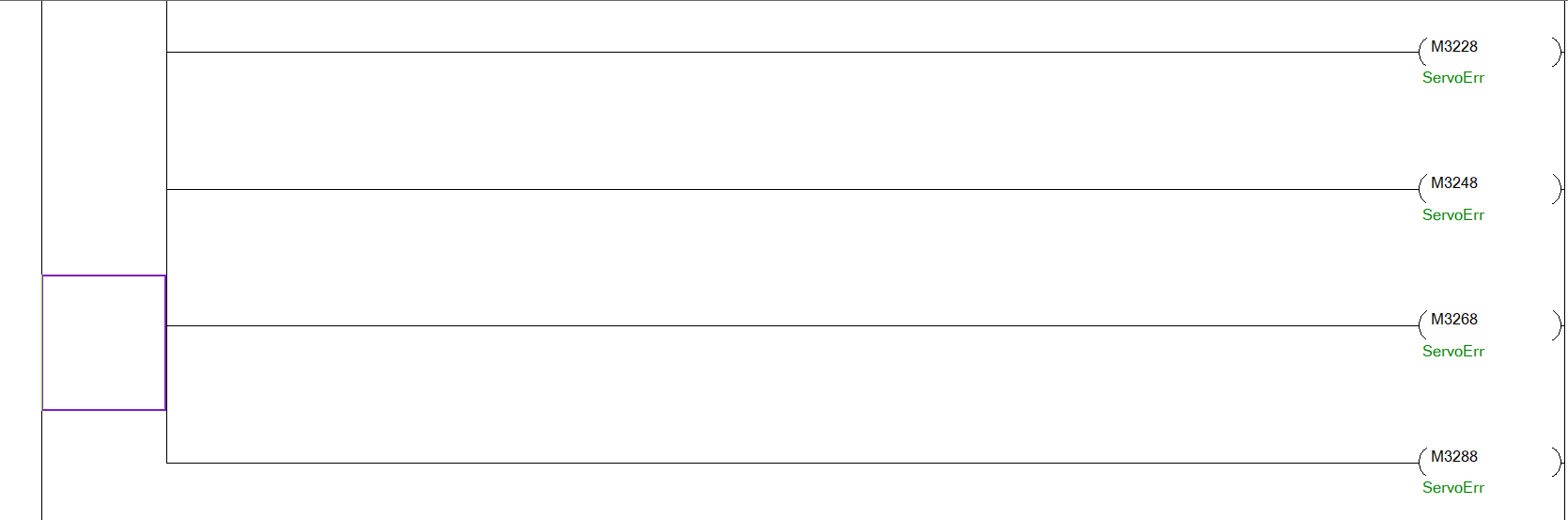
**Reset error:**

Trong quá trình điều khiển, sẽ không thể tránh khỏi những lỗi như: ngoài vùng hoạt động, low battery… và nhiều lỗi khác nữa. Khi gặp lỗi, module sẽ tự động bật lên và ngắt khả năng điều khiển từ bên phía phần mềm. Do vậy, việc khởi tạo lại trạng thái ban đầu, nhà sản xuất đã cung cấp những relay đặc biệt để thực hiện chức năng đó.

Có 2 cách để xem relay phục vụ cho chức năng gì: Kiểm bằng MT Developer2 và kiểm tra thông qua datasheet do nhà sản xuất cung cấp.

Trong hệ thống điều khiển, những relay dưới đây sẽ thực hiện reset error một số lỗi như sau:

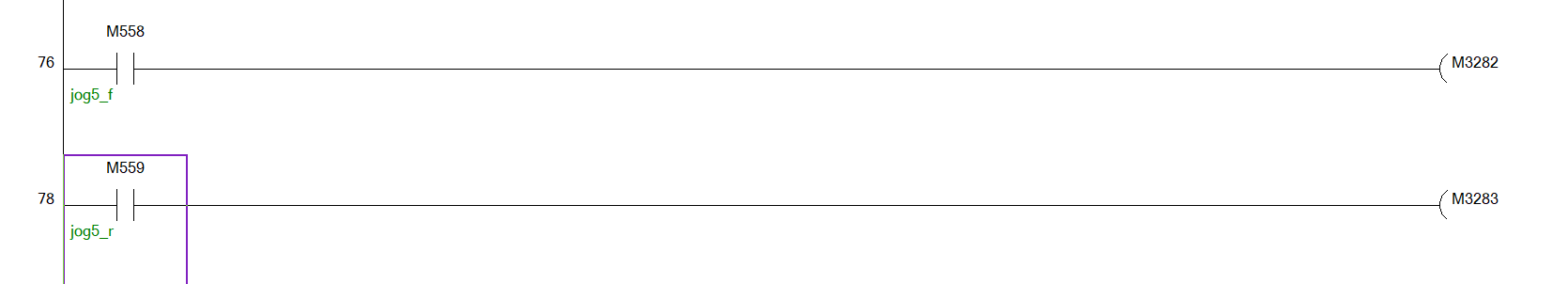




**Chạy jog**

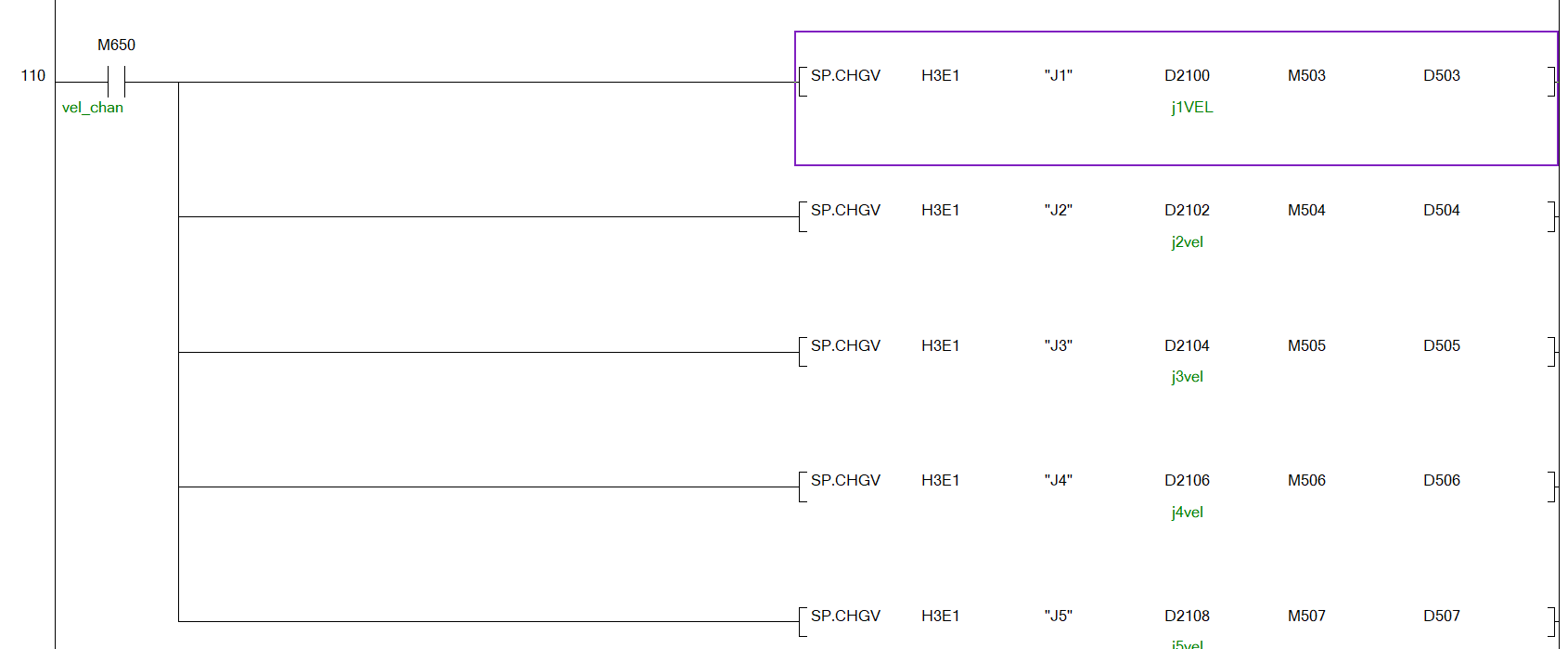
Ngoài ra, sẽ có những relay đặc biệt sẽ trigger thẳng trực tiếp đến servo motor để chạy jog cho từng động cơ.





**Thay đổi vận tốc tức thời**

Trong quá trình điều khiển cho một chương trình trong MT Developer2, ta sẽ phải định nghĩa từ đầu cho vận tốc của nó và thiết đặt vị trí cho từng khớp mong muốn và thực hiện bằng câu lệnh bằng SFCS để gọi hàm đó thực thi. Song, ta cũng có những vùng nhớ đặt biệt để thay đổi vận tốc của động cơ. Nhà sản xuất cũng cung cấp hàm **CHGV** để giúp thực thi điều đó:



**Quan sát hoàn thành chương trình**

Trong quá trình thực hiện 1 chương trình nào đó. Sẽ có những relay đặc biệt được bật lên để quan sát quá trình thực thi của động cơ, khi hoàn tất, những relays này sẽ được bật lên. Trong điều khiển, ta có thể sử dụng điều đó để kiểm tra xem rằng Robot đã thực hiện xong quy trình hay chưa:

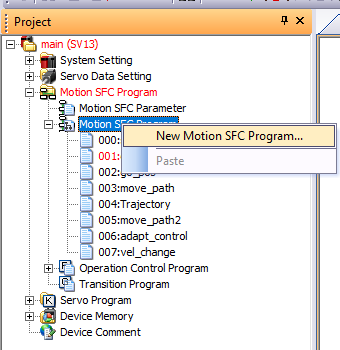
****

**5.1.3. Lập trình điều khiển robot trên MT Developer2**

Sau khi cấu hình xong cho module Q172HCPU, ta tiến hành lập trình cho các chương trình thực thi. Có 2 cách để tạo 1 chương trình: Tạo trực tiếp bằng **SFC program** hoặc tạo từng Servo Program và tổng hợp lại vào SFC program. Thường thì cách thứ nhất sẽ được sử dụng nhiều hơn do khả năng linh hoạt cũng như tiết kiệm thời gian.

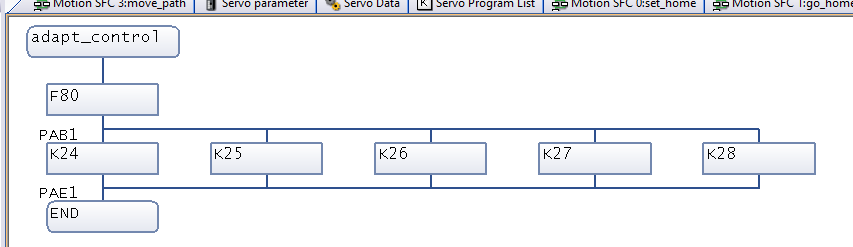
SFC Program đã được đề cập trên phần mềm GX Work2 ở phần các chương trình điều khiển cơ bản. Tương ứng với số chương trình trong phần mềm MT Developer2, sẽ là số chương trình nằm trong **K[x]** trong hàm SFCS của GX Work2.

**B1:** Tại tab **Motion SFC Program** ở thanh **Project,** ta xổ xuống thì sẽ thấy ở thư mục con **Motion SFC Program** 🡪 Chuột phải và chọn **New Motion SFC Program** như hình sau:



**B2:** Thiết lập cấu trúc chương trình

Về cấu trúc chương trình ta sẽ phải bắt đầu bằng 1 block **với tên được tạo** và kết thúc bằng 1 block **END.**



Về cấu trúc của 1 chương trình, ta sẽ thường dùng các block **F[x]** và **K[x]** để thực hiện các tác vụ:

**F[x]:** Là block để thực hiện các chức năng gán giá trị cho vùng nhớ.

**K[x]:** Là block dùng để thực hiện các hàm chức năng giúp trong việc di chuyển robot.

**5.1.3.1. Các chương trình trong hệ thống điều khiển**

**set\_home**

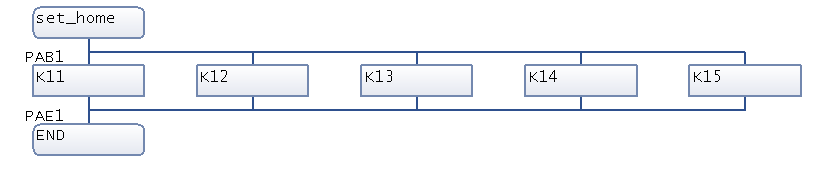
Trên thực tế, các dòng motor servo của các hãng Mitsubishi, ABB đều có khả năng lưu lại vị trí. Nhưng chỉ khi các servo module đều có nguồn dự phòng. Trong trường hợp có vấn đề với nguồn dự phòng, dữ liệu vị trí được lưu sẽ mất. Vì vậy, việc thiết đặt lại vị trí ban đầu cho robot là cần phải có trong hệ thống điều khiển.

