**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

**TÔ VIỆT DŨNG**

**LUẬN VĂN TỐT NGHIỆP**

**THIẾT KẾ HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA**

**KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**TP. HỒ CHÍ MINH, 2023**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

**TÔ VIỆT DŨNG – 1912513**

**LUẬN VĂN TỐT NGHIỆP**

**THIẾT KẾ HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA**

**DESIGN A SUPERVISOR SYSTEM FOR MULTIPLE ROBOTS**

**BASED ON OPC UA**

**KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA**

**GIẢNG VIÊN HƯỚNG DẪN**

**TS. NGUYỄN HOÀNG GIÁP**

**TP. HỒ CHÍ MINH, 2023**

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI

TRƯỜNG ĐẠI HỌC BÁCH KHOA –ĐHQG -HCM

Cán bộ hướng dẫn Khóa luận tốt nghiệp :

(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 1 :

(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 2 :

(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Khóa luận tốt nghiệp được bảo vệ tại Trường Đại học Bách Khoa, ĐHQG Tp.HCM ngày . . . . . tháng . . . . năm . . . . .

Thành phần Hội đồng đánh giá khoá luận tốt nghiệp gồm:

(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ khóa luận tốt nghiệp)

1.

2.

3.

4.

5.

Xác nhận của Chủ tịch Hội đồng đánh giá khóa luận tốt nghiệp và Chủ nhiệm Bộ môn sau khi luận văn đã được sửa chữa (nếu có).

**CHỦ TỊCH HỘI ĐỒNG CHỦ NHIỆM BỘ MÔN…………**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  **KHOA ĐIỆN – ĐIỆN TỬ**  **BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG** | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP. HCM, ngày….tháng…..năm……..* |

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP**

**CỦA CÁN BỘ HƯỚNG DẪN**

|  |  |  |
| --- | --- | --- |
| **Tên luận văn:** | | |
| **THIẾT KẾ HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA**  **DESIGN A SUPERVISOR SYSTEM FOR MULTIPLE ROBOTS BASED ON OPC UA** | | |
| **Nhóm Sinh viên thực hiện:** | | **Cán bộ hướng dẫn:** |
| Tô Việt Dũng | 1912513 | TS. Nguyễn Hoàng Giáp |
| **Đánh giá Luận văn**   1. Về cuốn báo cáo:   Số trang Số chương  Số bảng số liệu Số hình vẽ  Số tài liệu tham khảo Sản phẩm  Một số nhận xét về hình thức cuốn báo cáo:   1. Về nội dung luận văn: 2. Về tính ứng dụng: 3. Về thái độ làm việc của sinh viên:   **Đánh giá chung:** Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình  **Điểm từng sinh viên:**  **Tô Việt Dũng:………../10** | | |

|  |  |
| --- | --- |
|  | **Cán bộ hướng dẫn**  (Ký tên và ghi rõ họ tên) |

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  **KHOA ĐIỆN – ĐIỆN TỬ**  **BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG** | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP. HCM, ngày….tháng…..năm……..* |

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP**

**CỦA CÁN BỘ PHẢN BIỆN**

|  |  |  |
| --- | --- | --- |
| **Tên luận văn:** | | |
| **THIẾT KẾ HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA**  **DESIGN A SUPERVISOR SYSTEM FOR MULTIPLE ROBOTS BASED ON OPC UA** | | |
| **Nhóm Sinh viên thực hiện:** | | **Cán bộ phản biện:** |
| Tô Việt Dũng | 1912513 |  |
| **Đánh giá Luận văn**   1. Về cuốn báo cáo:   Số trang Số chương  Số bảng số liệu Số hình vẽ  Số tài liệu tham khảo Sản phẩm  Một số nhận xét về hình thức cuốn báo cáo:   1. Về nội dung luận văn: 2. Về tính ứng dụng: 3. Về thái độ làm việc của sinh viên:   **Đánh giá chung:** Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình  **Điểm từng sinh viên:**  **Tô Việt Dũng:………../10** | | |

|  |  |
| --- | --- |
|  | **Người nhận xét**  (Ký tên và ghi rõ họ tên) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  **KHOA ĐIỆN – ĐIỆN TỬ**  **BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG** | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc | |  | *TP. HCM, ngày….tháng…..năm……..* | |  |

**ĐỀ CƯƠNG CHI TIẾT**

|  |  |
| --- | --- |
| **TÊN LUẬN VĂN:** **THIẾT KẾ HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA** | |
| **Cán bộ hướng dẫn:** | |
| **Thời gian thực hiện:** Từ ngày………………..đến ngày…………… | |
| **Sinh viên thực hiện:**  **Tô Việt Dũng – 1912513** | |
| **Nội dung đề tài:**  Xây dựng một hệ thống tầng giám sát và điều khiển nhiều robot khác nhau sử dụng chuẩn truyền thông OPC UA, bao gồm: 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. Và ứng dụng hệ thống vừa được xây dựng này vào các robot thật để chạy thực nghiệm và đánh giá kết quả. | |
| **Kế hoạch thực hiện:**   |  |  | | --- | --- | | Công việc | Thời gian | | Xây dựng và phát triển các chức năng của OPC UA Server | 05/01/2023 – 15/01/2023 | | Xây dựng và phát triển các tính năng của OPC UA Client | 16/01/2023 – 05/02/2023 | | Xây dựng và phát triển phần mềm mô phỏng | 06/02/2023 – 28/02/2023 | | Xây dựng và phát triển phần mềm tích hợp OPC UA vào bộ điều khiển robot | 01/03/2023 – 20/03/2023 | | Xây dựng SQL Database và cải thiện tính năng bảo mật cho hệ thống | 21/03/2023 – 31/03/2023 | | Thực nghiệm các chức năng của hệ thống trên demo kit của robot SCARA và robot DELTA và đánh giá hệ thống | 01/04/2023 – 15/04/2023 | | Cải thiện và phát triển thêm một số tính năng cho hệ thống | 16/04/2023 – 23/04/2023 | | Thực nghiệm hệ thống trên robot thật, dùng phần mềm đo lường và đánh giá, cải thiện kết quả | 24/04/2023 – 12/05/2023 | | |
| **Xác nhận của Cán bộ hướng dẫn**  (Ký tên và ghi rõ họ tên) | TP. HCM, ngày….tháng …..năm…..  **Sinh viên**  (Ký tên và ghi rõ họ tên) |

**DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN**

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số …………………… ngày ………………….. của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

* 1. …………………………………………. – Chủ tịch.
  2. …………………………………………. – Thư ký.
  3. …………………………………………. – Ủy viên.
  4. …………………………………………. – Ủy viên.
  5. …………………………………………. – Ủy viên.

**LỜI CẢM ƠN**

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đối với quý thầy, cô trường đại học Bách Khoa nói chung và bộ môn Điều khiển - Tự động hóa nói riêng. Mọi người đã truyền đạt những kiến thức quý báu để em có thể hoàn thành đề tài luận văn này.

Em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Hoàng Giáp. Thầy đã chỉ dạy kiến thức chuyên môn cùng với phương pháp nghiên cứu khoa học đúng đắn để em có thể hoàn thành tốt đề tài luận văn của mình. Đối với em đó là sự may mắn rất lớn khi có thầy hướng dẫn, chỉ bảo trong một năm qua.

Xin cảm ơn công ty VAS CORPORATION đã tài trợ trang thiết bị để chúng em có thể hoàn thành được luận văn, cũng như là tiếp cận thêm với thiết bị thực tế trong công nghiệp. Cảm ơn những người đồng nghiệp, những người anh trong VAS CORPORATION đã hướng dẫn tận tình những kiến thức chuyên môn mà em còn thiếu sót.

Cảm ơn quý bạn bè đã giúp đỡ chúng mình trong quá trình học tập và thực hiện đề tài luận văn. Cuối cùng, chúng em xin gửi lời cảm ơn lần nữa tới quý thầy, cô, công ty và bạn bè. Vì kinh nghiệm và kiến thức còn hạn chế, khả năng tiếp cận vấn đề còn giới hạn nên không thể tránh khỏi những sai sót, em mong muốn nhận được sự quan tâm, góp ý và chia sẻ của quý thầy cô và bạn bè để luận văn này cũng như bản thân được hoàn thiện hơn.

TPHCM, ngày 16 tháng 5 năm 2023

Sinh viên

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 15](#_Toc135234877)

[1.1. Tính cấp thiết của luận văn 15](#_Toc135234878)

[1.2. Mục tiêu của luận văn 17](#_Toc135234879)

[1.3. Đối tượng và phạm vi nghiên cứu 17](#_Toc135234880)

[1.3.1. Đối tượng nghiên cứu 17](#_Toc135234881)

[1.3.2. Phạm vi nghiên cứu 18](#_Toc135234882)

[1.4. Nhiệm vụ và nội dung của luận văn 18](#_Toc135234883)

[1.5. Cấu trúc của luận văn 18](#_Toc135234884)

[CHƯƠNG 2. TỔNG QUAN VỀ HỆ THỐNG ROBOT CÔNG NGHIỆP VÀ CHUẨN TRUYỀN THÔNG OPC UA 19](#_Toc135234885)

[2.1. Khái niệm về robot công nghiệp và sự phát triển của robot công nghiệp trong sản xuất 19](#_Toc135234886)

[2.1.1. Khái niệm về robot công nghiệp 19](#_Toc135234887)

[2.1.2. Sự phát triển của robot công nghiệp trong sản xuất 20](#_Toc135234888)

[2.2. Khái niệm về hệ thống SCADA và cấu trúc của một hệ thống SCADA 21](#_Toc135234889)

[2.2.1. Khái niệm về hệ thống SCADA 21](#_Toc135234890)

[2.2.2. Cấu trúc của một hệ thống SCADA 23](#_Toc135234891)

[2.3. Tổng quan về chuẩn truyền thông OPC UA 24](#_Toc135234892)

[2.3.1. Khái niệm về OPC UA 24](#_Toc135234893)

[2.3.2. Các tính năng nổi bật của OPC UA 25](#_Toc135234894)

[CHƯƠNG 3. TỔNG QUAN VỀ CẤU TRÚC HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA 26](#_Toc135234895)

[3.1. Cấu trúc phần cứng 27](#_Toc135234896)

[3.2. Cấu trúc phần mềm 29](#_Toc135234897)

[CHƯƠNG 4. THIẾT KẾ CÁC PHẦN MỀM CHO HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA 30](#_Toc135234898)

[4.1. Thiết kế OPC UA Server 30](#_Toc135234899)

[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 30](#_Toc135234900)

[4.1.2. Cấu trúc OPC UA Server đã thiết kế 30](#_Toc135234901)

[4.1.3. Định dạng địa chỉ OPC UA Server và địa chỉ các node 31](#_Toc135234902)

[4.1.4. SQL Database của OPC UA Server 32](#_Toc135234903)

[4.2. Thiết kế OPC UA Client 33](#_Toc135234904)

[4.2.1. Giới thiệu về thư viện mở open62541 hỗ trợ thiết kế OPC UA Client 33](#_Toc135234905)

[4.2.2. Tổng quan các chức năng của phần mềm OPC UA Client 34](#_Toc135234906)

[4.2.3. Giao diện giám sát robot 35](#_Toc135234907)

[4.2.3.1. Giao diện DATA ACQUISITION 35](#_Toc135234908)

[4.2.3.2. Giao diện MONITOR 37](#_Toc135234909)

[4.2.3.3. Giao diện SIMULATION 38](#_Toc135234910)

[4.2.3.4. Giao diện HISTORY 46](#_Toc135234911)

[4.2.4. Giao diện điều khiển robot 48](#_Toc135234912)

[4.2.4.1. Giao diện JOG 51](#_Toc135234913)

[4.2.4.2. Giao diện PROGRAM 61](#_Toc135234914)

[4.2.4.3. Giao diện DATA 68](#_Toc135234915)

[4.2.4.4. Giao diện I/O 71](#_Toc135234916)

[4.2.5. Cơ chế bắt tay trong truyền nhận lệnh điều khiển giữa PC Client và bộ điều khiển robot 74](#_Toc135234917)

[4.3. Cải thiện tính năng bảo mật cho hệ thống được thiết kế 75](#_Toc135234918)

[4.4. Tích hợp OPC UA vào bộ điều khiển robot 78](#_Toc135234919)

[4.4.1. Quá trình tích hợp OPC UA vào bộ điều khiển robot 78](#_Toc135234920)

[4.4.2. Xử lý lệnh điều khiển ở bộ điều khiển và thực thi lệnh trên robot 80](#_Toc135234921)

[4.4.3. Cập nhật các thông số giám robot lên OPC UA Server 81](#_Toc135234922)

[CHƯƠNG 5. KẾT QUẢ THỰC HIỆN 82](#_Toc135234923)

[5.1. Thực nghiệm hệ thống 82](#_Toc135234924)

[5.2. Kết quả đạt được 84](#_Toc135234925)

[CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 86](#_Toc135234926)

[6.1. Kết luận 86](#_Toc135234927)

[6.2. Hướng phát triển 86](#_Toc135234928)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1: Sơ đồ sự bất đồng bộ truyền thông của các hãng robot khác nhau 17](#_Toc135235142)

[Hình 1.2: Sơ đồ giải pháp đồng bộ hóa truyền thông các hãng robot khác nhau 18](#_Toc135235143)

[Hình 3.1: Cấu trúc hệ thống giám sát và điều khiển nhiều robot sử dụng OPC UA 29](#_Toc135235144)

[Hình 3.2: Mô hình robot SCARA 30](#_Toc135235145)

[Hình 3.3: Mô hình robot DELTA 30](#_Toc135235146)

[Hình 3.4: Giao diện OPC UA Server 31](#_Toc135235147)

[Hình 3.5: Giao diện OPC UA Client trên PC 31](#_Toc135235148)

[Hình 4.1: Cấu trúc hình cây của OPC UA Server 33](#_Toc135235149)

[Hình 4.2: SQL Database của OPC UA Server 35](#_Toc135235150)

[Hình 4.3: Cấu trúc các khối chức năng của OPC UA Client trên PC 36](#_Toc135235151)

[Hình 4.4: Giao diện giám sát các thông số của robot 38](#_Toc135235152)

[Hình 4.5: Sơ đồ chức năng của các Timer giám sát thông số robot trên PC 39](#_Toc135235153)

[Hình 4.6: Giao diện MONITOR 40](#_Toc135235154)

[Hình 4.7: Sơ đồ quá trình mô phỏng robot 41](#_Toc135235155)

[Hình 4.8: Động học thuận robot DELTA 42](#_Toc135235156)

[Hình 4.9: Động học nghịch robot DELTA (1) 44](#_Toc135235157)

[Hình 4.10: Động học nghịch robot DELTA (2) 45](#_Toc135235158)

[Hình 4.11: Mô hình mô phỏng robot DELTA 46](#_Toc135235159)

[Hình 4.12: Động học thuận robot SCARA 46](#_Toc135235160)

[Hình 4.13: Mô hình mô phỏng robot SCARA 48](#_Toc135235161)

[Hình 4.14: Sơ đồ tổ chức giao diện HISTORY 48](#_Toc135235162)

[Hình 4.15: Giao diện Operation History 49](#_Toc135235163)

[Hình 4.16: Giao diện Alarm History 49](#_Toc135235164)

[Hình 4.17: Sơ đồ chức năng của giao diện điều khiển robot 50](#_Toc135235165)

[Hình 4.18: Giao diện chọn robot để điều khiển 51](#_Toc135235166)

[Hình 4.19: Giao diện nút Servo 51](#_Toc135235167)

[Hình 4.20: Lệnh bật/tắt Servo 52](#_Toc135235168)

[Hình 4.21: Giao diện nút chọn chế độ điều khiển 52](#_Toc135235169)

[Hình 4.22: Lệnh chọn chế độ điều khiển 52](#_Toc135235170)

[Hình 4.23: Giao diện nút khóa bộ điều khiển 52](#_Toc135235171)

[Hình 4.24: Lệnh mở/khóa bộ điều khiển robot 53](#_Toc135235172)

[Hình 4.25: Giao diện chọn chế độ bảo vệ 53](#_Toc135235173)

[Hình 4.26: Sơ đồ tổ chức giao diện JOG 53](#_Toc135235174)

[Hình 4.27: Giao diện JOG 54](#_Toc135235175)

[Hình 4.28: Giao diện chọn hệ tọa độ 54](#_Toc135235176)

[Hình 4.29: Giao diện chọn dụng cụ 55](#_Toc135235177)

[Hình 4.30: Sơ đồ giải thuật tính toán các giá trị thời gian trục robot di chuyển 56](#_Toc135235178)

