Thiết kế hệ thống giám sát, mô phỏng và điều khiển robot (SCADA) sử dụng chuẩn truyền thông OPC UA ứng dụng trong hệ robot công nghiệp

[Chương 1. Giới thiệu đề tài 4](#_Toc134278536)

[1.1. Tính cấp thiết của luận văn 4](#_Toc134278537)

[1.2. Mục tiêu của luận văn 6](#_Toc134278538)

[1.3. Đối tượng và phạm vi nghiên cứu 7](#_Toc134278539)

[1.3.1. Đối tượng nghiên cứu 7](#_Toc134278540)

[1.3.2. Phạm vi nghiên cứu 7](#_Toc134278541)

[1.4. Nhiệm vụ và nội dung của luận văn 8](#_Toc134278542)

[1.5. Cấu trúc của luận văn 8](#_Toc134278543)

[Chương 2. Tổng quan về hệ thống robot công nghiệp và chuẩn truyền thông OPC UA 9](#_Toc134278544)

[2.1. Khái niệm về robot công nghiệp và sự phát triển của robot công nghiệp trong sản xuất 9](#_Toc134278545)

[2.1.1. Khái niệm về robot công nghiệp 9](#_Toc134278546)

[2.1.2. Sự phát triển của robot công nghiệp trong sản xuất 9](#_Toc134278547)

[2.2. Khái niệm về hệ thống SCADA và cấu trúc của một hệ thống SCADA 11](#_Toc134278548)

[2.2.1. Khái niệm về hệ thống SCADA 11](#_Toc134278549)

[2.3. Các chuẩn truyền thông đang được sử dụng ở tầng SCADA trong các hệ thống robot công nghiệp 14](#_Toc134278550)

[2.3.1. Hãng ABB 14](#_Toc134278551)

[2.3.2. Hãng Yaskawa 14](#_Toc134278552)

[2.3.3. Hãng Mitsubishi 14](#_Toc134278553)

[2.4. Tổng quan về chuẩn truyền thông OPC UA 14](#_Toc134278554)

[2.4.1. Khái niệm về OPC UA 14](#_Toc134278555)

[2.4.2. Các tính năng nổi bật của OPC UA 15](#_Toc134278556)

[Chương 3. Tổng quan về cấu trúc hệ thống 16](#_Toc134278557)

[Chương 4. Thiết kế các phần mềm cho hệ thống SCADA trên máy tính 17](#_Toc134278558)

[4.1. Thiết kế OPC UA Server 17](#_Toc134278559)

[4.1.1. Giới thiệu về thư viện OPC UA Client & Server SDK hỗ trợ thiết kế OPC UA Server trên phần mềm Visual Studio 2022 17](#_Toc134278560)

[4.1.2. Cấu trúc OPC UA Server đã thiết kế 18](#_Toc134278561)

[4.1.3. Tính năng bảo mật của OPC UA Server 20](#_Toc134278562)

[4.1.4. SQL Database của OPC UA Server 21](#_Toc134278563)

[4.2. Thiết kế OPC UA Client 21](#_Toc134278564)

[4.2.1. Giới thiệu về thư viện mở open62541 hỗ trợ thiết kế OPC UA Client 21](#_Toc134278565)

[4.2.2. Chức năng của phần mềm OPC UA Client 22](#_Toc134278566)

[4.2.3. Giao diện giám sát của các robot trong hệ thống 23](#_Toc134278567)

[4.2.3.1. Giao diện giám sát các thông số của robot 23](#_Toc134278568)

[4.2.3.2. Giao diện MONITOR 24](#_Toc134278569)

[4.2.3.3. Giao diện SIMULATION 25](#_Toc134278570)

[4.2.3.4. Giao diện HISTORY 31](#_Toc134278571)

[4.2.4. Giao diện điều khiển robot trong hệ thống 34](#_Toc134278572)

[4.2.4.1. Giao diện JOG 37](#_Toc134278573)

[4.2.5.2. Giao diện PROGRAM 45](#_Toc134278574)

[4.2.5.3. Giao diện DATA 51](#_Toc134278575)

[4.2.5.4. Giao diện I/O 54](#_Toc134278576)

[Chương 5. Tích hợp OPC UA vào bộ điều khiển robot 57](#_Toc134278577)

[5.1. Giới thiệu tổng quan về giao tiếp giữa các tiến trình (Inter-process communication) và ứng dụng đọc/ghi dữ liệu của bộ điều khiển robot 58](#_Toc134278578)

[5.2. Cơ chế gửi lệnh điều khiển từ PC và lệnh phản hồi từ bộ điều khiển robot của hệ thống 59](#_Toc134278579)

[5.3. Xử lý lệnh điều khiển và thực thi lệnh 61](#_Toc134278580)

[5.4. Cập nhật các thông số giám robot lên OPC UA Server 63](#_Toc134278581)

[Chương 6. Kết quả và hướng phát triển 64](#_Toc134278582)

[6.1. Kết quả 64](#_Toc134278583)

# Chương 1. Giới thiệu đề tài

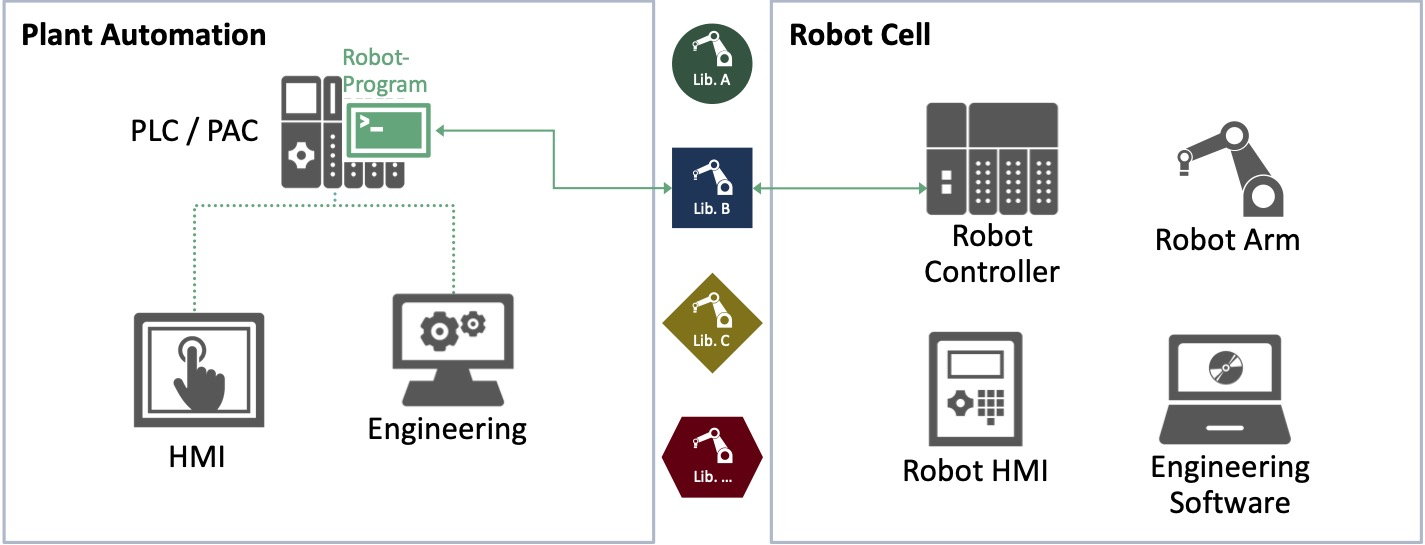
* 1. Tính cấp thiết của luận văn

Tình hình nghiên cứu:

* Ngoài nước:

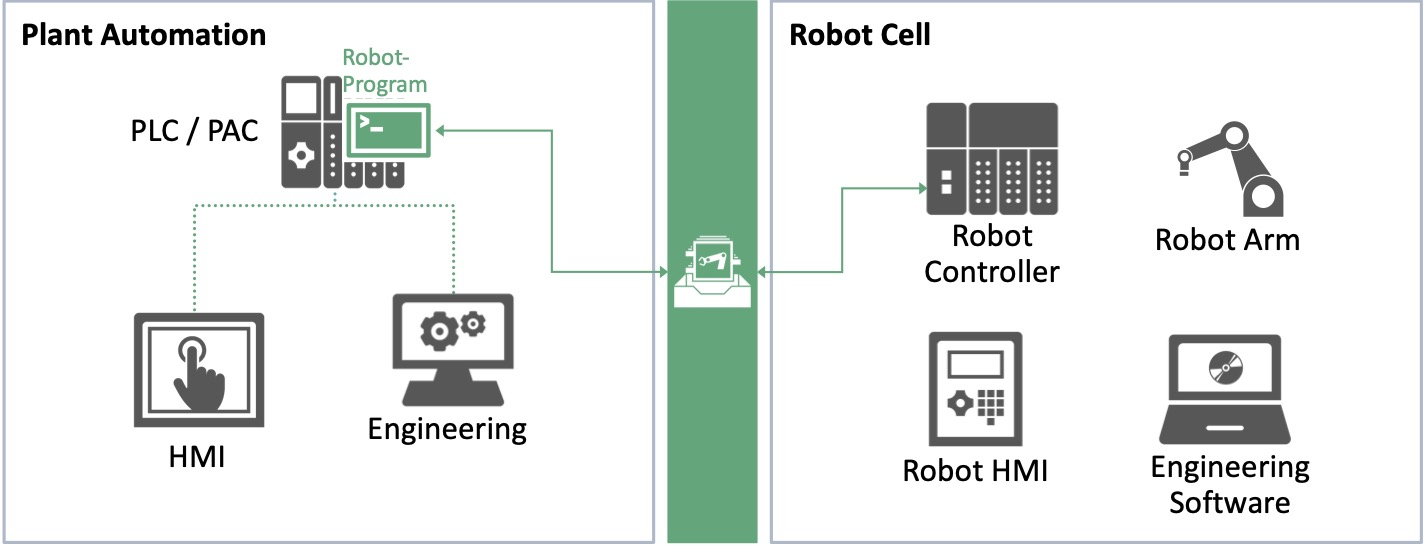
Trong những năm gần đây, cách mạng công nghiệp 4.0 đang thay đổi và có tác động mạnh mẽ đến các ngành sản xuất, trong đó có Robotics. Một trong những thách thức lớn nhất trong lĩnh vực này chính là sự thiếu thống nhất, thiếu đồng bộ trong kết nối truyền thông giữa các robot và các thiết bị điều khiển từ các nhà sản xuất khác nhau.

Ở trong mỗi nhà máy, một khối robot (robot cell) hoàn chỉnh bao gồm các thành phần như cánh tay robot, bộ điều khiển robot, phần mềm lập trình robot và 1 màn hình HMI. Một PLC/PAC có nhiệm vụ giám sát, điều khiển các khối robot này trong một dây chuyền sản xuất của nhà máy. Nhưng một trong những thách thức to lớn là các khối robot này được sản xuất từ các hãng khác nhau, mỗi hãng hỗ trợ riêng các thư viện để giao tiếp truyền thông, lập trình robot độc quyền của hãng đó. Điều này dẫn đến sự phức tạp để trong đồng bộ hóa việc giám sát, điều khiển các khối robot trong dây chuyền sản xuất. Ví dụ như, đối với Mistubishi sử dụng giao thức CC Link IE Field, trong khi Yaskawa sử dụng Profinet và Mechatrolink, tương tự ABB, KUKA, … cũng có kiến trúc và chuẩn giao tiếp riêng của họ.



Hình 1.0.1. Sơ đồ thể hiện sự bất đồng bộ trong truyền thông của các hãng robot khác nhau

Nhận ra vấn đề trên, 2011 ABB Robotics đã cung cấp giải pháp OPC Server cho robot và bộ điều khiển của hãng ABB, nhằm giải quyết vấn đề thiếu đồng bộ. Cấu trúc OPC Server khá mở, cho phép người dùng sử dụng hiệu quả phần mềm bất kể sự khác nhau về hệ thống phần mềm, phần cứng, hay bất kỳ công nghệ nào đều có khả năng sử dụng được Server. Tuy nhiên, giải pháp OPC Server này chưa phải mới nhất, còn thiếu các đặc điểm như bảo mật thông tin, mô hình hóa dữ liệu, khả năng mở rộng khi so sánh với chuẩn truyền thông OPC UA. Mặc dù vậy, các hệ thống phát triển với OPC UA để ứng dụng vào Robotics hiện nay vẫn chưa hoàn chỉnh. Kể cả ABB Robotics hay Mistubishi, các hệ thống này còn phải phụ thuộc vào dịch vụ của bên thứ ba, các sản phẩm này hơn nữa còn thiếu khả năng hoạt động đa nền tảng, và khó sử dụng.



Hình 1.0.2. Giải pháp đồng bộ hóa các chuẩn truyền thông của các hãng robot khác nhau

* Trong nước:

Chuẩn truyền thông OPC UA là chuẩn truyền thông khá mới và ưu việt được phát hành từ năm 2008, vì vậy ứng dụng của chuẩn này vẫn chưa được áp dụng nhiều và rõ ràng trong các nhà máy có hệ thống robot ở Việt Nam, chủ yếu mới được tích hợp trên PLC của hãng nước ngoài. Vì vậy hướng nghiên cứu này khá mới, có tính ứng dụng cao và có thể áp dụng ngay được vào các nhà máy, xí nghiệp vừa và nhỏ sử dụng hệ thống robot.

* 1. Mục tiêu của luận văn

Trong đề tài nghiên cứu này, em đã tập trung phát triển một kiến trúc truyền thông dựa trên chuẩn truyền thông OPC UA cho phép người dùng dễ dàng theo dõi và điều khiển các robot trong phòng thí nghiệm. Đề tài đã xây dựng hoàn thiện và phát triển một hệ thống quản lý, giám sát, mô phỏng, trao đổi dữ liệu và điều khiển robot bao gồm các robot, controller, cảm biến, máy tính server, máy tính client, …

Hệ thống có mục tiêu làm cải thiện, nâng cao năng suất nhà máy, xí nghiệp vì dữ liệu robot được đồng bộ, cập nhật đưa lên server, mô phỏng trước khi thực nghiệm để đảm bảo an toàn, hiệu quả cho robot, gửi lệnh trực tiếp từ người dùng đến bộ điều khiển trực tiếp từ người dùng đến bộ điều khiển robot với các lệnh có thể cài đặt và điều chỉnh bởi người dùng.

Hệ thống này áp dụng với lên Robot DELTA của VAS Corporation và Robot SCARA của RoboStar. Ở tầng SCADA, nhóm nghiên cứu sử dụng EtherCAT là chuẩn truyền thông thời gian thực để nâng cao chất lượng truyền thông cho Robot.

Luận văn khi hoàn thành sẽ phải đáp ứng được các mục tiêu sau đây:

* Hệ thống được thiết kế có khả năng giám sát, mô phỏng và điều khiển các robot một cách chính xác và ổn định trong môi trường công nghiệp sử dụng chuẩn truyền thông OPC UA.
* Xây dựng, phát triển các tính năng bảo mật, cơ sở dữ liệu cho hệ thống robot.
* Tích hợp vào bộ điều khiển robot và thực nghiệm giám sát, điều khiển hệ thống robot thật.
  1. Đối tượng và phạm vi nghiên cứu
     1. Đối tượng nghiên cứu
* Chuẩn truyền thông OPC UA
* Lý thuyết về kỹ thuật robot
* Giao thức truyền thông giữa máy tính Server, Client và bộ điều khiển robot
  + 1. Phạm vi nghiên cứu

Luận văn tập trung vào việc xây dựng một kiến trúc truyền thông ở tầng giám sát của hệ thống robot trong công nghiệp, tập trung chủ yếu vào phần mềm. Trong quá trình thực hiện Luận văn, cần tìm hiểu nhiều lĩnh vực khác nhau nhằm đáp ứng được yêu cầu Luận văn. Phạm vi nghiên cứu của luận văn bao gồm nhiều mảng kiến thức:

* Tìm hiểu chuẩn truyền thông OPC UA bao gồm: ứng dụng, tính năng nổi bật,…
* Ứng dụng OPC UA để xây dựng kiến trúc truyền thông ở lớp giám sát cho hệ thống robot trong công nghiệp
* Tìm hiểu lý thuyết về kỹ thuật robot: phương trình động học (thuận, nghịch), hoạch định quỹ đạo robot
* Tìm hiểu về SQL Database
* Xây dựng giải thuật điều khiển robot
* Tìm hiểu về tính năng bảo mật cho chuẩn truyền thông OPC UA
  1. Nhiệm vụ và nội dung của luận văn

Để thực hiện luận văn, cần hoàn thành các nhiệm vụ sau:

* Xây dựng một kiến trúc truyền thông ở tầng giám sát cho hệ thống robot trong công nghiệp sử dụng chuẩn truyền thông OPC UA
* Xây dựng và phát triển các tính năng cho phần mềm OPC UA Server
* Xây dựng và phát triển các tính năng cho phần mềm OPC UA Client
* Tích hợp OPC UA vào bộ điều khiển robot
* Cải thiện tính bảo mật truyền thông cho hệ thống robot.
  1. Cấu trúc của luận văn

Cấu trúc luận văn bao gồm:

Chương 1: Giới thiệu đề tài – trình bày khái quát về mục đích, đối tượng và phạm vi nghiên cứu của luận văn.

