

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

DƯƠNG NGUYỄN KHANG

LUẬN VĂN TỐT NGHIỆP
ỨNG DỤNG CÔNG NGHỆ ĐÁM MÂY VÀO VIỆC QUẢN LÝ
VÀ LẬP LỊCH ĐIỀU KHIỂN ROBOT TỪ XA

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

DƯƠNG NGUYỄN KHANG - 1911330

LUẬN VĂN TỐT NGHIỆP
ỨNG DỤNG CÔNG NGHỆ ĐÁM MÂY VÀO VIỆC QUẢN LÝ
VÀ LẬP LỊCH ĐIỀU KHIỂN ROBOT TỪ XA
CLOUD-BASED APPLICATION FOR REMOTE ROBOT
MANAGEMENT AND SCHEDULING
KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

GIẢNG VIÊN HƯỚNG DẪN
THS. TRẦN QUỐC TIẾN DŨNG

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

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN - ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

TP. HCM, ngày....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ HƯỚNG DẪN**

Tên luận văn:

**ỨNG DỤNG CÔNG NGHỆ ĐÁM MÂY VÀO VIỆC QUẢN LÝ
VÀ LẬP LỊCH ĐIỀU KHIỂN ROBOT TỪ XA
CLOUD-BASED APPLICATION FOR REMOTE ROBOT
MANAGEMENT AND SCHEDULING**

Nhóm Sinh viên thực hiện:

Cán bộ hướng dẫn:

Dương Nguyên Khang

1911330

ThS. Trần Quốc Tiến Dũng

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:

2. Về nội dung luận văn:

3. Về tính ứng dụng:

4. 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:

Dương Nguyên Khang:...../10

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

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN – ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

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:

Dương Nguyên Khang

1911330

Đánh giá Luận văn

5. 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:

6. Về nội dung luận văn:

7. Về tính ứng dụng:

8. 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:

Dương Nguyên Khang:...../10

Người nhận xét
(Ký tên và ghi rõ họ tên)

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN - ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

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: Dương Nguyên Khang – 1911330	
Nội dung đề tài: <p>Thiết kế một hệ thống phần mềm giám sát, mô phỏng và lập lịch hoạt động cho robot hoàn chỉnh đến từ sự kết hợp giữa các phần mềm trên mạng cục bộ và ứng dụng Web trên môi trường đám mây. Hệ thống có thể giám sát robot từ xa và lập lịch cho cánh tay robot thực hiện các nhiệm vụ nhất định. Ứng dụng vào một cánh tay robot ba bậc tự do để đánh giá kết quả.</p>	
Kế hoạch thực hiện:	
Công việc	Thời gian
Xây dựng và thiết kế giao diện trang Web giám sát	01/01/2023 – 15/01/2023

Phát triển các phần mềm OPC UA Server, OPC UA Client để giao tiếp với robot	16/01/2023 – 15/02/2023
Xây dựng cơ sở dữ liệu SQL	16/02/2023 – 28/02/2023
Tìm hiểu và triển khai cơ sở dữ liệu, Web App lên môi trường điện toán đám mây	01/03/2023 – 15/03/2023
Thiết kế cấu trúc để truyền đạt dữ liệu giữa Web App trên môi trường đám mây với phần mềm cục bộ.	16/03/2023 – 21/03/2023
Nghiên cứu phần cứng, điều khiển robot thực hiện các nhiệm vụ cơ bản	22/03/2023 – 15/04/2023
Đo đạc cánh tay robot, xây dựng phần mềm mô phỏng cho robot và các tính năng cơ bản của một phần mềm mô phỏng	16/04/2023 – 30/04/2023
Phát triển tính năng lập lịch hoạt động cho robot	01/05/2023 – 07/05/2023
Thực nghiệm kết hợp giữa phần cứng và phần mềm, cải thiện hoạt động của hệ thống	08/5/2023 – 18/05/2023

Xác nhận của Cán bộ hướng dẫn (Ký tên và ghi rõ họ tên)	TP. HCM, ngày....thángnăm..... 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.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	20
1.1 Mục đích và mục tiêu của luận văn.....	20
1.2 Đối tượng và phạm vi nghiên cứu	20
1.2.1 Đối tượng nghiên cứu	20
1.2.2 Phạm vi nghiên cứu.....	20
1.3 Cấu trúc của luận văn.....	21
CHƯƠNG 2. TỔNG QUAN VỀ ROBOT VÀ HỆ THỐNG SCADA	22
2.1 Khái niệm về cánh tay robot công nghiệp	22
2.2 Ứng dụng của công nghệ đám mây trong giám sát và thu thập thông tin.....	23
2.3 Động học robot	28
2.3.1 Một số khái niệm	28
2.3.2 Biểu diễn Denavit – Hartenberg của bài toán động học robot.....	30
2.3.3 Phương trình động học thuận của cánh tay robot 3 bậc tự do	32
2.3.4 Phương trình động học nghịch của cánh tay robot 3 bậc tự do	34
2.3.5 Hoạch định quỹ đạo	36
CHƯƠNG 3. TỔNG QUAN VỀ CẤU TRÚC HỆ THỐNG	38
CHƯƠNG 4. PHẦN CỨNG VÀ MẠCH ĐIỀU KHIỂN	40
4.1 Cánh tay robot 3 bậc tự do	40
4.2 Mạch điều khiển và các thành phần điện.....	40
4.3 Phần mềm trên board điều khiển	42
CHƯƠNG 5. PHẦN MỀM MÔ PHỎNG VÀ GIÁM SÁT ROBOT	45
5.1 Phần mềm OPC UA Server	45
5.1.1 Giới thiệu về thư viện OPC UA Client & Server SDK	45
5.1.2 Cấu trúc OPC UA Server đã thiết kế	45
5.1.3 Lưu trữ dữ liệu vào cơ sở dữ liệu SQL theo mô hình điện toán đám mây	
46	
5.2 Phần mềm mô phỏng và giám sát robot.....	48

5.2.1	Giới thiệu về OpenGL và thư viện Assimp.....	48
5.2.2	Thiết kế phần mềm mô phỏng.....	50
5.2.3	Chức năng giám sát và mô phỏng	56
CHƯƠNG 6. WEB APP GIÁM SÁT VÀ LẬP LỊCH ĐIỀU KHIỂN ROBOT.....		58
6.1	Giới thiệu về Azure Cloud	58
6.2	Thiết kế phần mềm web giám sát và lập lịch điều khiển.....	59
6.2.1	Giới thiệu về Node.js và Express.js.....	59
6.2.2	Thiết kế giao diện người dùng.....	60
6.2.3	Cấu trúc truyền nhận dữ liệu.....	62
CHƯƠNG 7. KẾT QUẢ, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN		64
7.1	Kết quả thực hiện	64
7.1.1	Điều khiển cánh tay robot 3 bậc tự do.....	64
7.1.2	Giám sát, quản lý robot từ xa bằng Web App trên đám mây	64
7.1.3	Phần mềm giám sát kết hợp mô phỏng.....	68
7.2	Đánh giá	70
7.3	Hướng phát triển	70
TÀI LIỆU THAM KHẢO		71

DANH MỤC HÌNH ẢNH

Hình 2.1: Cánh tay robot công nghiệp.....	22
Hình 2.2: Các thành phần cơ bản của một cánh tay robot công nghiệp.....	23
Hình 2.3: Công nghệ đám mây mở ra một kỷ nguyên mới cho robot.....	24
Hình 2.4: Kim tự tháp tự động hóa theo theo mô hình ANSI/ISA-95 (ISO 62264)	25
Hình 2.5: So sánh giữa SCADA truyền thống và SCADA trên đám mây.....	26
Hình 2.6: Khớp quay và khớp tịnh tiến	29
Hình 2.7: Các hệ tọa độ hoạt động của robot	30
Hình 2.8: Các thông số Denavit-Hartenberg	31
Hình 2.9: Chọn hệ trục tọa độ cho cánh tay robot ba bậc tự do	32
Hình 2.10: Mô hình 3D của cánh tay robot ba bậc tự do	33
Hình 3.1: Cấu trúc hệ thống được thiết kế	38
Hình 4.1: Mô hình cánh tay robot ba bậc tự do	40
Hình 4.2: Mạch điều khiển Arduino MEGA 2560	41
Hình 4.3: Mạch điều khiển RAMPS 1.4.....	41
Hình 4.4: Nguồn tổ ong DC 12V 5A	41
Hình 4.5: Mạch thời gian thực RTC DS 1307.....	42
Hình 4.6: Cảm biến nhiệt độ LM35.....	42
Hình 4.7: Vòng lặp chính trong chương trình trên mạch điều khiển.....	43
Hình 4.8: Hàm ngắt Timer 100ms.....	44
Hình 5.1: Thư viện OPC UA Client & Server SDK	45
Hình 5.1: Cấu trúc dữ liệu trong phần mềm OPC UA Server.....	46
Hình 5.2: Tổ chức dữ liệu trong cơ sở dữ liệu SQL	47
Hình 5.3: Triển khai cơ sở dữ liệu lên đám mây Azure.....	47
Hình 5.4: Thư viện OpenGL	48
Hình 5.5: Cơ chế hoạt động của thư viện Assimp	50
Hình 5.6: Đo kích thước của base.....	51
Hình 5.7: Đo kích thước của các khâu	51
Hình 5.8: Các bước vẽ mô hình 3D thực hiện chức năng động học thuận	52
Hình 5.9: Các bước vẽ mô hình 3D thực hiện chức năng động học nghịch	53
Hình 5.10: Các bước thực hiện chức năng viết chương trình mô phỏng, hoạch định quỹ đạo LSPB cho mô hình mô phỏng	54
Hình 5.11: Hàm ngắt Timer 1	55
Hình 5.12: Hàm ngắt Timer 2	55
Hình 5.13: Hàm ngắt Timer 3	56
Hình 5.14: Thư viện mã nguồn mở open62541	57
Hình 5.15: Sơ đồ thể hiện vị trí của phần mềm mô phỏng trong hệ thống	57
Hình 6.1: Dịch vụ của Microsoft Azure trên mô hình điện toán đám mây.....	58

Hình 6.2: Mã nguồn mở Node.js.....	59
Hình 6.3: Thống kê những framework backend được sử dụng phổ biến nhất vào tháng 1 năm 2023	60
Hình 6.4: Trang đăng nhập	61
Hình 6.5: Trang Dashboard	62
Hình 6.6: Cấu trúc truyền nhận dữ liệu giữa Web App và OPC UA Server	63
Hình 7.1 Điều khiển robot di chuyển tự do đến các điểm cho trước	64
Hình 7.2 Phần mềm OPC UA Server khởi tạo các node	64
Hình 7.3: Phần mềm OPC UA Server chạy ổn định khi thu thập trạng thái hoạt động của robot	65
Hình 7.4: Giao diện đăng nhập của Web App.....	66
Hình 7.5: Giao diện Dashboard của Web App.....	66
Hình 7.6: Danh sách các nhiệm vụ của robot được lưu trong cơ sở dữ liệu.....	67
Hình 7.7: Web App hoạt động ổn định trên điện thoại di động, truy cập từ một mạng bất kỳ	67
Hình 7.8 Lập thời gian bắt đầu và kết thúc nhiệm vụ cho robot từ Web App.....	68
Hình 7.9: Giao diện của phần mềm giám sát kết hợp mô phỏng.....	68
Hình 7.10: Chương trình mẫu đơn giản	69
Hình 7.11: Kết quả chạy mô phỏng	69

DANH MỤC BẢNG

Bảng 2.1: So sánh tính năng giữa SCADA truyền thống và SCADA trên đám mây	27
Bảng 2.2: Các khớp thường gặp trong robot.....	28
Bảng 2.3: Thông số D-H	33
Bảng 5.1: Tập lệnh của chương trình mô phỏng.....	53

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến quý thầy cô, trường Đại học Bách Khoa nói chung và bộ môn điều khiển và 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 Trần Quốc Tiến Dũng. Thầy đã hướng dẫn và chỉ dạy em tận tình 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 những ngày tháng qua.

Đặc biệt, em xin chân thành cảm ơn tất cả người thân và bạn bè, thầy cô đã không ngừng giúp đỡ, đồng hành cùng em trong suốt chặng đường vừa qua, trong những thời điểm khó khăn nếu không có sự ủng hộ và động viên của mọi người có lẽ em đã chưa thể hoàn thành được đề tài và trưởng thành được như ngày hôm nay.

Em xin chân thành cảm ơn!

TPHCM, ngày 17 tháng 5 năm 2023

Sinh viên

Dương Nguyên Khang

LỜI MỞ ĐẦU

Trong những năm gần đây, các ngành công nghiệp, sản xuất đã chứng kiến một sự dịch chuyển mạnh mẽ do tác động của cuộc cách mạng công nghiệp 4.0, các doanh nghiệp đang dần nhận ra tầm quan trọng của việc ứng dụng robot vào sản xuất. Robot không chỉ giúp tăng năng suất và chất lượng sản phẩm mà còn giúp tiết kiệm thời gian và chi phí. Tuy nhiên, để đạt được hiệu quả tối đa, việc giám sát, quản lý và điều khiển robot từ xa là điều cần thiết.