[Hình 4.31: Mô phỏng MATLAB các giá trị p, v, a theo thời gian trường hợp 1 58](#_Toc135235179)

[Hình 4.32: Mô phỏng MATLAB các giá trị p, v, a theo thời gian trường hợp 2 59](#_Toc135235180)

[Hình 4.33: Mô phỏng MATLAB các giá trị p, v, a theo thời gian trường hợp 3 60](#_Toc135235181)

[Hình 4.34: Mô phỏng MATLAB các giá trị p, v, a theo thời gian trường hợp 4 61](#_Toc135235182)

[Hình 4.35: Lệnh điều khiển trục của robot kèm theo thời gian di chuyển của trục 62](#_Toc135235183)

[Hình 4.36: Lệnh dừng trục đang di chuyển của robot 62](#_Toc135235184)

[Hình 4.37: Lệnh thay đổi tốc điều khiển các trục của robot 63](#_Toc135235185)

[Hình 4.38: Sơ đồ tổ chức giao diện PROGRAM 63](#_Toc135235186)

[Hình 4.39: Giao diện PROGRAM 64](#_Toc135235187)

[Hình 4.40: Sơ đồ thực hiện một chương trình tự động 65](#_Toc135235188)

[Hình 4.41: Giao diện lập trình chương trình robot 65](#_Toc135235189)

[Hình 4.42: Sơ đồ quá trình gửi chương trình đến bộ điều khiển robot 66](#_Toc135235190)

[Hình 4.43: Lệnh gửi các dòng lệnh của chương trình robot 67](#_Toc135235191)

[Hình 4.44: Lệnh hoàn thành gửi một chương trình robot 67](#_Toc135235192)

[Hình 4.45: Lệnh chạy một chương trình robot 68](#_Toc135235193)

[Hình 4.46: Lệnh tạm dừng một chương trình robot đang thực hiện 68](#_Toc135235194)

[Hình 4.47: Lệnh dừng hẳn một chương trình robot đang thực hiện 68](#_Toc135235195)

[Hình 4.48: Sơ đồ quá trình gửi điểm đã dạy robot xuống bộ điều khiển robot 69](#_Toc135235196)

[Hình 4.49: Lệnh gửi tọa độ của của điểm dạy robot 70](#_Toc135235197)

[Hình 4.50: Lệnh hoàn thành gửi các điểm dạy robot 70](#_Toc135235198)

[Hình 4.51: Sơ đồ tổ chức của giao diện DATA 71](#_Toc135235199)

[Hình 4.52: Giao diện danh sách điểm theo hệ tọa độ Joint 71](#_Toc135235200)

[Hình 4.53: Giao diện danh sách điểm theo hệ tọa độ World 72](#_Toc135235201)

[Hình 4.54: Giá trị tọa độ các điểm dạy robot được lưu vào các file \*.txt 73](#_Toc135235202)

[Hình 4.55: Sơ đồ tổ chức của giao diện I/O 73](#_Toc135235203)

[Hình 4.56: Giao diện giám sát Digital Input (DI) 74](#_Toc135235204)

[Hình 4.57: Giao diện giám sát và điều khiển Digital Output (DO) 74](#_Toc135235205)

[Hình 4.58: Sơ đồ quá trình thay đổi các giá trị DO 75](#_Toc135235206)

[Hình 4.59: Lệnh thay đổi giá trị DO 75](#_Toc135235207)

[Hình 4.60: Sơ đồ cơ chế bắt tay giữa PC Client và bộ điều khiển robot 76](#_Toc135235208)

[Hình 4.61: Lệnh phản hồi từ bộ điều khiển robot 77](#_Toc135235209)

[Hình 4.62: Sơ đồ quá trình xác thực giữa OPC UA Client và OPC UA Server 78](#_Toc135235210)

[Hình 4.63: Quá trình xác thực giữa Socket Client và Socket Server 79](#_Toc135235211)

[Hình 4.64: File \*.yml lưu trữ thông tin xác thực 80](#_Toc135235212)

[Hình 4.65: Sơ đồ quá trình tích hợp OPC UA vào bộ điều khiển robot 81](#_Toc135235213)

[Hình 4.66: Sơ đồ tổng quát quá trình xử lý lệnh nhận về và thực thi trên robot 82](#_Toc135235214)

[Hình 4.67: Sơ đồ cụ thể quá trình thực thi lệnh trên robot 83](#_Toc135235215)

[Hình 5.1: Sơ đồ kết nối của phương án 1 84](#_Toc135235216)

[Hình 5.2: Sơ đồ kết nối của phương án 2 85](#_Toc135235217)

[Hình 5.3: Sơ đồ các vùng làm việc của người dùng 86](#_Toc135235218)

**DANH MỤC BẢNG**

[Bảng 2.1: Bảng so sánh tính năng nổi bật của OPC UA so với Modbus và Profinet 30](#_Toc135235591)

[Bảng 5.1: Bảng thông số đo lường và đánh giá 89](#_Toc135235592)

# 

# **GIỚI THIỆU ĐỀ TÀI**

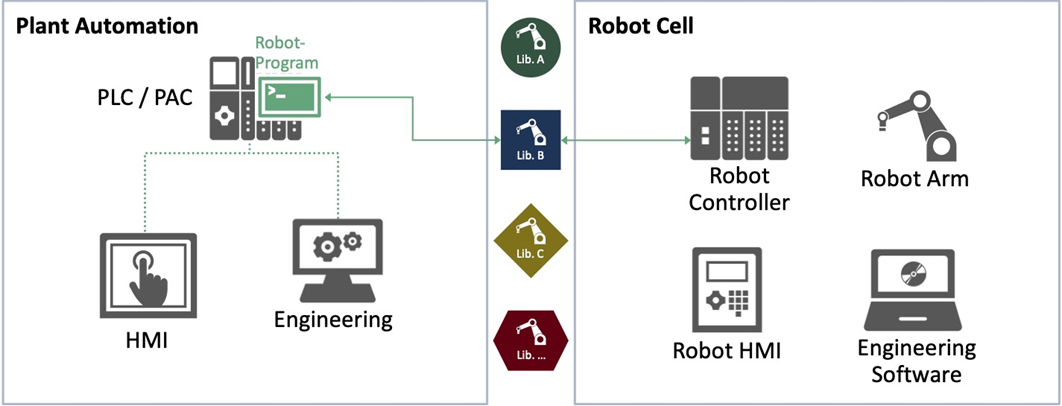
## **Tính cấp thiết của luận văn**

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.



Hình 1.1: Sơ đồ sự bất đồng bộ 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 1.2: Sơ đồ giải pháp đồng bộ hóa truyền thông 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.

## **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.

## **Đối tượng và phạm vi nghiên cứu**

### **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

### **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 tầng giám sát và điều khiển 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.
* Tìm hiểu về giao thức truyền thông để điều khiển robot.
* Tìm hiểu về tính năng bảo mật cho hệ thống.

## **Nhiệm vụ và nội dung của luận văn**

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

* Xây dựng một cấu trúc truyền thông ở tầng giám sát và điều khiển 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.

## **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 hệ thống robot hiện nay và lý thuyết về chuẩn truyền thông OPC UA

Chương 3: Tổng quan về cấu trúc hệ thống giám sát và điều khiển nhiều robot sử dụng OPC UA - trình bày cấu trúc tổng quan của hệ thống được thiết kế trong luận văn.

Chương 4: Thiết kế phần mềm cho hệ thống giám sát và điều khiển nhiều robot sử dụng OPC UA - trình bày các phần mềm được xây dựng bao gồm OPC UA Server, OPC UA Client trên PC, phần mềm tích hợp OPC UA vào bộ điều khiển robot.

Chương 5: Kết quả thực hiện – trình bày về quá trình thực nghiệm hệ thống, đo lường và đánh giá các kết quả đạt được

Chương 6: Kết luận và hướng phát triển - trình bày kết luận, ưu nhược điểm của hệ thống đã thực hiện được và đưa ra hướng nghiên cứu trong tương lai.

# **TỔNG QUAN VỀ HỆ THỐNG ROBOT CÔNG NGHIỆP VÀ CHUẨN TRUYỀN THÔNG OPC UA**

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

### **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.

### **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.
* Robot hàn xì tự động.
* Robot được ứng dụng trong ngành công nghiệp đúc để rót kim loại nóng chảy.

Các loại robot công nghiệp phổ biến

* 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.

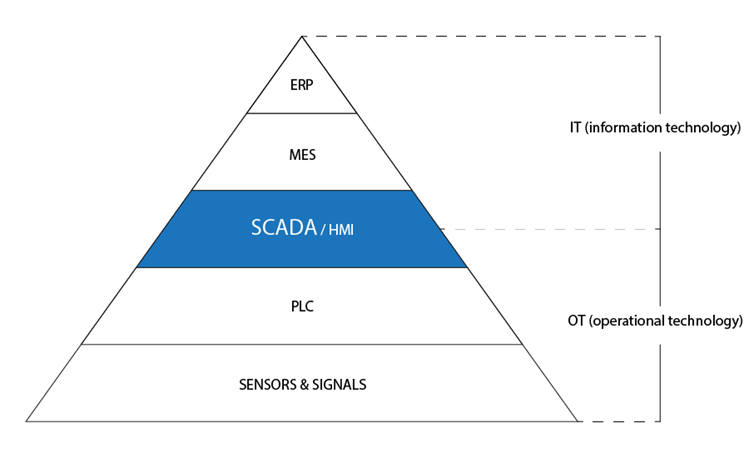
Các hãng sản xuất robot hàng đầu trên thế giới

* 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).

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

### **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.

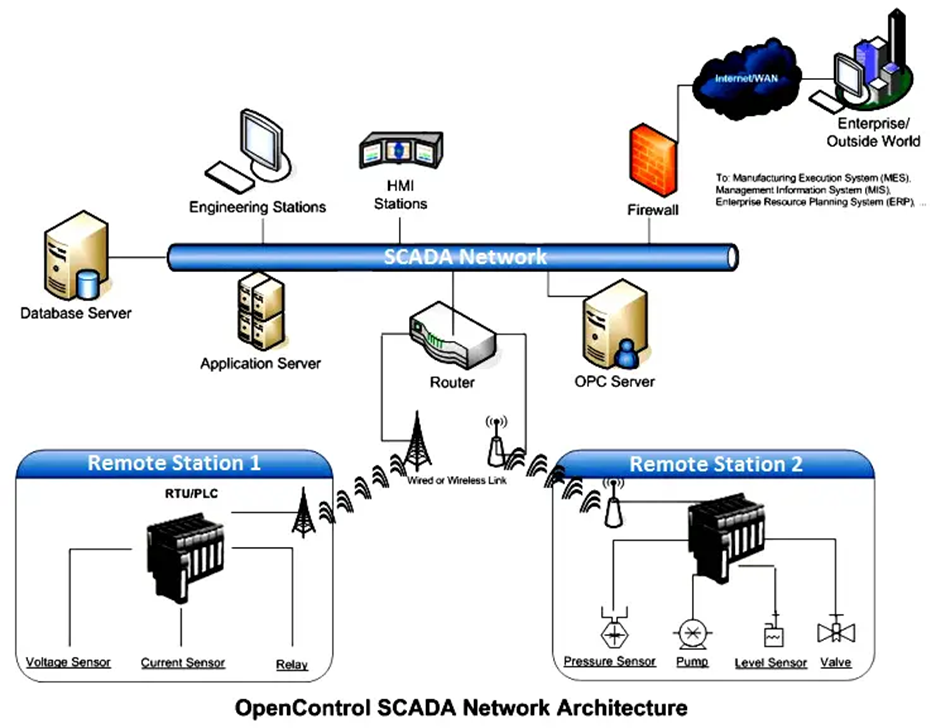


Hình 2.1: Vị trí của SCADA trong kim tự tháp tự dộng hóa

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.

### **Cấu trúc của một hệ thống SCADA**



*Hình 2.2: Cấu trúc tổng quát của một hệ thống SCADA*

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

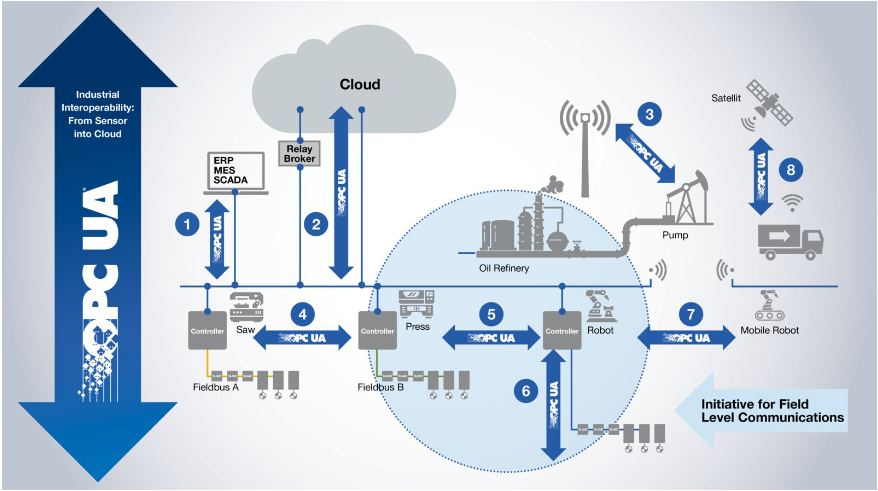
* Thiết bị từ xa (RTU-Remote Terminal Unit/PLC): là các thiết bị như RTU hay PLC thực hiện quá trình điều khiển trực tiếp các cơ cấu chấp hành, đồng thời dữ liệu thu thập được từ các cơ cấu chấp hành này sẽ được RTU/PLC truyền về máy chủ hoặc RTU/PLC nhận các lệnh điều khiển từ máy chủ để thực thi theo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

Hiện nay, Mistubishi sử dụng giao thức CC Link IE, tương tự ABB dùng Modbus TCP và Profinet, trong khi Yaskawa, KUKA, … cũng có kiến trúc và chuẩn giao tiếp riêng của họ.

## **Tổng quan về chuẩn truyền thông OPC UA**

### **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 2.3: Tổng quan các ứng dụng của OPC UA trong công nghiệp*

### **Các tính năng nổi bật của OPC UA**

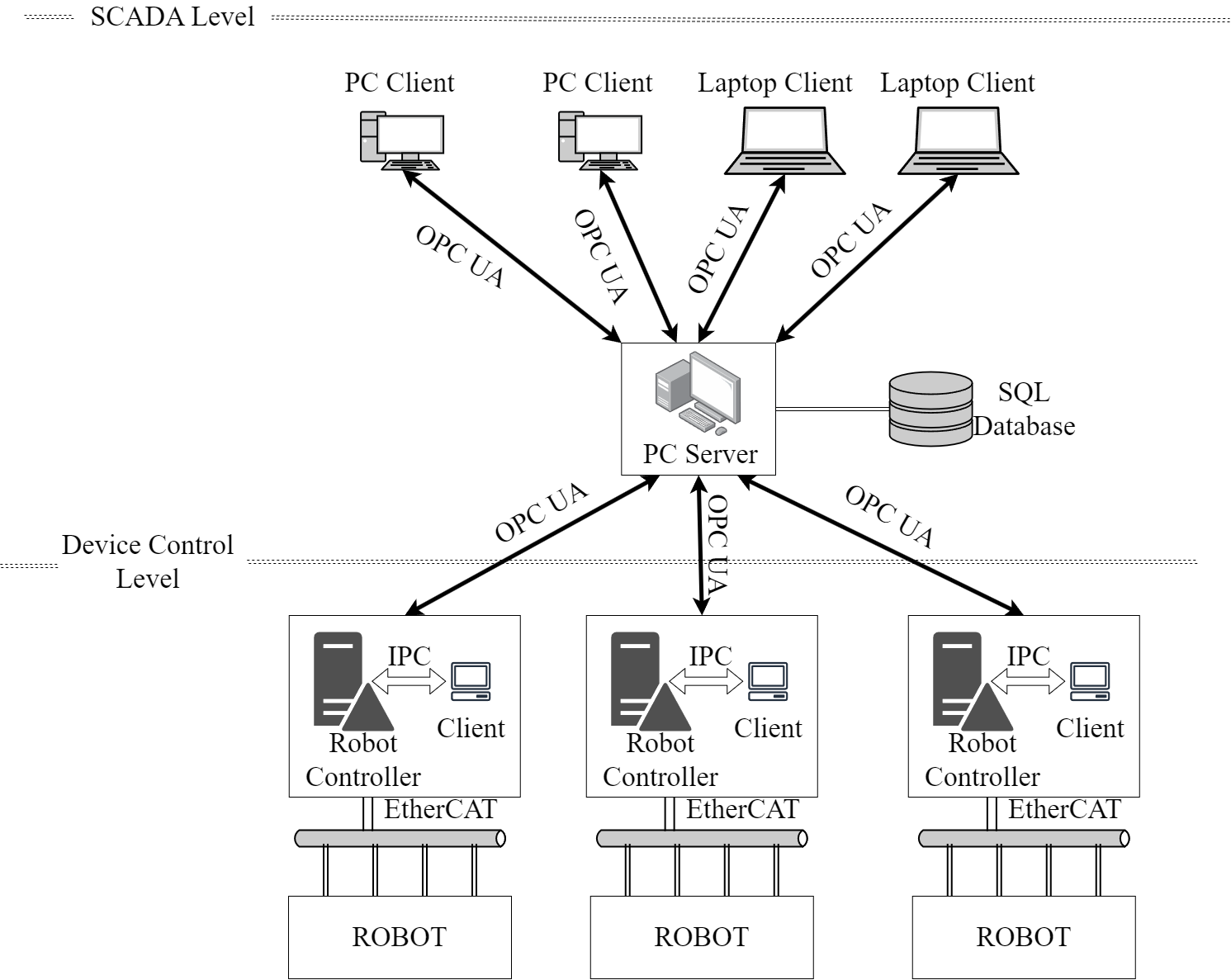
* Kết hợp tất cả các tính năng rời rạc của COM OPC Classic: Data Access, Historical Data Access, Alarm and Events, …
* Đa nền tảng: có thể sử dụng ở các phần cứng và phần mềm khác nhau
* Bảo mật thông tin cao: Session Encryption, Authentication, User Control, …
* Mô hình thông tin (Information Modeling): 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 .: Bảng so sánh tính năng nổi bật của OPC UA so với Modbus và Profinet

# **TỔNG QUAN VỀ CẤU TRÚC HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA**

Cấu trúc của hệ thống giám sát và điều khiển nhiều robot sử dụng chuẩn truyền thông OPC UA được trình bày như trong sơ đồ dưới đây:



Hình 3.1: Cấu trúc hệ thống giám sát và điều khiển nhiều robot sử dụng OPC UA

Hệ thống được thiết kế bao gồm:

* Cấu trúc phần cứng.
* Cấu trúc phần mềm.