Chương 2: Giới thiệu tổng quan về hệ thống robot công nghiệp và chuẩn truyền thông OPC UA - trình bày tổng quan về hệ thống robot công nghiệp, các chuẩn truyền thông được sử dụng trong các robot hiện nay và lý thuyết về chuẩn truyền thông OPC UA

Chương 3: Tổng quan về cấu trúc hệ thống - trình bày cấu trúc tổng quan của hệ thống được thiết kế trong luận văn.

Chương 4: Thiết kế phần mềm - trình bày các phần mềm được xây dựng bao gồm OPC UA Server, OPC UA Client trên máy tính.

Chương 5: Tích hợp OPC UA vào bộ điều khiển robot – trình bày chi tiết về cách đọc/ghi dữ liệu của robot ứng dụng OPC UA và giao thức truyền thông giữa máy tính và bộ điều khiển.

Chương 6: Kết quả và hướng phát triển - trình bày kết quả thu được và định hướng luận văn

# Chương 2. Tổng quan về hệ thống robot công nghiệp và chuẩn truyền thông OPC UA

## 2.1. Khái niệm về robot công nghiệp và sự phát triển của robot công nghiệp trong sản xuất

### 2.1.1. Khái niệm về robot công nghiệp

Robot công nghiệp có tên tiếng anh là Industrial Robotics là một loại máy được dùng trong sản xuất công nghiệp. Chúng được lập trình sẵn để có thể tự động di chuyển, làm việc bằng 2 hoặc nhiều trục liên kết với nhau. Hiện nay Robot ngày càng được con người nghiên cứu và cải tiến để có thể nâng cao năng suất làm việc và thay thế con người làm các công việc vất vả hoặc vào những nơi nguy hiểm. Dựa theo tiêu chuẩn ta có các định nghĩa Robot công nghiệp như:

* Dựa trên tiêu chuẩn RIA (Mỹ): Industrial Robotics là một hoặc 2 cánh tay vạn năng, chúng được lập trình để có thể thay thế con người làm những công việc khác nhau.
* Tiêu chuẩn AFNOR (Pháp): Người Pháp lại định nghĩa Robot công nghiệp là một cơ cấu chuyển động được lập trình sẵn thực hiện các công việc có tính tuần hoàn, chúng được làm việc dựa trên hệ trục tọa độ.

### 2.1.2. Sự phát triển của robot công nghiệp trong sản xuất

5 vai trò quan trọng của robot công nghiệp mang lại cho chúng ta:

* Nâng cao năng suất hoạt động
* Tiết kiệm không gian làm việc
* Tiết kiệm các chi phí không đáng có
* Hạn chế tối đa sai sót
* Đảm bảo an toàn sản xuất

Ứng dụng của robot công nghiệp trong các lĩnh vực:

* Robot sử dụng cho đóng gói, phân loại sản phẩm, thường sẽ dùng Robot song song.
* Robot được dùng để phun sơn trong xưởng sản xuất vỏ ô tô, máy bay….
* Robot công nghiệp dùng trong công nghệ ép nhựa.
* Được ứng dụng cho việc xếp dỡ, gắp hàng hóa.
* Robot hàn xì tự động.
* Robot được ứng dụng trong nghành công nghiệp đúc để rót kim loại nóng chảy.

Các loại robot công nghiệp phổ biến

Một số robot ứng với các công dụng chuyên dùng:

* Robot hàn: Sử dụng trong hàn xì
* Robot cắt: Dùng để cắt các loại vật liệu cứng, số lượng lớn.
* Robot gắp sản phẩm
* Robot sơn
* Robot Pallet: được dùng nhiều trong đóng các kiện hàng lên Pallet
* Robot công nghiệp hợp tác (Co-bot): Là dòng máy dùng để hợp tác chung với con người trong quá trình sản xuất hàng hóa.

Các hãng sản xuất robot hàng đầu trên thế giới

* Các hãng sản xuất robot tại Nhật Bản: Fanuc, Yaskawa Electric, Epson, Kawasaki, Panasonic.
* Các hãng sản xuất robot từ thị trường khác: ABB (Thụy Sĩ), KUKA (Đức - Nay thuộc tập đoàn Midea Trung Quốc), Adept Technology Inc (Mỹ – Nhật).

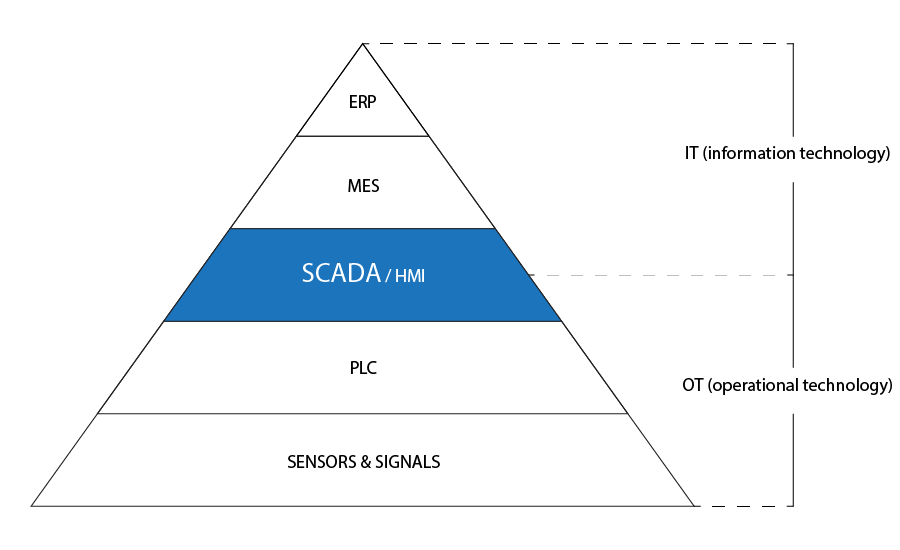


Hình 2.0.1. Robot công nghiệp trong dây chuyền lắp ráp ô tô

## 2.2. Khái niệm về hệ thống SCADA và cấu trúc của một hệ thống SCADA

### 2.2.1. Khái niệm về hệ thống SCADA

SCADA (Supervisory Control and Data Acquisition) là một phần mềm hệ thống được sử dụng để giám sát, điều khiển và thu thập thông tin dữ liệu của hệ thống phần cứng. Nó được sử dụng rộng rãi trong các nhà máy để giám sát và điều khiển các dây chuyền và máy móc sản xuất. Một cách tốt để hiểu hệ thống SCADA là gì và nó có thể được sử dụng ở đâu thì ta có thể hiểu điều này đơn giản hơn khi tìm hiểu về kim tự tháp tự động hóa.

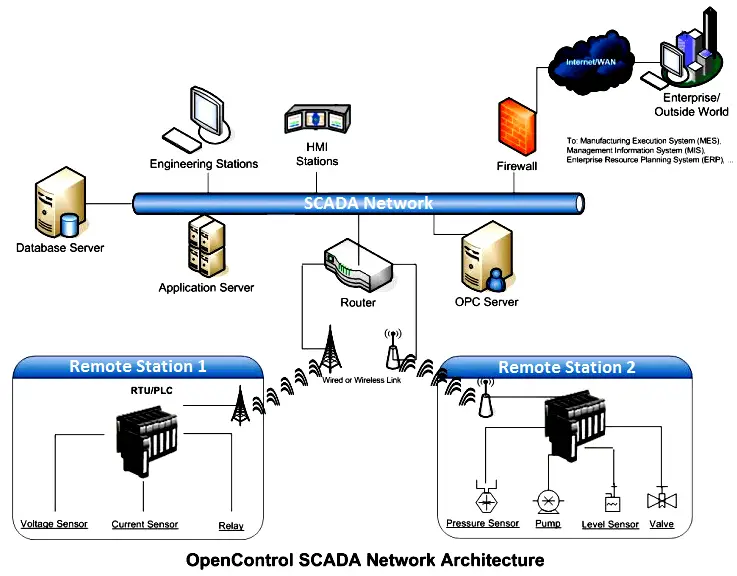


Hình 2.0.2. Kim tự tháp tự động hóa

Hệ thống SCADA được đặt ở cấp độ theo dõi và giám sát trong kim tự tháp tự động hóa. Kim tự tháp tự động hóa là một khái niệm được xuất bản trong ISA-95 và IEC 62264-3, nhằm cố gắng mô tả cách các hệ thống khác nhau hoạt động tương hỗ cùng nhau. Ở đỉnh kim tự tháp, bạn sẽ có tất cả các thông tin dữ liệu về hệ thống để xử lý về mặt kinh doanh, lập kế hoạch và hậu cần. Và ở dưới cùng, các thiết bị cơ cấu hành hoạt động. Hệ thống SCADA được đặt ngay giữa kim tự tháp tự động hóa, là nơi IT (information technology – công nghệ thông tin) gặp OT (operational technology – công nghệ vận hành).

Bên dưới hệ thống SCADA là tất cả các thiết bị hoạt động như PLC, bộ điều khiển robot, cảm biến, v.v. Công việc của SCADA thực sự là điều khiển và giám sát tất cả các thiết bị này. Nhưng đồng thời cũng gửi và nhận thông tin từ hệ thống MES hoặc ERP phía trên.

2.2.2. Cấu trúc của một hệ thống SCADA



Hình 2.0.3. Cấu trúc tổng quát của một hệ thống SCADA

Một hệ thống SCADA bao gồm các thành phần chính như sau:

Thiết bị từ xa (RTU-Remote Terminal Unit/PLC): là các thiết bị như RTU hay PLC thực hiện quá trình điều khiển trực tiếp các cơ cấu chấp hành, đồng thời dữ liệu thu thập được từ các cơ cấu chấp hành này sẽ được RTU/PLC truyền về máy chủ hoặc RTU/PLC nhận các lệnh điều khiển từ máy chủ để thực thi theo hệ thống mạng truyền thông.

Máy chủ trung tâm (MTU-Master Terminal Unit): là một máy chủ lưu trữ trung tâm của hệ thống, những thiết bị khách như máy trạm vận hành kết nối với máy chủ để sử dụng giao diện và cơ sở dữ liệu của hệ thống SCADA.

Máy trạm vận hành (Operator Workstation): là các máy tính để bàn, laptop với các phần mềm có giao diện người dùng (HMI-Human Machine Interface) được kết nối với máy chủ trung tâm của hệ thống. Các máy này có khả năng giám sát các thông số, gửi các lệnh điều khiển (với số lượng hạn chế) tới RTU để điều khiển các cơ cấu chấp hành thông qua máy chủ.

Hệ thống mạng truyền thông: Các thành phần trong hệ thống SCADA có kết nối với nhau qua các phương thức truyền thông có dây (dây cáp, …) hoặc không dây (sóng vệ tinh, …) hay kết hợp cả 2 phương thức trên, với các chuẩn truyền thông khác nhau**.**

## 2.3. Các chuẩn truyền thông đang được sử dụng ở tầng SCADA trong các hệ thống robot công nghiệp

### 2.3.1. Hãng ABB

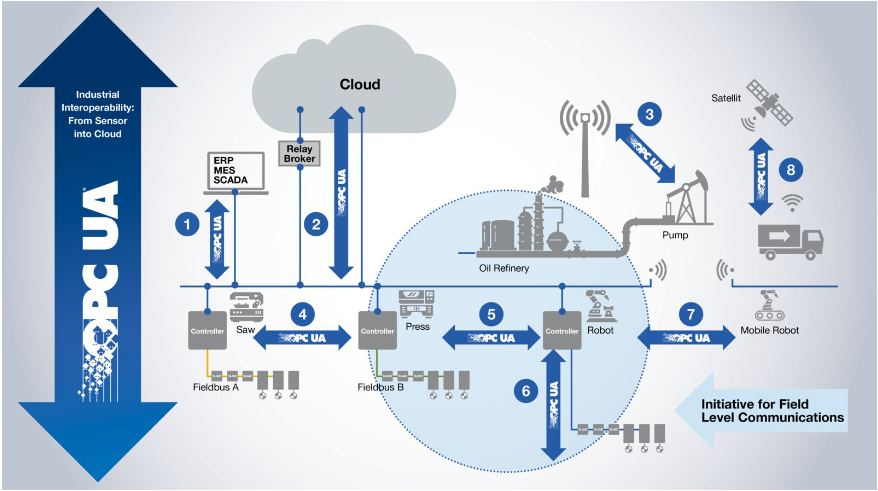
### 2.3.2. Hãng Yaskawa

### 2.3.3. Hãng Mitsubishi

## 2.4. Tổng quan về chuẩn truyền thông OPC UA

### 2.4.1. Khái niệm về OPC UA

Giao thức OPC UA, viết tắt của Open Platform Communications Unifed Architecture, được phát hành năm 2008, là một giao thức truyền thông đa nền tảng cho tự động hóa công nghiệp. OPC UA có tầm ứng dụng rộng rãi trong ngành tự động hóa công nghiệp, trao đổi thông tin dữ liệu từ các cơ cấu chấp hành như cảm biến đến các ứng dụng đám mây phía trên.



Hình 2.0.4. Các ứng dụng của OPC UA trong công nghiệp

### 2.4.2. Các tính năng nổi bật của OPC UA

- Kết hợp tất cả các tính năng rời rạc của COM OPC Classic: Data Access, Historical Data Access, Alarm and Events, …

- Đa nền tảng: có thể sử dụng ở các phần cứng và phần mềm khác nhau

- Bảo mật thông tin cao: Session Encryption, Authentication, User Control,…

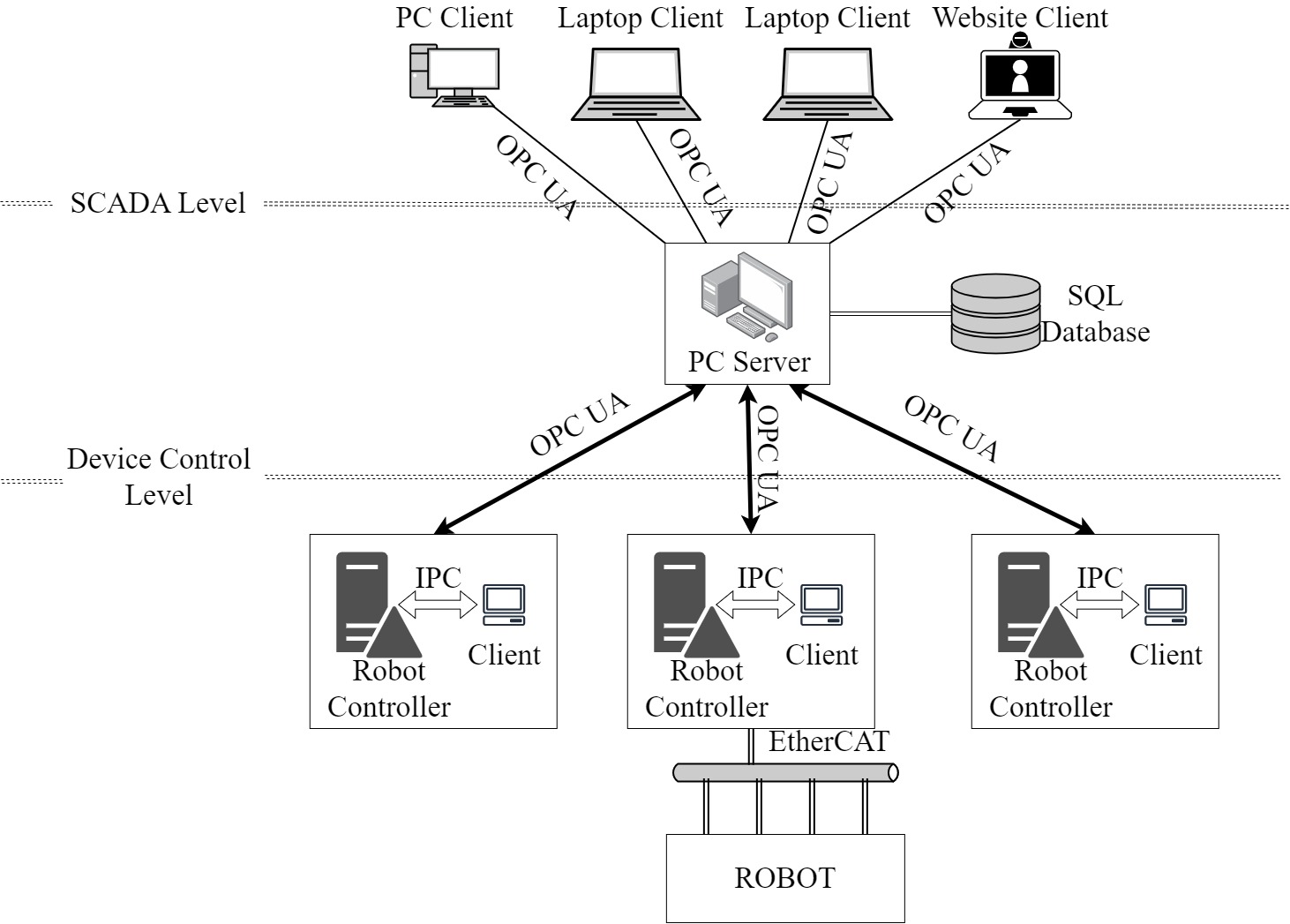
- Mô hình thông tin (Information Modeling): người dùng tự định nghĩa các cấu trúc thông tin phức tạp

|  |  |  |  |
| --- | --- | --- | --- |
|  | OPC UA | Modbus TCP | Profinet |
| Kiểu dữ liệu của biến: TagName / Register | TagName | Register | TagName |
| Định dạng của dữ liệu: Hướng đối tượng | Có | Không | Không |
| Giám sát dữ liệu thời gian thực | Có | Có | Có |
| Lưu trữ các cảnh báo, sự kiện và lịch sử | Có | Không | Có |
| Bảo mật | Có | Không | Có (hạn chế) |