Trong kỷ nguyên của sự kết nối, xu hướng giám sát dữ liệu từ xa và mô hình điện toán đám mây ngày càng được quan tâm bởi các doanh nghiệp ở các quy mô khác nhau. Đặc biệt trong lĩnh vực robotics, việc sử dụng các hệ thống giám sát và điều khiển từ xa trên môi trường đám mây giúp cho người quản lý có thể dễ dàng theo dõi quá trình sản xuất của robot, giải quyết các sự cố kỹ thuật và tối ưu hóa quy trình sản xuất. Từ đó có thể kiểm tra và lập lịch vận hành cho các robot từ xa mà không cần phải xuống tận từng khu vực sản xuất để kiểm tra, giúp cải thiện tối đa các lịch bảo trì và sửa lỗi không đáng có, giảm thiểu việc phải di chuyển đến hiện trường và đảm bảo an toàn cho nhân viên.

Vì vậy, việc ứng dụng robot vào sản xuất không chỉ đòi hỏi sự hiểu biết về robot mà còn đòi hỏi sự hiểu biết về các hệ thống giám sát và điều khiển từ xa. Trong luận văn này, em tập trung vào việc xây dựng một hệ thống phần mềm để giám sát, quản lý và lập lịch điều khiển cho robot cho phép doanh nghiệp có thể tương tác kiểm soát hoạt động của robot một cách dễ dàng và linh hoạt từ bất kỳ địa điểm nào trên các giao diện web hoặc các thiết bị di động có kết nối Internet, bên cạnh đó một cánh tay robot ba bậc tự do cũng được em sử dụng để kiểm chứng hệ thống phần mềm.

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Mục đích và mục tiêu của luận văn

Mục đích: Thiết kế một hệ thống phần mềm giám sát, mô phỏng và lập lịch hoạt động cho cánh tay robot ba bậc tự do hoàn chỉnh đến từ sự kết hợp giữa các phần mềm trên mạng cục bộ và ứng dụng Web trên môi trường đám mây. Hệ thống có thể giám sát robot từ xa và lập lịch cho cánh tay robot thực hiện các nhiệm vụ nhất định.

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

- Phát triển được một hệ thống phần mềm có khả năng giám sát, mô phỏng và lập lịch hoạt động cho robot từ xa một cách hoàn chỉnh và ổn định.
- Điều khiển được cánh tay robot có sẵn hoạt động theo mong muốn những tác vụ cơ bản.
- Thiết kế cơ sở dữ liệu, phát triển ứng dụng Web và triển khai trên môi trường đám mây.

1.2 Đối tượng và phạm vi nghiên cứu

1.2.1 Đối tượng nghiên cứu

- Cánh tay robot ba bậc tự do.
- Các phương trình động học của cánh tay robot.
- Lý thuyết về hoạch định quỹ đạo robot.
- Phương thức mô phỏng robot.
- Phương thức lưu trữ với cơ sở dữ liệu trên đám mây.
- Kiến trúc truyền thông giữa máy tính Server, các Client và phần mềm trên đám mây.

1.2.2 Phạm vi nghiên cứu

Luận văn tập trung nghiên cứu và thiết kế một hệ thống có sự tương tác giữa phần cứng và phần mềm và bao gồm nhiều thành phần kiến trúc cần nghiên cứu như sau:

- Tìm hiểu về cấu trúc cơ khí của một cánh tay robot có sẵn mô hình 3D.
- Đo đạc và phân tích để xây dựng phương trình động học cho cánh tay robot bằng phần mềm vẽ cơ khí SolidWorks.
- Cách thức hoạt động của các thành phần cơ, điện và kết hợp chúng lại thành một hệ thống phần cứng hoàn chỉnh.
- Phát triển giao diện người dùng phần mềm mô phỏng và ứng dụng Web.
- Thiết kế và xây dựng kiến trúc truyền thông giữa bộ điều khiển – máy tính server – máy tính client – web app trên đám mây và cơ sở dữ liệu.
- Xây dựng giải thuật đồng bộ truyền thông giữa bộ điều khiển robot và phần mềm trên máy tính.

1.3 Cấu trúc của luận văn

Cấu trúc luận văn được trình bày như sau:

- Chương 1: Giới thiệu đề tài, trình bày khái quát về mục đích, mục tiêu của luận văn, và giới thiệu đối tượng nghiên cứu, phạm vi nghiên cứu của luận văn.
- Chương 2: Tổng quan về robot công nghiệp, hệ thống SCADA cho robot công nghiệp và ứng dụng của IIoT trong các hệ thống robot, các khái niệm robot liên quan.
- Chương 3: Tổng quan hệ thống - trình bày về kiến trúc tổng thể của hệ thống, cách các thành phần tương tác với nhau.
- Chương 4: Phần cứng và mạch điều khiển – trình bày cụ thể về các thành phần cơ, điện sử dụng trong hệ thống phần cứng.
- Chương 5: Phần mềm mô phỏng và giám sát robot – trình bày phần mềm được xây dựng trên máy tính cục bộ.

- Chương 6: Web App giám sát và lập lịch điều khiển robot – trình bày phần mềm web được xây dựng trên đám mây.
- Chương 7: Kết quả, đánh giá và hướng phát triển – trình bày kết quả đạt được, đánh giá kết quả của luận văn và định hướng phát triển đề tài luận văn.

CHƯƠNG 2. TỔNG QUAN VỀ ROBOT VÀ HỆ THỐNG SCADA

2.1 Khái niệm về cánh tay robot công nghiệp

Cánh tay robot công nghiệp (tiếng Anh là Industrial Robot) là một tay máy tự động được sử dụng trong các quy trình sản xuất công nghiệp để thực hiện các tác vụ lặp đi lặp lại như gia công, hàn, lắp ráp, vận chuyển và đóng gói sản phẩm. Chúng được thiết kế để thay thế hoặc hỗ trợ công việc của con người trong môi trường sản xuất nhằm giảm chi phí lao động.

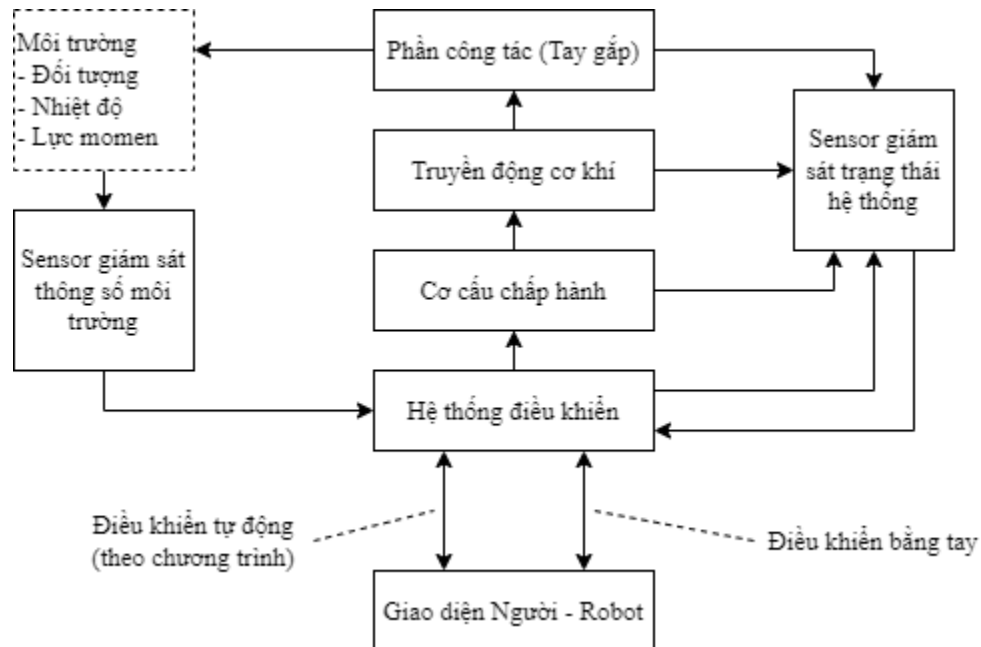


Hình 2.1: Cánh tay robot công nghiệp

Một cánh tay robot công nghiệp thông thường bao gồm các thành phần sau:

- Cơ khí: Bao gồm các khớp xoay, khớp nối, các khâu linh hoạt và tay gắp kết nối để thực hiện các tác vụ cụ thể. Đây là có thể xem là bộ khung xương của robot.

- Động cơ: Các động cơ điện hoặc thủy lực được sử dụng để giúp các khớp của robot chuyển động được. Động cơ được điều khiển bằng các hệ thống điện và truyền động để đạt được độ chính xác và hiệu suất cao.
- Hệ thống điều khiển: Hay là bộ điều khiển, kết nối đến các khớp, tín hiệu cảm biến để đưa ra quyết định điều khiển. Có thể xem là bộ não của robot, các hệ thống điều khiển tự động có thể sử dụng các thuật toán và phần mềm để định vị, điều khiển và phản hồi chuyển động của robot, giúp robot công nghiệp được điều khiển một cách tự động hoặc bằng tay.
- Cảm biến: Robot công nghiệp thường được trang bị các cảm biến để giám sát môi trường xung quanh và tương tác với các đối tượng. Các loại cảm biến phổ biến bao gồm cảm biến vị trí, cảm biến nhiệt độ, cảm biến tiếp xúc, cảm biến hình ảnh và cảm biến quang học.



Hình 2.2: Các thành phần cơ bản của một cánh tay robot công nghiệp

2.2 Ứng dụng của công nghệ đám mây trong giám sát và thu thập thông tin

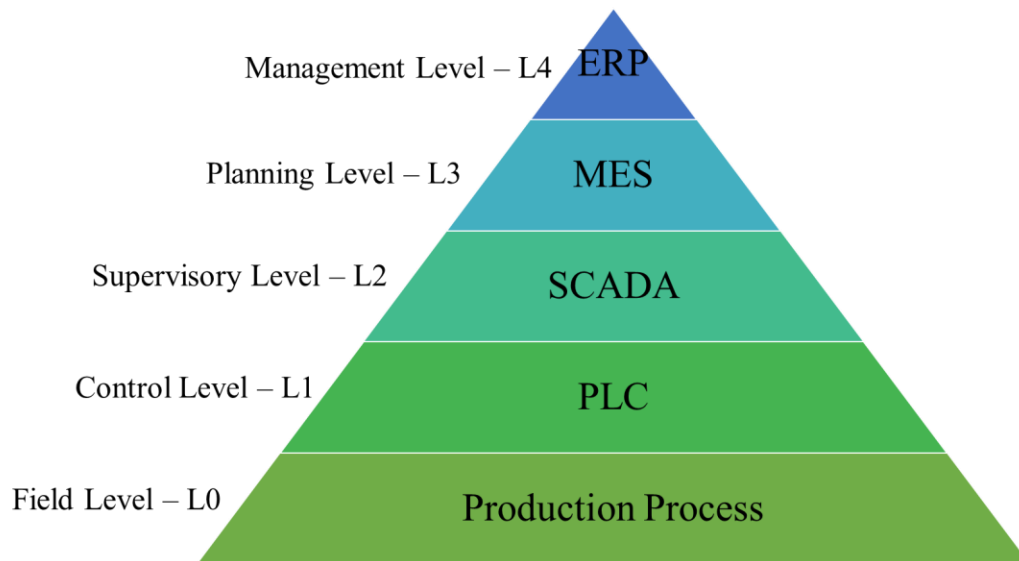
Công nghệ đám mây (tiếng Anh là Cloud Computing) đã đem lại những ưu điểm vượt trội trong việc quản lý và xử lý dữ liệu, cũng như cho phép truy cập và chia sẻ thông tin từ xa thông qua kết nối internet. Việc ứng dụng công nghệ đám mây

để thiết kế một hệ thống giám sát và thu thập thông tin (tiếng Anh là Supervisory Control and Data Acquisition – SCADA) đem lại nhiều lợi ích và tiềm năng nổi bật.



Hình 2.3: Công nghệ đám mây mở ra một kỷ nguyên mới cho robot

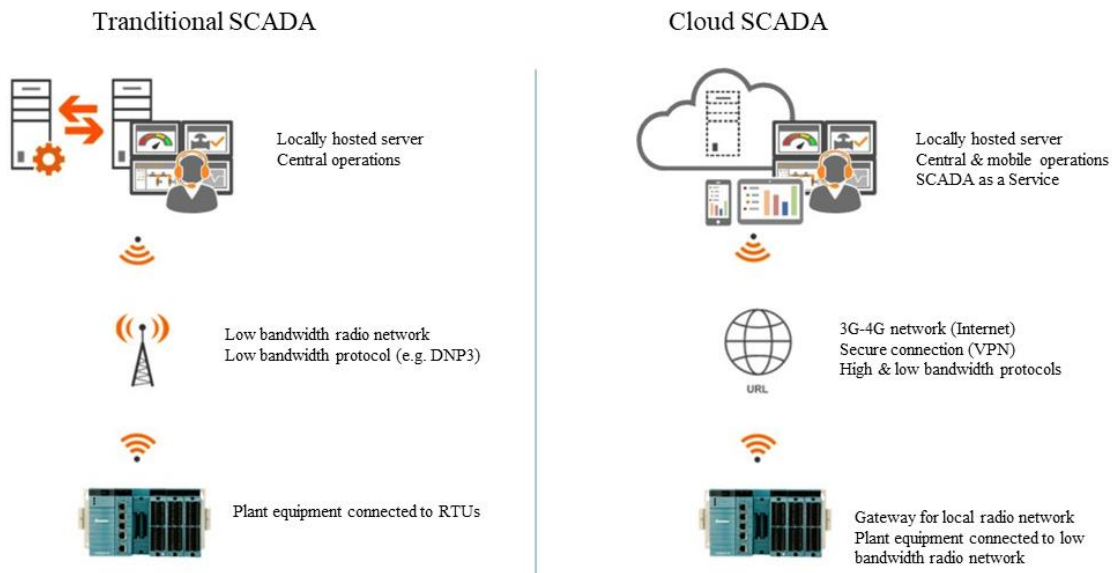
Trong kim tự tháp tự động hóa, SCADA đóng vai trò quan trọng và có vị trí trung tâm (theo mô hình ANSI/ISA-95), là một sự kết hợp giữa các thành phần phần cứng và phần mềm, cho phép giám sát và điều khiển hoạt động của các nhà máy một cách cục bộ hoặc từ xa, bên cạnh còn có vai trò ghi nhật ký, lưu trữ dữ liệu và cung cấp khả năng báo cáo và phân tích dữ liệu hoạt động của hệ thống. Khi ứng dụng công nghệ đám mây vào hệ thống SCADA, hệ thống công nghiệp có thể linh hoạt và dễ dàng mở rộng. Công nghệ đám mây còn cho phép chia sẻ dữ liệu và tính toán một cách hiệu quả và tiết kiệm chi phí.