## **Cấu trúc phần cứng**

* Hệ thống robot: bao gồm có robot DELTA và SCARA được điều khiển bằng các bộ EtherCat Master Robot Controller. Mỗi bộ điều khiển EtherCat Master Robot Controller này có chức năng điều khiển các Driver lái động cơ robot sử dụng chuẩn EtherCat. Dưới đây là mô hình các robot được sử dụng:



Hình .: Mô hình robot SCARA

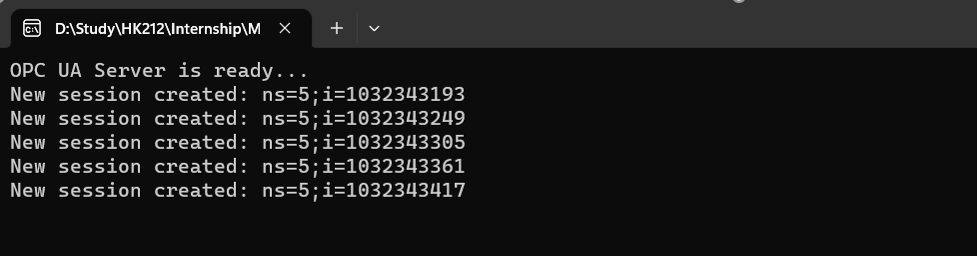


Hình .: Mô hình robot DELTA

* Các máy tính PC, laptop: dùng để chạy các phần mềm OPC UA Server và OPC UA Client

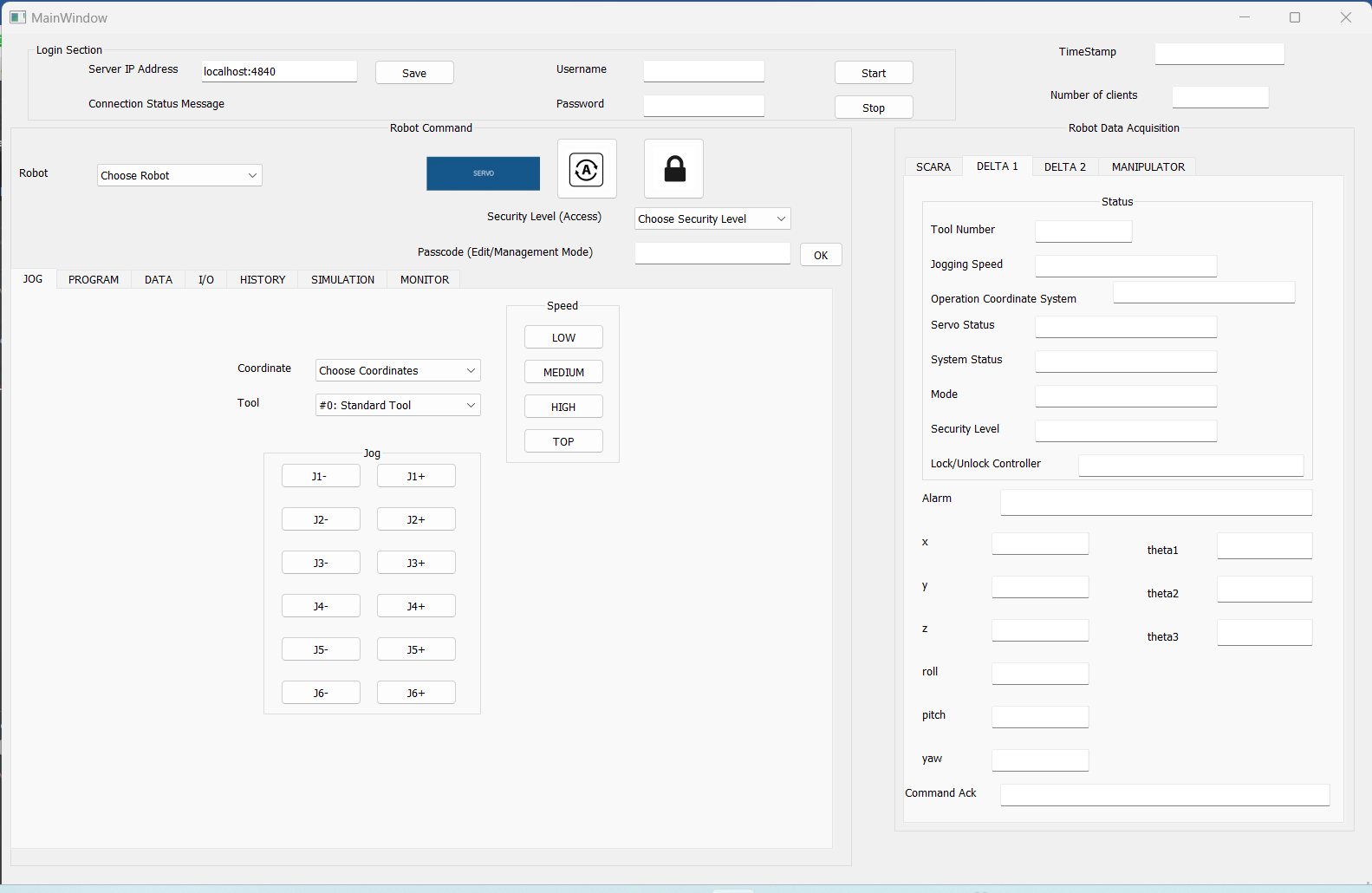
## **Cấu trúc phần mềm**

* Phần mềm OPC UA Server: là nơi lưu trữ thông tin của các OPC UA Client, lưu trữ các node dữ liệu của các robot trong hệ thống và có 1 SQL Database là cơ sở dữ liệu cho hệ thống.



Hình .: Giao diện OPC UA Server

* Phần mềm OPC UA Client: là một phần mềm có giao diện để người dùng giám sát, mô phỏng và điều khiển các robot.



Hình .: Giao diện OPC UA Client trên PC

* Phần mềm tích hợp OPC UA vào bộ điều khiển robot: xây dựng một cơ chế để đọc/ghi dữ liệu của bộ điều khiển và biến bộ điều khiển robot thành một OPC UA Client.

# **THIẾT KẾ CÁC PHẦN MỀM CHO HỆ THỐNG GIÁM SÁT VÀ ĐIỀU KHIỂN NHIỀU ROBOT SỬ DỤNG OPC UA**

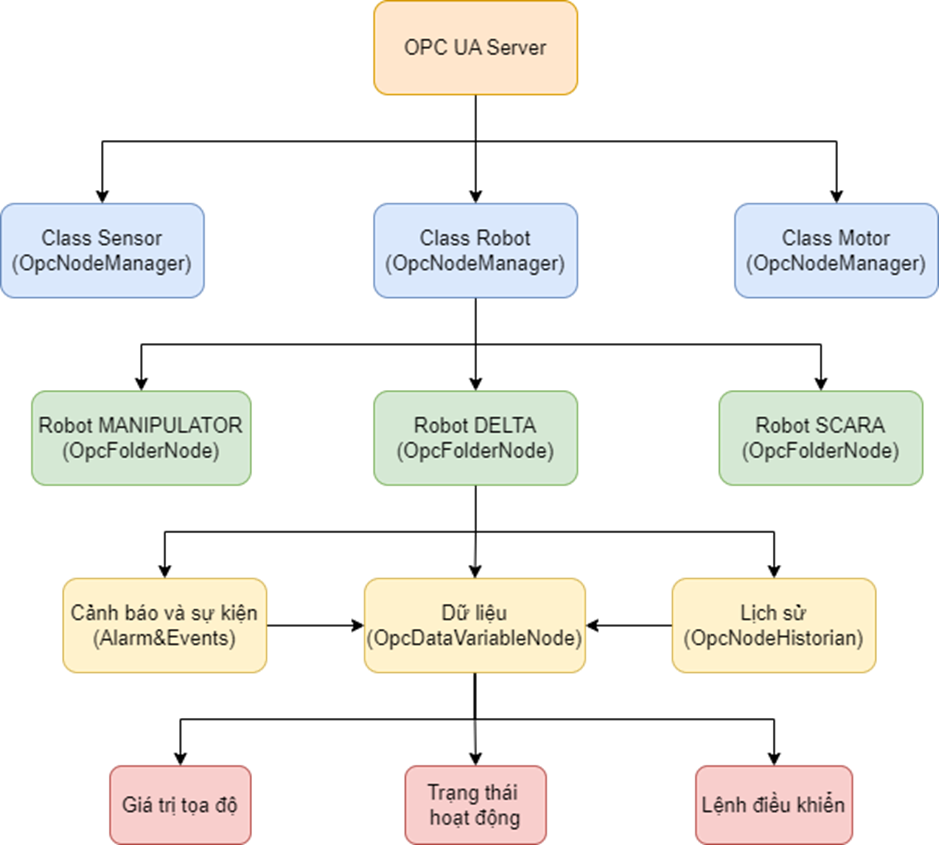
## **Thiết kế OPC UA Server**

### **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 được cung cấp 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.

### **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 em nghiên cứu xây dựng:



Hình .: Cấu trúc hình cây của OPC UA Server

### **Định dạng địa chỉ OPC UA Server và địa chỉ các node**

Khi kết nối vào OPC UA Server, OPC UA Client cần phải biết địa chỉ của Server. Địa chỉ OPC UA Server có định dạng như sau:

|  |
| --- |
| “opc:tcp//[IPAddress]:[Port]” |

Trong đó:

* IPAddress: Địa chỉ IP của máy chủ chạy OPC UA Server.
* Port = 4840: đây là port được đặt riêng cho OPC UA

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] |

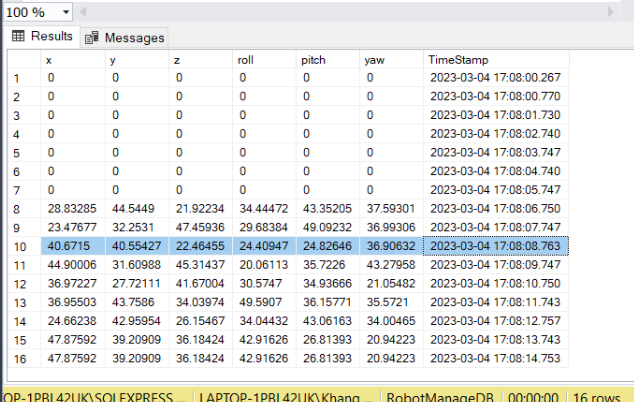
Trong đó:

* 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.

### **SQL Database của OPC UA Server**

Em tìm hiểu sử dụng một Cơ sở dữ liệu (Database) SQL Server để lưu trữ dữ liệu thông số của robot, từ những dữ liệu đó có thể phân tích hoạt động của các robot, qua đó dự đoán hư hỏng và đề suất các dịch vụ bảo trì. Để làm được điều đó, em phải kết nối OPC UA Server đến SQL Server Database thông qua SqlConnection với địa chỉ kết nối được lưu trong một Connection String. Sau khi đã thiết lập kết nối đến Database, OPC UA Server sẽ sử dụng các lệnh truy vấn trực tiếp đến Database để ghi vào đó dữ liệu của các robot.



Hình .: SQL Database của OPC UA Server

## **Thiết kế OPC UA Client**

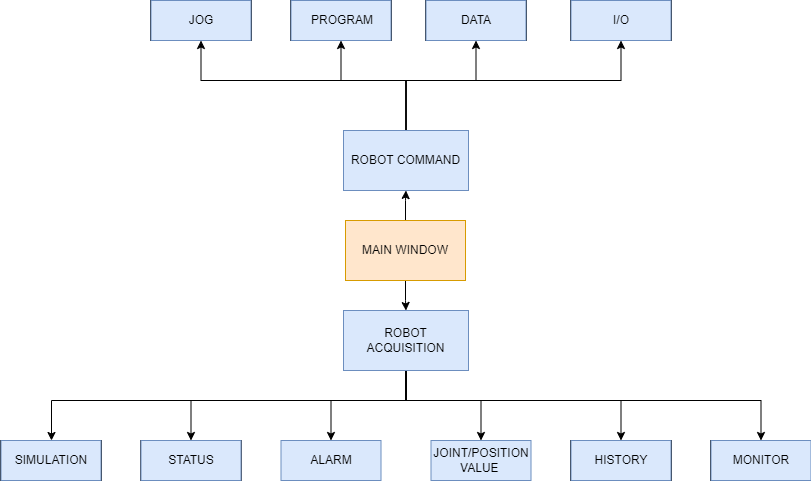
### **Giới thiệu về thư viện mở open62541 hỗ trợ thiết kế OPC UA Client**

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, được kiểm chứng qua các bài báo về độ bảo mật của thư viện này, đảm bảo sự tin cậy cho người dùng.

### **Tổng quan các chức năng của phần mềm OPC UA Client**

OPC UA Client trên PC có đầy đủ các chức năng giám sát của 1 hệ thống SCADA, như thu thập dữ liệu từ robot, gửi lệnh điều khiển như bộ điều khiển nhiều robot.

Sơ đồ tổng quát các khối chức năng của một OPC UA Client trên PC:



Hình .: Cấu trúc các khối chức năng của OPC UA Client trên PC

Trong đó có 2 khối chức năng chính:

* Robot Acquisition (Khối giám sát robot):
* Dữ liệu của robot: trạng thái, cảnh báo, giá trị tọa độ
* History: lịch sử hoạt động,
* Monitor: đồ thị các góc khớp
* Simulation: mô phỏng online hành vi của 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

Một OPC UA Client trên PC hoặc laptop được gọi tắt là một PC Client.

### **Giao diện giám sát robot**

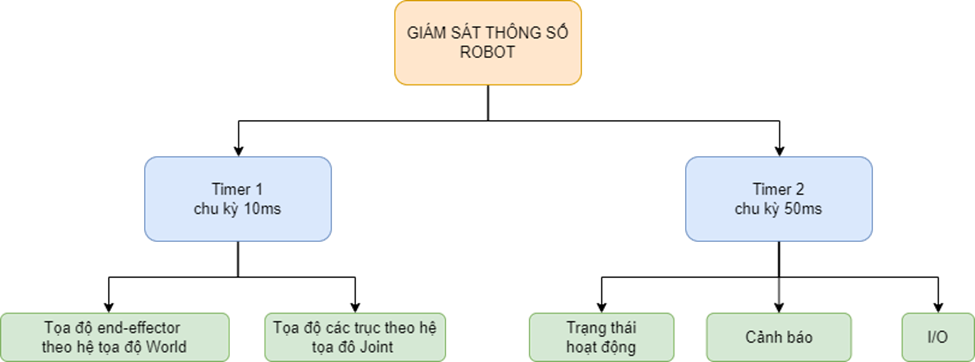
#### Giao diện DATA ACQUISITION

Một PC Client có thể giám sát đồng thời các robot khác nhau, mỗi Tab giám sát thông số của một robot với các thông số cần quan sát được thể hiện ở phần giao diện cụ thể như hình dưới đây:



Hình .: Giao diện giám sát các thông số của 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.