Bảng so sánh tính năng nổi bật của OPC UA so với Modbus/Profinet

# Chương 3. Tổng quan về cấu trúc hệ thống

Cấu trúc của hệ thống được trình bày như trong sơ đồ dưới đây:



Hình 3.0.1. Cấu trúc của hệ thống SCADA cho hệ robot sử dụng OPC UA

Hệ thống được thiết kế bao gồm:

* Phần cứng có sẵn:
* Hệ thống robot: bao gồm có robot DELTA và SCARA được điều khiển bằng các bộ EtherCat Master Robot Controller. Các bộ điều khiển này sử dụng chuẩn EtherCat kết nối dây Ethernet trực tiếp xuống các Driver lái động cơ robot.
* Các máy tính PC, laptop: dùng để chạy các phần mềm OPC UA Server và các OPC UA Client. Chuẩn truyền thông OPC UA hỗ trợ kết nối có dây (kết nối qua dây Ethernet) giữa Server và các Client hoặc kết nối không dây khi Server và Client kết nối trong cùng 1 lớp mạng
* Phần mềm:
* Phần mềm OPC UA Server: chứa các node dữ liệu của các robot trong hệ thống và có 1 SQL Database lưu trữ cơ sở dữ liệu cho hệ thống SCADA.
* Phần mềm OPC UA Client: có giao diện người dùng quan sát các thông số, trạng thái, mô phỏng và gửi các lệnh điều khiển xuống robot.
* Phần mềm tích hợp OPC UA vào bộ điều khiển robot: xây dựng một API để đọc/ghi dữ liệu của bộ điều khiển và xây dựng bộ điều khiển robot thành một OPC UA Client.

Chương 4. Thiết kế các phần mềm cho hệ thống SCADA trên máy tính

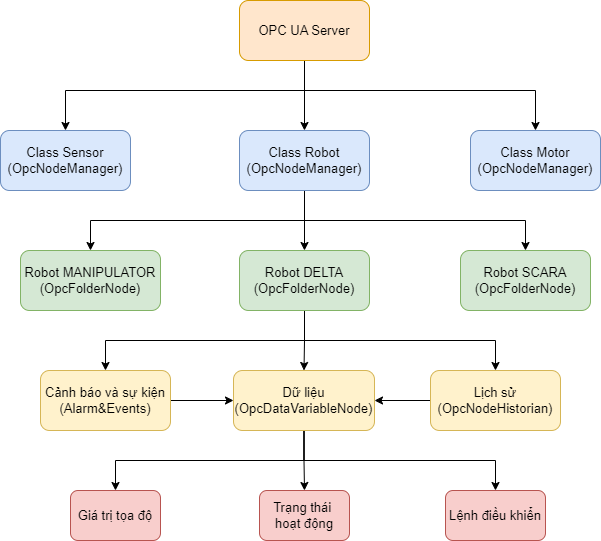
## 4.1. Thiết kế OPC UA Server

### 4.1.1. Giới thiệu về thư viện OPC UA Client & Server SDK hỗ trợ thiết kế OPC UA Server trên phần mềm Visual Studio 2022

OPC UA Client & Server SDK là một thư viện nằm trong Nuget Packages của Visual Studio 2022, hỗ trợ các tính năng của chuẩn truyền thông OPC UA. Đây là một công cụ dễ dàng phát triển cho người dùng sử dụng .NET Framework và .NET Standard, cung cấp .NET API đơn giản và quen thuộc. Thư viện này cung cấp những cấu trúc dữ liệu và giải thuật tối ưu cho một khối lượng dữ liệu lớn phải lưu trữ và xử lý của server, các tính năng bảo mật cao như mã hóa thông tin và xác thực thông tin. Ngoài ra, thư viện này còn cung cấp các chương trình mẫu được hỗ trợ cho người dùng rất hữu ích.

### 4.1.2. Cấu trúc OPC UA Server đã thiết kế

Em đã sử dụng thư viện được trình bày ở trên để tạo server. Trong server khởi tạo các node dữ liệu để các client truy xuất vào đọc và ghi. Server được lập trình bằng ngôn ngữ C#. Cách thức thực hiện server: tạo một class robot là một node quản lý (OpcNodeManager) để quản lý chung nhiều robot khác nhau, bên trong chứa các node thư mục (OpcFolderNode) được xem như các thư mục để chứa dữ liệu của từng robot. Trong mỗi node thư mục, tạo các node dữ liệu (OpcDataVariableNode) để lưu trữ dữ liệu với nhiều kiểu khác nhau của robot. Ngoài ra, còn có các chức năng khác như cảnh báo và sự kiện (Alarm&Events) hoặc lịch sử (History) để kết hợp với các node dữ liệu, từ đó khai thác các tính năng ưu việt của OPC UA. Ngoài ra, em cũng tạo thêm 2 class nữa là class Motor và class Sensor có thể phục vụ thêm cho các mục đích sau này của hệ thống. Hình dưới thể hiện mô hình cây được nhóm nghiên cứu xây dựng:



Hình 4.1. Cấu trúc hình cây của OPC UA Server

Bên cạnh đó, khi muốn truy cập vào một node dữ liệu của một robot nào đó để đọc/ghi dữ liệu, OPC UA Client cần phải biết địa chỉ chính xác của node đó. Địa chỉ của một node dữ liệu mà OPC UA Client dùng để truy cập có định như sau:

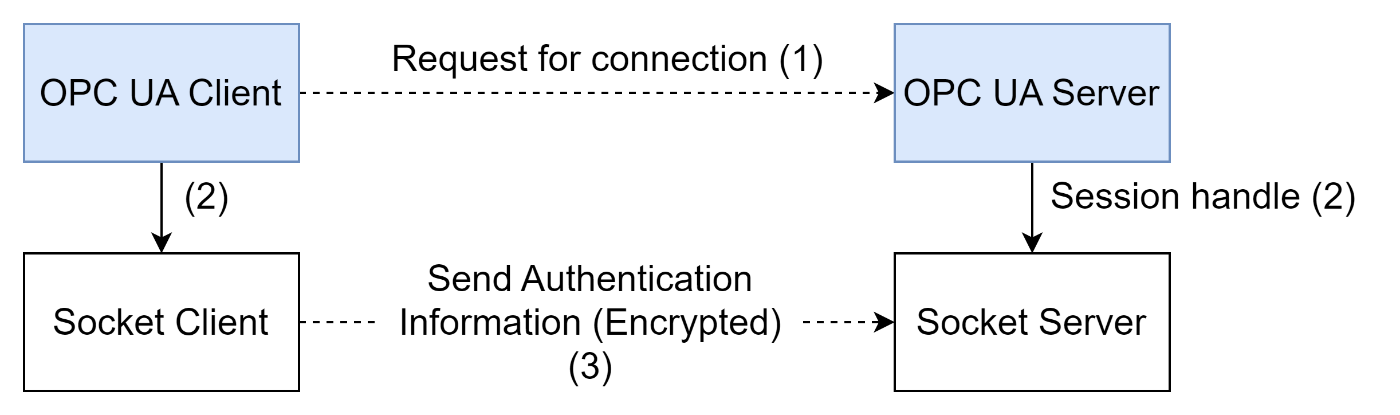
|  |
| --- |
| ns=OrderofClass;s=NameofRobot/NameofDataNode |

Trong đó:

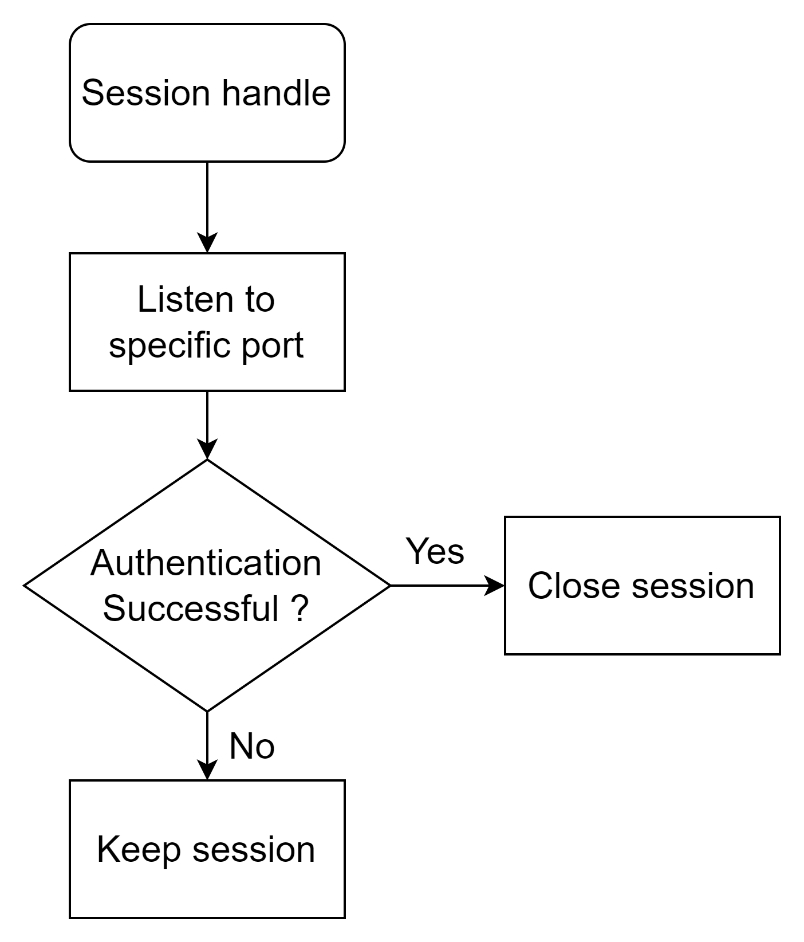
* OrderofClass: Thứ tự của class. Trong trường hợp này là class Robot thì ns=4. Còn nếu như class Motor thì ns=2 và class Sensor thì ns=3.
* NameofRobot: Tên của robot chứa node dữ liệu muốn truy cập. Ví dụ như Robot1, Robot2, Robot Delta, …
* NameofDataNode: Tên của node dữ liệu đó.

Đặc biệt, ta có thể tự đặt tên cho NameofRobot và NameofDataNode một cách tùy ý, không cần phải theo bất kỳ cấu trúc hay định dạng nào. Còn OrderofClass sẽ được đặt theo thứ tự, vì khi tạo các class em tạo theo thứ tự: class Motor 🡪 class Sensor 🡪 class Robot.

### 4.1.3. Tính năng bảo mật của OPC UA Server

OPC UA Server có chức năng xác thực mỗi khi có OPC UA Client kết nối tới Server. 

Hình 4.0.2. Cơ chế xác thực



Hình 4.0.3. Cơ chế xác thực 2

### 4.1.4. SQL Database của OPC UA Server

## 4.2. Thiết kế OPC UA Client

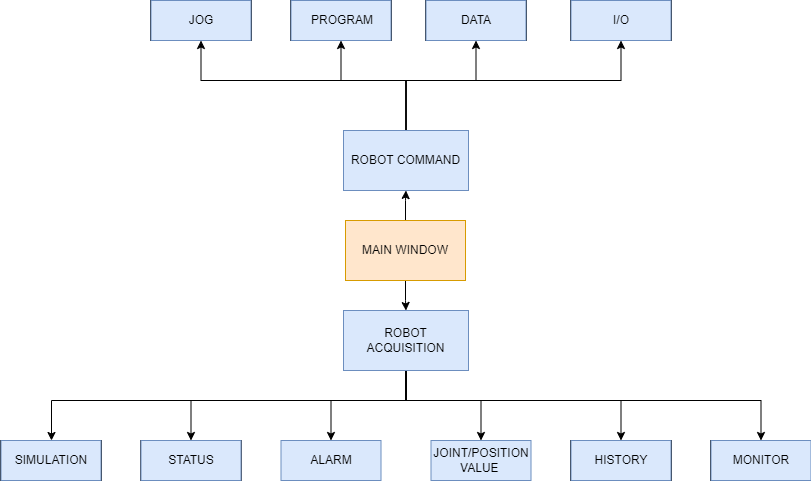
* + 1. Giới thiệu về thư viện mở open62541 hỗ trợ thiết kế OPC UA Client

Với mục đích tìm kiếm một thư viện để xây dựng một OPC UA Client với đầy đủ các chức năng giám sát, mô phỏng, điều khiển hệ thống robot, ngoài ra thì thư viện này còn phải tương thích để tích hợp vào bộ điều khiển robot được lập trình bằng ngôn ngữ C++, em đã tìm kiếm và sử dụng thư viện mã nguồn mở open62541 [], là một trong những thư viện mã nguồn mở phổ biến nhất của OPC UA. Sau khi xem xét tất cả các thư viện mở khả dụng của OPC UA, em nhận thấy rằng open62541 là sự lựa chọn đáng tin cậy và phù hợp nhất với các ứng dụng của hệ thống em đang thực hiện. Thư viện open62541 cũng là một thư viện mở được sử dụng nhiều nhất trên GitHub, với nhóm tác giả đông đảo và cộng đồng hỗ trợ tích cực. Ngoài ra, nhóm tác giả ở bài báo [] không không tìm thấy các lỗ hổng bảo mật của thư viện này, đảm bảo sự tin cậy cho người dùng.

* + 1. Chức năng của phần mềm OPC UA Client

Về mặt chức năng, PC Client có đầy đủ các chức năng giám sát của 1 hệ thống SCADA, như thu thập dữ liệu từ robot, gửi lệnh điều khiển robot như robot controller.