Hình 2.4: Kim tự tháp tự động hóa theo mô hình ANSI/ISA-95 (ISO 62264)

Trong thực tế, các doanh nghiệp khi thiết kế một hệ thống SCADA có thể theo hai xu hướng: SCADA truyền thống và SCADA trên đám mây.

- **SCADA truyền thống:** Khi thiết kế SCADA truyền thống, chúng ta thường bắt đầu xoay quanh với thiết bị đầu cuối từ xa (hay còn được gọi là Remote Terminal Units – RTUs) và/hoặc bộ điều khiển logic có thể lập trình (Programmable Logic Controllers – PLCs). RTUs và PLCs là các vi xử lý, giao tiếp và tương tác với các bảng điều khiển tương tác với người dùng (hay còn được gọi là Human Machine Interface – HMI) và kết nối bằng dây với các thiết bị cấp trường như bơm, van, các động cơ và cảm biến, v.v. Tùy thuộc vào PLC, kiến trúc truyền thông của mạng lưới SCADA có thể khác nhau, thông qua các giao thức công nghiệp như Modbus hoặc EtherNet/IP. Hệ thống SCADA truyền thống có khả năng ghi nhật ký và lưu trữ dữ liệu hoạt động một cách cục bộ, do đó yêu cầu cần có các máy chủ riêng biệt để kết nối với phần mềm SCADA.



Hình 2.5: So sánh giữa SCADA truyền thống và SCADA trên đám mây

- SCADA trên đám mây (còn được gọi là Cloud SCADA): Là một hệ thống SCADA dựa trên mô hình điện toán máy mây (Cloud Computing), cho phép người dùng truy cập từ bất kỳ đâu và bất kỳ thiết bị nào có kết nối Internet thông qua các ứng dụng web. Tùy thuộc vào hệ thống được thiết kế, kiến trúc truyền thông của hệ thống SCADA trên đám mây sẽ khác nhau, các giao thức hỗ trợ có thể bao gồm giao thức HTTP, hoặc OPC/UA, v.v. Hệ thống SCADA trên đám mây có khả năng ghi nhật ký và lưu trữ dữ liệu hoạt động trên đám mây, dung lượng lưu trữ không phụ thuộc vào các thiết bị máy tính tại mạng cục bộ như SCADA truyền thống, và có thể mở rộng không giới hạn theo mong muốn của doanh nghiệp. Với Cloud SCADA, người dùng có thể theo dõi và quản lý các hoạt động công nghiệp từ xa, tiết kiệm thời gian, công sức và tăng tính linh hoạt trong việc quản lý quy trình và hệ thống công nghiệp. Nó cũng cung cấp khả năng tích hợp với các hệ thống khác như hệ thống quản lý dự trữ (Enterprise Resource Planning - ERP) và hệ thống quản lý bảo trì (Computerized Maintenance Management System - CMMS) để tối ưu hóa quản lý toàn diện của doanh nghiệp.

Table 1 Bảng 2.1: So sánh tính năng giữa SCADA truyền thống và SCADA trên đám mây

Tiêu chí	SCADA truyền thống	SCADA dựa trên mô hình điện toán đám mây
Cơ sở hạ tầng	Triển khai trong mạng cục bộ hoặc mạng nội bộ của nhà máy, với các máy chủ và thiết bị giám sát được đặt tại chỗ.	Sử dụng mô hình đám mây, dựa trên cơ sở hạ tầng đám mây và có thể truy cập từ bất kỳ đâu thông qua Internet.
Vị trí lưu trữ dữ liệu	Dữ liệu được lưu trữ và xử lý trong hệ thống cục bộ, thường là trên các máy chủ và thiết bị trong mạng nội bộ.	Lưu trữ dữ liệu trên đám mây, cho phép truy cập và quản lý dữ liệu từ xa thông qua Internet.
Khả năng mở rộng	Bị giới hạn về khả năng mở rộng, vì nó dựa trên cơ sở hạ tầng cục bộ và tài nguyên có hạn.	Có khả năng mở rộng linh hoạt dựa trên tính chất đám mây, cho phép thêm mới thiết bị và mở rộng quy mô một cách dễ dàng.
Tiện ích và truy cập từ xa	Yêu cầu người dùng phải có kết nối mạng nội bộ để truy cập và quản lý hệ thống.	Cho phép truy cập từ xa thông qua Internet, giúp người dùng có thể giám sát và điều khiển hệ thống từ bất kỳ đâu và từ bất kỳ thiết bị nào có kết nối Internet.
Quản lý dữ liệu	Lưu trữ dữ liệu trong các cơ sở dữ liệu cục bộ	Sử dụng cơ sở dữ liệu đám mây, cho phép lưu


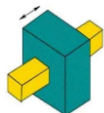
		trữ lớn và dễ dàng mở rộng.
Chi phí triển khai	Đòi hỏi đầu tư ban đầu lớn để xây dựng cơ sở hạ tầng cục bộ bao gồm máy chủ, thiết bị phần cứng, mạng nội bộ và phần mềm.	Không phải đầu tư ban đầu lớn cho cơ sở hạ tầng cục bộ mà cung cấp mô hình dịch vụ (SaaS), trả phí theo mô hình thuê bao hoặc trả tiền theo sử dụng.
Chi phí duy trì và cập nhật	Yêu cầu các bản cập nhật phần mềm, bảo trì phần cứng và các hoạt động quản lý hệ thống liên quan khác, chi phí do người dùng chịu trách nhiệm.	Các bản cập nhật phần mềm và bảo trì hệ thống được thực hiện bởi nhà cung cấp đám mây, giúp giảm bớt gánh nặng chi phí và công việc quản lý của người dùng.

2.3 Động học robot

2.3.1 Một số khái niệm

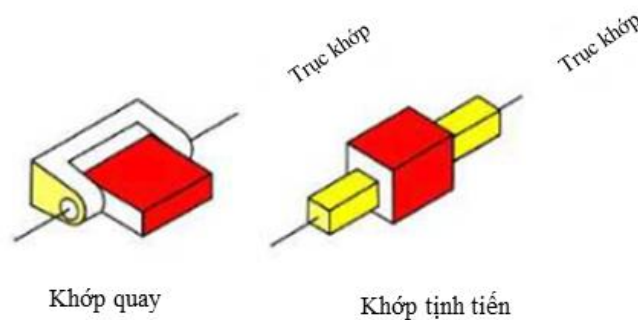
- Khâu: Các vật rắn riêng lẻ hình thành nên robot gọi là khâu, khâu robot có thể là khâu cứng hoặc mềm và chuyển động tương đối với các khâu khác.

Bảng 2.2: Các khớp thường gặp trong robot

Kiểu khớp	Số bậc tự do	Chuyển động	Sơ đồ động học
Khớp quay	1	Quay tròn	
Khớp tịnh tiến	1	Thẳng	

Khớp trụ	2	Trụ	
Khớp cầu	3	Cầu	
Khớp vít	1	Xoắn ốc	
Khớp phẳng	3	Phẳng	

- Khớp: Hai khâu được nối với nhau thông qua khớp mà khi chuyển động tương quan có thể biểu diễn bởi một hệ trục. Trong hầu hết các loại robot, thường thấy các loại khớp được sử dụng là khớp quay (R) và khớp tịnh tiến (T).



Hình 2.6: Khớp quay và khớp tịnh tiến

- Bậc tự do (tiếng Anh gọi là Degree of Freedom – DoF) là số lượng tham số tự do hay khả năng di chuyển của cơ cấu. Số bậc tự do của robot chính là khả năng chuyển động của robot trong hệ tọa độ gắn liền với điểm tham chiếu. Vì robot hoạt động trong không gian nên số bậc tự do được tính theo công thức sau:

$$F = \lambda(n - j - 1) + \sum_i f_i - f_p$$

Trong đó:

F: bậc tự do của robot,

λ : số bậc tự do của không gian mà robot hoạt động,

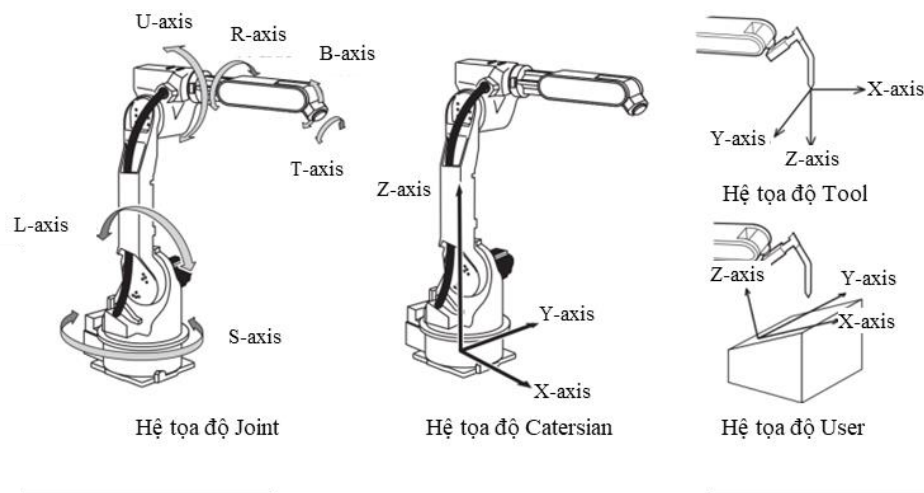
m: số khâu của robot bao gồm cả khâu cố định,

j: số khớp của robot

f_i : số chuyển động cho phép của khớp i,

f_p : số bậc tự do thừa trong robot.

- Hệ trục tọa độ: Khi làm việc với robot, thông thường chúng ta cần lưu ý bốn loại hệ trục tọa độ bao gồm hệ tọa độ Joint, hệ tọa độ Catersian, hệ tọa độ Tool và hệ tọa độ User.



Hình 2.7: Các hệ tọa độ hoạt động của robot

2.3.2 Biểu diễn Denavit – Hartenberg của bài toán động học robot

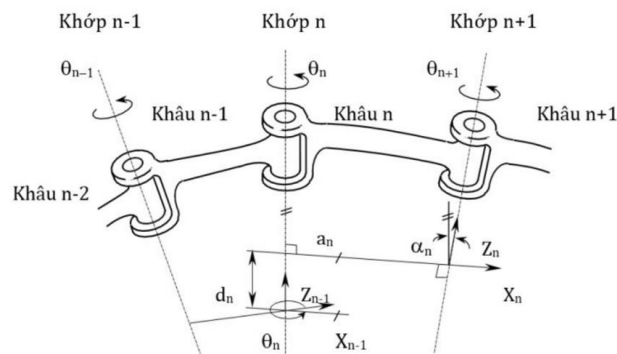
Mô hình hóa Denavit – Hartenberg (viết tắt là phương pháp D-H) là cách biểu diễn sử dụng các phép biến đổi để mô tả vị trí và hướng tương đối của các khớp giữa các hệ tọa độ. Một robot có n khớp và $n+1$ khâu. Chỉ số các khớp bắt đầu từ 1 đến n còn khâu bắt đầu từ 0 đến n . Khớp (i) kết nối khâu (i-1) và khâu (i). Để chọn hệ trục tọa độ cho các khâu ta theo sát nguyên tắc như sau:

- Trục z_i nằm dọc theo trục của khớp (i+1).

- Trục x_i nằm dọc theo đường vuông góc chung hướng từ z_{i-1} đến z_i . Trong trường hợp 2 trục giao nhau thì chọn $\vec{x}_i = \vec{z}_{i-1} \times \vec{x}_i$, trục $\vec{y}_i = \vec{z}_i \times \vec{x}_i$.

Xác định bộ thông số Denavit-Hartenberg:

- Gọi a_i là khoảng cách giữa trục z_{i-1} và trục z_i dọc theo trục x_i .
- Gọi α_i là góc xoay z_{i-1} quanh trục x_i đến song song với z_i .
- Gọi d_i là khoảng cách dọc theo trục z_{i-1} giữa x_{i-1} và x_i .
- Gọi θ_i là góc xoay x_{i-1} quanh trục z_i đến song song với trục x_i .