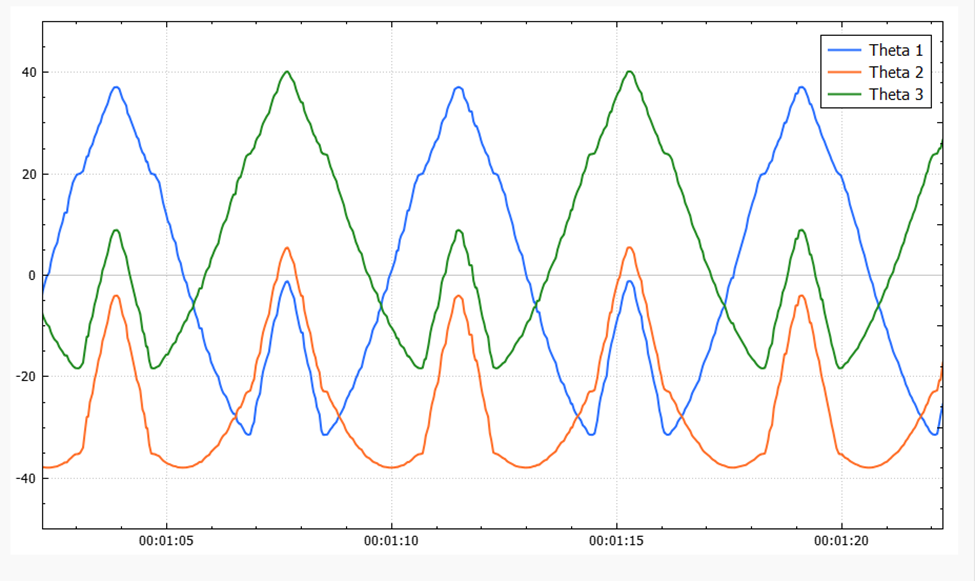


Hình .: Sơ đồ chức năng của các Timer giám sát thông số robot trên PC

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).

#### Giao diện MONITOR

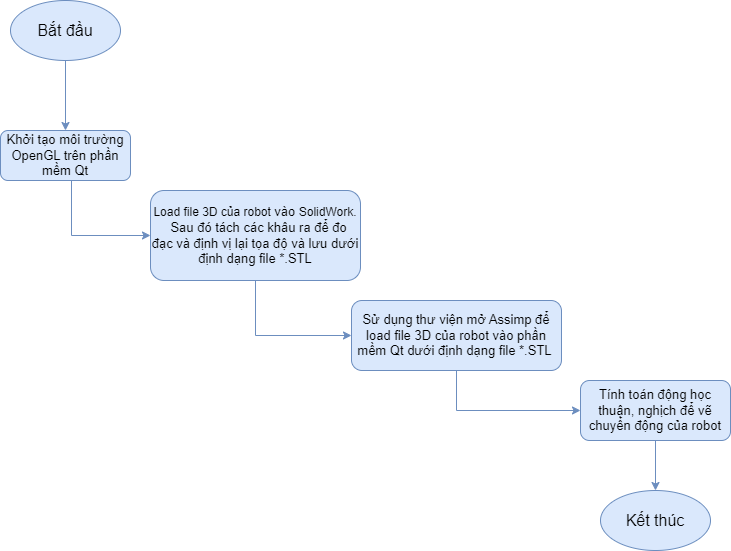
Em sử dụng chức năng vẽ đồ thị của thư viện QCustomPlot hỗi trợ trên Qt. Thư viện này cung cấp các chức năng hiển thị và vẽ đồ thị cho các dữ liệu thời gian thực, phù hợp cho việc thực hiện vẽ các góc quay của các trục robot khi có sự thay đổi và thể hiện lên đồ thị góc theo thời gian (độ/s). Dữ liệu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



Hình .: Giao diện MONITOR

#### Giao diện SIMULATION

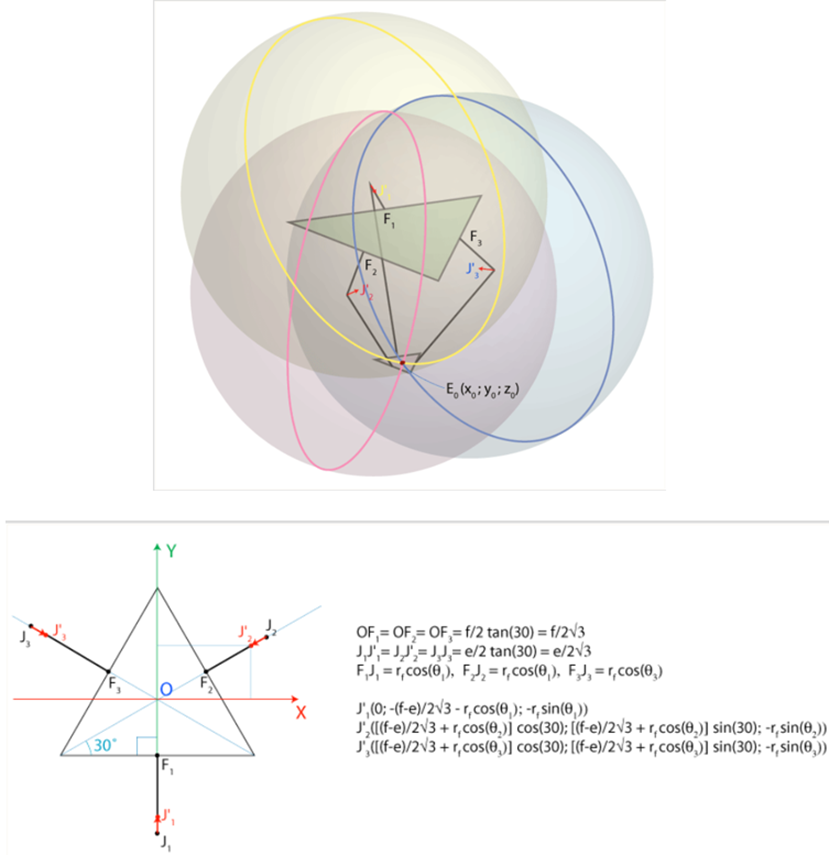
Hệ thống cung cấp một giao diện mô phỏng 3D cho cả robot SCARA và robot DELTA, giúp cho người dùng có thể dễ dàng và trực quan hơn trong quan sát chuyển động của robot bên cạnh mô hình thực. Các bước chính để thực hiện phần mô phỏng này được mô tả qua sơ đồ sau:



Hình .: Sơ đồ quá trình mô phỏng 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



Hình .: Động học thuận robot DELTA

Đặt tọa độ J1’, J2’, J3’ lần lượt là (x1, y1, z1), (x2, y2, z2), (x3, y3, z3)



Ta có

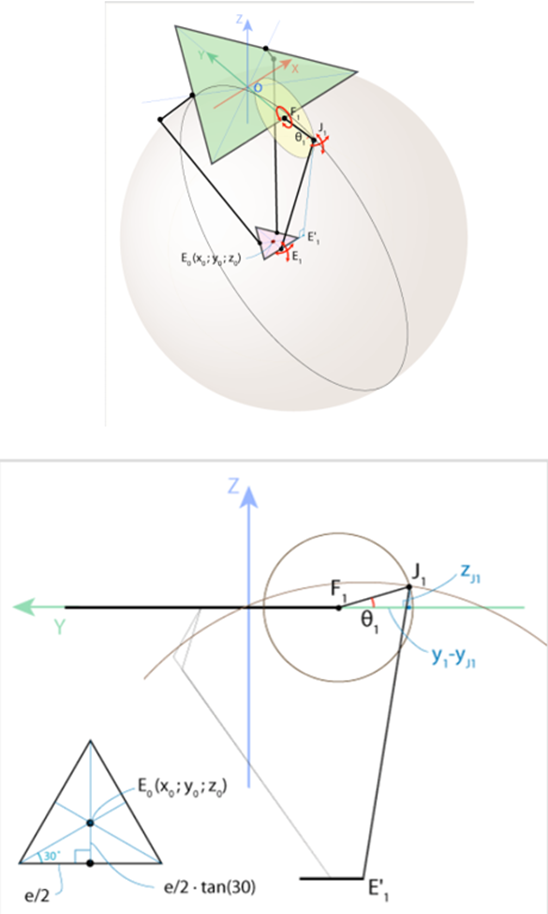


Thế (7) và (8) vào (1):



Ta tính được z0, sau đó thay ngược z = z0 vào lần lượt (7) và (8) ta tính được x0 và y0. Từ đó ta có tọa độ end-effector E0(x0, y0, z0).

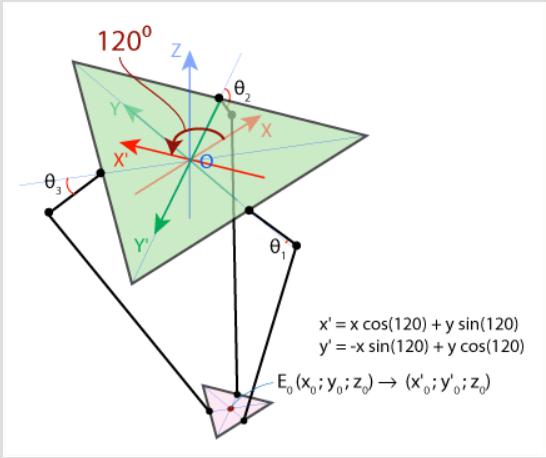
* Lời giải động học nghịch cho robot DELTA



Hình .: Động học nghịch robot DELTA (1)

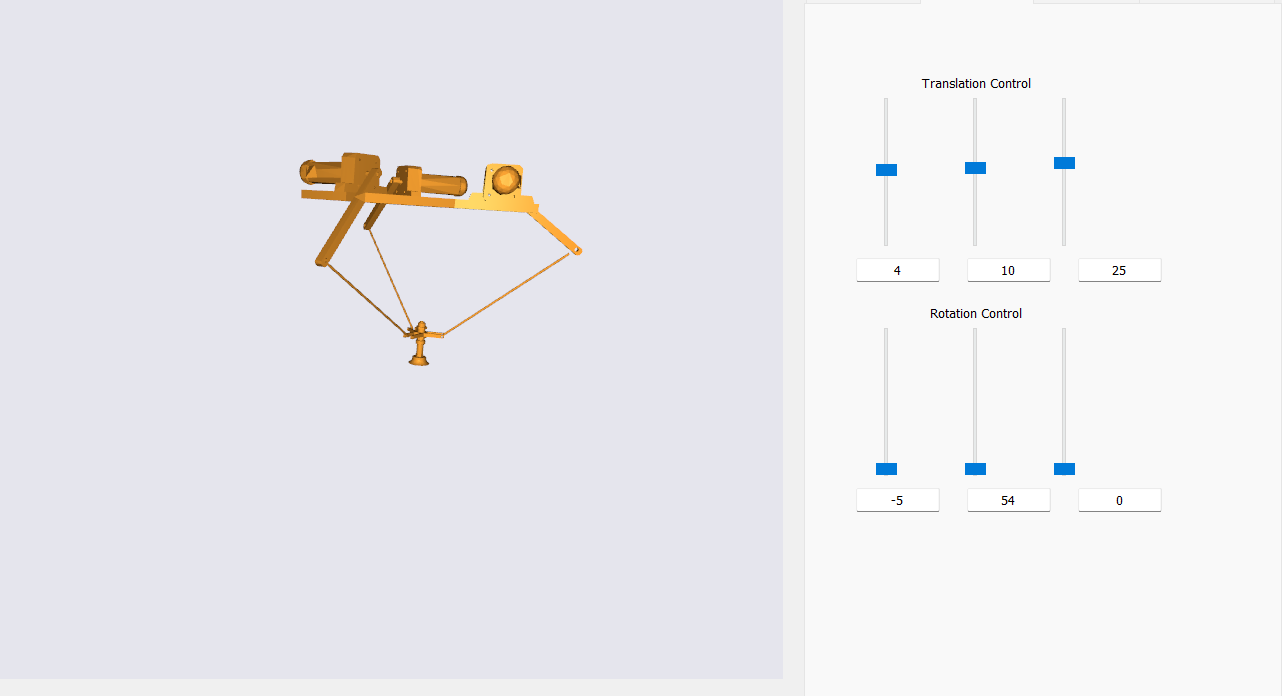


Để tính θ2 và θ3 ta thực hiện xoay mặt phẳng XY quanh trục Z 1200 ngược chiều kim đồng hồ hoặc cùng chiều kim đồng hồ như hình sau:



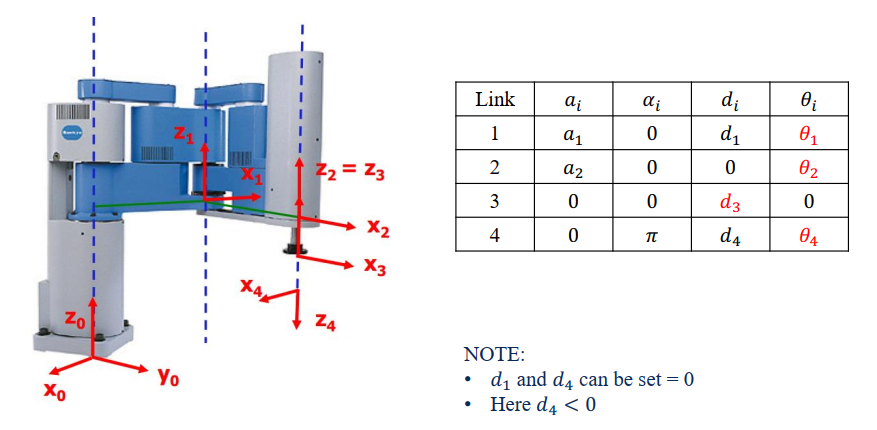
Hình .: Động học nghịch robot DELTA (2)

Và thực hiện việc tính toán với công thức tương tự như θ1.