Sơ đồ khối tổng quát của một PC Client:



Hình 4.0.4. Cấu trúc của OPC UA Client trên PC

Trong đó có 2 khối chức năng chính:

- Robot Acquisition (Khối giám sát robot): giám sát các thông tin dữ liệu của robot (trạng thái, cảnh báo, giá trị tọa độ, lịch sử hoạt động, đồ thị các góc khớp, mô phỏng online/offline các robot)

-Robot Command (Khối điều khiển robot): điều khiển robot

+ Jog: di chuyển chế độ bằn tay cho robot

+ Program: chạy các chương trình lập trình sẵn cho robot

+ Data: lưu các điểm đã được dạy cho robot

+ I/O: bật tắt và hiển thị các I/O của robot

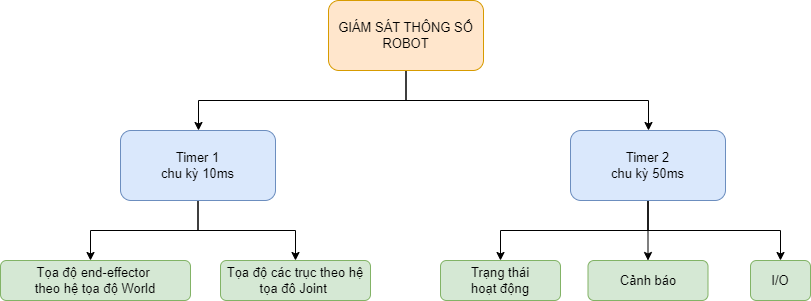
* + 1. Giao diện giám sát của các robot trong hệ thống
       1. Giao diện giám sát các thông số của robot

Một OPC UA Client có thể giám sát đồng thời các robot khác nhau, cụ thể ở đây em thực hiện giám sát hệ thống 2 robot với các thông số cần quan sát được thể hiện ở phần 4.2.2. Giao diện cụ thể sẽ được thể hiện ở dưới đây:



Hình 4.0.5. Giao diện các thông số giám sát robot

Ở giao diện này, người dùng sẽ quan sát được các trạng thái hoạt động của robot như chỉ số dụng cụ (Tool Number), tốc độ điều khiển các trục robot(Jogging Speed), hệ tọa độ của robot (Operation Coordinate System), trạng thái bật/tắt Servo (Servo Status), trạng thái hoạt động của hệ thống (System Status), chế độ điều khiển bằng tay/tự động (Mode), chế độ bảo vệ ở chế độ vận hành/chỉnh sửa/quản lý (Security Level), trạng thái mở/khóa của bộ điều khiển robot (Lock/Unlock Controller). Bên cạnh đó, người dùng có thể quan sát các cảnh báo khi robot gặp lỗi (Alarm) và các thông số vị trí hiện tại của tọa độ điểm cuối của robot hay góc xoay của các trục robot. Ngoài ra, người dùng còn có thể xem được lệnh phản hồi của bộ điều khiển robot khi đã gửi thành công lệnh điều khiển (Command Ack). Các thông số này được cập nhật bằng các Timer trên PC.

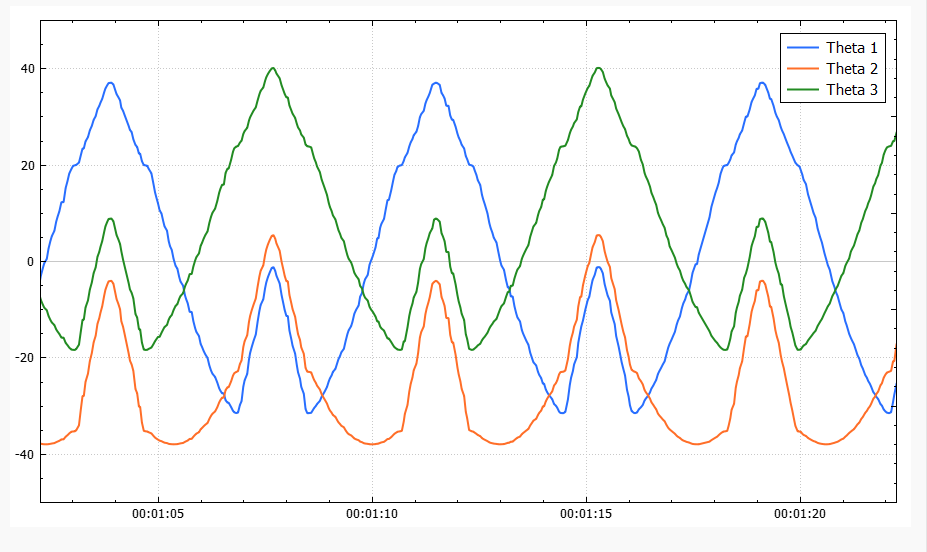


Hình 4.0.6. Sơ đồ các Timer giám sát thông số robot trên PC

Việc sử dụng 2 Timer với mục đích để dễ dàng quản lý dữ liệu và giảm thiểu việc đọc và xử lý dữ liệu của 1 Timer vì các giá trị tọa độ thay đổi liên tục nên cần sử dụng một Timer với chu kỳ ngắn (10ms) còn các thông số khác ít thay đổi liên tục hơn sẽ được cập nhật với Timer có chu kỳ dài hơn (có thể 50ms-100ms).

* + - 1. Giao diện MONITOR

Em sử dụng chức năng vẽ đồ thị của thư viện QCustomPlot hỗi trợ trên Qt. Thư viện này cung cấp các chức năng hiển thị và vẽ đồ thị cho các dữ liệu thời gian thực, phù hợp cho việc thực hiện vẽ các góc quay của các trục robot khi có sự thay đổi và thể hiện lên đồ thị góc theo thời gian (độ/s). Dữ liệu góc của các trục được cập nhật liên tục từ phần giám thông sát thông số robot và vẽ dùng để lên đồ thị.

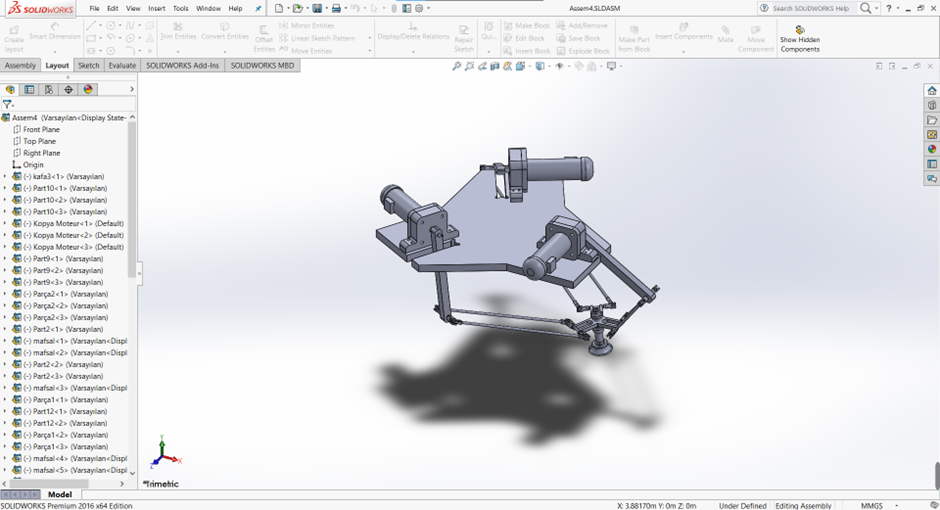


Hình 4.0.7. Giao diện MONITOR

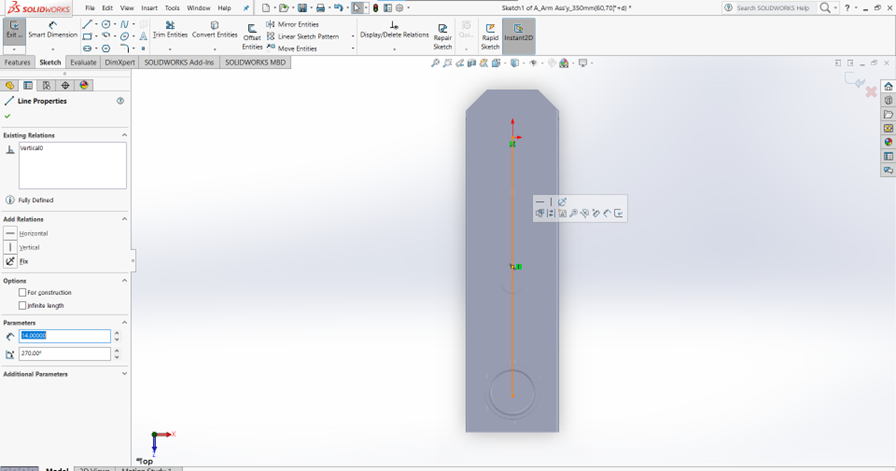
* + - 1. Giao diện SIMULATION

Hệ thống cung cấp một giao diện mô phỏng 3D cho cả robot SCARA và robot DELTA, giúp cho người dùng có thể trực quan hơn trong quan sát bên cạnh mô hình thực. Đầu tiên, em thực hiện đo đạc thông số của các Robot bằng phần mềm Solidwork như hình dưới đây:

* Robot DELTA



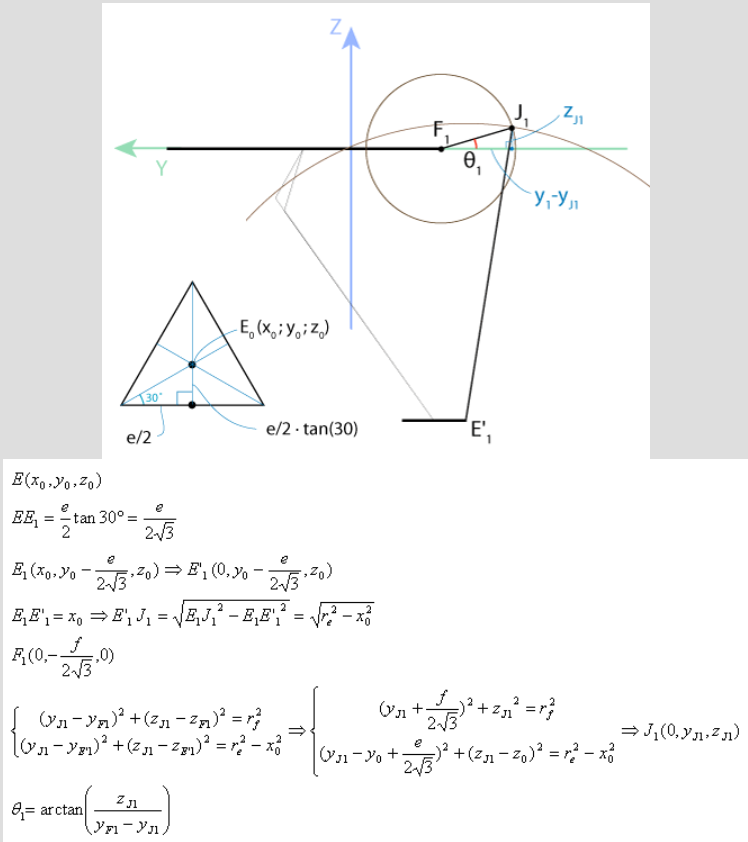
Mô hình đo đạc của robot DELTA



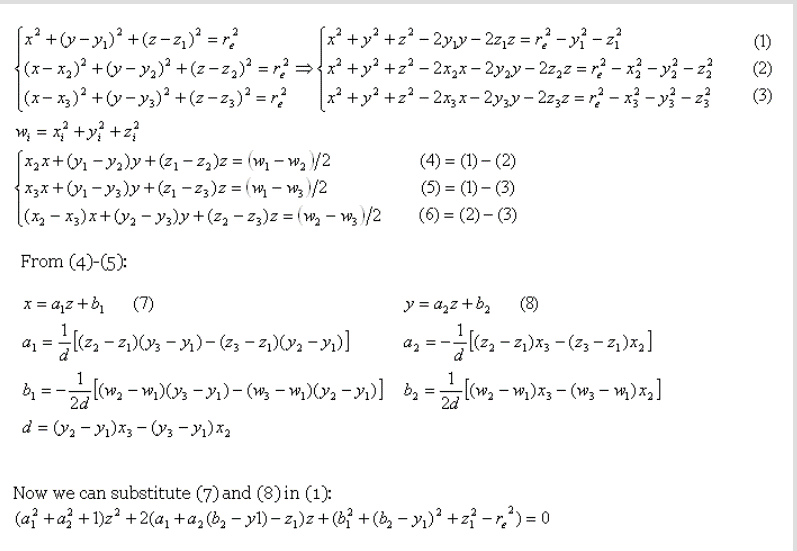
hình đo đạc các khâu của robot

Để Robot có thể thực hiện được các tác vụ, cần giải bài toán động học thuận và động học nghịch cho Robot.

* Động học thuận và nghịch cho Robot DELTA:



Lời giải động học thuận cho robot DELTA

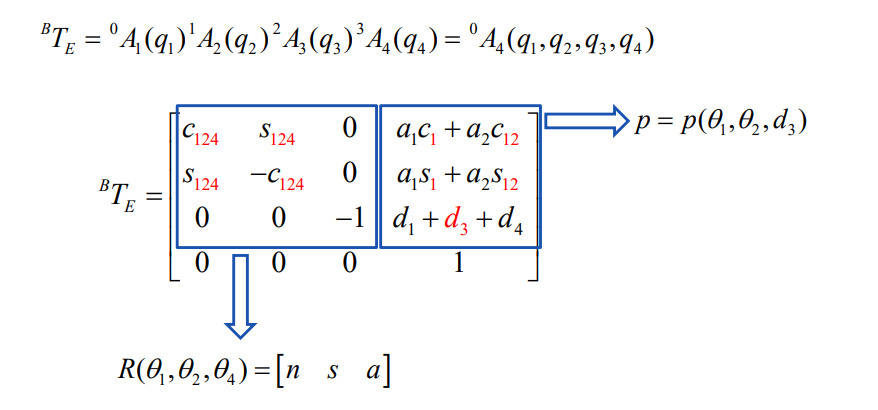


Lời giải động học nghịch cho robot DELTA

* Robot SCARA

Mô hình đo đạc của robot SCARA

Hình đo đạc các khâu của robot SCARA

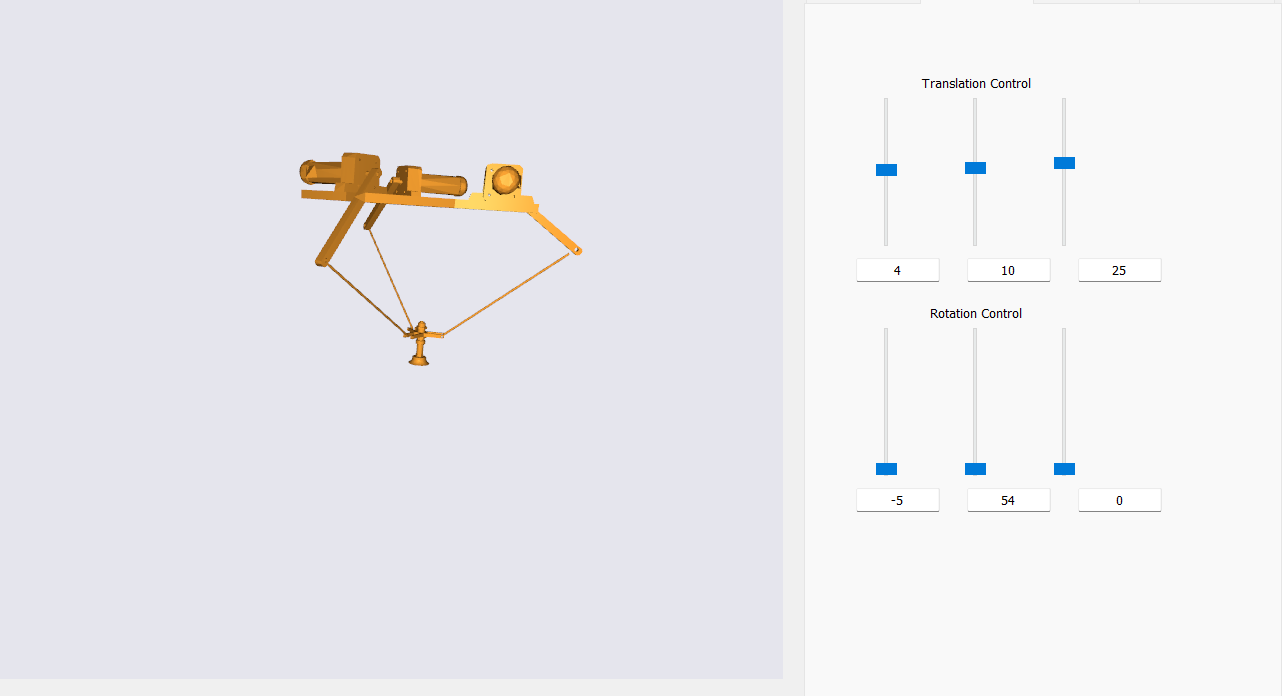


Lời giải động học thuận của robot SCARA

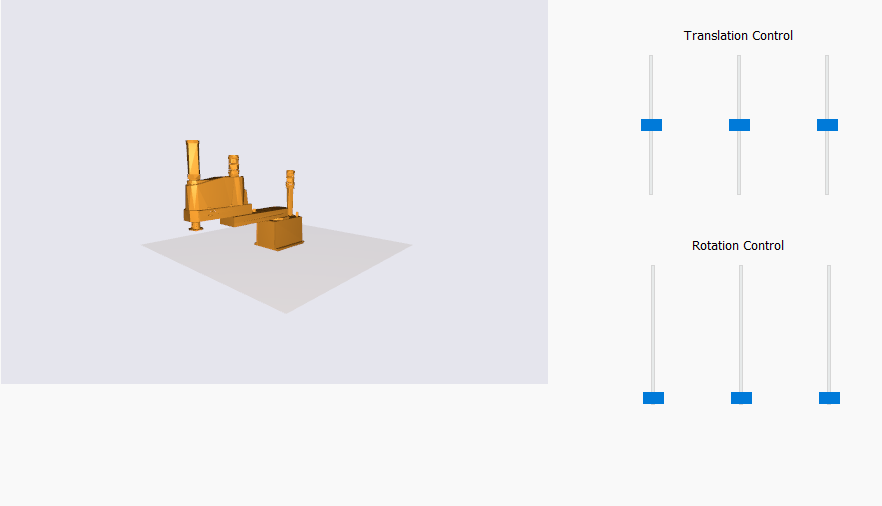


Lời giải động học nghịch của robot SCARA

Tiếp theo, em sử dụng thư viện Assimp, là một mã nguồn mở có khả năng import file 3D và sử dụng OpenGL để vẽ lại các phần tử trên giao diện phần mềm. Phần mềm chiếu người dùng thay đổi hướng nhìn, mô phỏng hoạch định quỹ đạo cho robot để xem hành vi của robot một cách trực quan, trước khi quyết định áp dụng xuống mô hình robot thật.