Hình 2.8: Các thông số Denavit-Hartenberg

Xác định biến khớp:

- Nếu khớp (i) là khớp quay thì θ_i là biến khớp.
- Nếu khớp (i) là khớp tịnh tiến thì d_i là biến khớp.

Như vậy, mối quan hệ giữa hai hệ tọa độ nối tiếp nhau (i-1) và (i) được xác định bởi phép biến đổi và ma trận chuyển đổi như sau:

- Quay quanh z_{i-1} một góc θ_i .
- Tịnh tiến dọc theo z_{i-1} một đoạn d_i .
- Tịnh tiến dọc theo x_{i-1} một góc α_i .

Ma trận chuyển đổi giữa hai hệ tọa độ:

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

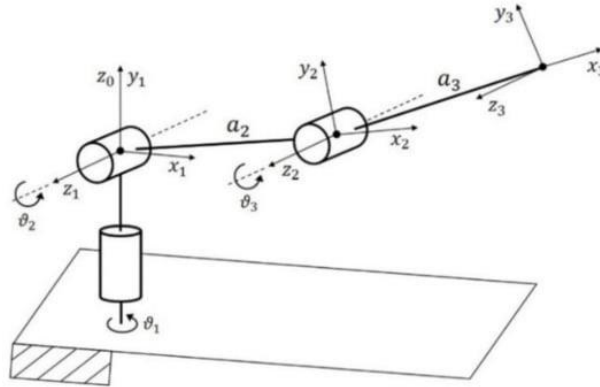
2.3.3 Phương trình động học thuận của cánh tay robot 3 bậc tự do

Bài toán động học thuận của robot mô tả hướng và vị trí của phần tay gấp dưới dạng hàm số của các biến khớp. Giả sử ta có một tay máy với $(n+1)$ khâu và (n) khớp. Vị trí và hướng của phần tay gấp so với hệ tọa độ gốc được mô tả bằng vector định vị p^0 (biểu diễn tọa độ của tay gấp) và hướng của các vector chỉ phương n, s, a . Phép chuyển đổi tọa độ được biểu diễn bằng ma trận chuyển đổi thuần nhất:

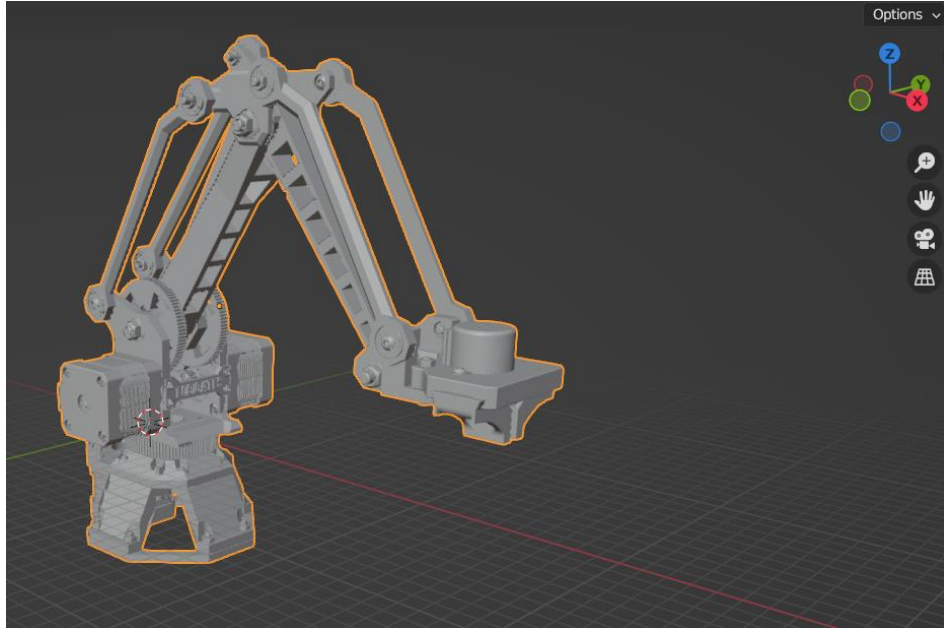
$${}^0T(q) = \begin{bmatrix} n^0(q) & s^0(q) & a^0(q) & p^0(q) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Trong đó, q là vector n phần tử, bao gồm các biến khớp, p là vector định vị tọa độ của tay gấp trong không gian, n, s, a là các vector xác định hướng của robot.

Cánh tay robot được sử dụng trong luận văn có ba khớp xoay, áp dụng quy tắc Denavit – Hartenberg, chọn hệ trục tọa độ như sau:



Hình 2.9: Chọn hệ trục tọa độ cho cánh tay robot ba bậc tự do



Hình 2.10: Mô hình 3D của cánh tay robot ba bậc tự do

Bảng biểu diễn D – H với các thông số thực tế $a_1 = 130.44^{mm}$, $a_2 = 165.04^{mm}$, $a_3 = 144.57^{mm}$:

Bảng 2.3: Thông số D-H

Link	a_i	α_i	d_i	θ_i
1	a_1	$\pi/2$	0	θ_1^*
2	a_2	0	0	θ_2^*
3	a_3	0	0	θ_3^*

Các ma trận chuyển đổi được tính như sau:

$${}^0T_1 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & a_1\sin\theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3$$

$$= \begin{bmatrix} \cos\theta_1 \cos(\theta_2 + \theta_3) & -\cos\theta_1 \sin(\theta_2 + \theta_3) & \sin\theta_1 & \cos\theta_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) \\ \sin\theta_1 \cos(\theta_2 + \theta_3) & -\sin\theta_1 \sin(\theta_2 + \theta_3) & -\cos\theta_1 & \sin\theta_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) \\ \sin(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 & a_2\sin\theta_2 + a_3\sin(\theta_2 + \theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Như vậy tọa độ cuối của tay gắp robot đã được xác định tại vị trí (p_x, p_y, p_z) và hướng được định bởi ba vector $\vec{n}, \vec{o}, \vec{a}$ từ giá trị các biến khớp $[\theta_1, \theta_2, \theta_3]$.

2.3.4 Phương trình động học nghịch của cánh tay robot 3 bậc tự do

Trái ngược với bài toán động học thuận, bài toán động học nghịch cho ta biết giá trị các biến khớp $[\theta_1, \theta_2, \theta_3]$, từ ma trận chuyển đổi 0T_3 .

Ta cần tìm giá trị các biến khớp $[\theta_1, \theta_2, \theta_3]$ từ các phương trình sau:

$$\begin{cases} p_x = \cos\theta_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) & (1) \\ p_y = \sin\theta_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) & (2) \\ p_z = a_2\sin\theta_2 + a_3\sin(\theta_2 + \theta_3) & (3) \end{cases}$$

Dựa vào các phương trình trên, ta có những biến đổi cơ bản sau:

- Biến đổi 1:

$$p_x^2 + p_y^2 + p_z^2 = a_1^2 + a_2^2 + a_3^2 + 2a_1a_2\cos\theta_2 + 2a_2a_3\cos\theta_3 + 2a_1a_3\cos(\theta_2 + \theta_3)$$

$$\Leftrightarrow p_x^2 + p_y^2 + p_z^2 + a_1^2 - a_2^2 - a_3^2 = 2a_1(a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)) + 2a_2a_3\cos\theta_3 \quad (4)$$

- Biến đổi 2:

$$p_z = a_2\sin\theta_2 + a_3\sin(\theta_2 + \theta_3)$$

$$\begin{aligned}
&\Leftrightarrow p_z = \sin\theta_2(a_2 + a_3\cos\theta_3) + a_3\cos\theta_2\sin\theta_3 \\
&\Leftrightarrow \frac{p_z}{\sqrt{(a_2 + a_3\cos\theta_3)^2 + (a_3\sin\theta_3)^2}} = \sin(\theta_2 + \vartheta), \quad (5) \\
&\text{với } \vartheta = \text{Atan2}(a_3\sin\theta_3, a_2 + a_3\cos\theta_3)
\end{aligned}$$

Lời giải bài toán động học nghịch:

Nếu $p_y = 0$:

Nếu $p_x = 0$:

$$\begin{aligned}
p_x = p_y = 0 &\Rightarrow p_x^2 + p_y^2 = [a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)]^2 = 0 \\
&\Rightarrow a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3) = -a_1 \\
&\Rightarrow p_z^2 + a_1^2 = a_2^2 + a_3^2 + 2a_2a_3\cos\theta_3 \\
&\Rightarrow \theta_3
\end{aligned}$$

Với giá trị vừa tìm được của θ_3 , thế vào (5) ta tìm được θ_2 , và trường hợp này có vô số nghiệm θ_1 .

Nếu $p_x \neq 0$:

$$\begin{aligned}
p_x^2 + p_y^2 &= [a_1 + a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3)]^2 = 0 \Rightarrow \sin\theta_1 = 0 \\
&\Rightarrow \theta_1 = 0^0 \text{ hoặc } \theta_1 = 180^0
\end{aligned}$$

Nếu $\theta_1 = 0^0$:

$$\begin{aligned}
(1) &\Leftrightarrow p_x - a_1 = a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3) \\
&\Rightarrow (p_x - a_1)^2 + p_z^2 = a_2^2 + a_3^2 + 2a_2a_3\cos\theta_3 \Rightarrow \theta_3.
\end{aligned}$$

Với giá trị vừa tìm được của θ_3 , thế vào (5) ta tìm được θ_2 .

Nếu $\theta_1 = 180^0$:

$$\begin{aligned}
(1) &\Leftrightarrow p_x + a_1 = a_2\cos\theta_2 + a_3\cos(\theta_2 + \theta_3) \\
&\Rightarrow (p_x + a_1)^2 + p_z^2 = a_2^2 + a_3^2 + 2a_2a_3\cos\theta_3 \Rightarrow \theta_3.
\end{aligned}$$

Với giá trị vừa tìm được của θ_3 , thế vào (5) ta tìm được θ_2 .

Nếu $p_y \neq 0$:

$$\theta_1 = \text{atan2}(p_x, p_y)$$

$$(4) \Rightarrow p_x^2 + p_y^2 + p_z^2 + a_1^2 - a_2^2 - a_3^2 - 2a_1 \frac{p_y}{\sin\theta_1} = 2a_2a_3\cos\theta_3 \Rightarrow \theta_3.$$

Với giá trị vừa tìm được của θ_3 , thế vào (5) ta tìm được θ_2 .

2.3.5 Hoạch định quỹ đạo

Hoạch định quỹ đạo nhằm xác định quy luật chuyển động của các biến khớp để điều khiển robot bắt buộc phải qua các điểm trung gian và tổng hợp chung thành một quỹ đạo xác định. Việc này nhằm giúp robot hoạt động hiệu quả trong môi trường công nghiệp, thực hiện các công việc như hàn, sơn, v.v.

Trong luận văn ở phần thiết kế phần mềm mô phỏng, chức năng mô phỏng ngoại tuyến (offline) cho robot, em đã sử dụng phương pháp quy hoạch quỹ đạo có vận tốc hình thang, là một kiểu quy hoạch quỹ đạo cơ bản cho robot. Quỹ đạo (được biểu diễn bởi hàm số $q(t)$) gồm hai đoạn parabol ở hai đầu và đoạn tuyến tính ở giữa (LSPB – Linear Segment with Parabolic Blend). Gia tốc là hằng số a_{max} . Vận tốc gồm hai đoạn dốc ở đầu và cuối, vận tốc ở đoạn giữa trung gian có giá trị là hằng số v_{max} , thời gian tăng tốc là t_1 bằng với thời gian giảm tốc.