Hình .: Mô hình mô phỏng robot DELTA

* Robot SCARA
* Lời giải động học thuận của robot SCARA



Hình .: Động học thuận robot SCARA



Trong đó:

Ma trận xoay:



Tọa độ điểm cuối:

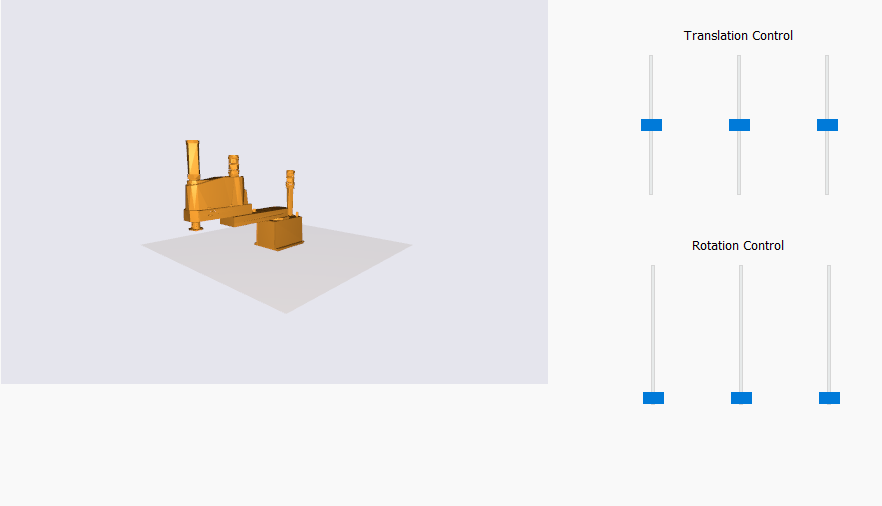


* Lời giải động học nghịch của robot SCARA (theo phương pháp đại số):



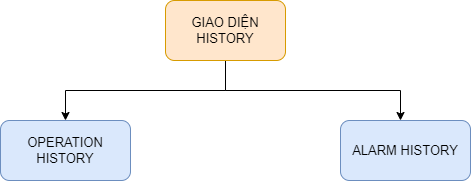
Trong đó:





Hình .: Mô hình mô phỏng robot SCARA

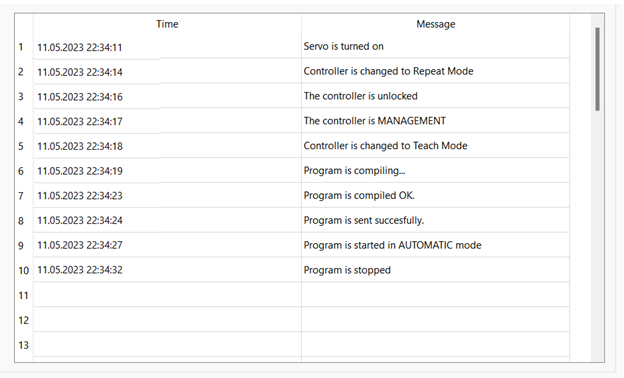
#### Giao diện HISTORY



Hình .: Sơ đồ tổ chức giao diện HISTORY

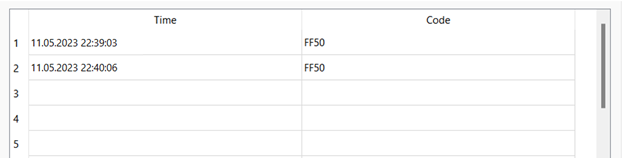
Giao diện History gồm 2 phần:

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



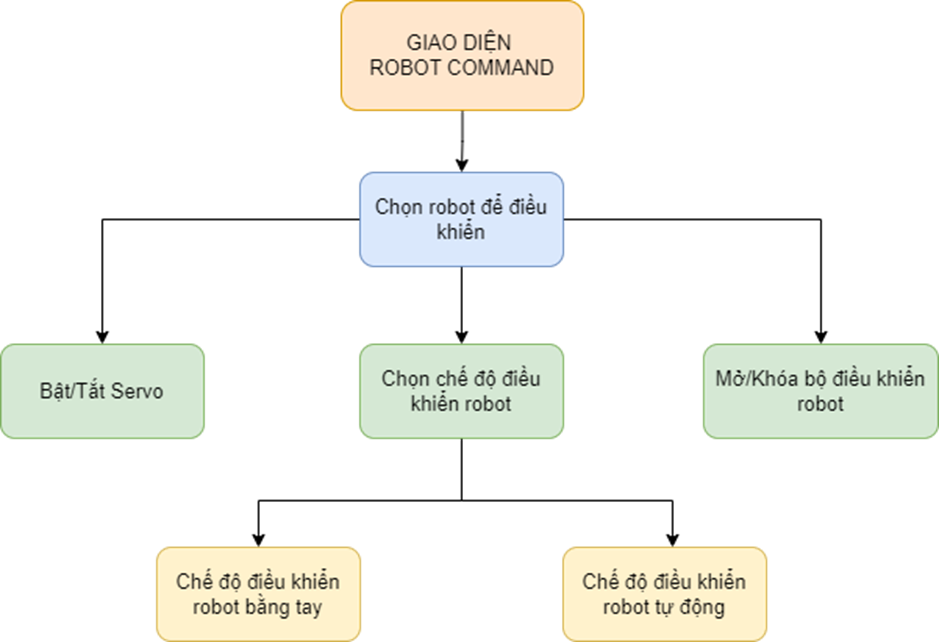
Hình .: Giao diện Operation History

* 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 .: Giao diện Alarm History

### **Giao diện điều khiển robot**



Hình .: Sơ đồ chức năng của giao diện điều khiển robot

Để gửi lệnh từ PC Client đến bộ điều khiển robot, em sử dụng một định dạng khung truyền lệnh điều khiển như sau:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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

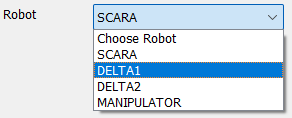
Khung dữ liệu được bắt đầu bằng ký tự bắt đầu STX, tiếp theo là tên lệnh, các đối số được ngăn cách nhau bằng ký tự “,”, kế tiếp là CRC để kiểm tra lỗi và ký tư kết thúc ETX.

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;  } |

Để hiểu cụ thể hơn, giao diện điều khiển robot bao gồm:

* Chọn robot để điều khiển:



Hình .: Giao diện chọn robot để điều khiển

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



Hình .: Giao diện nút Servo

Khung truyền:

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

Tên lệnh: SVON

Argument\_1: 0/1 (tắt/bật)



Hình .: Lệnh bật/tắt Servo

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



Hình .: 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 .: Lệnh chọn chế độ điều khiển

* Mở/khóa bộ điều khiển robot:



Hình .: Giao diện nút khóa bộ điều khiển

Khung truyền:

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

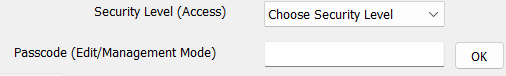
Tên lệnh: LOCK

Argument\_1: 0/1 (mở khóa/khóa)



Hình .: Lệnh mở/khóa bộ điều khiển robot

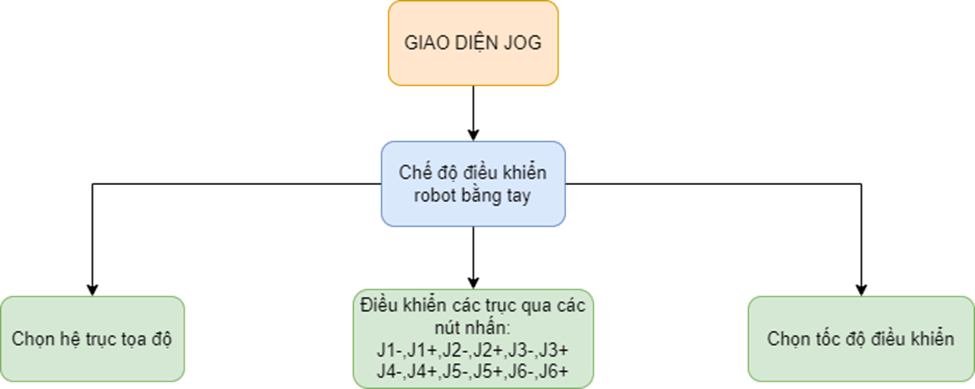
* Chọn chế độ bảo vệ: vận hành, chỉnh sửa, quản lý. Ở 2 chế độ chỉnh sửa và quản lý, phải nhập mật khẩu để có thể sử dụng chế độ này.



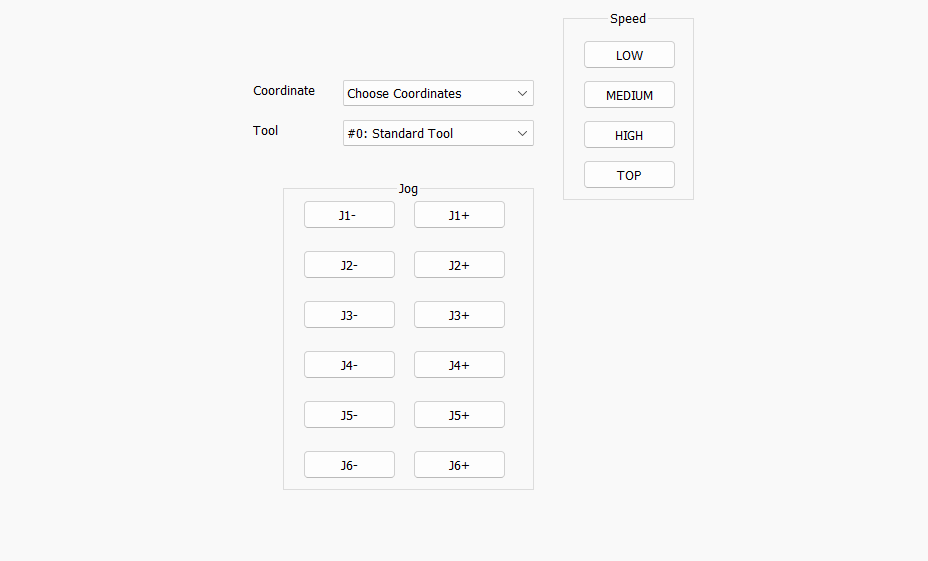
Hình .: Giao diện chọn chế độ bảo vệ

* Và các Tab với các chức năng điều khiển khác nhau như giao diện JOG, PROGRAM, DATA, I/O.

#### Giao diện JOG



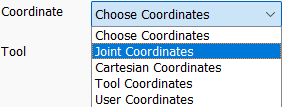
Hình .: Sơ đồ tổ chức giao diện JOG



Hình .: Giao diện JOG

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

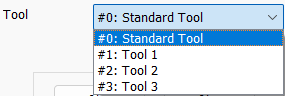
* Chọn hệ tọa độ



Hình .: 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

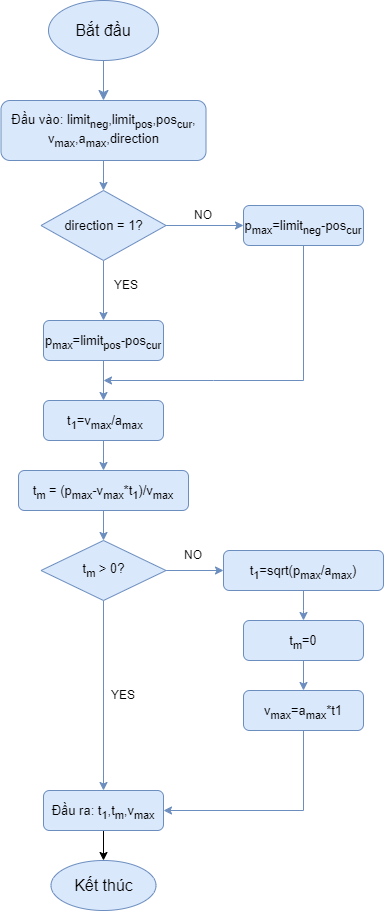


Hình .: Giao diện chọn dụng cụ

* 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 .: 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

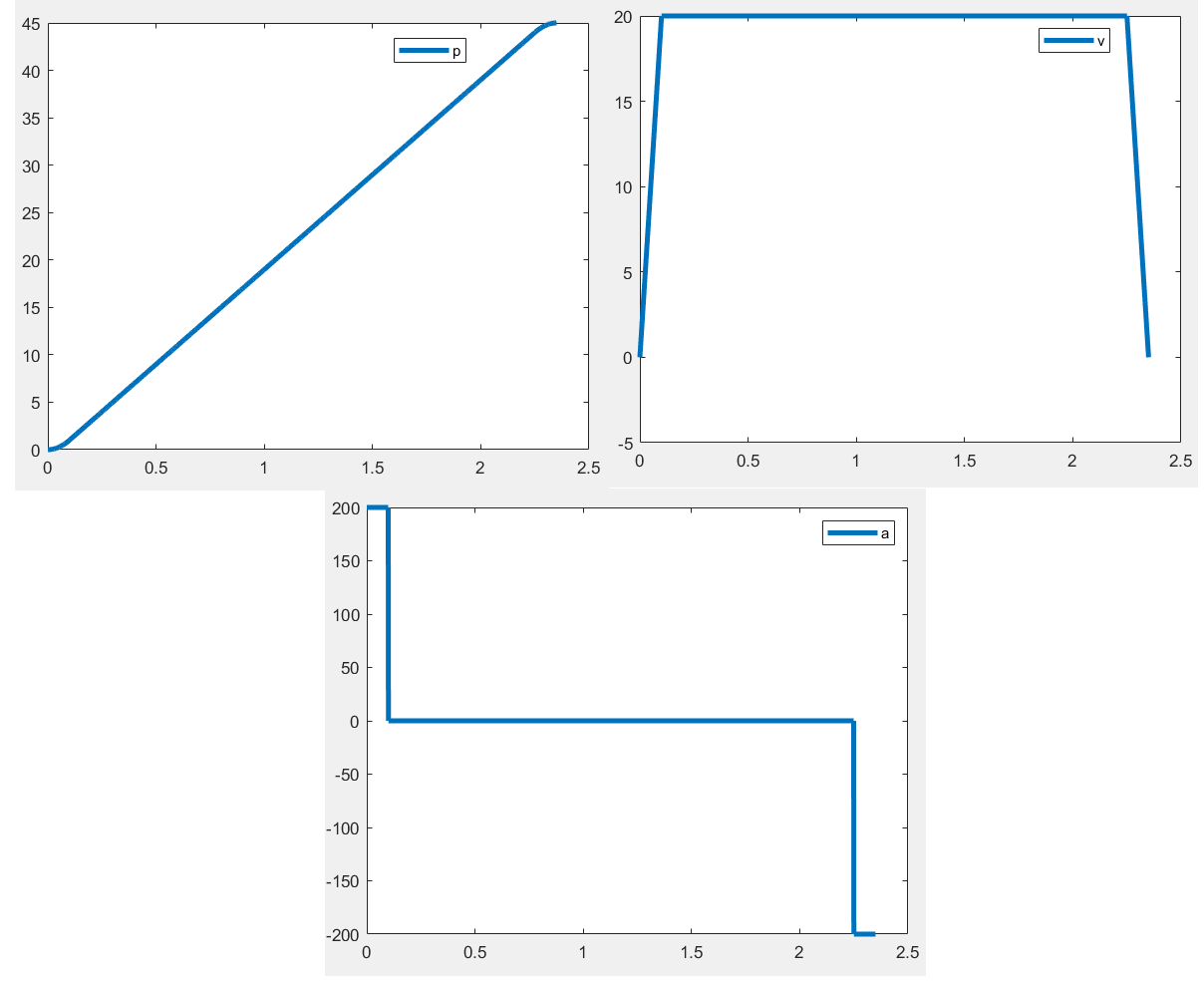
t1: thời gian di chuyển tăng tốc/giảm tốc

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)

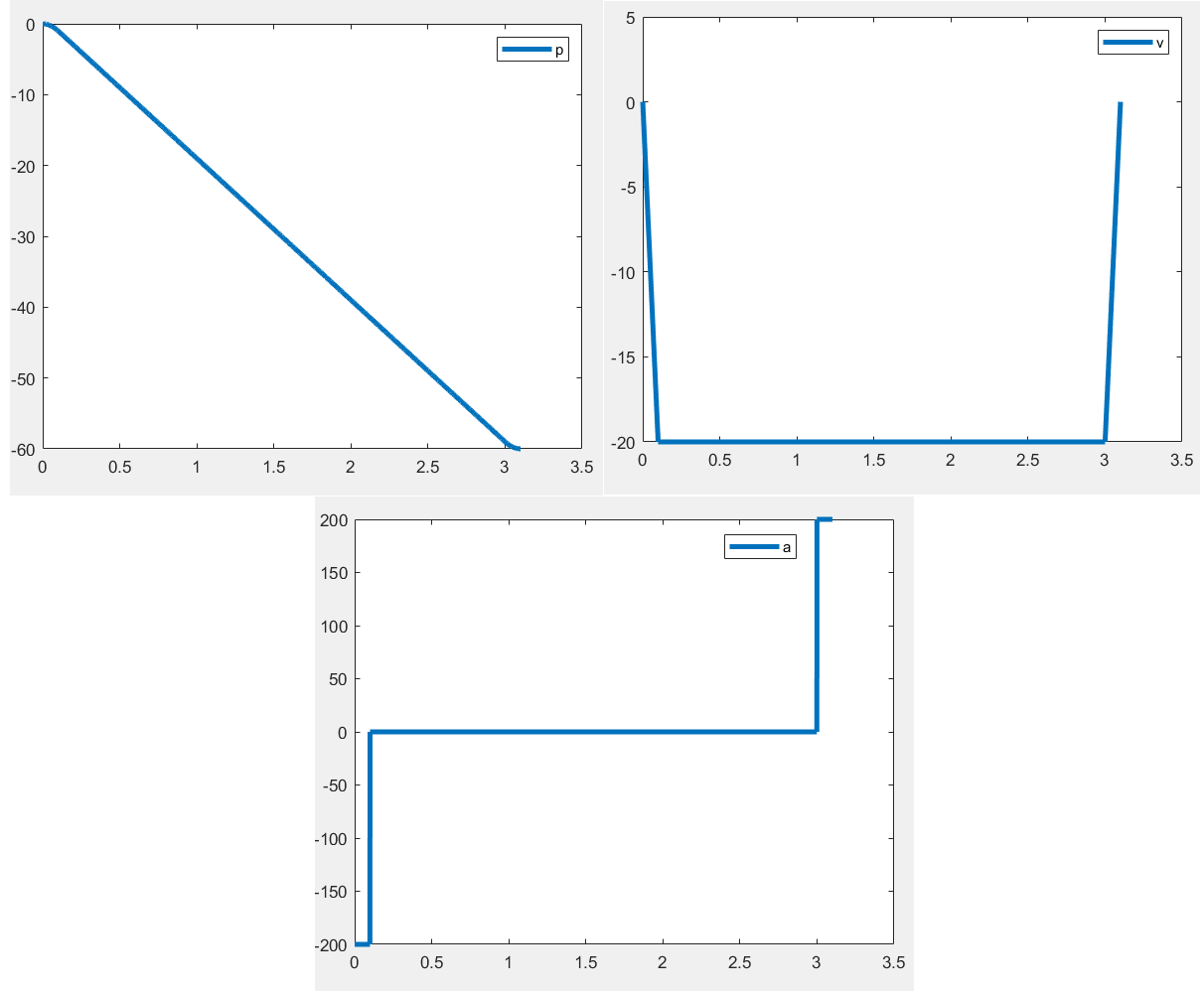
t1 = 0.1, tm = 2.15 (s):