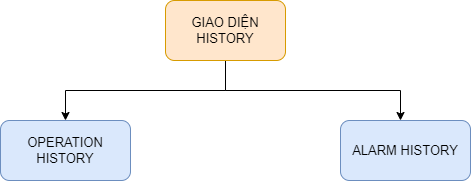


Mô hình mô phỏng robot DELTA



Mô hình mô phỏng robot SCARA

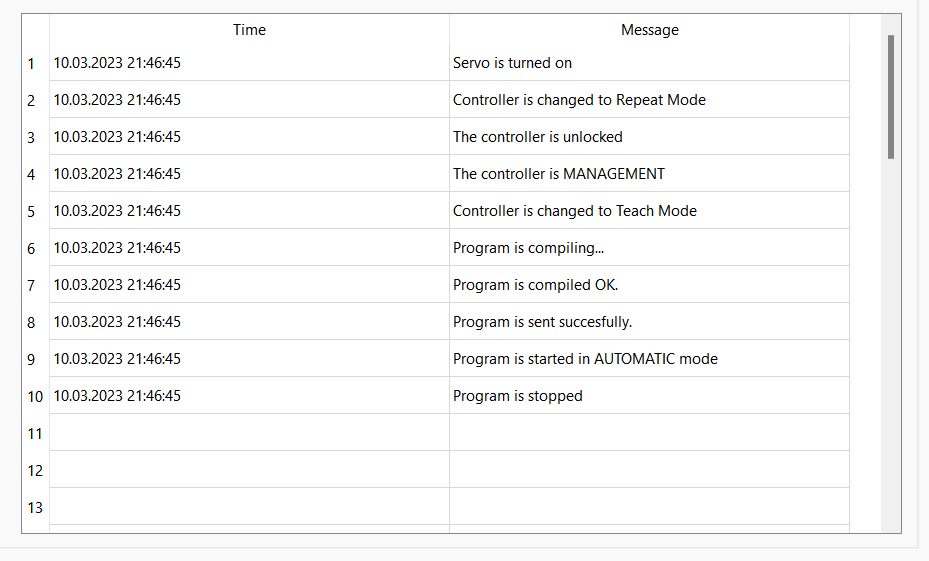
* + - 1. Giao diện HISTORY



Hình 4.0.8. Sơ đồ giao diện HISTORY

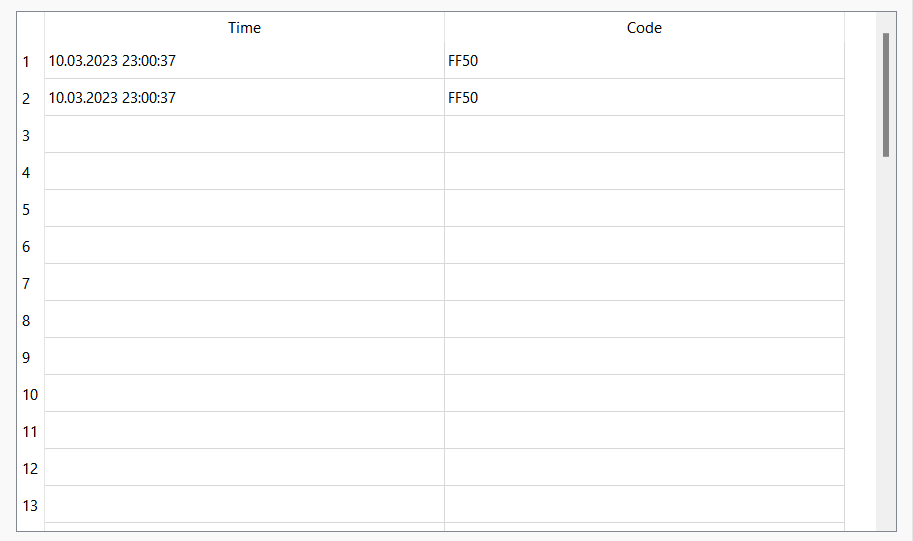
Giao diện History gồm 2 phần:

* Giao diện Operation History: hiển thị tất cả lịch sử vận hành của người dùng kèm mốc thời gian.



Hình 4.0.9. giao diện Operation History

* Giao diện Alarm History: hiển thị chi tiết lịch sử mã cảnh báo kèm mốc thời gian xảy ra cảnh báo, giúp người dùng chẩn đoán và xử lý.



Hình 4.0.10. Giao diện Alarm History

4.2.3. Khung truyền PC Client khi giao tiếp với bộ điều khiển robot

Khung truyền từ PC Client đến bộ điều khiển robot:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | Command | , | Argument\_1 | , | … | , | Argument\_n | ; | CRC | ETX(0x03) |

Trong đó:

* STX là kí tự bắt đầu
* Command là tên lệnh
* Argument\_i là đối số
* CRC là mã kiểm tra lỗi
* ETX là kí tự kết thúc

Khung dữ liệu được bắt đầu bằng ký tự bắt đầu STX, tiếp theo là tên lệnh, các đối số được ngăn cách nhau bằng ký tự “,”, kế tiếp là CRC để kiểm tra lỗi và ký tư kết thúc ETX.

Quy tắc tạo ra CRC để kiểm tra lỗi:

unsigned short MakeCrc(unsigned short\* wPacketData, uint16 usPacketSize)

{

unsigned short wCheckSum = 0;

for(int i = 0; i < usPacketSize ; i++)

{

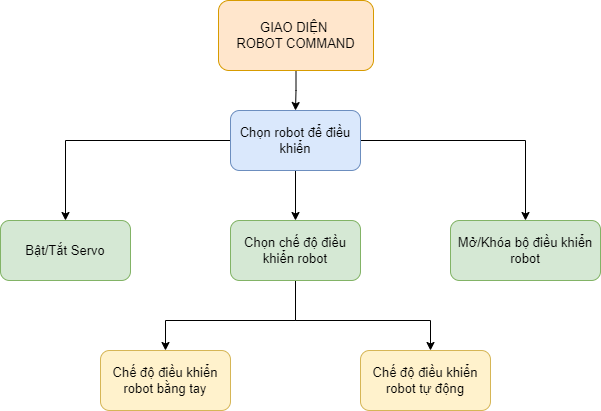
wCheckSum = wCheckSum ^ wPacketData[i];

}

return wCheckSum;

}

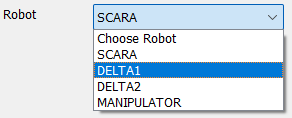
* + 1. Giao diện điều khiển robot trong hệ thống



Hình 4.0.11. Giao diện điều khiển robot tổng quát

Trước khi đến với từng giao diện điều khiển riêng biệt, giao diện điều khiển robot gồm:

* Chọn robot để điều khiển:



Hình 4.0.12. Giao diện chọn robot dể điều khiển

* Các nút điều khiển quan trọng chính như:
* Nút bật/tắt Servo:



Hình 4.0.13. Giao diện nút Servo

* Khung truyền

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | SVON | , | Argument\_1 | ; | CRC | ETX(0x03) |

Tên lệnh: SVON

Argument\_1: 0/1 (tắt/bật)



Hình 4.0.14. Minh họa lệnh bật/tắt Servo

* Chọn chế độ điều khiển bằng tay/tự động:



Hình 4.0.15. Giao diện nút chọn chế độ điều khiển

* Khung truyền

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | MODE | , | Argument\_1 | ; | CRC | ETX(0x03) |

Tên lệnh: MODE

Argument\_1: 0/1 (bằng tay/tự động)



Hình 4.0.16. Minh họa lệnh chọn chế độ điều khiển robot

* Mở/khóa bộ điều khiển robot:



Hình 4.0.17. Giao diện nút khóa bộ điều khiển

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | LOCK | , | Argument\_1 | ; | CRC | ETX(0x03) |

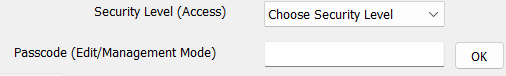
Tên lệnh: LOCK

Argument\_1: 0/1 (mở khóa/khóa)



Hình 4.0.18. Minh họa lệnh mở/khóa bộ điều khiển robot

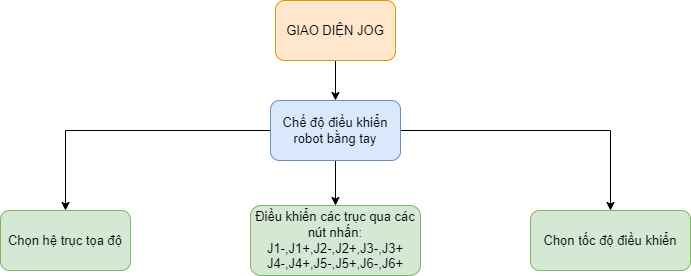
* Chọn chế độ bảo vệ: vận hành, chỉnh sửa, quản lý. Ở 2 chế độ chỉnh sửa và quản lý, phải nhập mật khẩu để có thể sử dụng chế độ này.



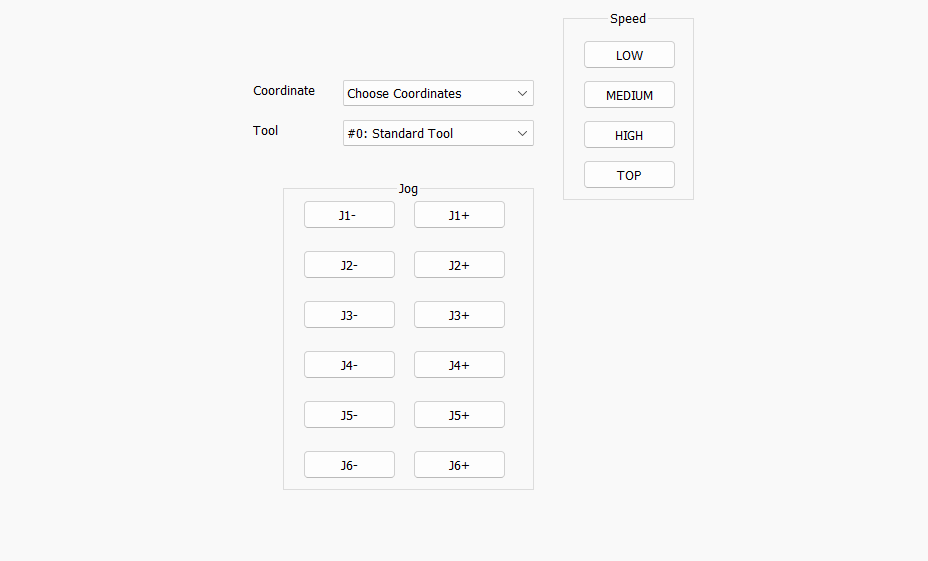
Hình 4.0.19. Giao diện chọn chế độ bảo vệ

Tiếp theo, từng giao diện cụ thể như JOG, PROGRAM, DATA, I/O sẽ có những chức năng điều khiển khác nhau.

* + - 1. Giao diện JOG



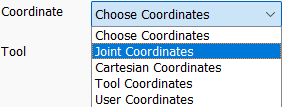
Hình 4.0.20. Sơ đồ tổ chức giao diện JOG



Hình 4.0.21. Giao diện JOG

Giao diện JOG này được sử dụng ở chế độ điều khiển robot bằng tay, bao gồm:

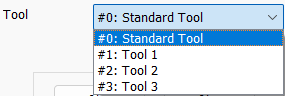
* Chọn hệ tọa độ



Hình 4.0.22. Giao diện chọn hệ tọa độ

Ở đây, ta có thể chọn 4 hệ tọa độ tương ứng: hệ tọa độ Joint, hệ tọa độ Cartesian, hệ tọa độ Tool, hệ tọa độ User.

* Chọn chỉ số dụng cụ: 0-63

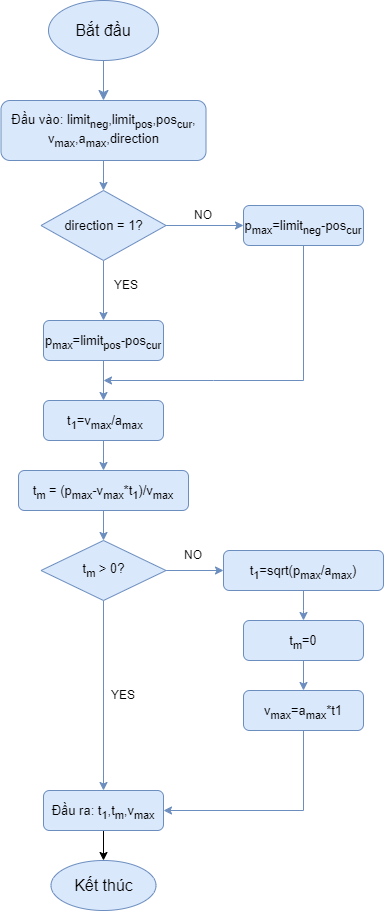


Hình 4.0.23. Giao diện chọn dụng cụ

* Nhóm nút Jog:

Thực hiện lệnh điều khiển các trục của robot theo chiều âm và chiều dương. Nhóm nút này có thể điều khiển robot có 6 trục động cơ.

Lệnh này cần thực hiện quy hoạch quỹ đạo hình thang cơ bản cho các trục của robot, tính toán thời gian di chuyển của các trục tương ứng rồi gửi kèm theo lệnh điều khiển. Sơ đồ giải thuật cho phần tính toán này với các điều kiện: vận tốc ban đầu và vận tốc kết thúc bằng 0, độ lớn của gia tốc khi tăng tốc bằng độ lớn của gia tốc khi giảm tốc nhưng ngược dấu nhau.