Phương trình của quỹ đạo:

$$t_1 = t_2 = \frac{v_{max}}{a_{max}}$$

$$t_m = \frac{p_{max} - at_1^2}{v_{max}} = \frac{p_{max} - v_{max}t_1}{v_{max}}$$

$$p(t) = \begin{cases} \frac{1}{2}at^2 & , \text{với } 0 < t \leq t_1 \\ \frac{1}{2}at_1^2 + v_{\max}t & , \text{với } t_1 < t \leq t_1 + t_m \\ \frac{1}{2}at_1^2 + v_{\max}(t_1 + t_m) + v_{\max}t - \frac{1}{2}at^2 & , \text{với } t_1 + t_m < t \leq t_1 + t_m + t_2 \end{cases}$$

Phương trình vận tốc:

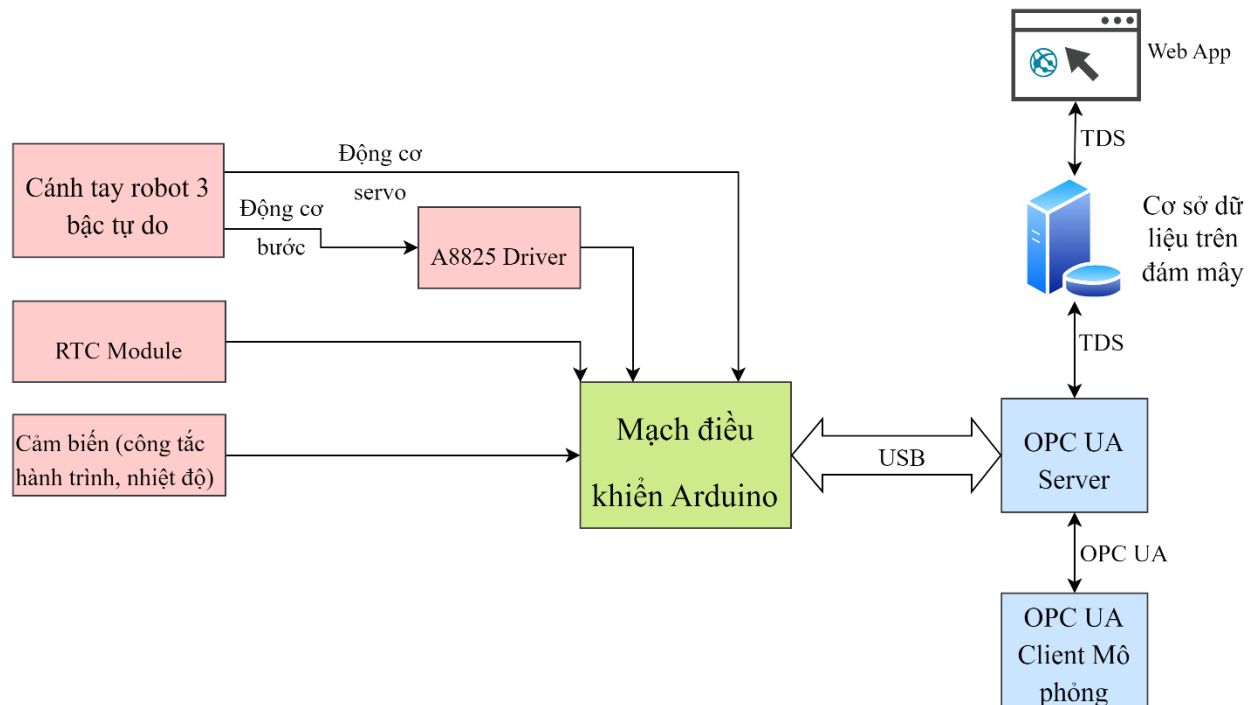
$$v(t) = \begin{cases} a_{\max}t, & \text{với } 0 < t \leq t_1 \\ a_{\max}t_1, & \text{với } t_1 < t \leq t_1 + t_m \\ v_{\max} - a_{\max}t, & \text{với } t_1 + t_m < t \leq t_1 + t_m + t_2 \end{cases}$$

Phương trình gia tốc:

$$a(t) = \begin{cases} a_{\max}, & \text{với } 0 < t \leq t_1 \\ 0, & \text{với } t_1 < t \leq t_1 + t_m \\ -a_{\max}, & \text{với } t_1 + t_m < t \leq t_1 + t_m + t_2 \end{cases}$$

CHƯƠNG 3. TỔNG QUAN VỀ CẤU TRÚC HỆ THỐNG

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



Hình 3.1: Cấu trúc hệ thống được thiết kế

Hệ thống được thiết kế bao gồm:

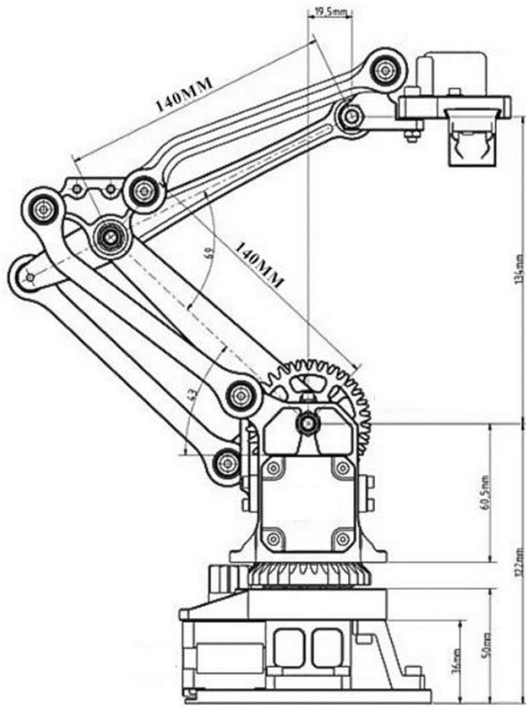
- Phần cứng:
 - Cánh tay robot ba bậc tự do: Là kết cấu cơ khí, thực hiện các nhiệm vụ gấp, thả vật và di chuyển trong không gian.
 - Mạch điều khiển xuất xung điều khiển đến các driver để điều khiển các động cơ bước của robot, điều khiển động cơ servo để đóng mở tay kẹp của robot, thu thập các giá trị ngõ vào khác và giao tiếp với máy tính.
 - A8825 Driver: Nhận xung điều khiển từ mạch điều khiển và điều khiển động cơ bước.

- Phần mềm:
 - Phần mềm ở trên mạch điều khiển: Thực hiện chức năng giao tiếp với máy tính thông qua kết nối USB để xuất xung điều khiển động cơ
 - Phần mềm OPC UA Server: Là trung tâm của hệ thống, lưu trữ các giá trị trạng thái của robot khi hoạt động.
 - Phần mềm OPC UA Client: Phần mềm có chức năng mô phỏng và giám sát robot.
 - Cơ sở dữ liệu trên mô hình điện toán đám mây: Lưu trữ thông tin vận hành của robot trên dịch vụ đám mây của Azure Microsoft.
 - Web App: Phần mềm được triển khai trên mô hình điện toán đám mây cho phép giám sát trạng thái vận hành của robot từ xa và có thể lập lịch để robot vận hành theo thời gian nhất định.

CHƯƠNG 4. PHẦN CỨNG VÀ MẠCH ĐIỀU KHIỂN

4.1 Cánh tay robot 3 bậc tự do

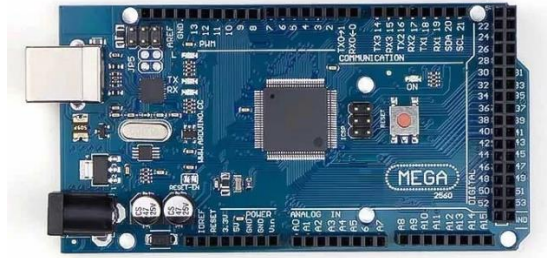
Trong luận văn này em sử dụng một cánh tay robot 3 bậc tự do làm phần cứng minh họa, bao gồm 3 trục xoay.



Hình 4.1: Mô hình cánh tay robot ba bậc tự do

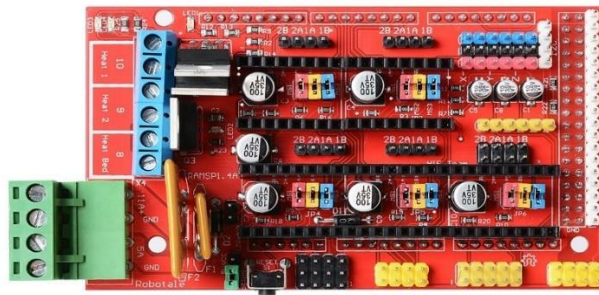
4.2 Mạch điều khiển và các thành phần điện

- Mạch điều khiển chính cho robot được em sử dụng trong luận văn là mạch Arduino Mega 2560, là một bo mạch điều khiển phổ biến trong họ điều khiển Arduino. Arduino Mega 2560 được xây dựng trên vi điều khiển ATmega2560 có tốc độ xử lý 16MHz và số lượng I/O khá lớn (54 chân I/O trong đó gồm 14 chân số tương tự analog và 40 chân kỹ thuật số digital).



Hình 4.2: Mạch điều khiển Arduino MEGA 2560

- Kết hợp với Arduino Mega 2560, em sử dụng thêm mạch Shield Ramps 1.4 (RepRap Arduino Mega Pololu Shield) để hỗ trợ Arduino Mega 2560 điều khiển các động cơ bước, các cảm biến được kết nối đơn giản hơn.



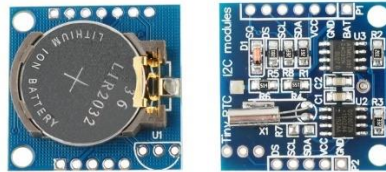
Hình 4.3: Mạch điều khiển RAMPS 1.4

- Nguồn tổ ong DC 12V 5A để cấp nguồn cho mạch điều khiển.



Hình 4.4: Nguồn tổ ong DC 12V 5A

- Driver A8825: Là một driver điều khiển động cơ bước phổ biến với khả năng điều chỉnh dòng động cơ, tốc độ và hướng quay.
- Module thời gian thực RTC DS 1307: Là một module thông dụng được sử dụng để đồng bộ và duy trì thời gian thực trong các ứng dụng điện tử, lưu trữ thông tin về giờ, phút, giây, ngày, tháng, năm và cả năm nhuận.



Hình 4.5: Mạch thời gian thực RTC DS 1307

- Cảm biến nhiệt độ LM35: Là một cảm biến nhiệt độ nhỏ, gọn và dễ sử dụng. LM35 có độ chính xác cao và khả năng đáp ứng nhanh.