Hình .: 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)

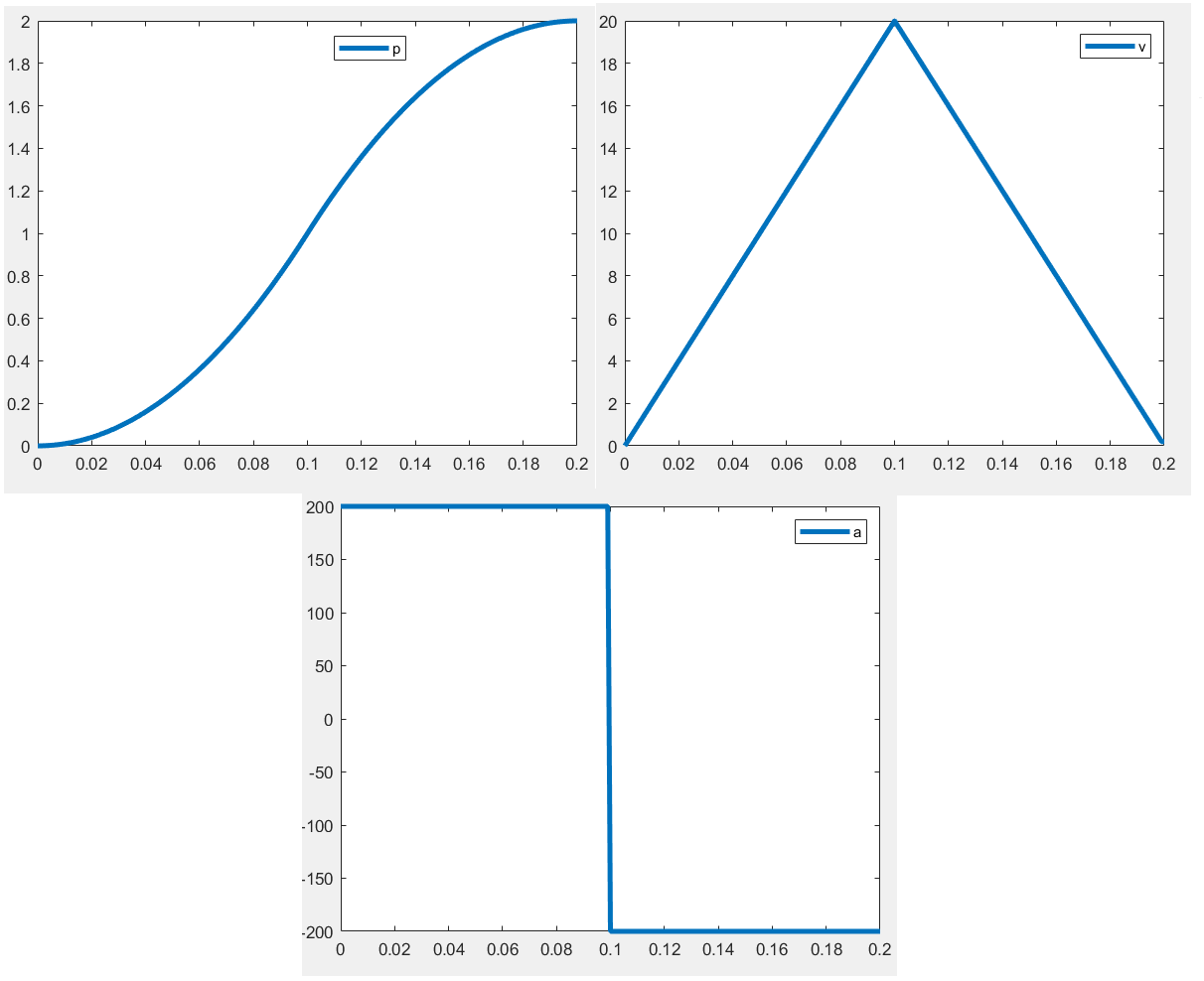
t1 = 0.1, tm = 2.9 (s):



Hình .: 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)

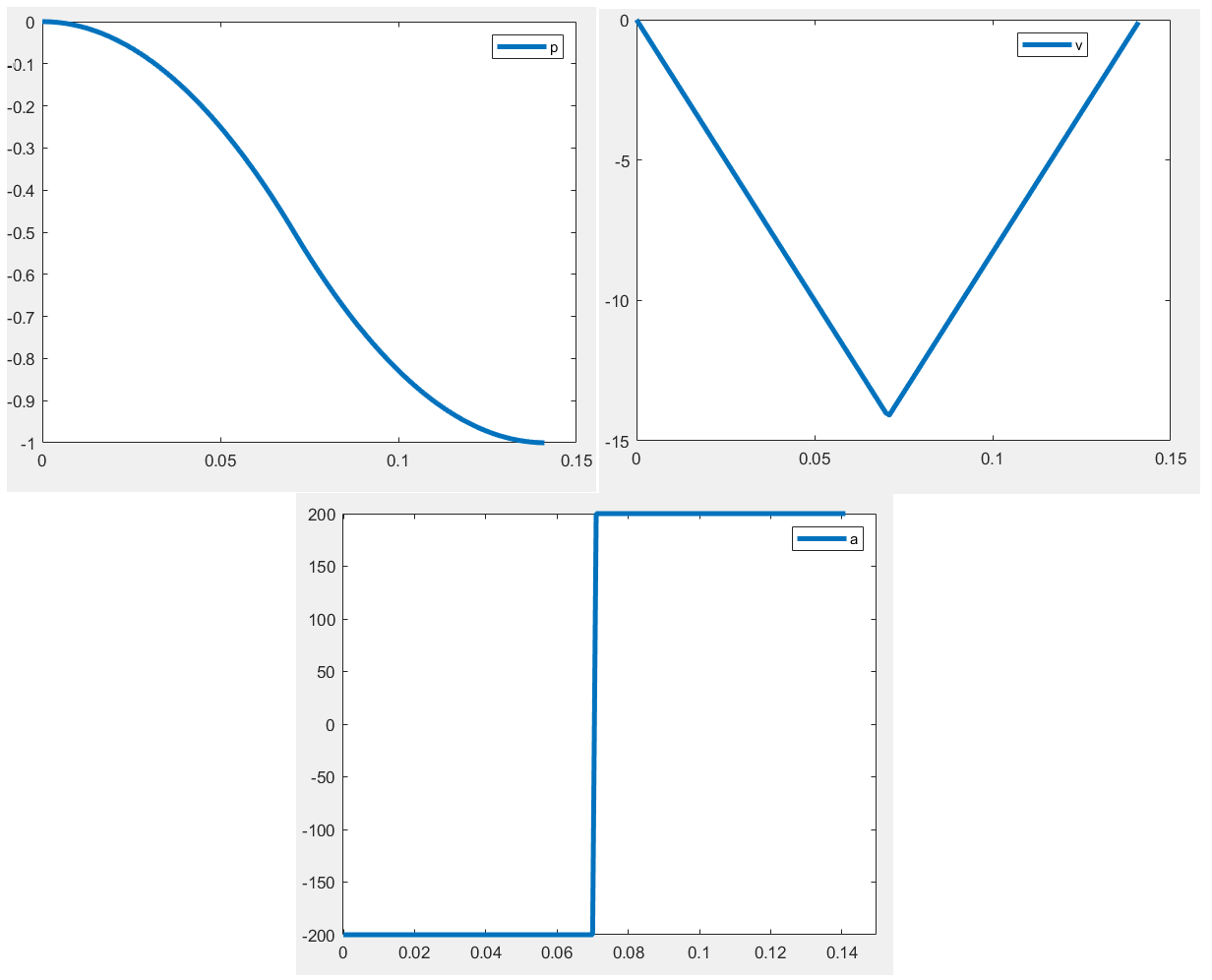
t1 = 0.1, tm = 0 (s):



Hình .: 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 .: 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 .: Lệnh điều khiển trục của robot kèm theo thời gian di chuyển của trục

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 .: 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

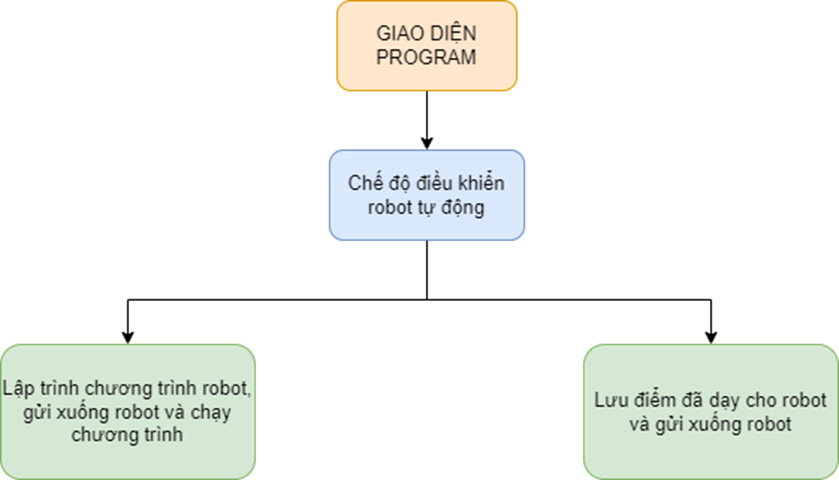
Argumnet\_1: tốc độ điều khiển (0/1/2/3 – thấp/trung bình/cao/rất cao)



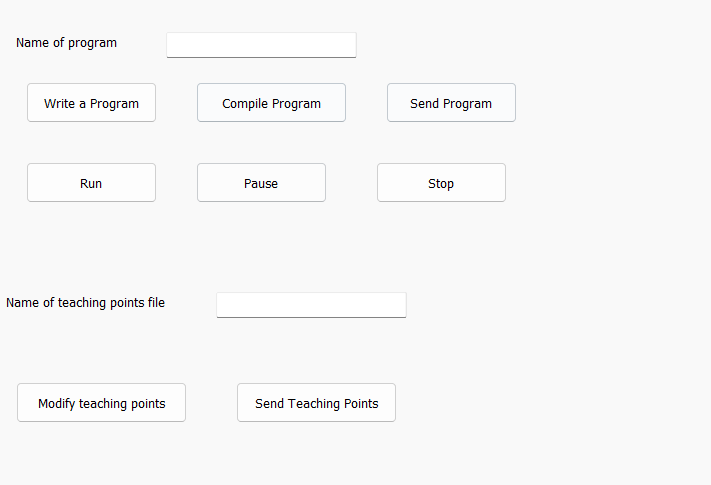
Hình .: Lệnh thay đổi tốc điều khiển các trục của robot

#### Giao diện PROGRAM

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

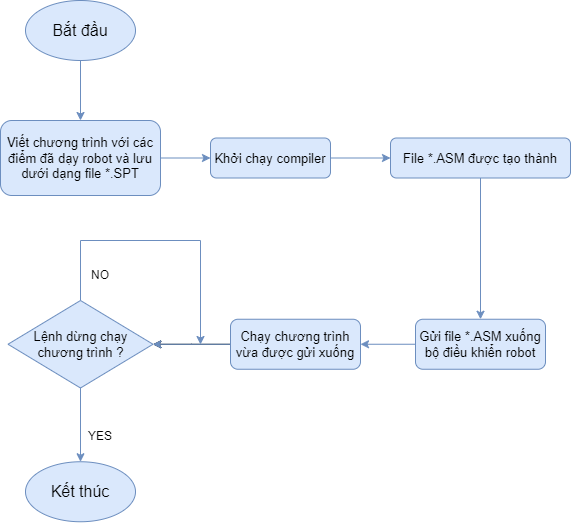


Hình .: Sơ đồ tổ chức giao diện PROGRAM

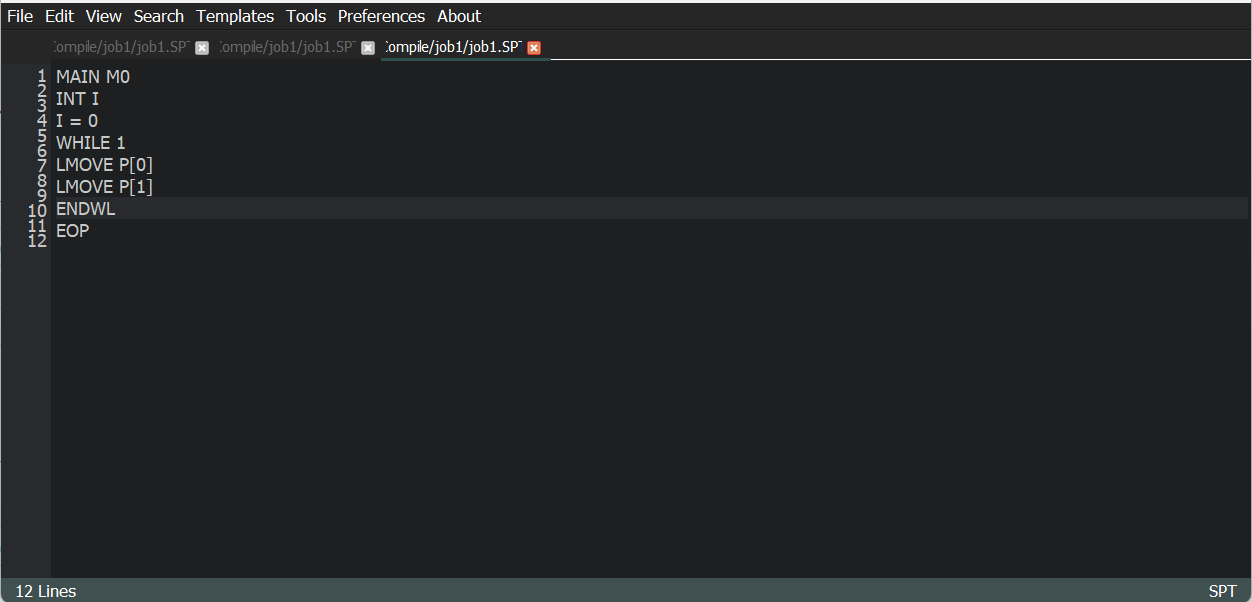


Hình .: Giao diện PROGRAM

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

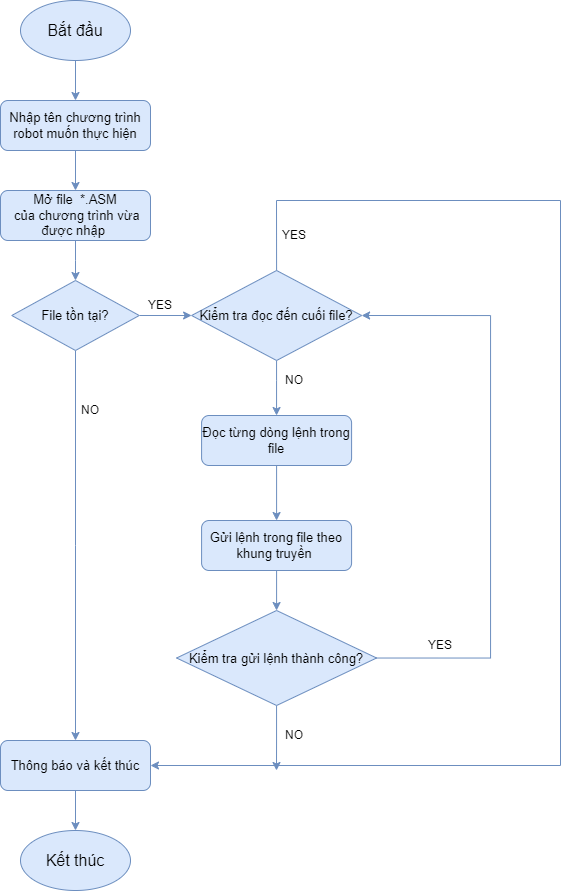


Hình .: Sơ đồ thực hiện một chương trình tự động



Hình .: Giao diện lập trình chương trình robot

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 .: 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 .: Lệnh gửi các dòng lệnh của chương trình 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 .: Lệnh 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 .: 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 .: 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) |