Hình 4.0.24. Sơ đồ giải thuật tính toán các giá trị thời gian trục robot di chuyển

Trong đó:

limitneg = - 45o: giá trị giới hạn theo chiều âm của trục

limitpos = 45o: giá trị giới hạn theo chiều dương của trục

pmax: khoảng cách trục cần phải di chuyển đến điểm giới hạn

vmax: vận tốc tối đa của trục

amax: gia tốc tối đa của trục

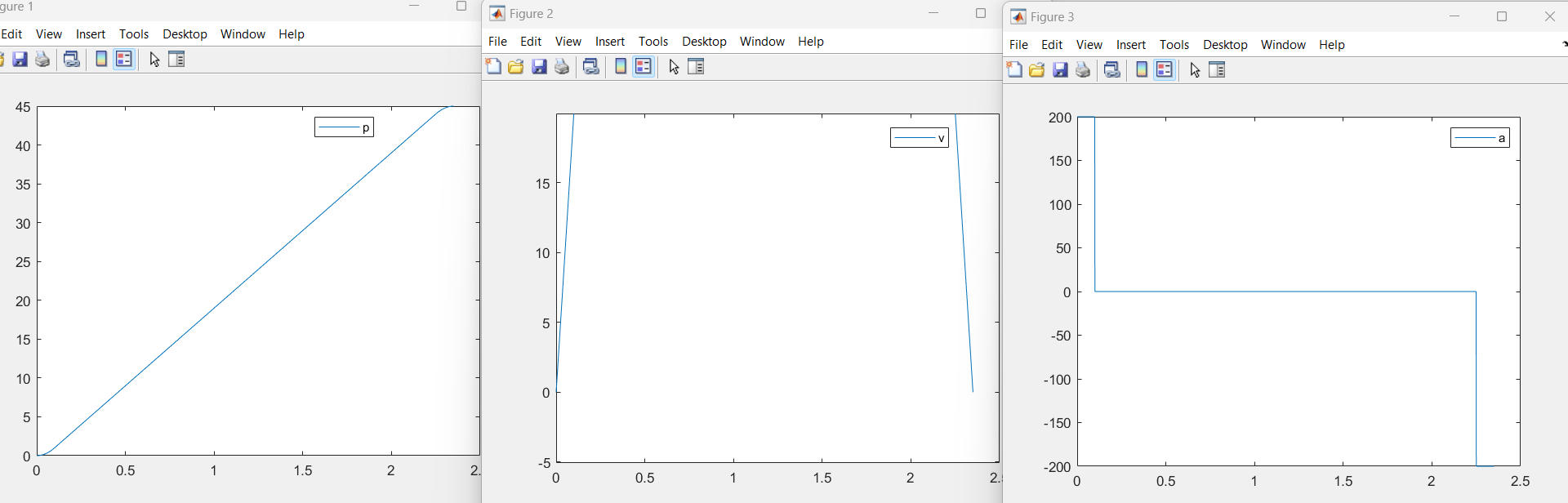
t1: thời gian di chuyển tăng tốc/giảm tốc

tm: thời gian di chuyển với vận tốc không đổi

Em thực hiện mô phỏng trên MATLAB 4 trường hợp cơ bản khi điều khiển các trục của robot:

- Trường hợp 1: p>0, v>0, a>0. Chọn pmax = 45, vmax = 20, amax = 200 (quỹ đạo vận tốc hình thang)

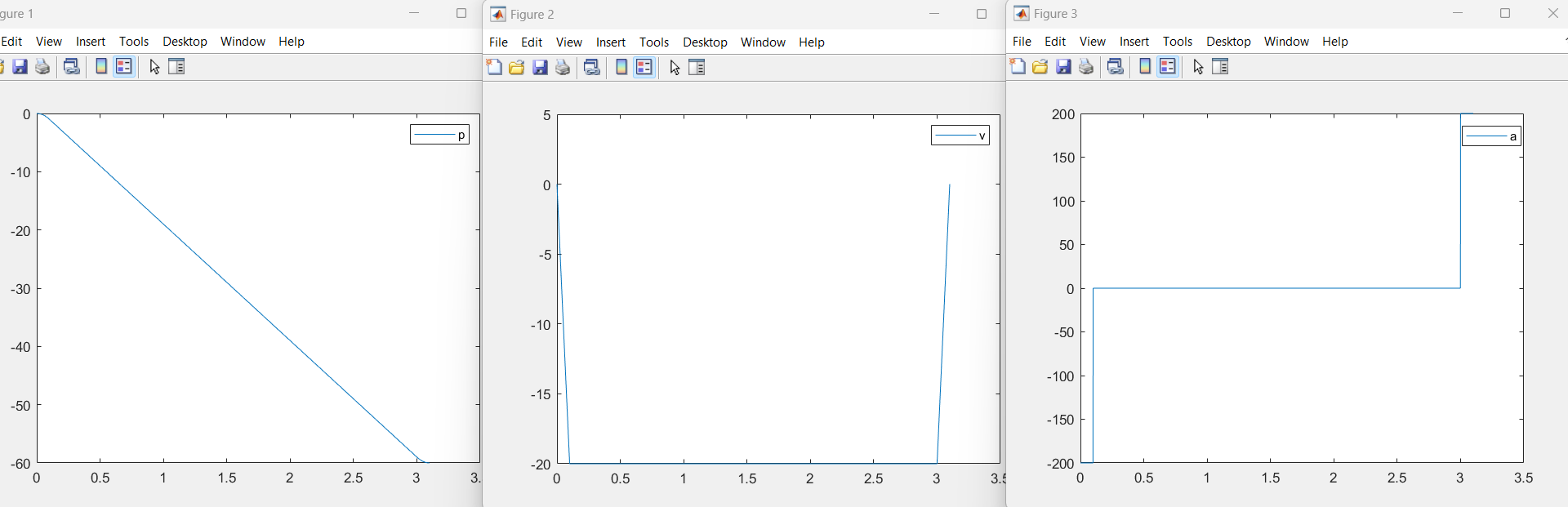
t1 = 0.1, tm = 2.15 (s)



Hình 4.0.25. Mô phỏng MATLAB các giá trị p, v, a theo thời gian trường hợp 1

- Trường hợp 2: p<0, v<0, a<0. Chọn pmax = -60, vmax = -20, amax = -200 (quỹ đạo vận tốc hình thang)

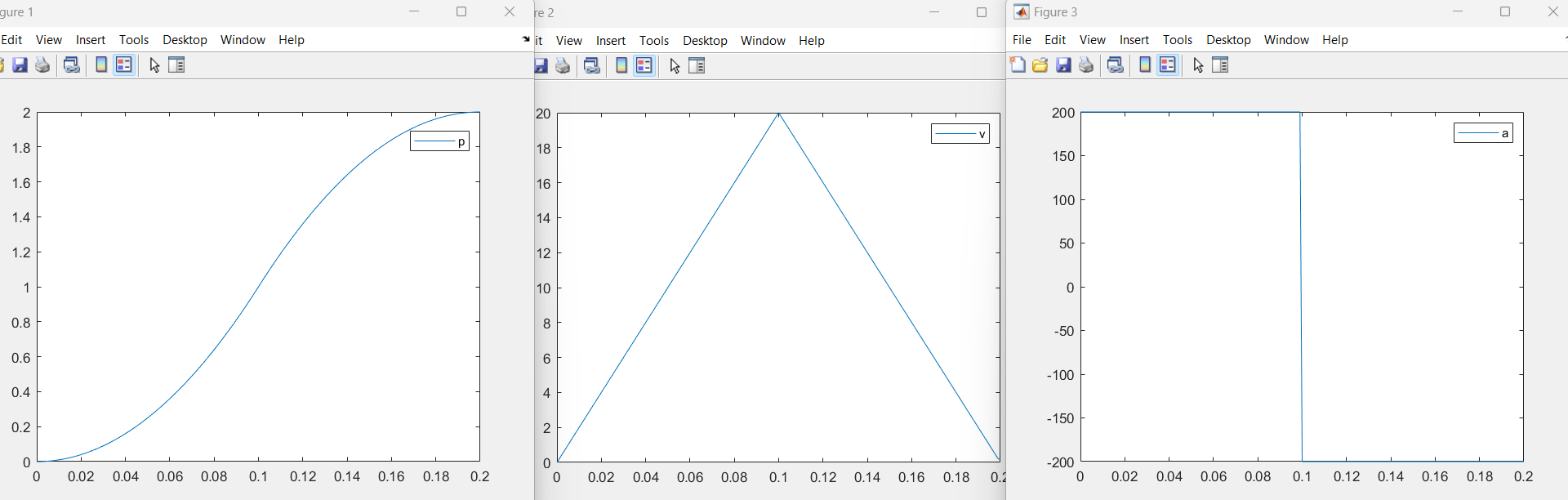
t1 = 0.1, tm = 2.9 (s)



Hình 4.0.26. Mô phỏng MATLAB các giá trị p, v, a theo thời gian trường hợp 2

+ Trường hợp 3: p>0, v>0, a>0. Chọn pmax = 2, vmax = 20, amax = 200 (quỹ đạo vận tốc hình tam giác)

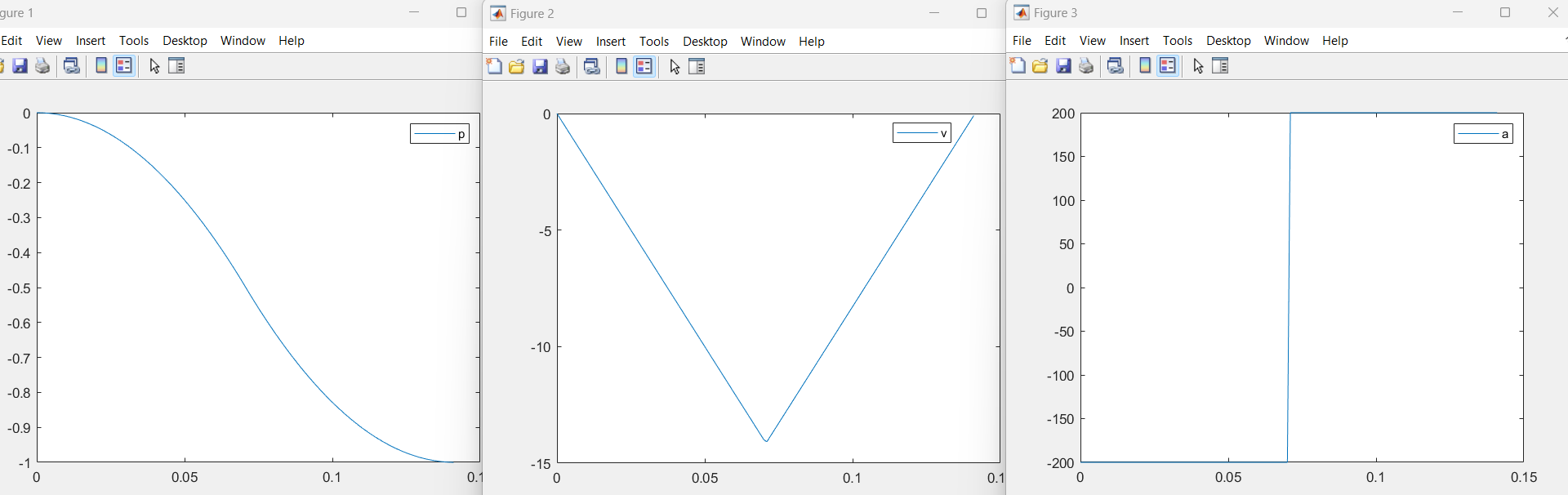
t1 = 0.1, tm = 0 (s)



Hình 4.0.27. Mô phỏng MATLAB các giá trị p, v, a theo thời gian trường hợp 3

+ Trường hợp 4: p <0, v<0, a<0: Chọn pmax = -1, vmax = -20, amax = -200 (quỹ đạo vận tốc hình tam giác)

t1 = 0.0707, tm = 0 (s)



Hình 4.0.28. Mô phỏng MATLAB các giá trị p, v, a theo thời gian trường hợp 4

Khi nhấn nút giữ một nút trong nhóm nút Jog, lệnh điều khiển trục động cơ sẽ được gửi đi như sau:

* Khung truyền:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | JOGJ | | | , | Argument\_1 | | | , | Argument\_2 | | | , | Argument\_3 | | | , | Argument\_4 | | , |
| Argument\_5 | | , | Argument\_6 | | | , | Argument\_7 | | | , | Argument\_8 | | | ; | CRC | | | ETX(0x03) | |

Tên lệnh: JOGJ

Argument\_1: hệ tọa độ (0/1/2/3 – hệ tọa độ Joint/hệ tọa độ Cartesian/hệ tọa độ Tool/hệ tọa độ User); Argument\_2: số thứ tự trục (0/1/2/3/4/5 – trục 1/trục 2/trục 3/trục 4/trục 5/trục 6); Argument\_3: chiều quay của trục (-1/+1 – chiều âm/chiều dương); Argument\_4: thời gian t1 – thời gian tăng tốc cũng là thời gian giảm tốc; Argument\_5: thời gian tm - thời gian đi với tốc độ không đổi; Argument\_6: vmax - vận tốc tối đa đạt được; Argument\_7: acc - gia tốc khi tăng tốc; Argument\_8: dec - gia tốc khi giảm tốc. Cả 3 đối số Argument\_6, Argument\_7, Argument\_8 đều được xét dấu âm/dương tương ứng với chiều quay âm/dương.



Hình 4.0.29. Minh họa lệnh điều khiển trục của robot kèm theo thời gian di chuyển của trục đó

Khi thả tay ra khỏi nút vừa được nhấn, lệnh dừng trục động cơ đang quay không di chuyển nữa sẽ được gửi đi:

* Khung truyền:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | STOP | , | Argument\_1 | ; | CRC | ETX(0x03) |

Tên lệnh: STOP

Argument\_1: số thứ tự trục (0/1/2/3/4/5 – trục 1/trục 2/trục 3/trục 4/trục 5/trục 6)



Hình 4.0.30. Minh họa lệnh dừng trục đang di chuyển của robot

* Nhóm nút Speed:

Nhóm nút này dùng để thay đổi tốc độ điều khiển các trục của robot.

Khung truyền:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | SPED | , | Argument\_1 | ; | CRC | ETX(0x03) |

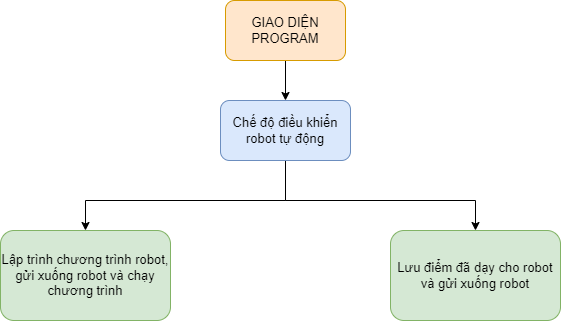
Tên lệnh: SPED

Argumnet\_1: tốc độ điều khiển (0/1/2/3 – thấp/trung bình/cao/rất cao)



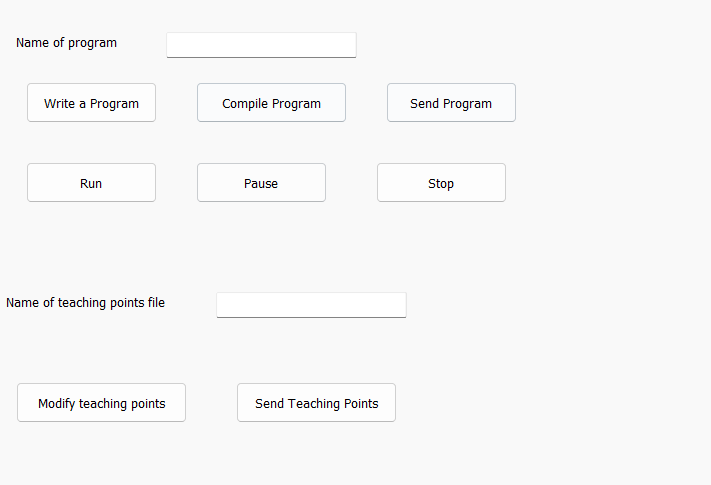
Hình 4.0.31. Minh họa lệnh thay đổi tốc điều khiển các trục của robot

#### 4.2.5.2. Giao diện PROGRAM



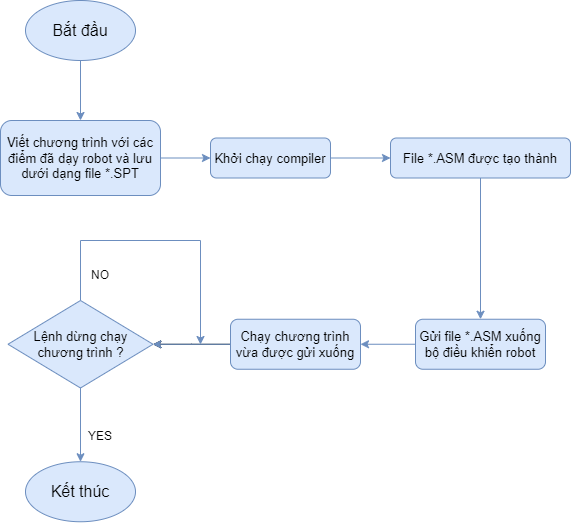
Hình 4.0.32. Sơ đồ tổ chức giao diện PROGRAM

Giao diện PROGRAM được sử dụng ở chế độ tự động.