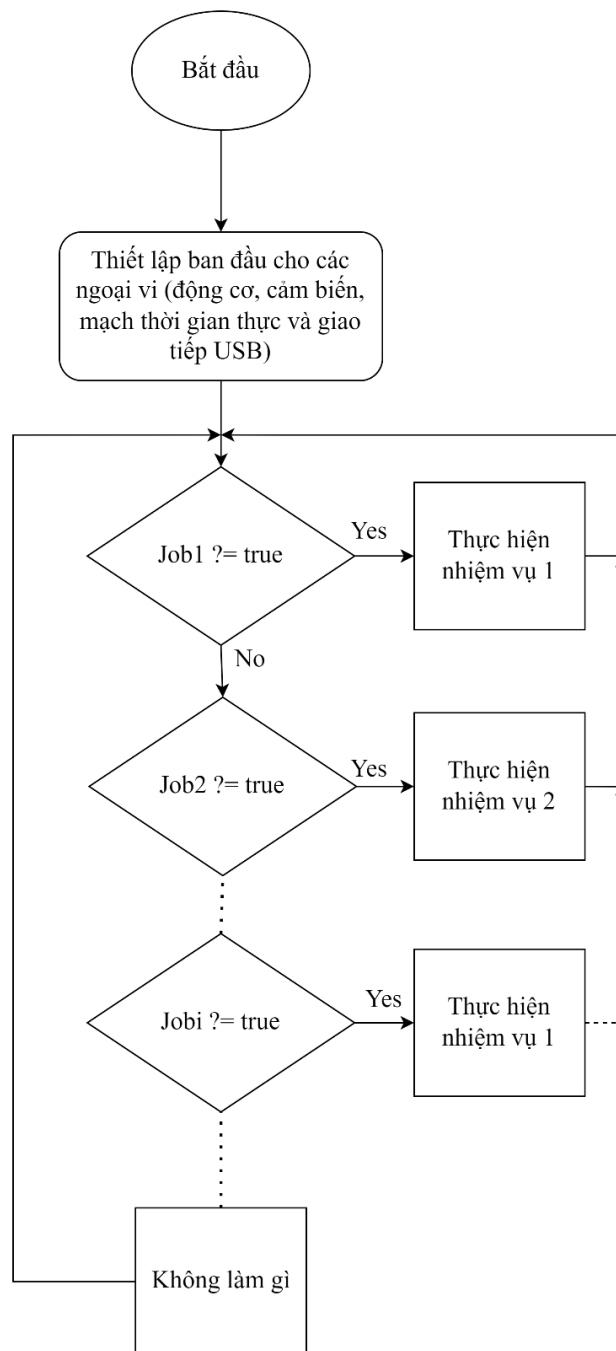
Hình 4.6: Cảm biến nhiệt độ LM35

4.3 Phần mềm trên board điều khiển

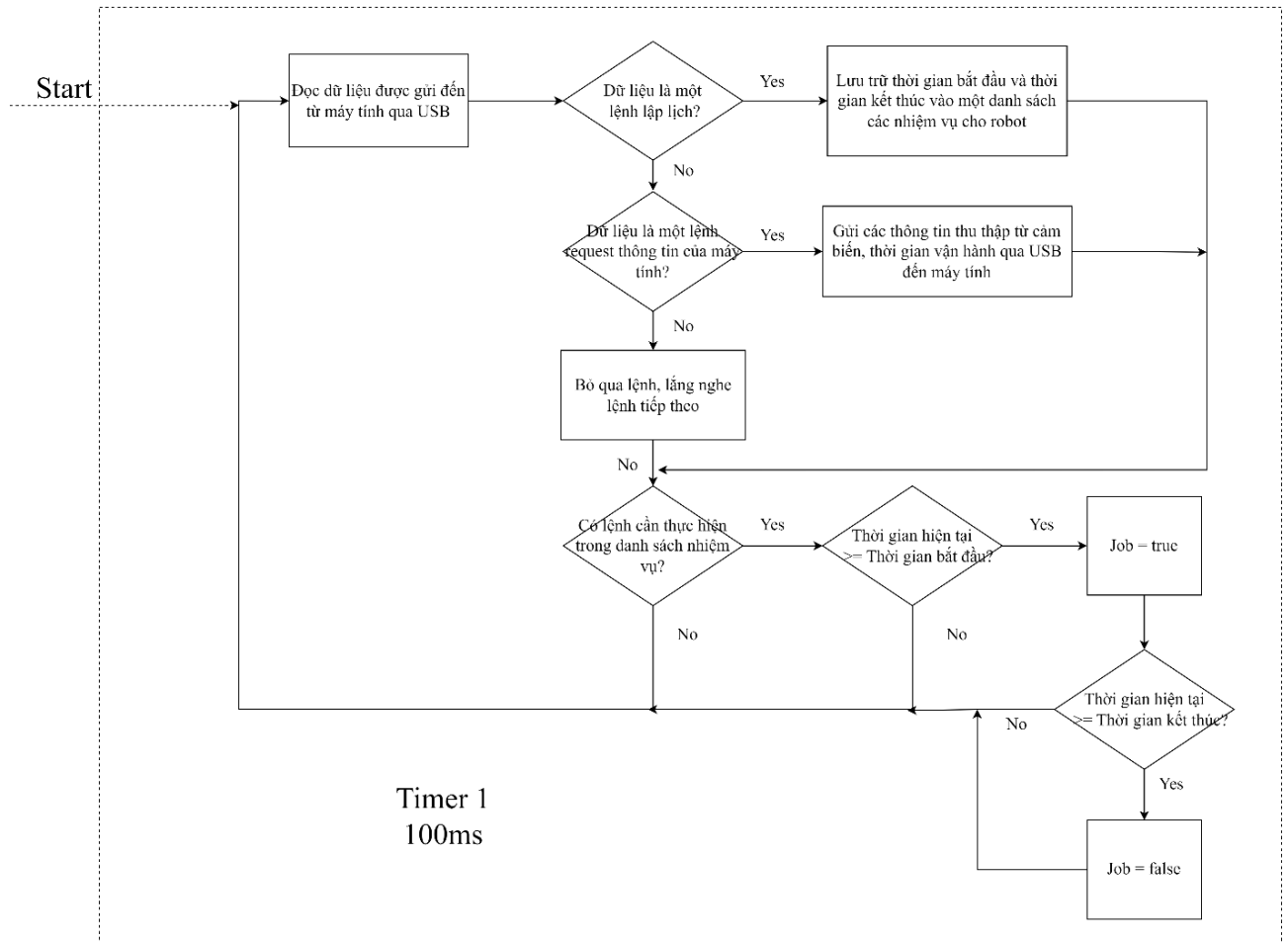
Chính vì hệ thống có nhiều thông tin cần thu thập và trao đổi với máy tính, nên có thể xảy ra các trường hợp thông tin bị thiếu nếu như xử lý chương trình điều khiển không tốt.

Về cơ bản, Arduino sẽ lắng nghe lệnh từ máy tính thông qua giao tiếp USB. Máy tính sẽ yêu cầu Arduino gửi thông tin thu thập được đến OPC UA Server, và khi

có thông tin lập lịch được thiết lập bởi Web App, OPC UA Server sẽ gửi lệnh xuống mạch điều khiển để Arduino ghi nhận và hoạt động vào thời gian xác định



Hình 4.7: Vòng lặp chính trong chương trình trên mạch điều khiển

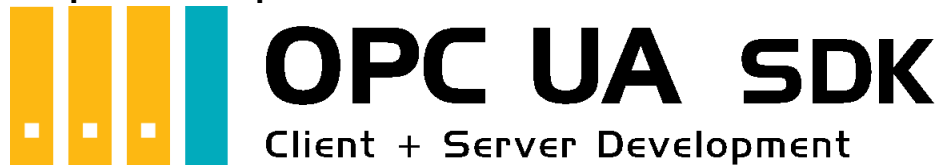


Hình 4.8: Hàm ngắt Timer 100ms

CHƯƠNG 5. PHẦN MỀM MÔ PHỎNG VÀ GIÁM SÁT ROBOT

5.1 Phần mềm OPC UA Server

5.1.1 Giới thiệu về thư viện OPC UA Client & Server SDK



Hình 5.1: Thư viện OPC UA Client & Server SDK

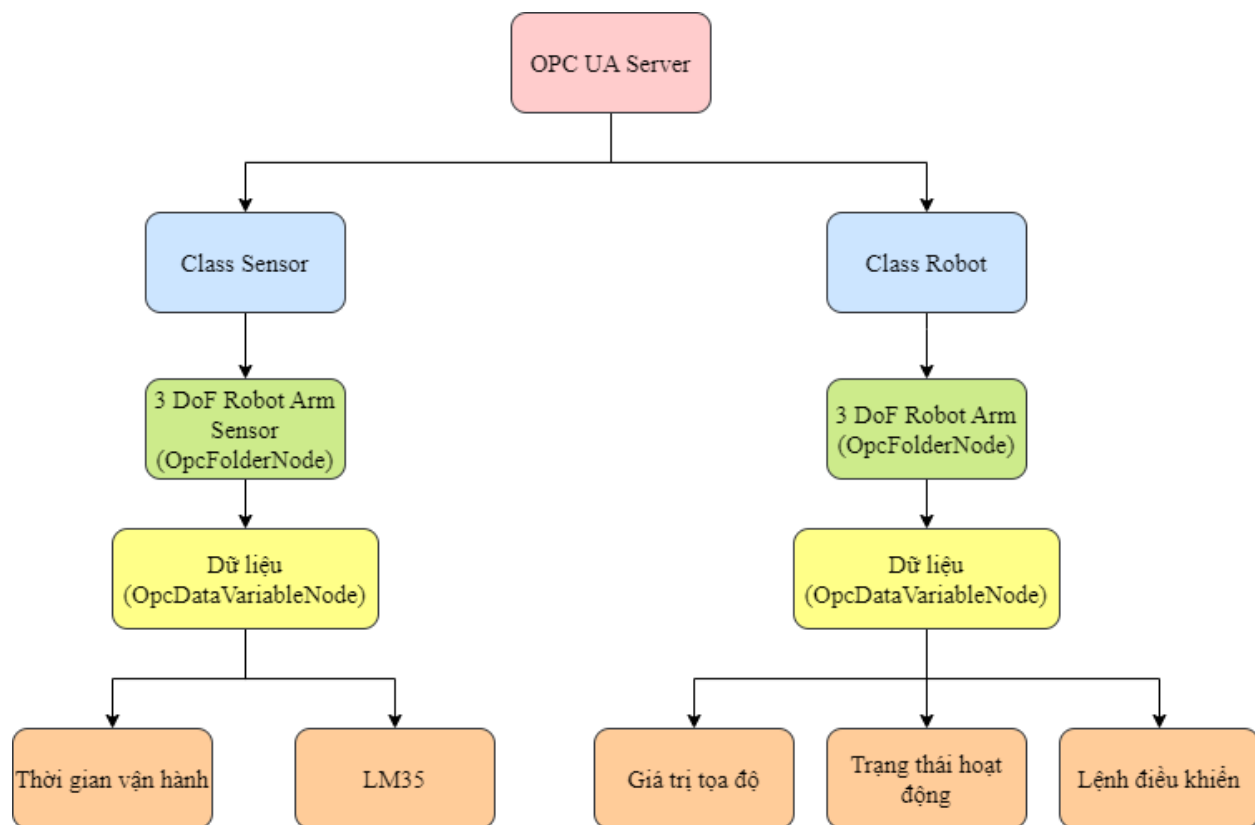
Thư viện OPC UA Client & Server SDK là một tập hợp các công cụ, thư viện và API (Application Programming Interface) hỗ trợ để phát triển các ứng dụng với chuẩn giao thức OPC UA (Open Platform Communications Unified Architecture). Đây là một giao thức truyền thông đa nền tảng trong lĩnh vực tự động hóa công nghiệp, có thể dễ dàng nhúng vào phần mềm và phần cứng để phục vụ các ứng dụng truyền thông.

Thư viện cung cấp các chức năng trong việc lập trình xây dựng và triển khai các ứng dụng OPC UA Client và Server. Một số tính năng nổi trội thường được người dùng khai thác như kết nối, gửi và nhận dữ liệu, quản lý phiên, định dạng dữ liệu, bảo mật và xác thực, v.v.

5.1.2 Cấu trúc OPC UA Server đã thiết kế

Trong luận văn em đã sử dụng thư viện để phát triển một ứng dụng OPC UA Server, lập trình với dạng một ứng dụng C# Console App trên Visual Studio 2022. Server có khả năng định dạng dữ liệu, là trung tâm phân phối các thông số hoạt động của robot trên một máy chủ đặt tại nhà máy và các OPC UA Client khác có thể truy cập đến Server.

Dữ liệu trong OPC UA Server được tổ chức như sau:



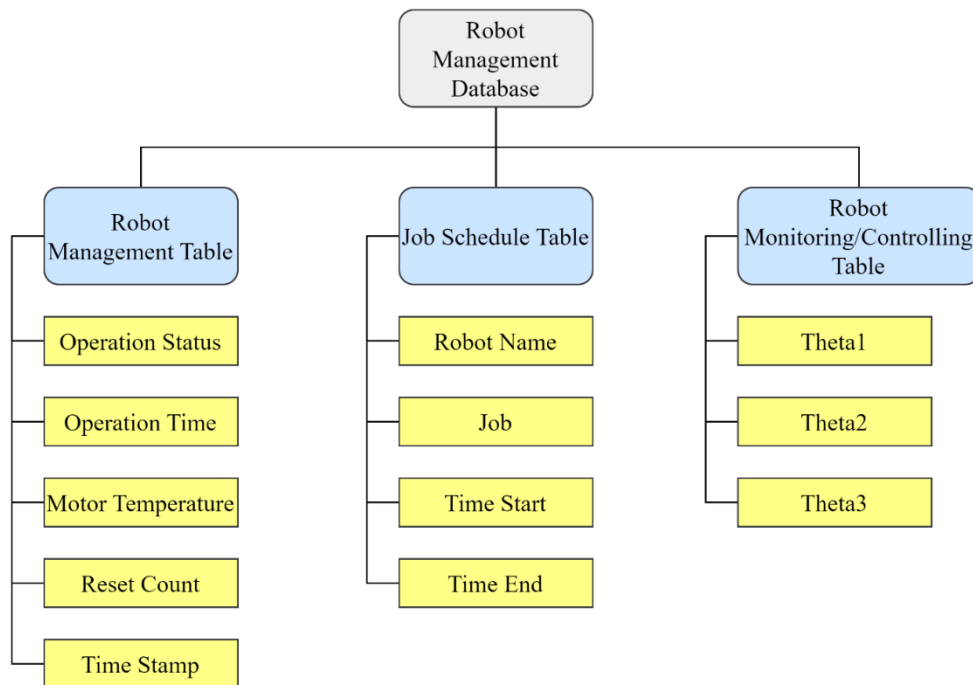
Hình 5.1: Cấu trúc dữ liệu trong phần mềm OPC UA Server

Trong đó, dữ liệu được chia thành từ lớp (Class), mỗi robot khác nhau có một node quản lý khác nhau (OpcNodeManager), từ đó, em tạo các node dữ liệu để lưu trữ thông tin vận hành của mỗi robot (OpcDataVariableNode).

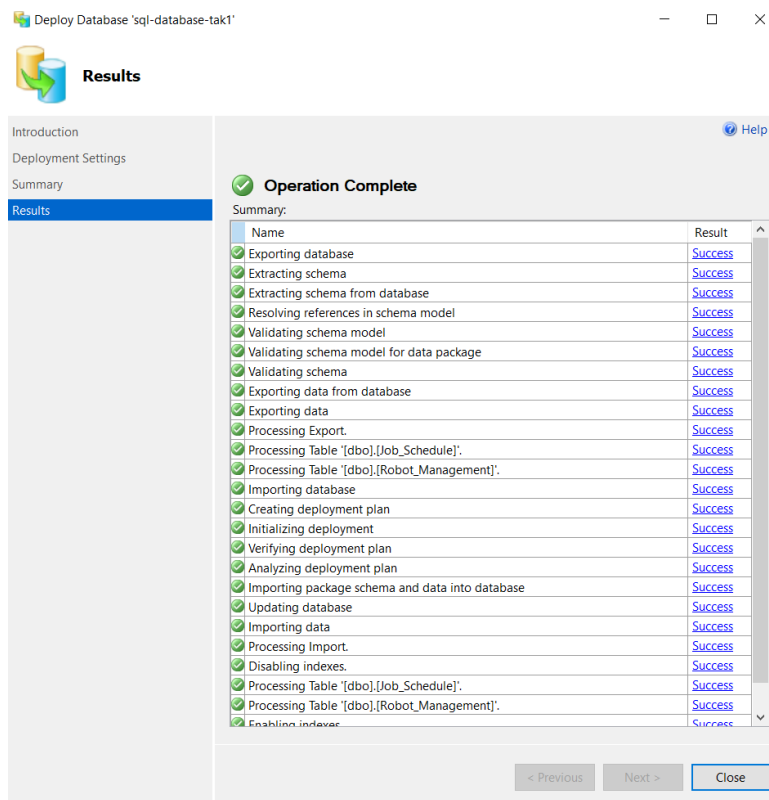
5.1.3 Lưu trữ dữ liệu vào cơ sở dữ liệu SQL theo mô hình điện toán đám mây

Cơ sở dữ liệu SQL theo mô hình điện toán đám mây (tên tiếng Anh là Cloud SQL Database), là một hệ thống tập hợp các dữ liệu tổ chức theo một cấu trúc nhất định và được lưu trữ trên đám mây. Để làm việc với cơ sở dữ liệu, trước hết cần có hệ quản trị cơ sở dữ liệu. Em đã hiện thực điều này bằng phần mềm SQL Server Management Studio và triển khai dữ liệu lên dịch vụ cơ sở dữ liệu đám mây được cung cấp bởi Microsoft Azure (tên tiếng Anh là Azure Cloud Database).