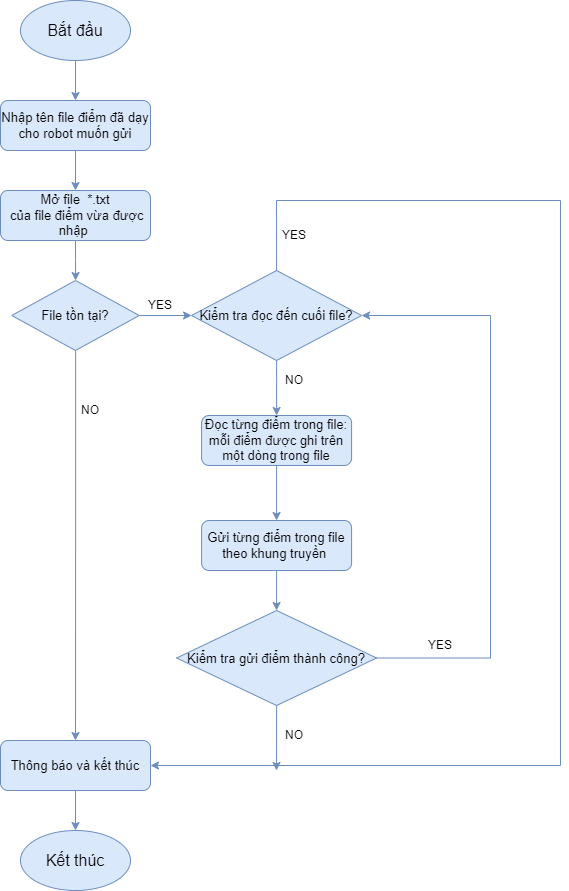
Tên lệnh: SPRG

Argument\_1: Tên chương trình robot đang thực hiện



Hình .: 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 .: 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 .: Lệnh gửi tọa độ của của điểm dạy robot

Khung truyền:

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

Tên lệnh: SEPT

Argument\_1: Tên file điểm được gửi



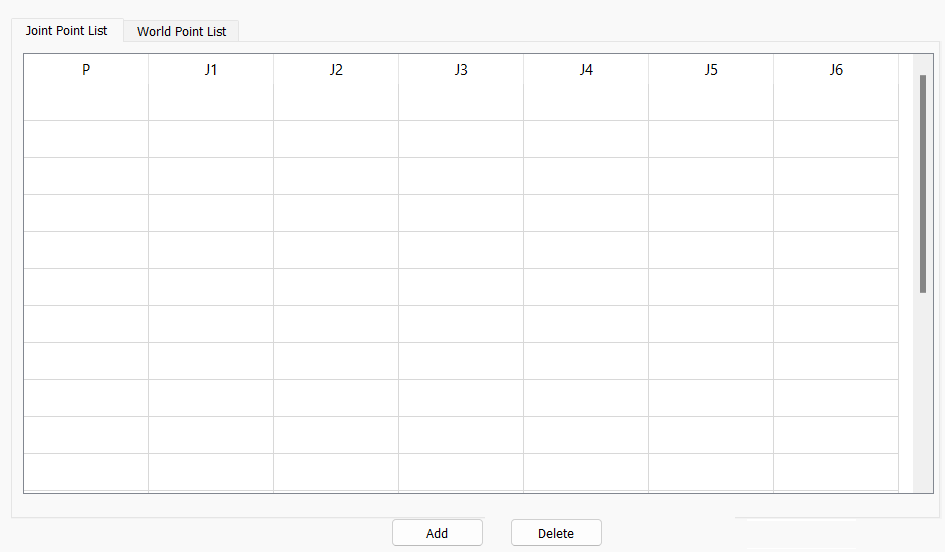
Hình .: Lệnh hoàn thành gửi các điểm dạy robot

#### Giao diện DATA

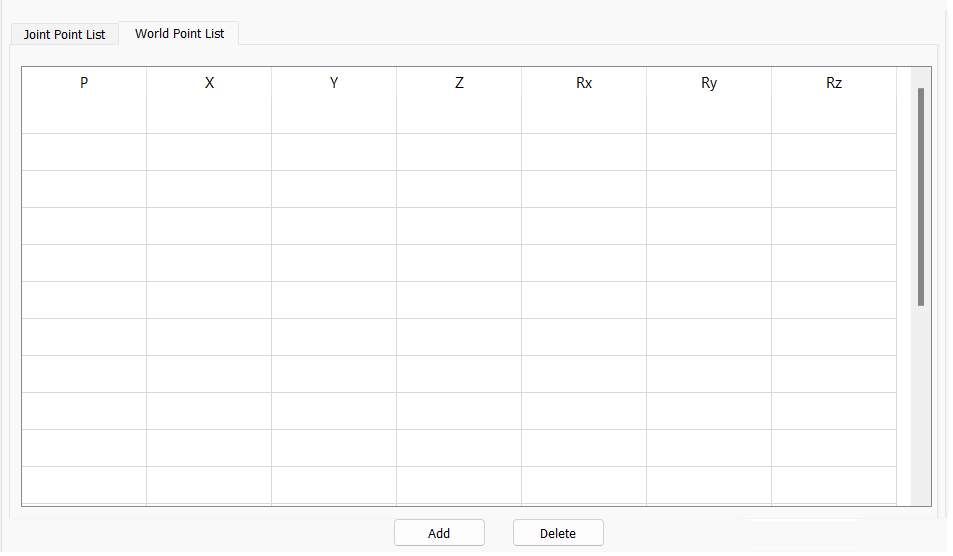
Giao diện này được sử dụng để lữu trữ các điểm dạy robot vào các bảng giá trị.



Hình .: Sơ đồ tổ chức của giao diện DATA



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



Hình .: 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.

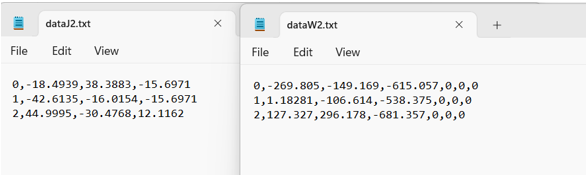
Giá trị tọa độ các điểm dạy robot được lưu vào các file \*.txt theo 2 hệ tọa độ Joint và World theo định dạng:

* Hệ tọa độ Joint:

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

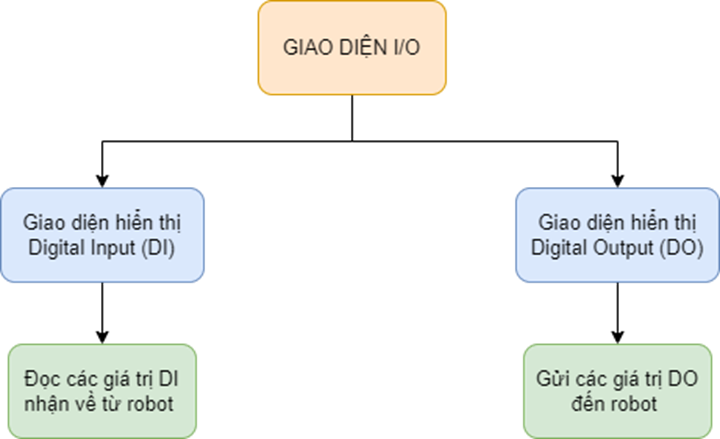
* Hệ tọa độ World:

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



Hình .: Giá trị tọa độ các điểm dạy robot được lưu vào các file \*.txt

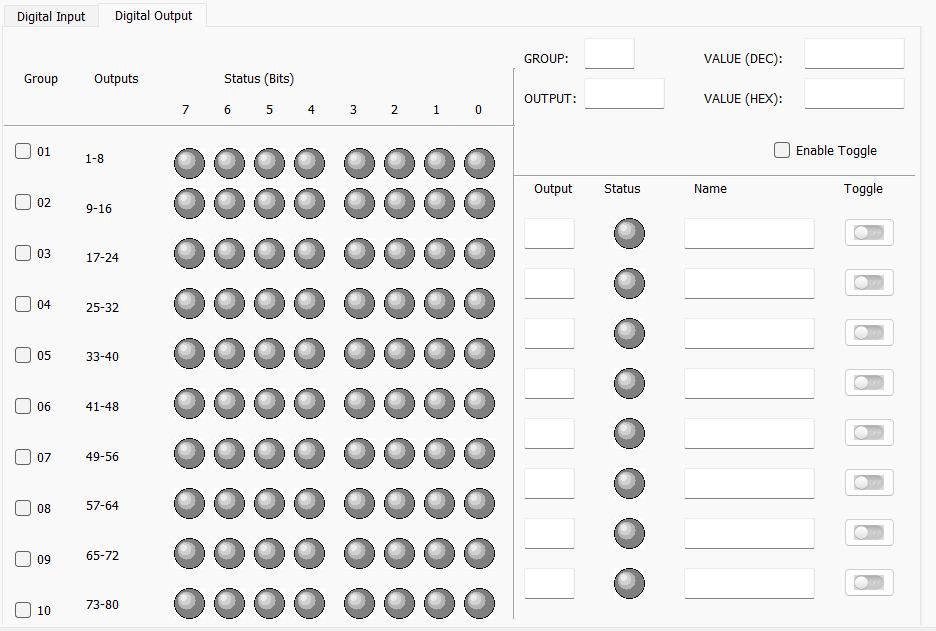
#### Giao diện I/O



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

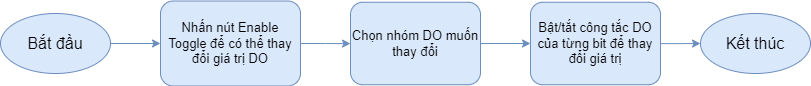


Hình .: Giao diện giám sát Digital Input (DI)



Hình .: 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 .: 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)

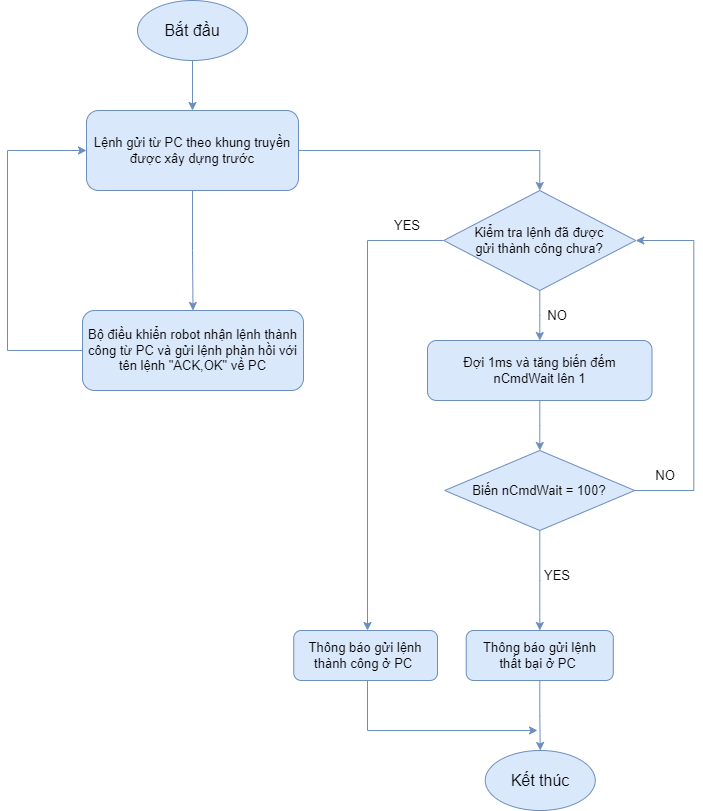
Argument\_2: Thứ tự bit (0-79)

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



Hình .: Lệnh thay đổi giá trị DO

### **Cơ chế bắt tay trong truyền nhận lệnh điều khiển giữa PC Client và bộ điều khiển robot**



Hình .: Sơ đồ cơ chế bắt tay giữa PC Client và bộ điều khiển robot

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

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



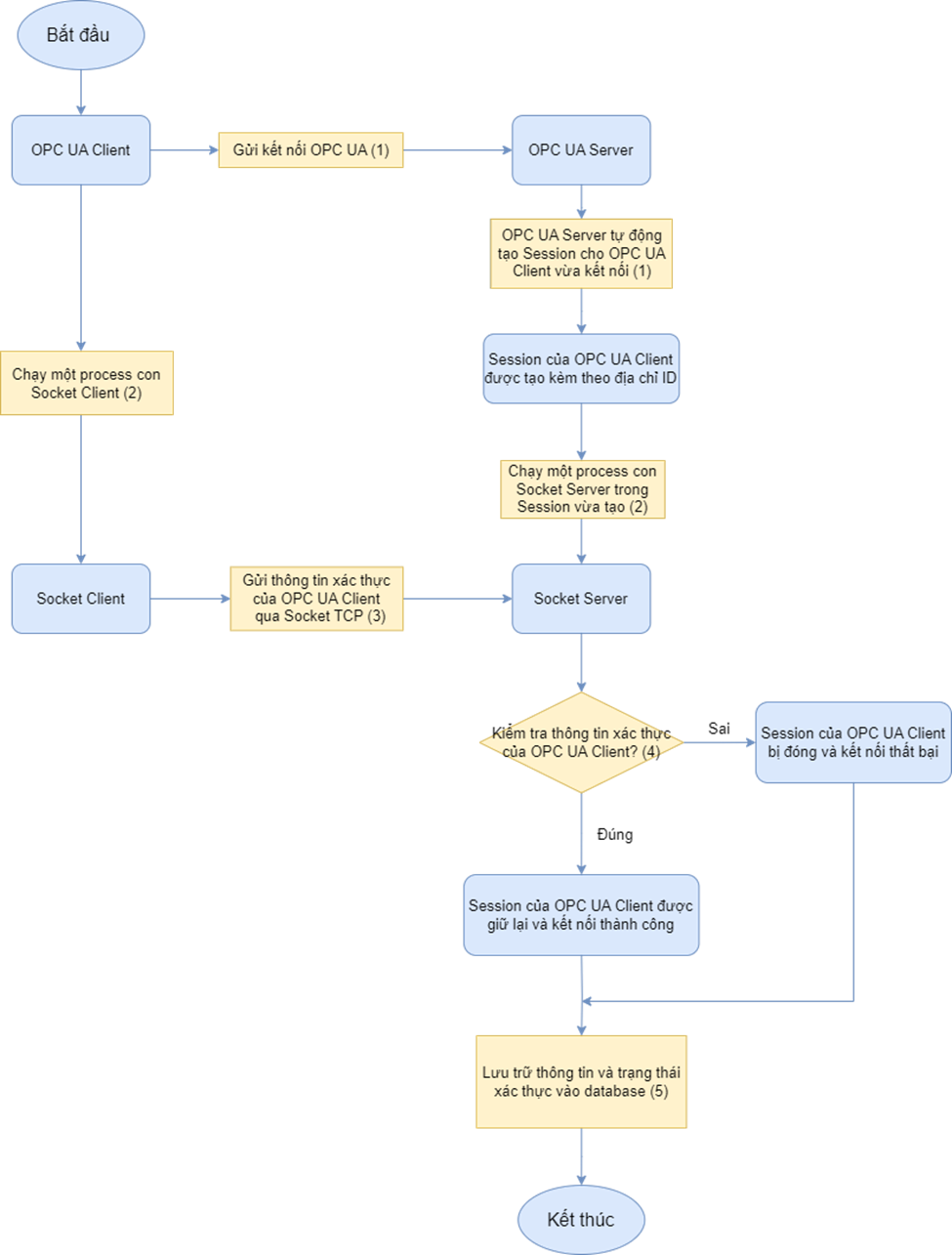
Hình .: Lệnh phản hồi từ bộ điều khiển robot

## **Cải thiện tính năng bảo mật cho hệ thống được thiết kế**

Chuẩn truyền thông OPC UA có tính năng xác thực dựa trên 3 cơ chế chính:

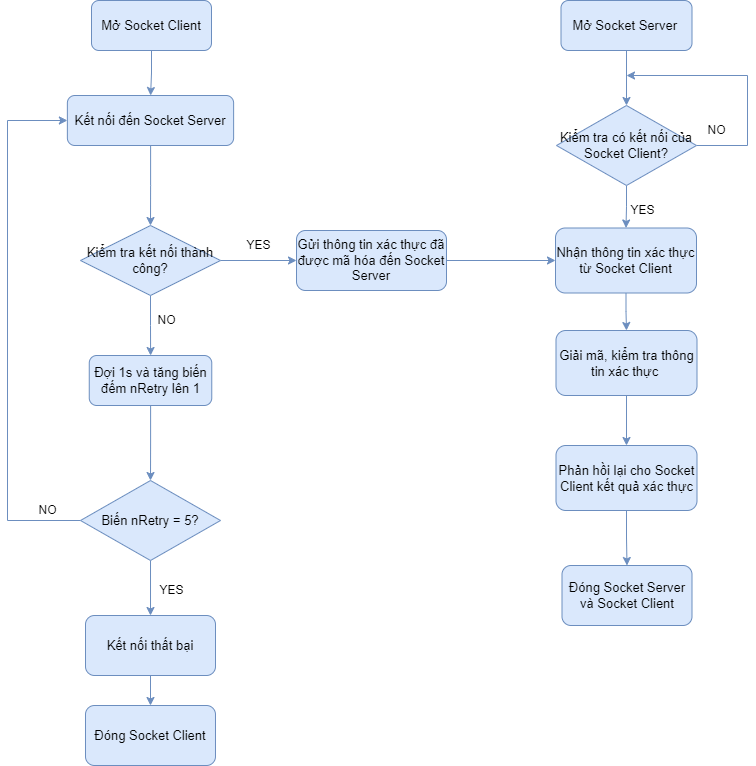
* Đăng nhập vô danh (Anonymous login): bất kỳ OPC UA Client nào cũng có thể kết nối vào OPC UA Server.
* Đăng nhập bằng tên đăng nhập/mật khẩu (Username/Password): OPC UA Client cần có tên đăng nhập/mật khẩu mới có thể kết nối vào OPC UA Server.
* Đăng nhập dựa trên chứng chỉ x509 (x509 certificates): chứng chỉ x509 như một ID chứa các thông tin xác thực của OPC UA Client khi kết nối vào OPC UA Server.