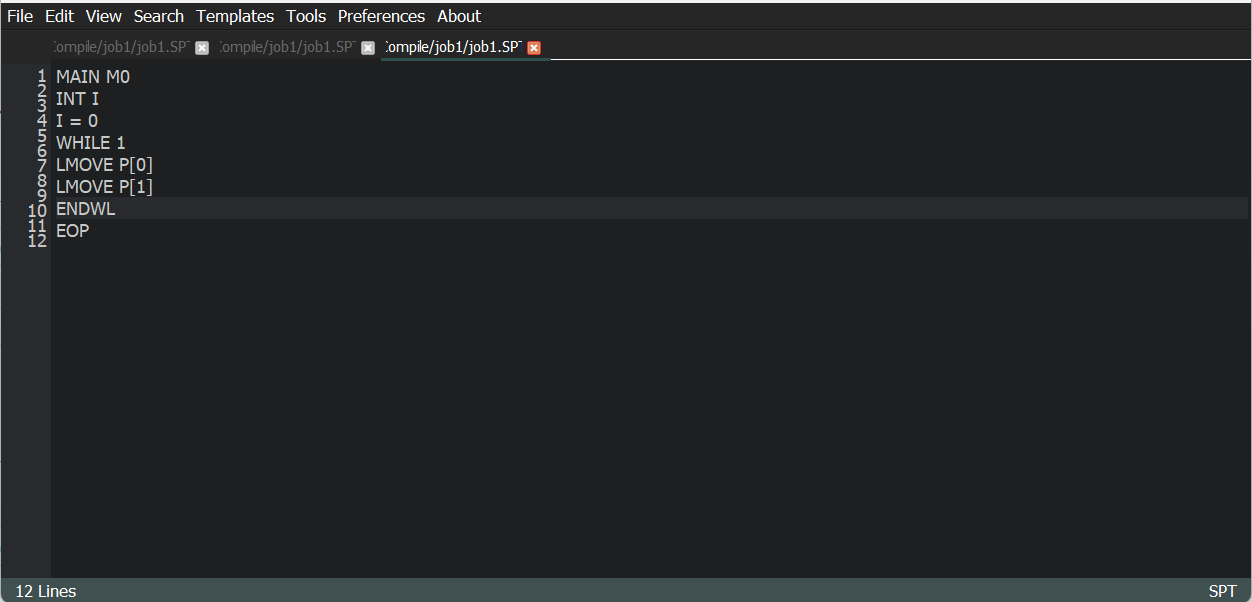


Hình 4.0.33. Giao diện PROGRAM

Quá trình tạo một chương trình robot, gửi xuống bộ điều khiển và chạy tự động chương trình này sẽ được mô tả khái quát như sơ đồ dưới đây:



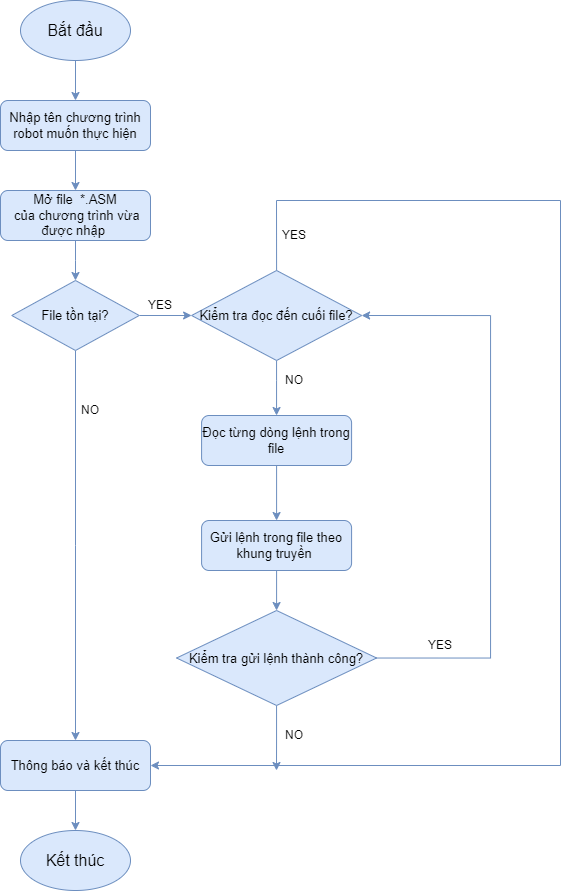
Hình 4.0.34. Sơ đồ thực hiện một chương trình tự động



Hình 4.0.35. Giao diện lập trình chương trình robot

* Cụ thể từng lệnh ở giao diện PROGRAM sẽ được trình bày dưới đây:

Lệnh gửi chương trình robot xuống bộ điều khiển bằng nút nhấn Send Program. Khi nhấn nút, chương trình sẽ được gửi đi như sau:



Hình 4.0.36. Sơ đồ quá trình gửi chương trình đến bộ điều khiển robot

* Khung truyền:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | ASMW | , | Argument\_1 | , | Argument\_2 | ; | CRC | ETX(0x03) |

Tên lệnh: ASMW

Argument\_1: Dòng lệnh trong file \*.ASM

Argument\_2: Thứ tự của dòng lệnh được gửi trong file \*.ASM



Hình 4.0.37. Minh họa một dòng lệnh được gửi từ PC đến bộ điều khiển robot

Sau khi gửi xong tất cả các dòng lệnh trong file \*.ASM, một lệnh xác nhận gửi thành công toàn bộ file sẽ được gửi đến bộ điều khiển robot.

Khung truyền:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | SEPG | , | DONE Argument\_1 | ; | CRC | ETX(0x03) |

Tên lệnh: SEPG

Argument\_1: Tên chương trình đã được gửi



Hình 4.0.38. Minh họa lệnh đã hoàn thành gửi một chương trình robot

Tiếp theo, nhấn nút Run để chạy chương trình vừa được gửi.

Khung truyền:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | RPRG | , | Argument\_1 | ; | CRC | ETX(0x03) |

Tên lệnh: RPRG

Argument\_1: Tên chương trình robot được thực hiện



Hình 4.0.39. Minh họa lệnh chạy một chương trình robot

Ta có thể dừng tạm thời chương trình đang thực hiện bằng nút nhấn Pause:

* Khung truyền:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | PPRG | , | Argument\_1 | ; | CRC | ETX(0x03) |

Tên lệnh: PPRG

Argument\_1: Tên chương trình robot đang thực hiện



Hình 4.0.40. Minh họa lệnh tạm dừng một chương trình robot đang thực hiện

Hoặc ta có thể dừng hẳn chương trình đang thực hiện bằng nút nhấn Stop:

* Khung truyền:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | SPRG | , | Argument\_1 | ; | CRC | ETX(0x03) |

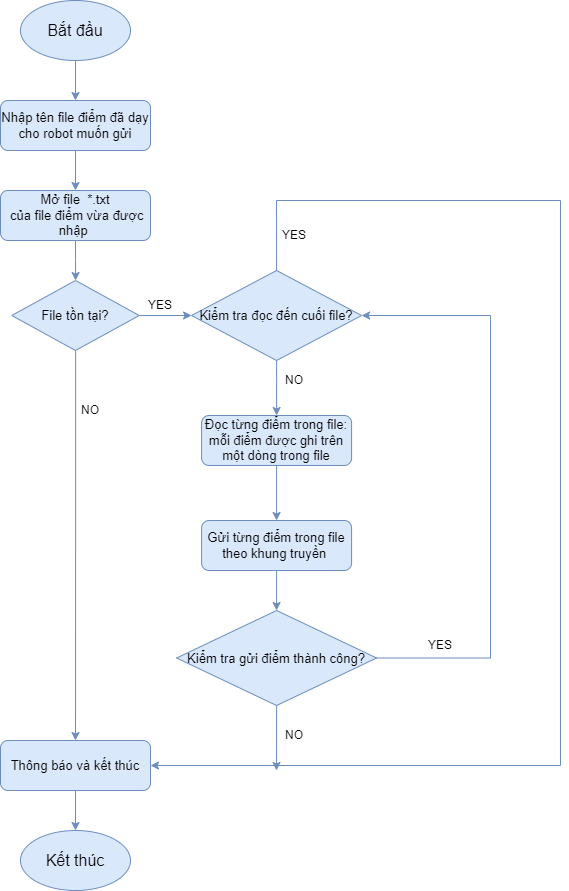
Tên lệnh: SPRG

Argument\_1: Tên chương trình robot đang thực hiện



Hình 4.0.41. Minh họa lệnh dừng hẳn một chương trình robot đang thực hiện

Ngoài ra, ở giao diện PROGRAM còn có 2 nút nhấn Modify Teaching Points và Send Teaching Points. Nút nhấn Modify Teaching Points được dùng để chỉnh sửa các điểm dạy cho robot như thay đổi thứ tự, xóa các điểm đó trong file text. Nút nhấn Send Teaching Points để gửi các điểm đã dạy xuống bộ điều khiển robot để lưu trữ và cho mục đích lập trình sau này (nếu cần thiết). Quá trình gửi các điểm đã dạy cho robot xuống bộ điều khiển robot tương tự như việc gửi chương trình robot.



Hình 4.0.42. Sơ đồ quá trình gửi điểm đã dạy robot xuống bộ điều khiển robot

* Khung truyền:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | TPTS | , | Argument\_1 | , | Argument\_2 | ; | CRC | ETX(0x03) |

Tên lệnh: TPTS

Argument\_1: Các điểm được gửi trong file \*.txt

Argument\_2: Thứ tự của điểm được gửi trong file \*.txt

Sau khi gửi xong tất cả các điểm trong file \*.txt, một lệnh xác nhận gửi thành công toàn bộ file sẽ được gửi đến bộ điều khiển robot.



Hình 4.0.43. Minh họa tọa độ của một điểm theo hệ tọa độ Joint được gửi

* Khung truyền:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | SEPT | , | DONE Argument\_1 | ; | CRC | ETX(0x03) |

Tên lệnh: SEPT

Argument\_1: Tên file điểm được gửi



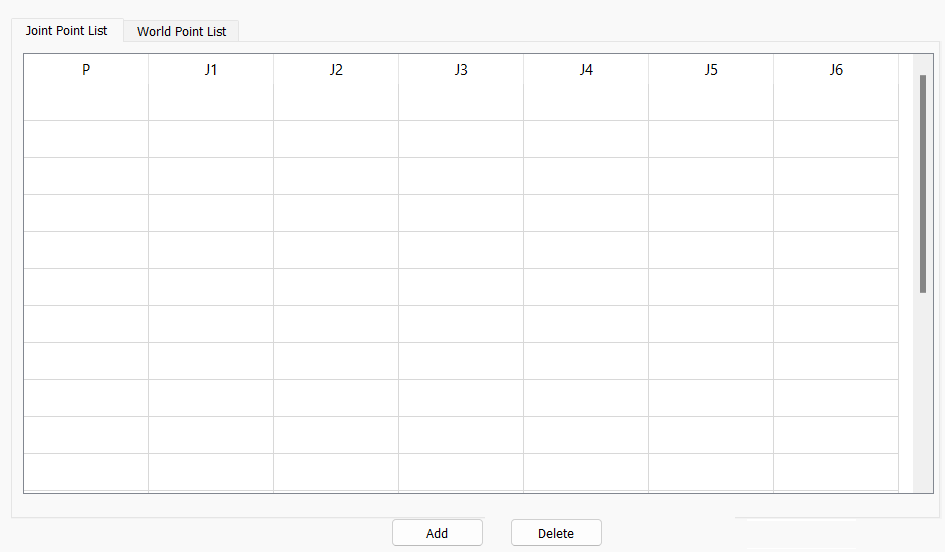
Hình 4.0.44. Minh họa lệnh đã hoàn thành gửi các điểm dạy robot

#### 4.2.5.3. Giao diện DATA

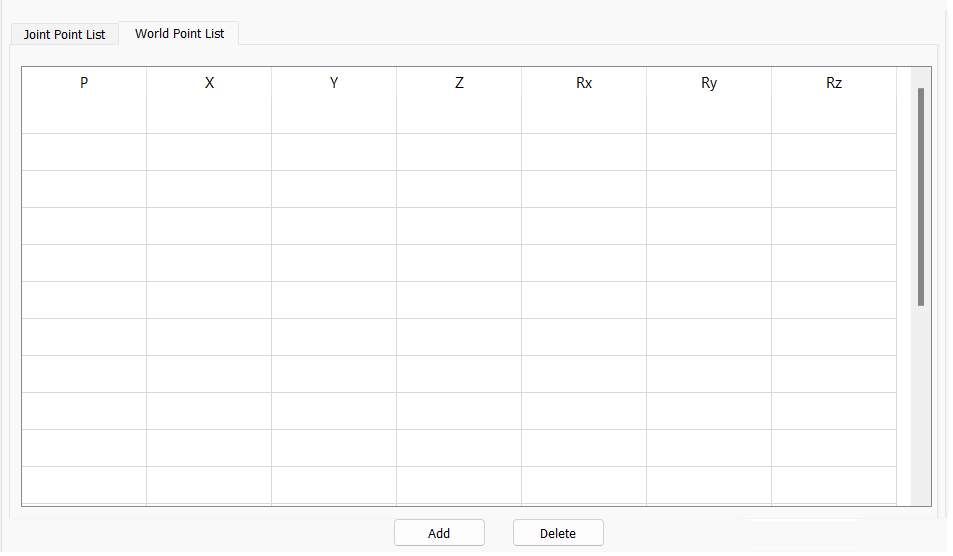
Giao diện này được sử dụng để lữu trữ các điểm dạy robot vào các bảng giá trị.



Hình 4.0.45. Sơ đồ tổ chức của giao diện DATA



Hình 4.0.46. Giao diện danh sách điểm theo hệ tọa độ Joint



Hình 4.0.47. Giao diện danh sách điểm theo hệ tọa độ World

Giao diện DATA bao gồm các nút nhấn chính:

* Nút nhấn Add: lưu giá trị tọa độ hiện tại của robot vào 2 danh sách trên.
* Nút nhấn Delete: xóa tất cả các giá trị tọa độ trong 2 danh sách trên.
* Ngoài ra, ta có thể ghi các giá trị tọa độ các điểm được lưu trong 2 danh sách trên vào các file \*.txt tương ứng để gửi xuống bộ điều khiển robot bằng cách nhấn vào các điểm đó trong các bảng danh sách.

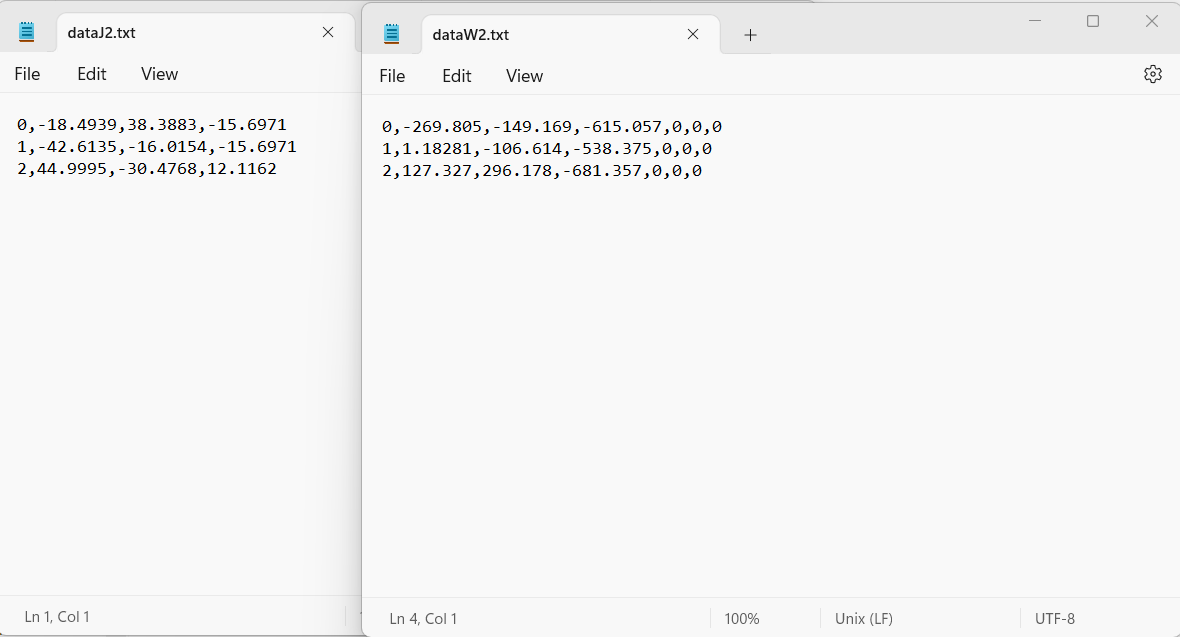
Giá trị tọa độ các điểm dạy robot được lưu vào các file \*.txt theo 2 hệ tọa độ Joint và World theo định dạng:

* Hệ tọa độ Joint:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thứ tự | , | J1 | , | J2 | , | J3 | , | J4 | , | J5 | , | J6 |

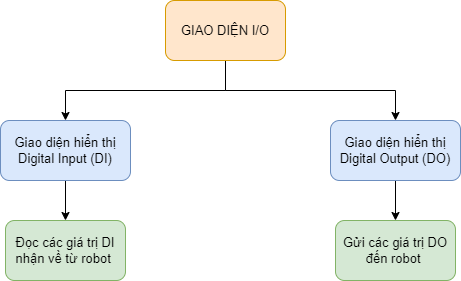
* Hệ tọa độ World:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thứ tự | , | X | , | Y | , | Z | , | Rx | , | Ry | , | Rz |



Hình 4.0.48. Giá trị tọa độ các điểm dạy robot được lưu vào file \*.txt theo các hệ tọa độ tương ứng