MT Developer2 cho phép người dùng sử dụng lệnh để cài đặt lại vị trí ban đầu cho robot.



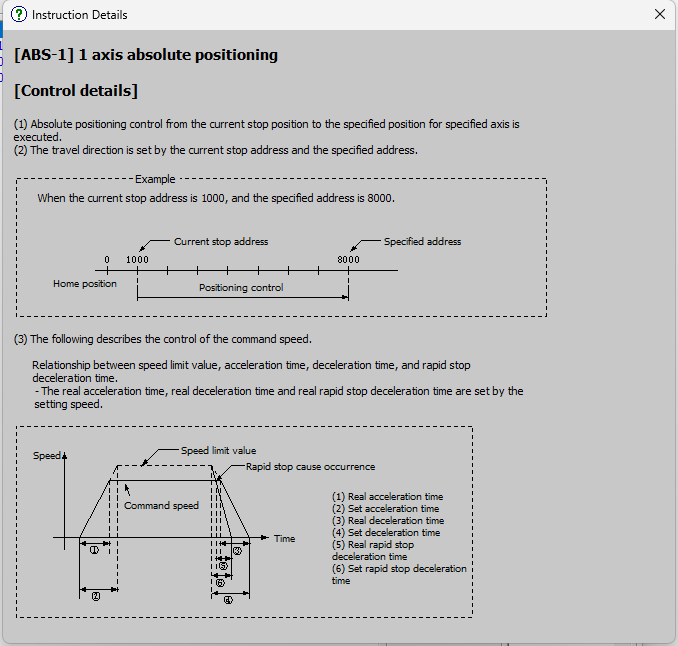
Lấy ví dụ như chương trình K11, ta dùng lệnh **ZERO** 🡪 **Axis [x]** với **x** là trục/khớp mà ta muốn thiết đặt vị trí ban đầu cho robot.

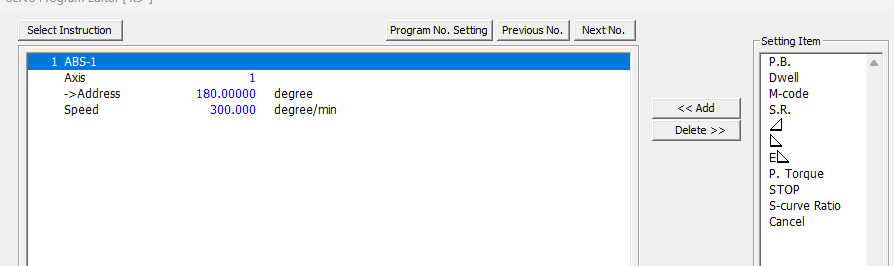
Thiết lập tương tự cho các Axis/khớp 2 🡪 5 tương ứng với K12 🡪 K15, ta có được một chương trình SFC hoàn chỉnh phục vụ cho việc Set home như sau:



**go\_home**

Ta dùng lệnh ABS-1 (Axis abosulte positioning) để điều khiển cho từng khớp đến vị trí tuyệt đối. Ta có thể tra trên datasheet của nhà sản xuất hoặc có thể xem trong tab Instruction Details.



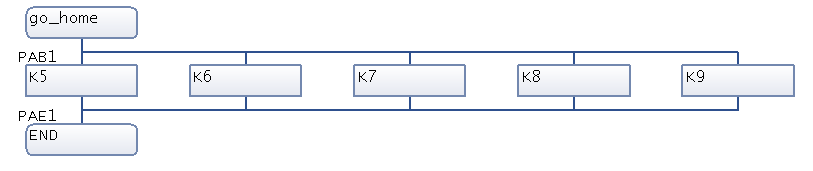


Hàm ABS-1 có rất nhiều thành phần cho phép điều khiển mà ta có thể thêm vào (Add) cũng như xóa (Delete) nằm ở bên ô **Setting Item.** Song để điều khiển được robot đến vị trí mong muốn, cơ bản ta chỉ cần yếu tố là **vị trí (Address)** và **vận tốc (Speed)**.

**Address:** Là vị trí mà ta mong muốn robot tới. Như đã nêu ở phần lưu ý trên phần thiết đặt, vị trí ban đầu của robot sẽ là 180.00000 degree.

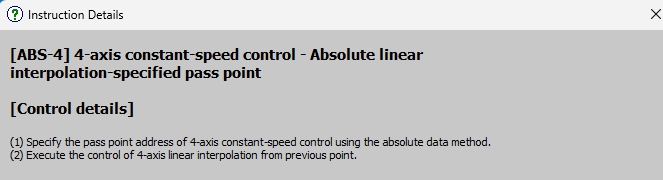
**Speed:** Với đơn vị là degree, thì ta cũng sẽ có thể chọn đơn vị vận tốc trong MT Developer2 là degree/min.

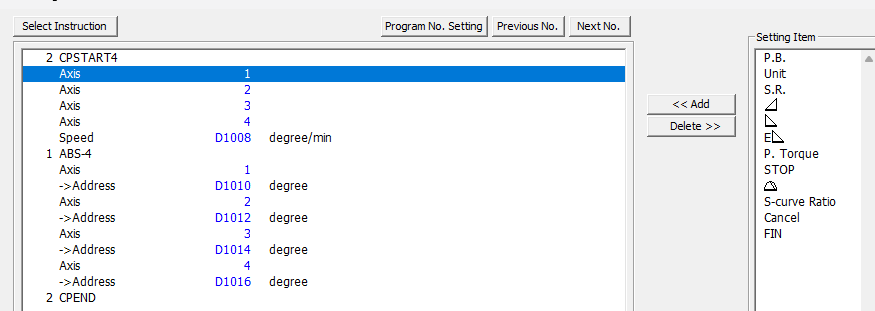
Tương ứng cho các trục/khớp từ 2 🡪 5 , ta thay đổi ở phần Axis và chọn vận tốc tương ứng với giá trị Address không đổi. Ta có 1 chương trình hoàn thiệt như sau:



**go\_pos**

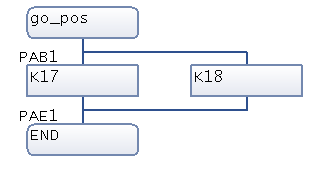
Ngoài ra với việc chia sẻ được vùng nhớ với Q02HCPU, ta có thể dựa vào đó mà đưa vào các giá trị góc mong muốn để có thể điều khiển được robot đến vị trí mình mong muốn bằng việc thay đổi giá trị **Address** trong các hàm bằng các vùng nhớ có thể thay đổi được vận tốc và vị trí với sự kết hợp sử dụng hàm ABS-4 (Axis constant speed control).