Vì lý do các OPC UA Client được xây dựng trên thư viện mã nguồn mở open62541, một số tính năng xác thực này còn lỗi và chưa hoàn thiện. Vì vậy, em thiết kế một cơ chế xác thực riêng để tiến hành xác thực các OPC UA Client muốn kết nối vào OPC UA Server.



Hình .: Sơ đồ quá trình xác thực giữa OPC UA Client và OPC UA Server

Thứ tự thực hiện các bước chuyển tiếp được đánh số như trong sơ đồ. Ở bước chuyển tiếp thứ 3 thì Socket Server sẽ được mở với địa chỉ bao gồm địa chỉ IP của máy Server và địa chỉ port người dùng tự định nghĩa. Socket Client kết nối đến Socket Server bằng địa chỉ IP và port đó. Kết nối và truyền nhận thông tin xác thực giữa Socket Server và Socket Client được trình bày cụ thể hơn ở sơ đồ sau:



Hình .: Quá trình xác thực giữa Socket Client và Socket Server

Trong đó: Thông tin xác thực bao gồm tên đăng nhập (username), mật khẩu (password) và chìa khóa (salt value).

Sau khi kết nối thành công, thông tin xác thực sẽ lưu được lưu trữ bao gồm Session Id và trạng thái xác thực thành công hay thất bại của OPC UA Client vào database.



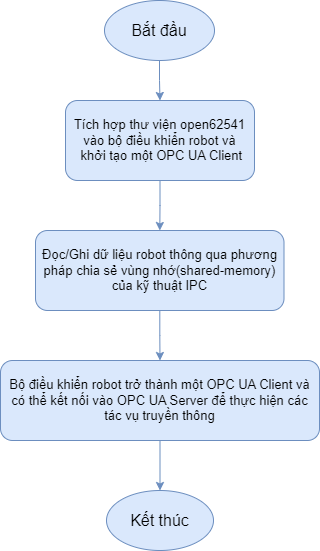
Hình .: File \*.yml lưu trữ thông tin xác thực

## **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.

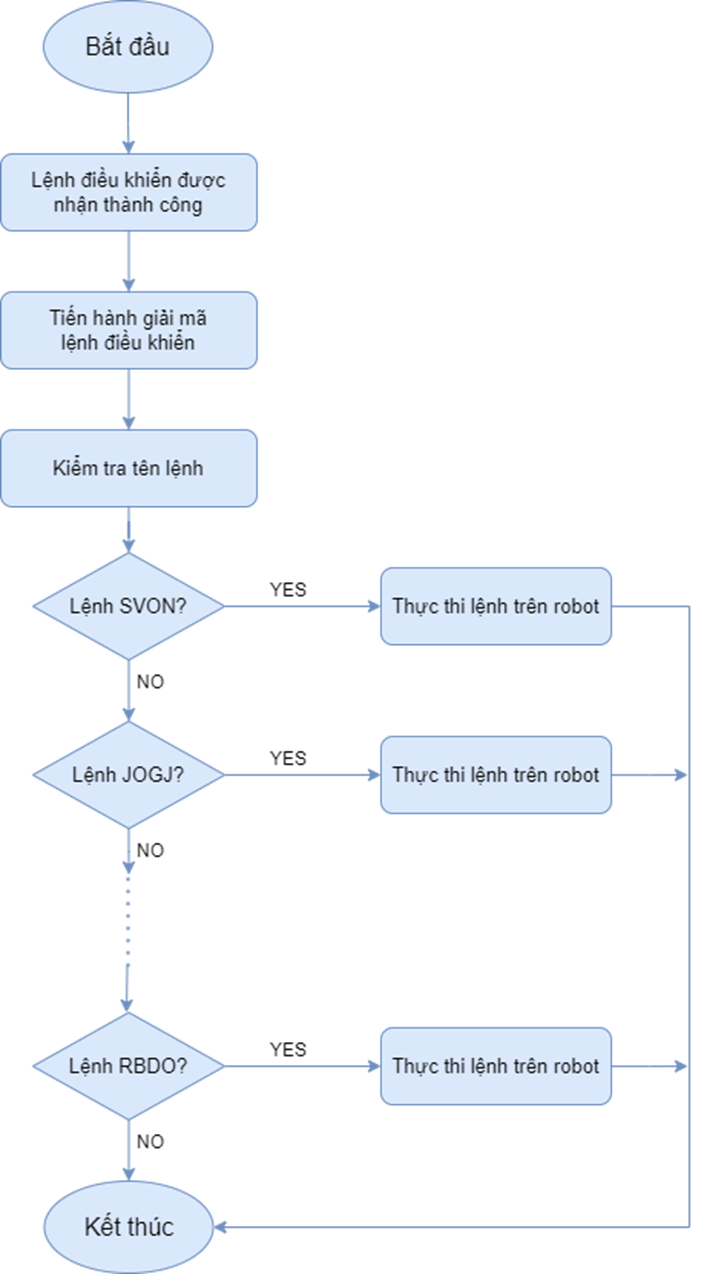
### **Quá trình tích hợp OPC UA vào bộ điều khiển robot**

Đầu tiên, em tích hợp thư viện mở open62541 và khởi tạo một OPC UA Client cho bộ điều khiển robot. Sau đó, em sử dụng phương pháp chia sẻ vùng nhớ (shared-memory) của kỹ thuật IPC để truy cập đọc/ghi dữ liệu của bộ điều khiển robot. Phương pháp 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 thực hiện mỗi khi bộ điều khiển robot được trở thành một OPC UA Client kết nối vào OPC UA Server cho đến khi ngắt kết nối. Quá trình tích hợp được mô tả khái quát qua sơ đồ sau:



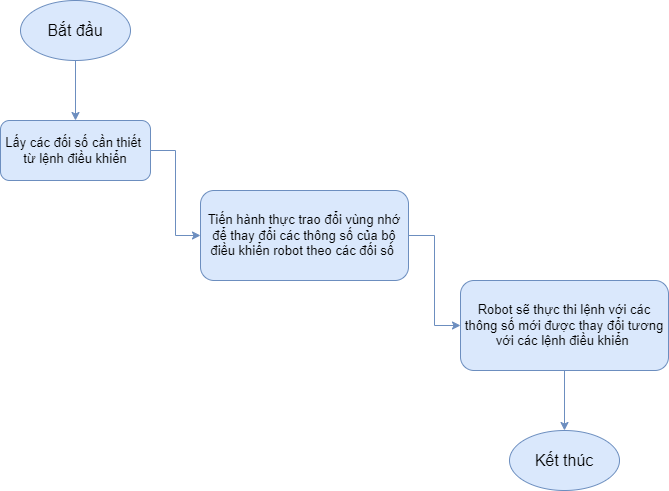
Hình .: Sơ đồ quá trình tích hợp OPC UA vào bộ điều khiển robot

### **Xử lý lệnh điều khiển ở bộ điều khiển và thực thi lệnh trên robot**



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

Phần thực thi lệnh sẽ được trình bày cụ thể hơn qua sơ đồ sau:



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

Qua nghiên cứu và tìm hiểu, em thấy việc sử dụng Timer là khả thi nhất trong việc nhận lệnh từ PC Client. Vì vậy mỗi khi nhận được lệnh, Timer sẽ tắt và sau khi thực hiện xong lệnh thì Timer mới được bật lại để thực hiện chờ để nhận lệnh mới. Quá trình này được lặp lại trong quá trình hoạt động của hệ thống. Thời gian lấy mẫu của Timer sẽ được em tùy chỉnh cho phù hợp.

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

Ngoài chức năng nhận lệnh điều khiển từ PC Client 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.

# **KẾT QUẢ THỰC HIỆN**

## **Thực nghiệm hệ thống**

Em tiến hành thực nghiệm hệ thống giám sát và điều khiển nhiều robot dựa trên 2 phương án kết nối của hệ thống được trình bày dưới đây:

* Phương án 1: Kết nối có dây Ethernet thông qua Ethernet Hub:



Hình .: Sơ đồ kết nối của phương án 1

* Phương án 2: Kết nối không dây qua Internet (Server và các Client kết nối vào chung một lớp mạng)



Hình .: Sơ đồ kết nối của phương án 2

Khi thực nghiệm lên hệ thống, em chia ra 3 vùng làm việc chính của người dùng có sơ đồ như sau:



Hình .: Sơ đồ các vùng làm việc của người dùng

Trong đó:

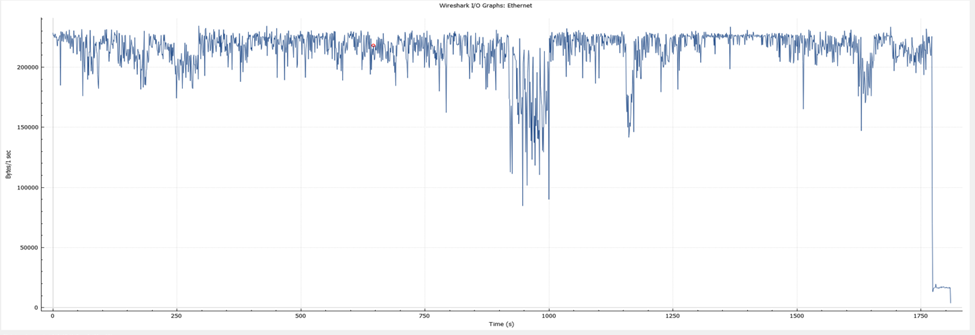
* Vùng #1: Giao diện giám sát, điều khiển chính hệ robot
* Vùng #2: Hệ robot thật được giám sát và điều khiển
* Vùng #3: Giao diện mô phỏng trực tuyến robot

## **Kết quả đạt được**

Sau khi thực nghiệm cả 2 phương án trên, em nhận thấy ở phương án 1 hệ thống hoạt động ổn định hơn, truyền thông dữ liệu được liên tục và thông suốt. Ngược lại ở phương án 2, vì em chỉ thực nghiệm hệ thống trên một mạng Wifi thông thường (Typical Wireless Network), không phải Wifi công nghiệp (Industrial Wireless Network) nên dữ liệu được truyền thông có trễ và không được liền mạch (có thể bị mất dữ liệu trong quá trình truyền thông) do tính chất và tốc độ mạng Wifi thông thường không được đảm bảo. Vì vậy, các kết quả dưới đây được em thu thập và đánh giá ở phương án 1.

Các kết quả đạt được:

* Xây dựng được một hệ thống giám sát và điều khiển nhiều robot sử dụng chuẩn truyền thông OPC UA với các chức năng giám sát, mô phỏng và gửi lệnh điều khiển robot. Bên cạnh đó, hệ thống còn có tính năng bảo mật và cơ sở dữ liệu.
* Hệ thống hoạt động ổn định khi ứng dụng lên mô hình robot thật.
* Đo lường và đánh giá các thông số của chuẩn truyền thông OPC UA trong quá trình hoạt động của hệ thống



|  |  |
| --- | --- |
| Time(s) | 30:10 |
| Interface | Ethernet |
| Dropped packets | 0 (0.0%) |

Bảng .: Bảng thông số đo lường và đánh giá

Từ các kết quả đánh giá trên, việc thực nghiệm hệ thống robot sử dụng chuẩn truyền thông OPC UA đã có thể đạt được các yêu cầu về hiệu năng và độ ổn định.

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **Kết luận**

Ý nghĩa của luận văn: Luận văn đã xây dựng và phát triển được một hệ thống giám sát và điều khiển nhiều robot dựa trên chuẩn truyền thông OPC UA và đã được ứng lên hệ thống robot thật như mục tiêu đã đặt ra của đề tài.

* Ưu điểm: Với những tính năng ưu việt của OPC UA, em đã xây dựng được một cấu trúc hệ thống giám sát và điều khiển nhiều robot bao gồm có OPC UA Server, OPC UA Client và tích hợp OPC UA vào các robot. Hệ thống này khắc phục được tình trạng bất đồng bộ trong vận hành sản xuất của các nhà máy có nhiều robot từ các hãng khác nhau.
* Nhược điểm: Hệ thống vẫn chưa khai thác hết các tính năng của OPC UA do thư viện mở open62541 còn khá nhiều lỗi và chưa được ứng dụng lên nhà máy có nhiều robot hơn để thực nghiệm và kiểm tra hiệu năng và tính ổn định của hệ thống.

## **Hướng phát triển**

Hệ thống chỉ mới bước đầu thực nghiệm trên quy mô nhỏ gồm 2-3 robot trong thời gian nhất định. Sau đây là một số hướng phát triển đề tài trong tương lai:

* Mở rộng hệ thống có nhiều robot và thực nghiệm lâu hơn để có thể đánh giá thêm về hiệu năng và tính ổn định của hệ thống được xây dựng.
* Bổ sung thêm các tính năng cho các phần mềm của hệ thống.
* Phát triển hệ thống lên các tầng quản lý cao hơn như tầng MES, ERP và đưa dữ liệu lên Cloud và thiết kế Web.

**TÀI LIỆU THAM KHẢO**

[1] OPC Foundation, OPC Technologies – Unified Architecture (UA).

<https://opcfoundation.org/about/opc-technologies/opc-ua/>

[2] OPC UA SDK for .NET, Development Guides. <https://docs.traeger.de/en/software/sdk/opc-ua/net/start#opc-ua-sdkfor-net>

[3] Open62541, open62541 release 1.3 Documentation.

<https://www.open62541.org/doc/open62541-1.3.pdf>

[4] VAS Corporation, XMC-ROBOT EtherCAT Master Robot Controller User’s Manual

[5] OPC Foundation, Analyzing OPC UA Communications with Wireshark. <https://opcconnect.opcfoundation.org/2017/02/analyzing-opc-uacommunications-with-wireshark/>

[6] Assimp, Model Loading. <https://learnopengl.com/ModelLoading/Assimp>

[7] H. -I. Lin and Y. -C. Hwang, “Integration of Robot and IIoT over the OPC Unified Architecture,” 2019 International Automatic Control Conference (CACS), Keelung, Taiwan, 2019, pp. 1-6.

[8] Pauker, F., Ayatollahi, I., & Kittl, B. (2014). OPC UA for machine tending industrial robots. In Proc. of the 2nd Intl. Conf. on Advances In Robotics Engineering (pp. 79–83). <http://hdl.handle.net/20.500.12708/67093>

[9] Delta robot kinematics. <https://hypertriangle.com/~alex/delta-robot-tutorial/>

[10] Simple Snippets, Creating MS SQL Database & Tables & adding Connection String to web.config file in ASP.NET. <https://simplesnippets.tech/creating-ms-sql-database-tables-adding-connection-string-to-web-config-file-in-asp-net/>