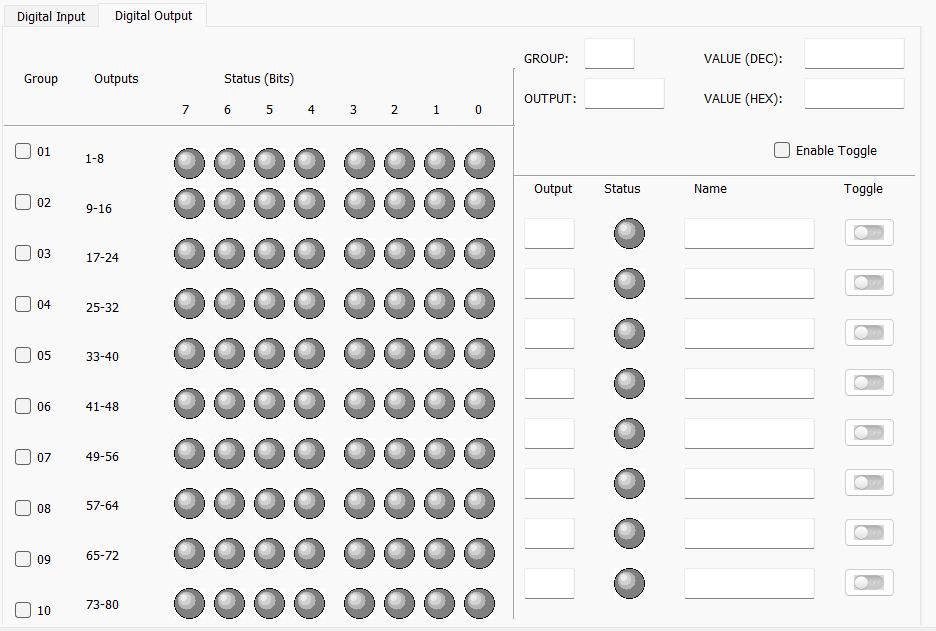
#### 4.2.5.4. Giao diện I/O



Hình 4.0.49. Sơ đồ tổ chức của giao diện I/O

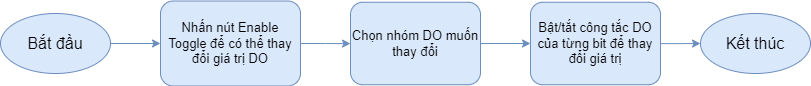


Hình 4.0.50. Giao diện giám sát Digital Input (DI)



Hình 4.0.51. Giao diện giám sát và điều khiển Digital Output (DO)

Ở giao diện giám sát và điều khiển Digital Output (DO), ta có thể thay đổi giá trị các DO như sau:



Hình 4.0.52. Sơ đồ quá trình thay đổi các giá trị DO

Lệnh gửi giá trị DO được thay đổi xuống bộ điều khiển robot:

* Khung truyền:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STX(0x02) | RBDO | , | Argument\_1 | , | Argument\_2 | , | Argument\_3 | ; | CRC | ETX(0x03) |

Tên lệnh: RBDO

Argument\_1: Nhóm DO (1-10)

Argument\_2: Thứ tự bit (0-79)

Argument\_3: Giá trị bit (0/1: tắt/bật)

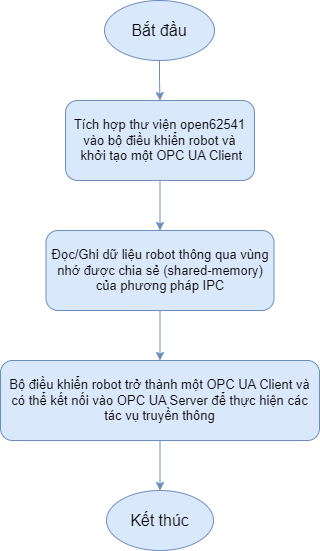


Hình 4.0.53. Minh họa lệnh thay đổi giá trị DO

Chương 5. Tích hợp OPC UA vào bộ điều khiển robot

Vì bộ điều khiển robot được lập trình bằng ngôn ngữ C++, em sử dụng thư viện open62541 để tích hợp OPC UA vào bộ điều khiển robot để biến nó thành một OPC UA Client. Bên cạnh đó, em sử dụng phương pháp giao tiếp giữa các tiến trình (Inter-process communication - IPC) để đọc/ghi dữ liệu của bộ điều khiển. Phần tích hợp OPC UA vào bộ điều khiển robot chủ yếu xử lý back-end nên em sẽ không xây dựng giao diện.

Quá trình tích hợp được thực hiện như lưu đồ sau:

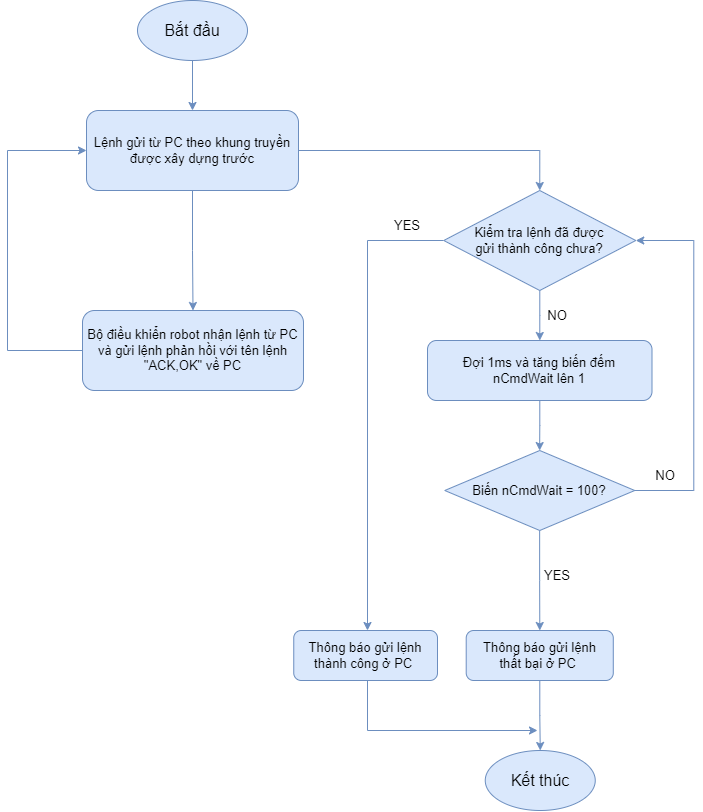


Hình 5.0.1. Sơ đồ quá trình tích hợp OPC UA vào bộ điều khiển robot

## 5.1. Giới thiệu tổng quan về giao tiếp giữa các tiến trình (Inter-process communication) và ứng dụng đọc/ghi dữ liệu của bộ điều khiển robot

Em sử dụng kỹ thuật chia sẻ vùng nhớ (shared-memory) của phương pháp IPC để truy cập đọc/ghi dữ liệu của bộ điều khiển robot. Kỹ thuật này bao gồm việc tạo ra một vùng nhớ chung cho phép cả OPC UA Client được khởi tạo ở bộ điều khiển và bộ điều khiển có thể truy cập vào, vùng nhớ này như một bộ nhớ đệm cho 2 quá trình trên giao tiếp với nhau được liền mạch. Khi 2 quá trình này được gắn vào vùng nhớ được chia sẻ (shared-memory segment), chúng có khả năng đọc và ghi vào chung một vùng nhớ. Vùng nhớ này được tồn tại đến khi OPC UA Client được ngắt kết nối ra khỏi OPC UA Server, cùng lúc đó thì các quá trình trên sẽ được tách ra khỏi vùng nhớ chung này và vùng nhớ chung cũng sẽ được tự động xóa vì không còn cần thiết nữa. Quá trình này được lặp đi lặp lại khi bộ điều khiển robot được trở thành một OPC UA Client kết nối vào OPC UA Server cho đến khi ngắt kết nối.

* 1. Cơ chế gửi lệnh điều khiển từ PC và lệnh phản hồi từ bộ điều khiển robot của hệ thống



Hình 5.0.2. Sơ đồ thể hiện quá trình gửi và nhận lệnh của hệ thống

Lệnh phản hồi của bộ điều khiển robot khi nhận được lệnh điều khiển từ PC

* Khung truyền:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STX(0x02) | ACK,OK | ; | CRC | ETX(0X03) |

Tên lệnh: ACK,OK

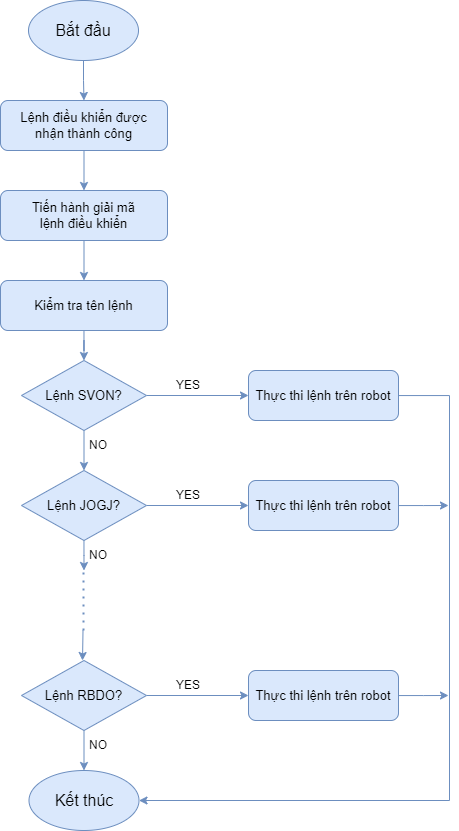


Hình 5.0.3: Minh họa lệnh phản hồi của lệnh điều khiển bật Servo

Mã kiểm tra lỗi CRC ở bộ điều khiển robot sẽ được tạo ra tương tự như cách tạo ra mã CRC ở PC khi gửi xuống. Sau đó, 2 mã CRC ở PC gửi xuống và ở bộ điều khiển robot tạo ra sẽ được so sánh với nhau, nếu giống nhau thì bộ điều khiển sẽ phản hồi lại PC nhận lệnh thành công, nếu khác sẽ hiện thông báo lỗi ở và phản hồi lại PC nhận lệnh thất bại.

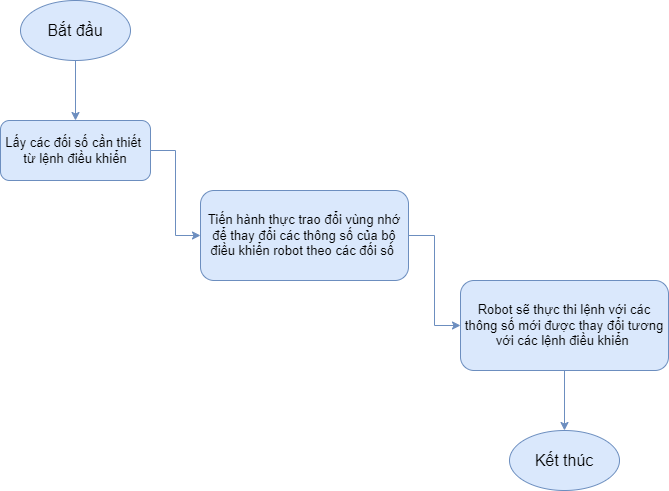
* 1. Xử lý lệnh điều khiển và thực thi lệnh

Sau khi nhận được lệnh thành công, gửi lệnh phản hồi cho PC thì bộ điều khiển robot tiến hành xử lý lệnh vừa nhận được: giải mã chuỗi để lấy tên lệnh điều khiển, các đối số phục vụ cho việc thực các lệnh điều khiển cụ thể của robot.



Hình 5.0.4: Sơ đồ tổng quát quá trình xử lý lệnh nhận về và thực thi trên robot

Phần thực thi lệnh sẽ được khái quát hóa bằng sơ đồ sau:



Hình 5.5: Sơ đồ cụ thể quá trình thực thi lệnh trên robot

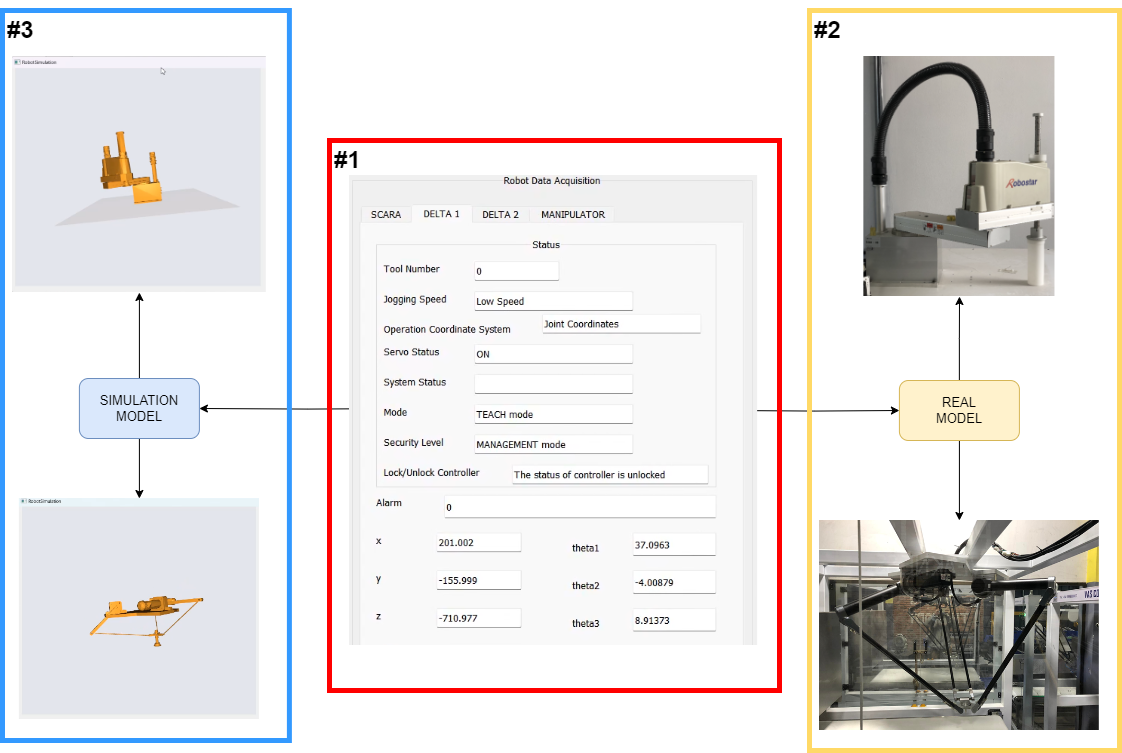
* 1. Cập nhật các thông số giám robot lên OPC UA Server

Ngoài chức năng nhận lệnh điều khiển từ PC và thực thi lệnh đó trên robot, bộ điều khiển còn có chức cập nhật các thông số giám sát robot lên OPC UA Server, để từ các OPC UA Client khác có thể đồng thời giám sát được hoạt động của robot. Các thông số được cập nhật như: chỉ số dụng cụ (Tool Number), tốc độ điều khiển các trục robot(Jogging Speed), hệ tọa độ của robot (Operation Coordinate System), trạng thái bật/tắt Servo (Servo Status), trạng thái hoạt động của hệ thống (System Status), chế độ điều khiển bằng tay/tự động (Mode), chế độ bảo vệ ở chế độ vận hành/chỉnh sửa/quản lý (Security Level), trạng thái mở/khóa của bộ điều khiển robot (Lock/Unlock Controller) cũng đã được đề cập giới thiệu ở các phần trước. Ngoài ra, các thông số về giá trị tọa độ hiện tại, cảnh báo của robot hay các giá trị thay đổi của I/O cũng được cập nhật. Các thông số này được liên tục gửi lên OPC UA Server bằng Timer của bộ điều khiển với chu kỳ 30ms để đảm bảo thông tin của robot được cập nhật liên tục.

Chương 6. Kết quả và hướng phát triển

## 6.1. Kết quả

Hệ thống giám sát, mô phỏng và điều khiển robot được áp dụng lên hệ robot ở lab bao gồm các robot DELTA và SCARA được trình bày khái quát như hình sau:



Hình 6.1. Sơ đồ khái quát hệ thống được xây dựng

Hệ thống được SCADA được xây dựng bao gồm 3 phần chính:

* Phần #1: Giao diện giám sát, điều khiển chính hệ robot
* Phần #2: Giao diện mô phỏng online hành vi của robot
* Phần #3: Hệ robot thật đang được giám sát và điều khiển

<https://autorobots.vn/ung-dung-cua-robot-cong-nghiep/>

<https://mesidas.com/scada/>