***Lưu ý:*** *Address sẽ cần 32 bit để lưu giá trị. Ta cần phải có 2 vùng nhớ 16 bit. Như hình trên, với việc khai báo D1010 thì mặc định hàm sẽ đọc từ D1010 -> D1011 để lưu vị trí vào module servo để điều khiển robot. Tương tự với các Axis còn lại*

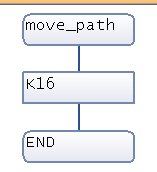
Nhưng với việc robot trong điều khiển đang có 5 bậc tự do, trong khi hàm ABS-4 chỉ hỗ trợ 4 trục. Ta có thể giải quyết vấn đề này bằng cách kết hợp thêm hàm ABS-1 để điều khiển cho khớp còn lại. Chương trình hoàn chỉnh như sau:



**move\_path**

Ngoài ra bằng việc kết hợp các chuỗi hàm ABS-4, ta có thể điều khiển được vị trí và vận tốc di chuyển tuân theo quỹ đạo mà ta quy định. Trong chương trình điều khiển, quỹ đạo đang được chia thành 100 điểm tải xuống robot để điều khiển.

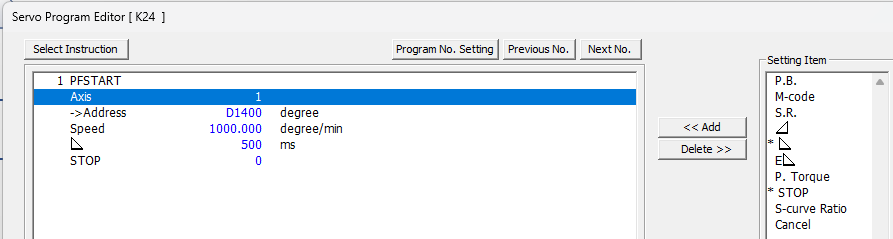
Ta có chương trình hoàn thiện như hình:





**adapt\_control**

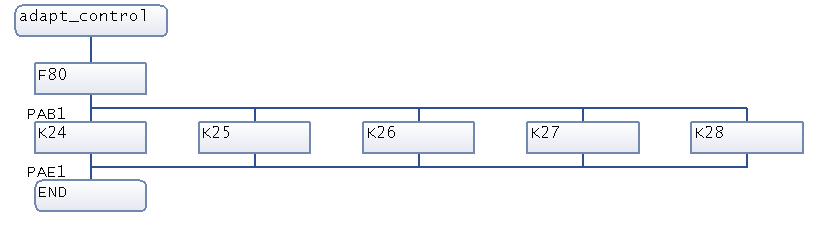
Ta cũng sẽ có những vùng nhớ đặc biệt giúp cập nhật vị trí muốn đến một cách tức thời cho robot. Bằng việc truy cập vào chúng, module servo sẽ có thể nhận diện và đáp ứng vị trí sau một khoảng thời gian mà ta cài đặt



Như trên hình, vùng nhớ D1400 là vùng nhớ đặc biệt dành cho trục 1, ta có thể cài đặt vận tốc mong muốn cũng như thời gian đáp ứng sao cho tương thích.

*Lưu ý: Chế độ adapt control sẽ được bật khi ta phải thiết đặt lại cờ PFSTART bằng việt kích hoạt relay ảo M3200 thì mới hoạt động được.*

Chương trình hoàn chỉnh của **adapt control**:



5.2. Xây dựng phần mềm cho băng tải

5.3. Xây dựng phần mềm cho thiết bị theo dõi chuyển động

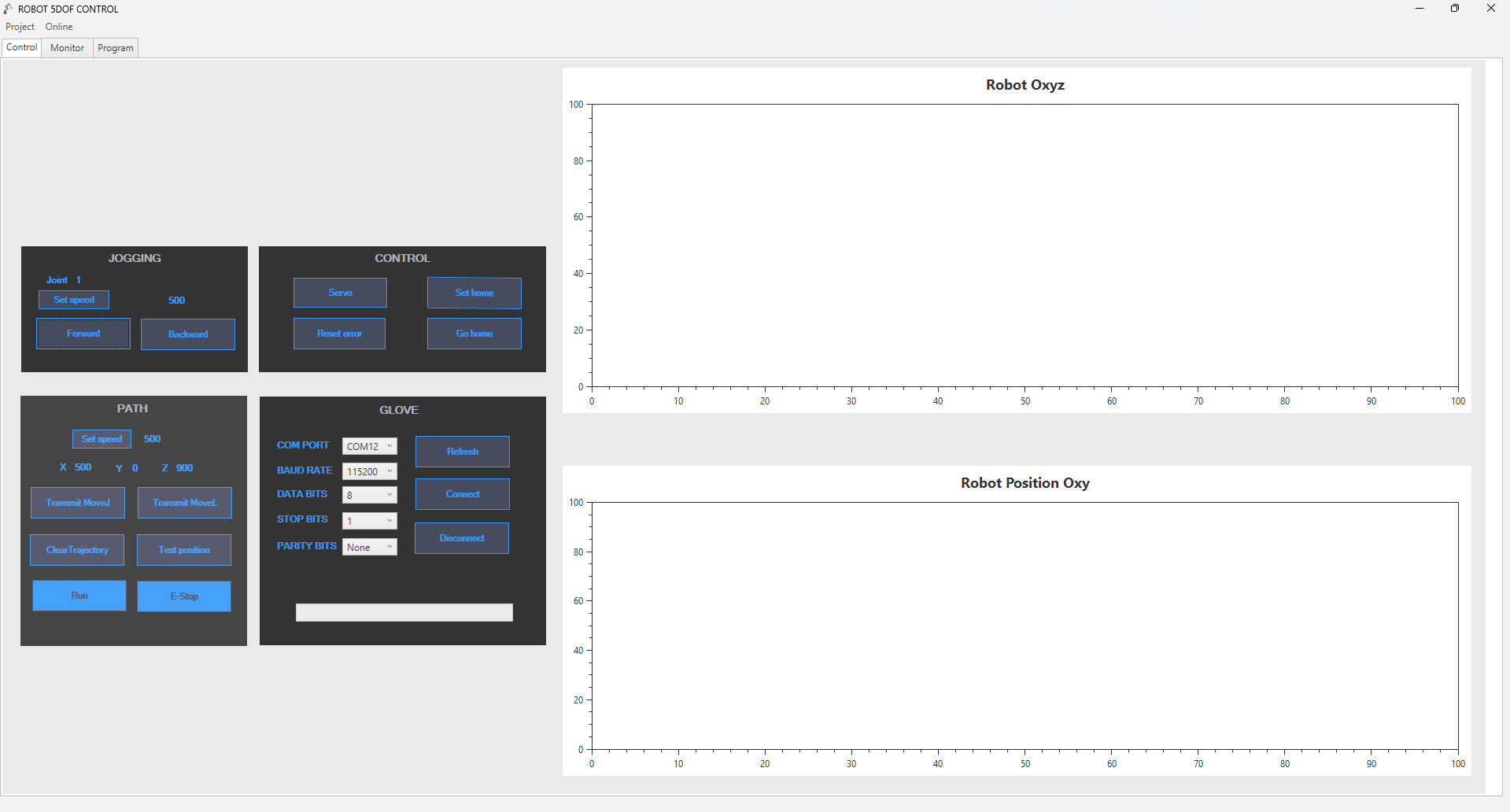
5.4. Xây dựng giao diện giám sát, điều khiển cho hệ thống bằng WPF