Cấu trúc thông tin của hệ thống robot này được tổ chức trong cơ sở dữ liệu như sau:



Hình 5.2: Tổ chức dữ liệu trong cơ sở dữ liệu SQL



Hình 5.3: Triển khai cơ sở dữ liệu lên đám mây Azure

Mặc định trong quá trình thiết lập kết nối, giao thức truyền thông được SQL Server sử dụng là sự kết hợp giữa Transmission Control Protocol (TCP/IP) và Tabular Data Stream (TDS). Trong đó TCP/IP đóng vai trò là giao thức vận chuyển cơ bản trong kết nối giữa các ứng dụng truy vấn đến cơ sở dữ liệu, còn TDS là giao thức chịu trách nhiệm cho việc truy vấn thực sự đến cơ sở dữ liệu như thực hiện các lệnh SELECT, INSERT, UPDATE hay DELETE.

5.2 Phần mềm mô phỏng và giám sát robot

5.2.1 Giới thiệu về OpenGL và thư viện Assimp

- Giới thiệu về OpenGL:

OpenGL (được gọi đầy đủ là Open Graphic Library), chủ yếu được dùng như một API (được gọi đầy đủ là Application Programming Interface) cho phép người dùng các hàm và quy tắc cho phép người dùng phát triển các ứng dụng đồ họa chất lượng cao trên nhiều nền tảng khác nhau, được phát triển bởi Khronos Group.



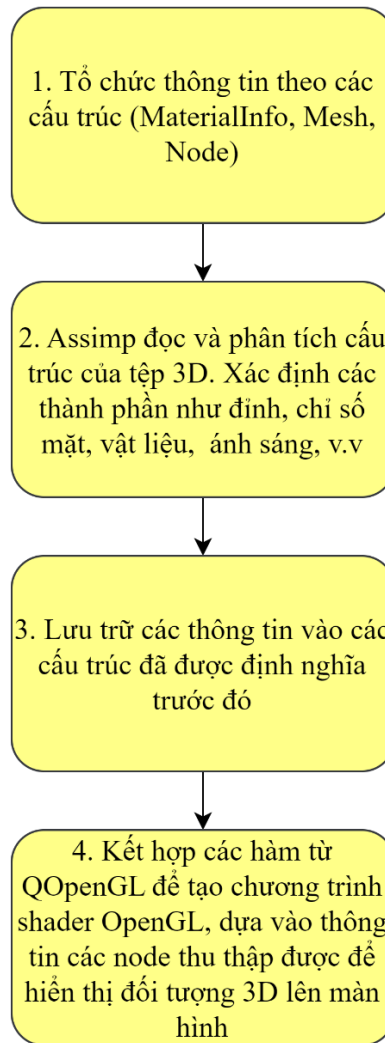
Hình 5.4: Thư viện OpenGL

Trong luận văn này, em sử dụng OpenGL với môi trường IDE Qt Creator, là một mã nguồn mở đa nền tảng hỗ trợ người dùng phát triển những dự án một cách dễ dàng và được cộng đồng hỗ trợ rất đáng tin cậy. Qt còn cung cấp một số hàm và lớp để làm việc với OpenGL mà không cần phụ thuộc vào các API như GLFW (OpenGL Framework) hay GLUT (OpenGL Utility Toolkit). Trong Qt, cơ chế hoạt động của Open GL được triển khai thông qua một số thành phần chính như:

- QOpenGLWidget: Là lớp con của QWidget, dùng để tạo cửa sổ hiển thị OpenGL, cung cấp một ngữ cảnh OpenGL để vẽ các đối tượng 3D.

- QOpenGLContext: Đại diện cho ngữ cảnh OpenGL, lớp này chịu trách nhiệm chính tổng việc quản lý tài nguyên OpenGL như các buffer, texture và shader program.
- QOpenGLFunctions: Cung cấp một số hàm OpenGL cơ bản.
- QOpenGLShaderProgram: Dùng để tải, biên dịch và liên kết các chương trình shader OpenGL, bao gồm các vertex shader, fragment shader và geometry shader.
- Giới thiệu về thư viện Assimp:

Thư viện Assimp (tên đầy đủ là Open Asset Import Library) là một thư viện mã nguồn mở được sử dụng để tải các tệp có định dạng 3D vào trong ứng dụng đồ họa. Thư viện hỗ trợ nhiều định dạng tệp phổ biến như OBJ, STL, v.v. Cơ chế hoạt động của thư viện Assimp:



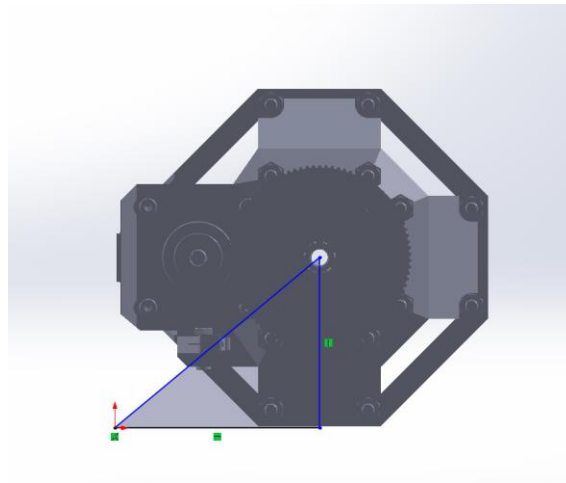
Hình 5.5: Cơ chế hoạt động của thư viện Assimp

5.2.2 Thiết kế phần mềm mô phỏng

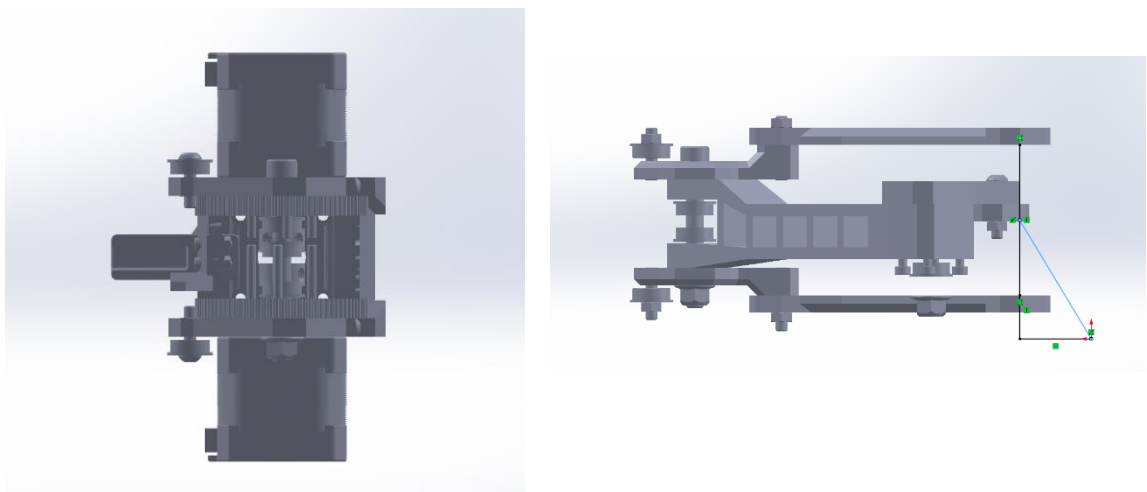
Phần mềm được phát triển có đầy đủ những tính năng cơ bản trong việc mô phỏng trực tuyến và ngoại tuyến cho robot hoạt động. Quá trình xây dựng phần mềm mô phỏng cần trải qua các bước: Đo đạc robot, đo đạc file 3D của robot; Tính toán động học thuận và nghịch cho robot; Phát triển phần động học của robot và hoạch định quỹ đạo trên Qt Creator bằng ngôn ngữ C++; Phát triển các tính năng cơ

bản khác của một phần mềm (có thể thay đổi góc nhìn, vẽ đồ thị, trực quan hóa đường đi của End Effector, hoạch định quỹ đạo offline)

- Đo đạc robot: Em sử dụng phần mềm SolidWorks 2018 và Blender khi làm việc với các file 3D của robot trong luận văn. Phần mềm SolidWorks giúp em tách biệt các phần riêng biệt của Robot, và cho kết quả đo đạc để phục vụ việc tính toán động học sau này. Phần mềm Blender giúp em định lại tọa độ của các phần của robot.

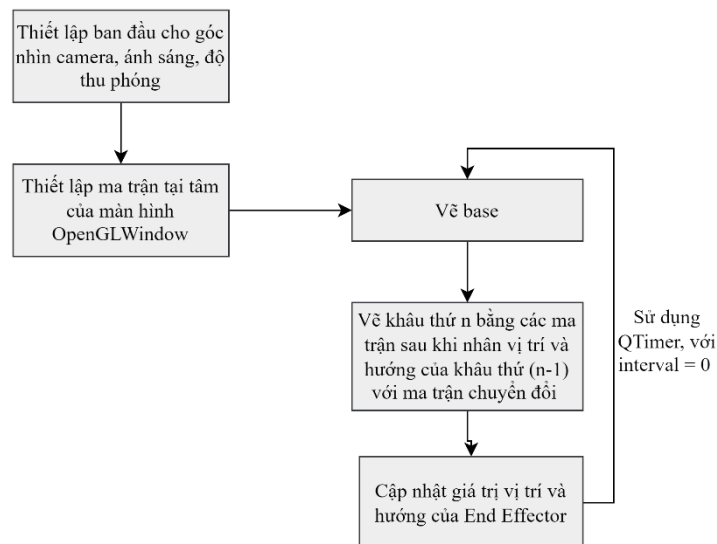


Hình 5.6: Đo kích thước của base



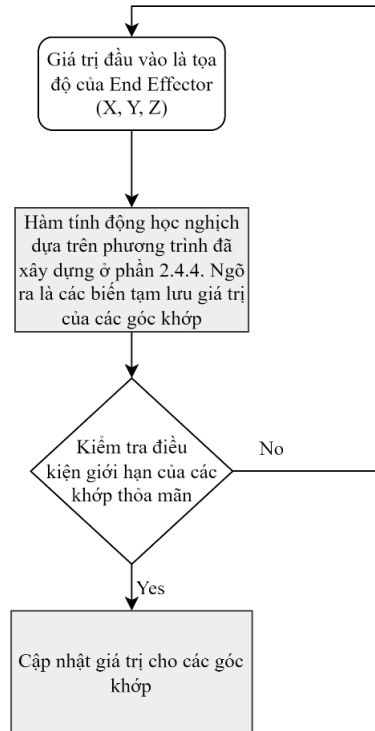
Hình 5.7: Đo kích thước của các khâu

- Tính toán động học thuận và nghịch của robot, hoạch định quỹ đạo robot:
Được trình bày trong phần 2.4.3, 2.4.4 và 2.4.5.
- Phát triển phần động học thuận, nghịch và hoạch định quỹ đạo của robot bằng ngôn ngữ C++.



Hình 5.8: Các bước vẽ mô hình 3D thực hiện chức năng động học thuận

- Trong Qt, QTimer là một lớp để tạo và quản lý các sự kiện thời gian (timers). Nếu không truyền tham số Interval, hoặc Interval bằng 0, QTimer sẽ phát tín hiệu timeout ngay lập tức khi gọi start() và không có khoảng thời gian đợi. Điều này có nghĩa là tín hiệu timeout sẽ được phát ra ngay sau lệnh start().