### **5.4.1. Giao diện điều khiển với WPF**

**Tổng quan về giao diện điều khiển:** Giao diện điều khiển robot sẽ được chia làm tab chính: Control, Monitor và Program. Tương ứng với từng vùng sẽ mang một chức năng khác nhau:

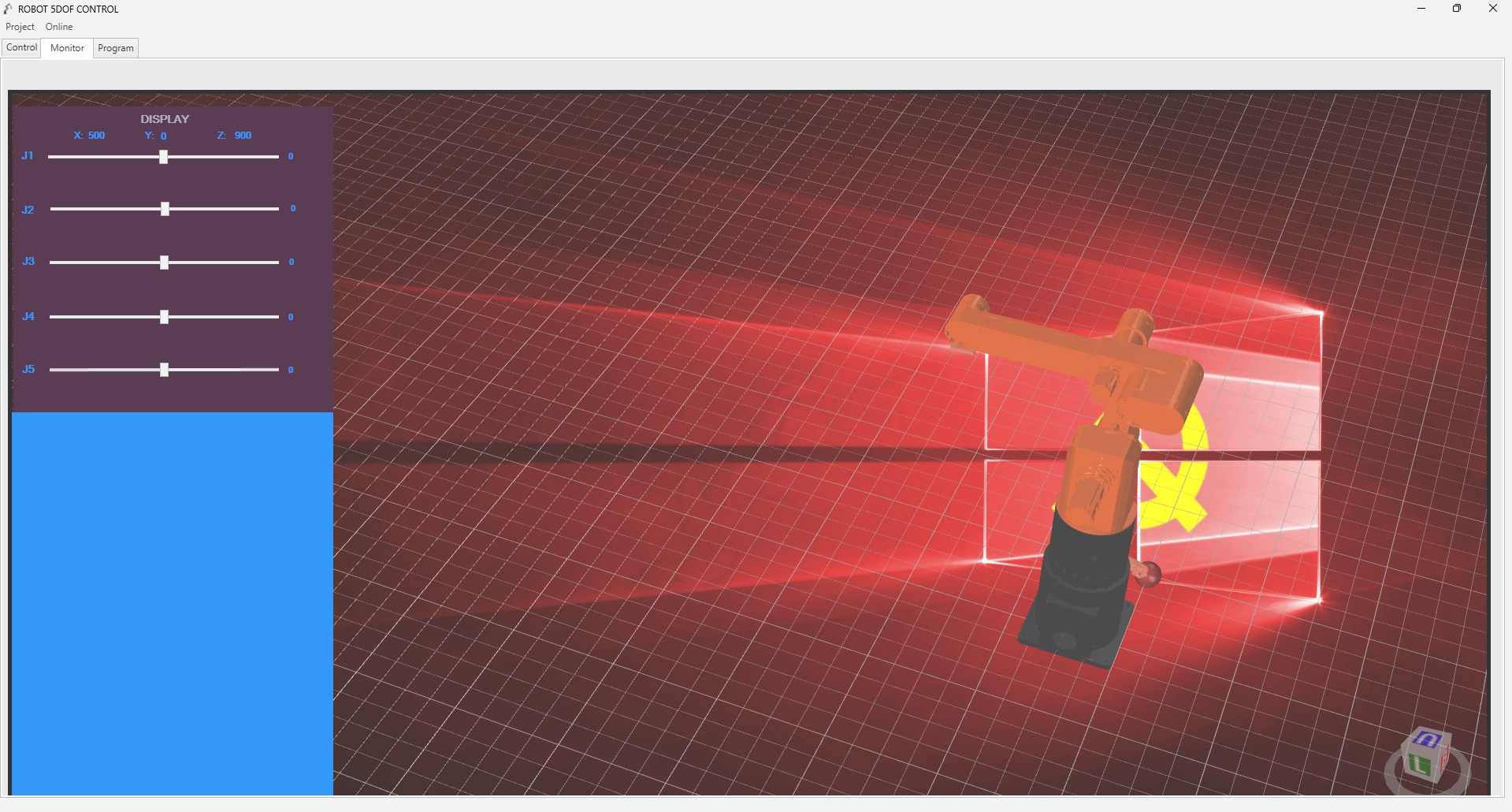
**Control:**

* Chạy Jog cho từng khớp của robot.
* Thực hiện những chức năng cơ bản như: Bật tắt Servo, Set home, Go home và Reset error.
* Thực hiện những lệnh chạy path như MoveJ, MoveL, Test Position và Clean Trajectory.
* Cổng kết nối UART để kết nối với thiết bị theo dõi chuyển động cánh tay người cũng như PLC Siemen.
* Đồ thị vận tốc và vị trí phục vụ cho việc theo dõi chuyển động.

****

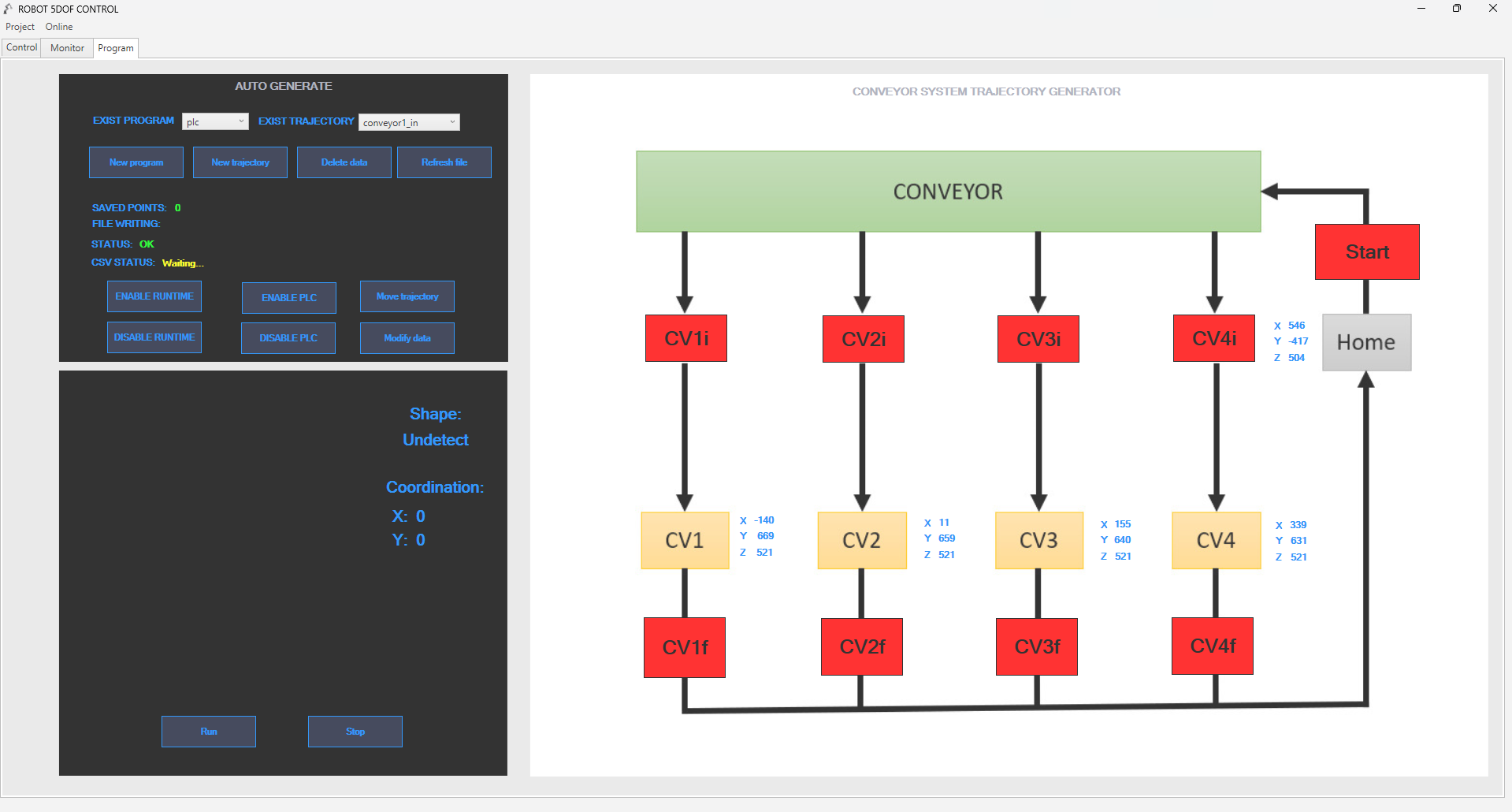
**Monitor:**

* Dùng để quan sát cũng như mô phỏng chuyển động của robot thực tế so với mô phỏng bằng robot 3D và Slider.
* Quan sát được trạng thái robot trong quá trình điều khiển thông qua textbox bao gồm như trạng thái nhấn các nút điều khiển, kiểm tra lỗi conversion động học nghịch robot cho các khớp từ hệ tọa độ…

****

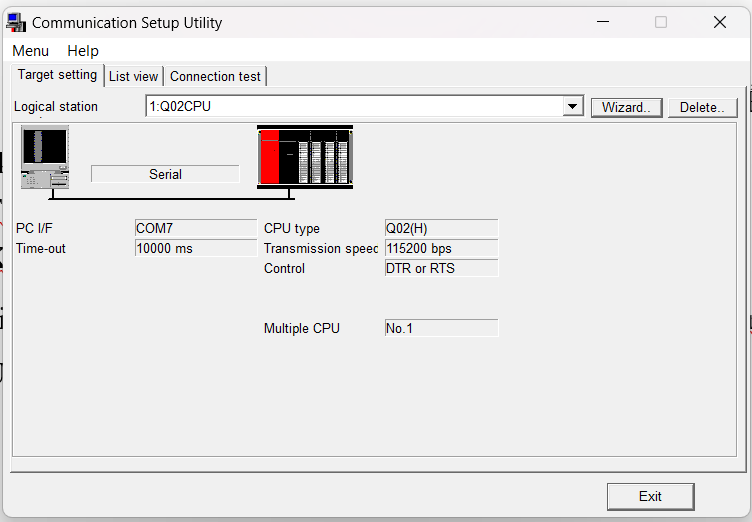
**Program:**

* Bật/Tắt chế độ điều khiển robot thông qua hệ thống điều khiển chuyển động và chức năng lưu quỹ đạo của robot tự động và bằng thủ công.
* Thể hiện được trạng thái số điểm được lưu lại vào tệp csv chứa quỹ đạo hiện tại.
* Tạo/Xóa tệp hiện có trong nơi chứa quỹ đạo của robot.
* Thể hiện được hình ảnh đã qua xử lý và trả ra kết quả nhận diện được từ camera về hình dạng cũng như trọng tâm từ camera.
* Bật/tắt chế độ điều khiển tự động hệ thống bốc xếp sản phẩm.

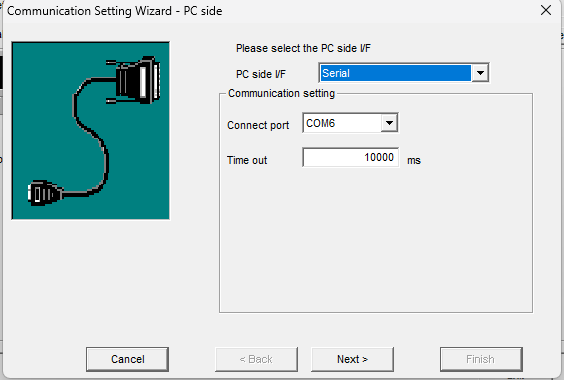
****

### **5.4.2. Cách kết nối thư viện PLC Mitsubishi với máy tính**

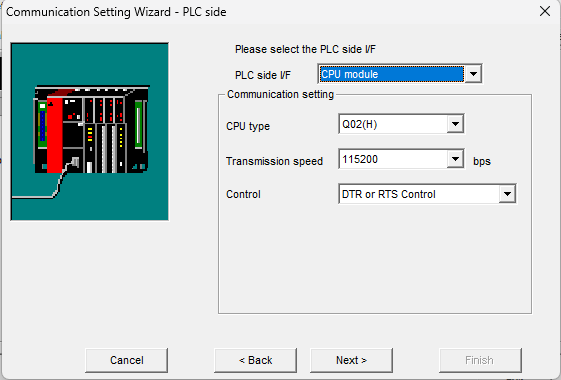
PLC Mitsubishi hỗ trợ thư viện **ActUtlTypeLib** thông qua phần mềm Communication Setup Utility để có thể kết nối được giữa PLC và C# thông qua serial port RS232.

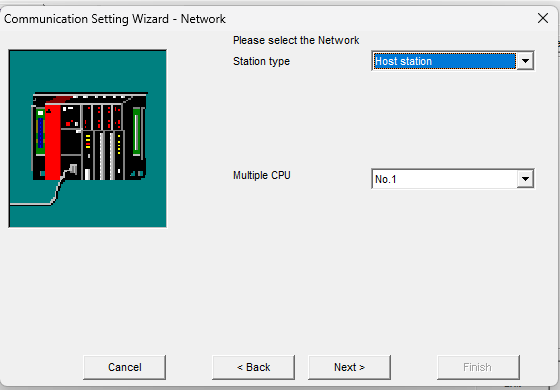


*Lưu ý: Khi mở phần mềm Communication Setupt Utility, phải mở bằng quyền Admin thì mới điều chỉnh được thông số trong phần mềm.*

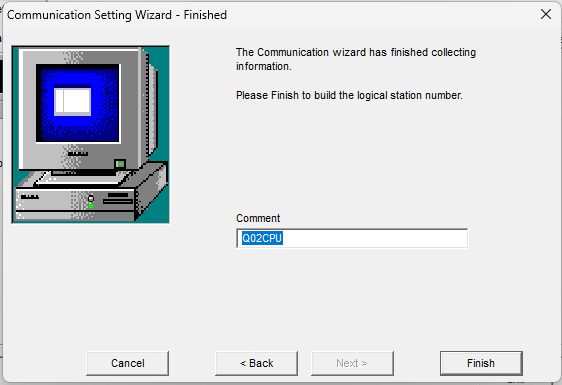
**

**Connect port** sẽ phụ thuộc vào cổng COM khi cắm dây RS232 vào máy tính. Ta có thể dễ dàng kiểm tra bằng Device Manager.



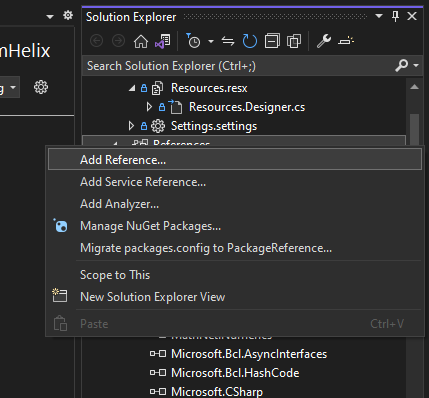


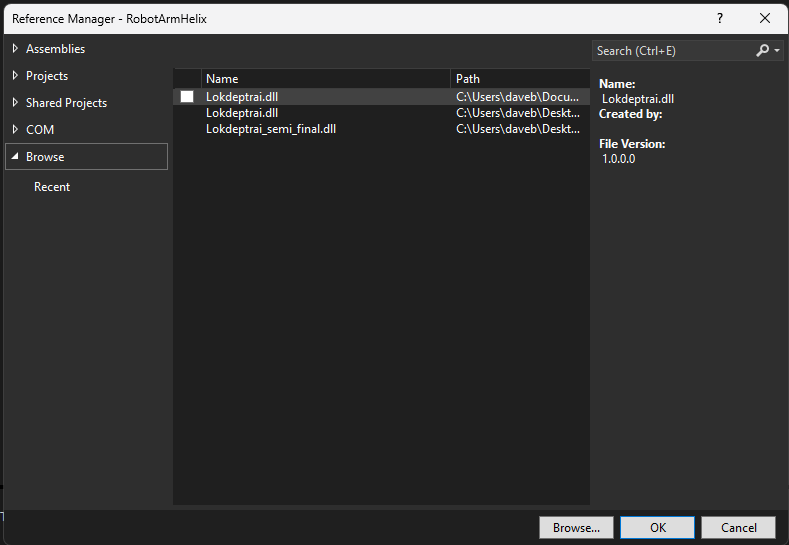
**Multiple CPU:** sẽ phụ thuộc vào vị trí lắp đặt module Q02HCPU 🡪 Hiện tại đang được đặt ở slot 1 🡪 **No.1**



Sau khi setup trên phần mềm, ta tiến hành cài đặt thư viện sử dụng trong WPF:

**B1: Add referrence vào trong chương trình:** Tại cửa sổ lập trình ở ô Solution Explorer góc bên phải 🡪 Chuột phải vào References 🡪 Add reference… 🡪 Tại cửa sổ Reference Manager 🡪 Chọn Browse 🡪 Ta chọn đường dẫn đến thư viện *ActUtlTypeLib.dll* 🡪 Nhấn OK và hoàn tất.





B2: ta khai báo cách kết nối trên C# bằng một số câu lệnh trong WPF ở tệp code-behind (.xaml.cs):

*using ActUtlTypeLib;*

Sau khi thiết đặt xong tất cả các cấu hình, ta đã hoàn tất việc kết nối giữa module Q02HCPU và có thể điều khiển relay ảo cũng như thay đổi các vùng nhớ thông qua phần mềm WPF.

5.4.3. Các phương pháp điều khiển quỹ đạo trong WPF

5.4.3.1. Phương pháp điều khiển theo đường thẳng

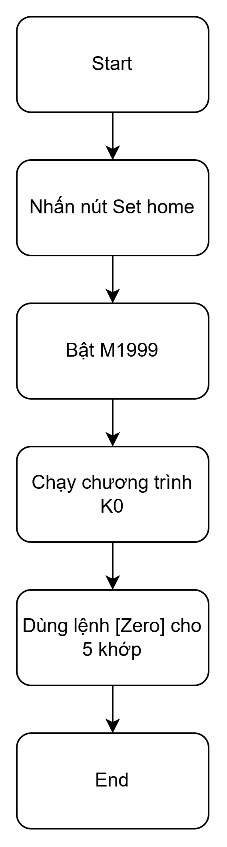
Từ dữ liệu đầu vào là 2 đầu của đoạn thẳng ta có thể xác định được phương trình và từ đó tìm ra đồ thị vị trí của 5 trục khi thực hiện quỹ đạo. Để có thể thực hiện chuyển động của robot bám theo đường thẳng mong muốn ta cần phải điều khiển vị trí của 5 trục theo đồ thị đã tính toán dựa trên quỹ đạo cần di chuyển. Ta tiến hành chia quỹ đạo thành các đoạn nhỏ, mỗi đoạn nhỏ sẽ sử dụng hàm Constant Speed để tiến hành di chuyển. Số điểm chia càng nhiều, quỹ đạo bám theo sẽ càng chính xác.

5.4.3.2. Phương pháp điều khiển theo hệ thống theo dõi chuyển động

Trong PLC, nhà sản xuất cho phép ta can thiệp vào những vùng nhớ đặc biệt giúp thiết đặt các giá trị góc một cách tức thời cho động cơ, để điều khiển động cơ ngay tức thì. Dựa vào đó ta sẽ dùng hệ thống theo dõi chuyển động để cập nhật vị trí tức thời cho robot. Từ đó sẽ tạo ra 1 quỹ đạo với một đường cong bất kỳ bám theo chuyển động của tay người. Tăng khả năng linh hoạt và chính xác của robot.

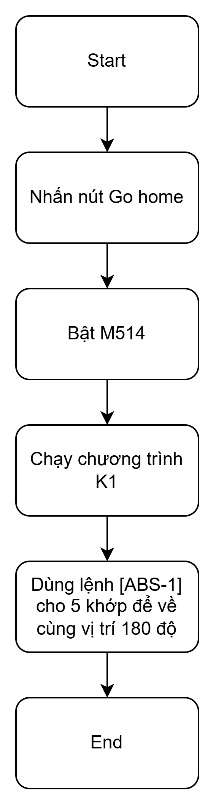
5.4.3.3. Giải thuật cho các câu lệnh điều khiển trên WPF

Set home



*Lưu đồ cho lệnh Set Home*

**Go Home**



*Lưu đồ cho lệnh Go Home*

**Chạy đơn điểm (MoveJ)**



*Lưu đồ cho lệnh MoveJ*

**Chạy đa điểm (MoveL)**



*Lưu đồ cho lệnh MoveL*

### **5.4.4. WPF với thiết bị theo dõi chuyển động**

**Lưu đồ giải thuật:**

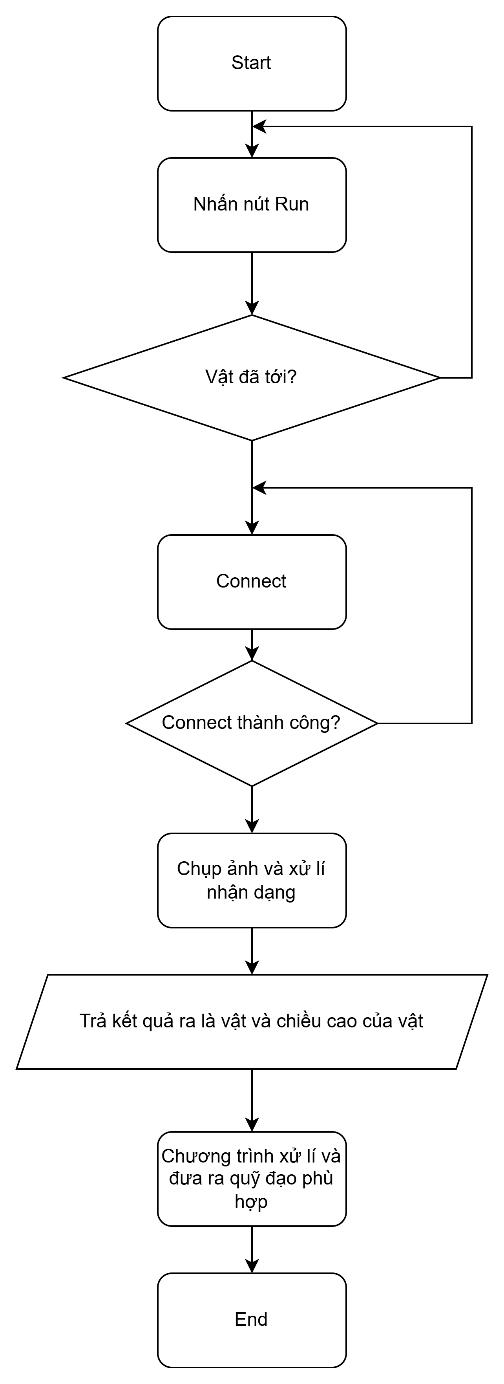


*Lưu đồ cho lệnh chạy Runtime*

Như đã trình bày ở chương trước, sau khi ta nhận được frame truyền từ thiết bị theo dõi chuyển động, ta tiến hành bóc tách và xử lý từng byte trong frame từ đó có được vị trí mong muốn và vận tốc của găng tay. Từ đó tiến hành tính toán động học nghịch và áp dụng Jacobi nghịch thì ta sẽ có được chuyển động của robot sẽ đáp ứng tức thời cả về vị trí và vận tốc của cánh tay robot.

### **5.4.5. WPF với camera nhận dạng vật ảnh O2D222**

**Lưu đồ giải thuật:**



Bằng việc giao tiếp thông qua giao thức TCP/IP, ta có thể kết nối được với camera và dựa vào những thuật toán xử lí ảnh đã nêu ở chương trước. Ta có thể áp dụng vào hệ thống điều khiển phục vụ cho hệ thống điều khiển tự động.

### **5.4.6. WPF với PLC Siemens**

**Lưu đồ hoạt động:**



Bằng việc giao tiếp với PLC Siemen với WPF thông qua UART, ta có thể sử dụng nó như một phương thức truyền thông giữa chúng để từ đó có thể thiết lập nên 1 hệ thống điều khiển tự động hình thành nên một quy trình điều khiển.

# CHƯƠNG 6: KẾT QUẢ - THỰC NGHIỆM – HƯỚNG PHÁT TRIỂN