Hình 5.9: Các bước vẽ mô hình 3D thực hiện chức năng động học nghịch

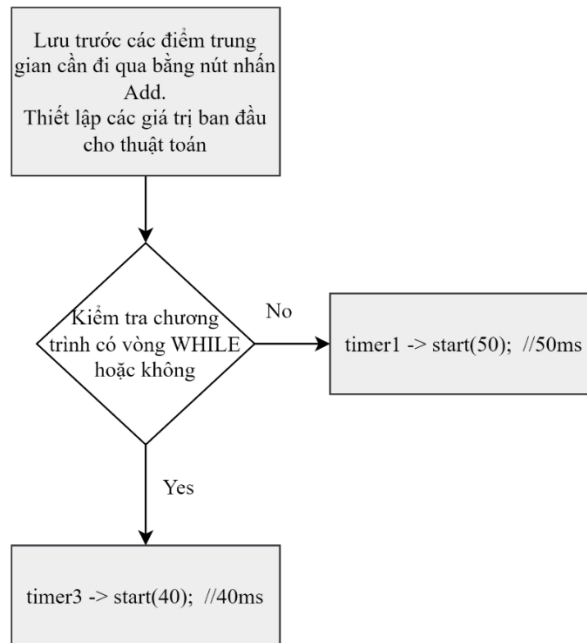
Phần quy hoạch đường thẳng LSPB, người dùng có thể thêm các điểm robot đi qua bằng nút nhấn 'Add'. Giải thuật quy hoạch sẽ dựa trên công thức được xây dựng ở phần 2.4.5. Mặc định chương trình sẽ quy hoạch đường thẳng qua hai điểm được chọn. Trong trường hợp mô phỏng mong muốn robot di chuyển liên tục qua nhiều điểm, người dùng có thể viết một chương trình để phần mềm thực hiện theo với các tập lệnh như sau:

Bảng 5.1: Tập lệnh của chương trình mô phỏng

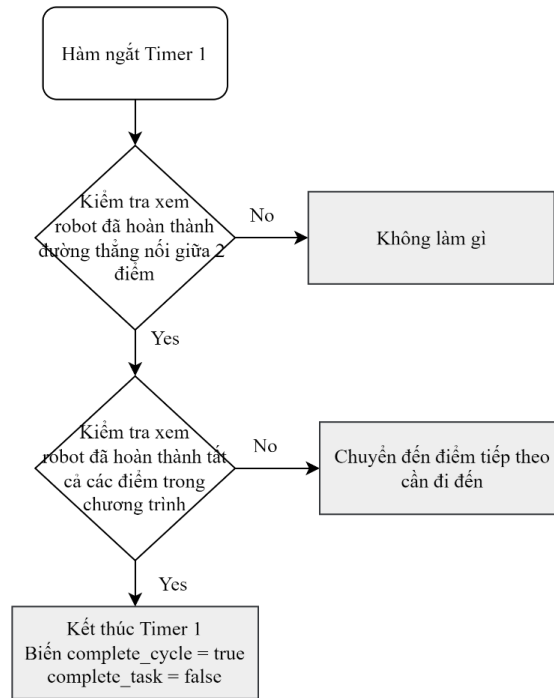
Câu lệnh	Chức năng
WHILE 1	Bắt đầu vòng lặp
ENDWL	Kết thúc vòng lặp
LMOVE P[i]	Quy hoạch robot theo đường thẳng từ vị trí hiện tại đến vị trí điểm P[i]

EOP	Kết thúc chương trình
-----	-----------------------

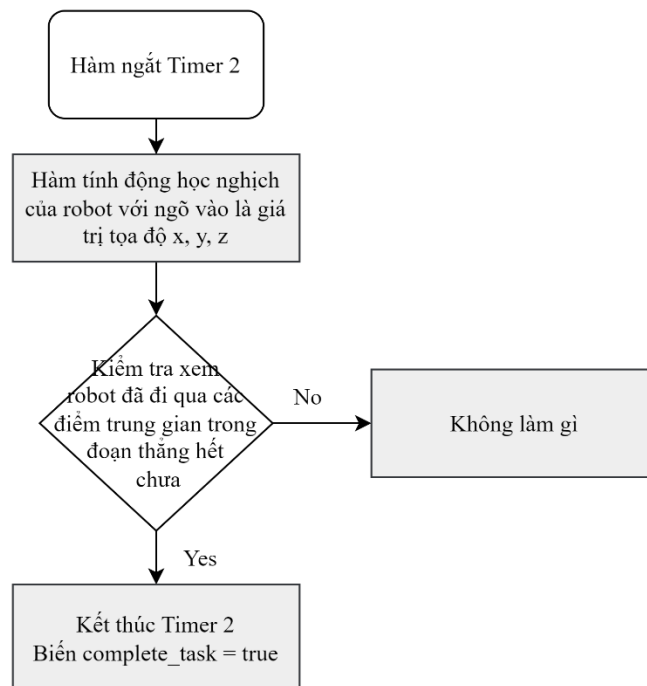
Giải thuật được kiểm soát bởi ba ngắt QTimer, ngắt Timer2 có nhiệm vụ đảm bảo Robot quy hoạch theo quỹ đạo thẳng giữa 2 điểm, ngắt Timer1 có nhiệm vụ đảm bảo Robot đi quy tất cả các điểm được đề cập trong chương trình, ngắt Timer3 có nhiệm vụ lặp lại chương trình nếu vòng WHILE xuất hiện.



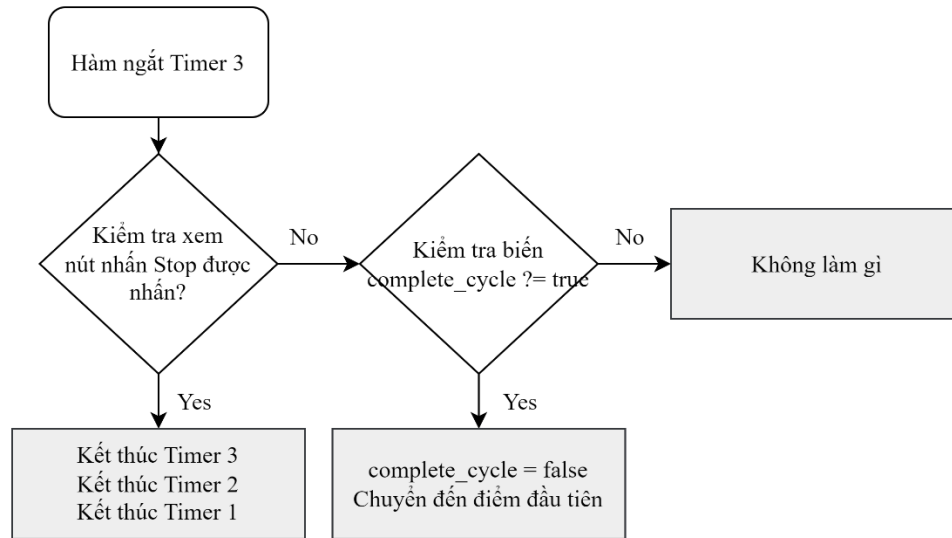
Hình 5.10: Các bước thực hiện chức năng viết chương trình mô phỏng, hoạch định quỹ đạo LSPB cho mô hình mô phỏng



Hình 5.11: Hàm ngắt Timer 1



Hình 5.12: Hàm ngắt Timer 2



Hình 5.13: Hàm ngắt Timer 3

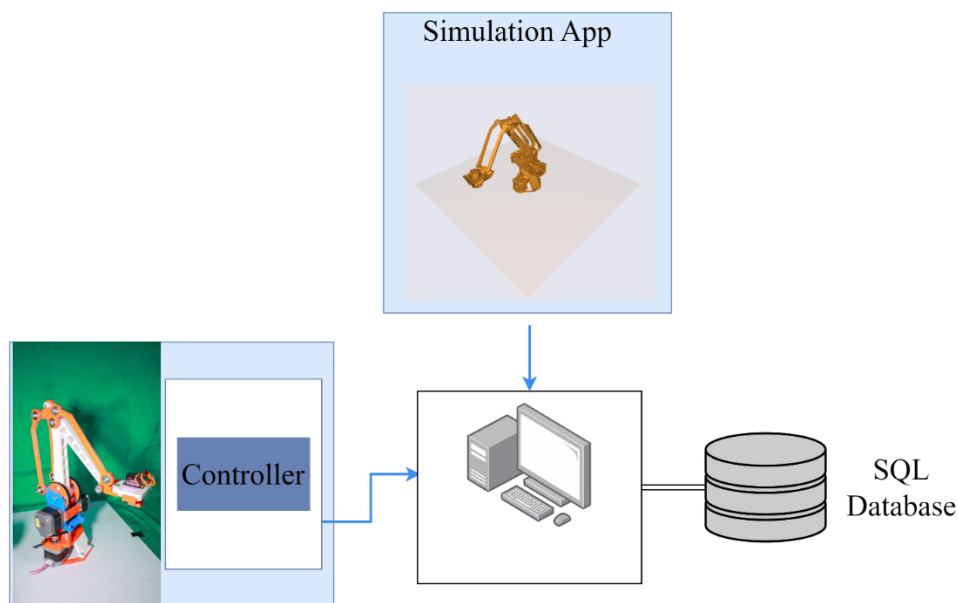
- Phát triển các tính năng cơ bản khác của một phần mềm (có thể thay đổi góc nhìn, vẽ đồ thị, trực quan hóa đường đi của End Effector, hoạch định quỹ đạo offline) Em đã phát triển một số tính năng cơ bản như trên để phục vụ người dùng có thể sử dụng phần mềm dễ dàng hơn bằng các hàm và lớp được cung cấp bởi Qt Creator như QMouseEvent, phát hiện tương tác bằng chuột để kéo thay thả đổi góc nhìn, thu phóng và QCustomPlot để hỗ trợ vẽ đồ thị các biến khớp.

5.2.3 Chức năng giám sát và mô phỏng

Phần mềm mô phỏng đóng vai trò như một OPC UA Client, kết nối đến phần mềm OPC UA Server trong cấu trúc chung của hệ thống để trao đổi dữ liệu. Đối với vai trò này của phần mềm mô phỏng, em sử dụng open62541 như là một thư viện mã nguồn mở để hỗ trợ xây dựng tính năng OPC UA. Thư viện open62541 được hỗ trợ và sử dụng bởi một cộng đồng người dùng khá lớn, do đó những tính năng của thư viện cũng được cập nhật liên tục và hiệu quả để phát triển.

Hình 5.14: Thư viện mã nguồn mở open62541

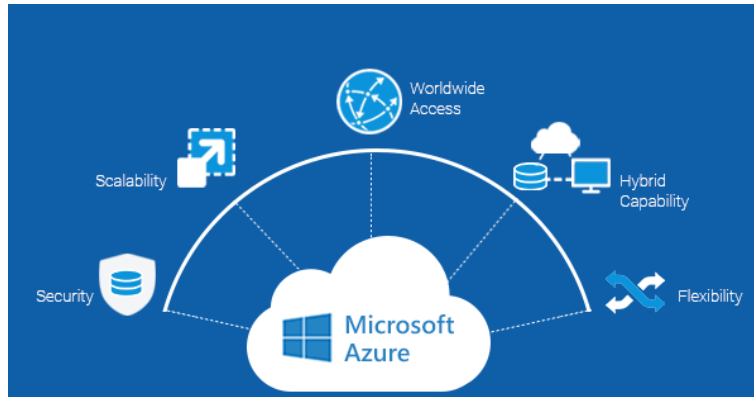
Như đã trình bày ở phần tổ chức thông tin trong cơ sở dữ liệu và phần mềm OPC UA Server, OPC UA Server khởi tạo và lưu trữ giá trị các biến khớp trong quá trình robot hoạt động, phần mềm mô phỏng khi truy cập đến các giá trị này thông qua OPC UA Server có thể cập nhật giá trị biến khớp tức thời và mô phỏng trực tuyến lại hành vi của robot.



Hình 5.15: Sơ đồ thể hiện vị trí của phần mềm mô phỏng trong hệ thống

CHƯƠNG 6. WEB APP GIÁM SÁT VÀ LẬP LỊCH ĐIỀU KHIỂN ROBOT

6.1 Giới thiệu về Azure Cloud



Hình 6.1: Dịch vụ của Microsoft Azure trên mô hình điện toán đám mây

Azure Cloud là một nền tảng điện toán đám mây được Microsoft phát triển, và nó cung cấp một loạt các dịch vụ trên đám mây linh hoạt, mạnh mẽ và dễ dàng mở rộng. Nền tảng Azure cho phép doanh nghiệp và các người dùng cá nhân xây dựng, triển khai và quản lý ứng dụng của mình bằng rất nhiều tính năng và dịch vụ quan trọng như:

- Máy ảo (tiếng Anh là Virtual Machines): Khả năng tạo máy ảo linh hoạt, cho phép người dùng chạy các ứng dụng và hệ điều hành khác nhau trên môi trường đám mây.
- Dịch vụ Đa phương tiện (tiếng Anh là Media Service): Dịch vụ cho phép lưu trữ, mã hóa, xử lý và phân phối nội dung đa phương tiện như video, âm thanh và hình ảnh.
- Dịch vụ Cơ sở dữ liệu (tiếng Anh là Database Service): Cơ sở dữ liệu được quản lý và lưu trữ trên đám mây, bao gồm cơ sở dữ liệu SQL, cơ sở dữ liệu Cosmos và các dịch vụ quản lý cơ sở dữ liệu khác.
- Dịch vụ Internet vạn vật (tiếng Anh là Internet of Things IoT): Azure cung cấp dịch vụ IoT cho phép người dùng kết nối, giám sát và quản lý các thiết bị IoT,

đồng thời còn có khả năng thu thập, phân tích và sử dụng dữ liệu từ các thiết bị kết nối.

- Ngoài ra, người dùng cũng có thể mở rộng Dịch vụ AI bằng cách sử dụng các mô hình AI Azure Machine Learning (máy học Azure).

Azure Cloud hỗ trợ nhiều ngôn ngữ lập trình và nền tảng, bao gồm Node.js, Java, Python, .NET, v.v. Sử dụng Azure Cloud giúp giảm thiểu chi phí vận hành và bảo trì hệ thống, tính năng bảo mật và độ tin cậy cao, do đó Azure Cloud là một lựa chọn lý tưởng cho các doanh nghiệp và tổ chức muốn triển khai ứng dụng và dịch vụ của mình trên đám mây một cách an toàn và hiệu quả.

6.2 Thiết kế phần mềm web giám sát và lập lịch điều khiển

6.2.1 Giới thiệu về Node.js và Express.js

Node.js, hay còn được gọi là Node, là một mã nguồn mở đa nền tảng và là một môi trường chạy mã JavaScript. Thông thường Node.js được sử dụng để phát triển phần lớp dưới (back-end services) hay còn được gọi là API (Application Programming Interface). Phần lớp dưới này được chạy trên máy chủ, điều khiển luồng dữ liệu và tương tác với các ứng dụng như Web App chạy trên một trình duyệt web, hoặc một ứng dụng Mobile App chạy trên thiết bị di động mà tại đó cung cấp giao diện để người dùng tương tác.

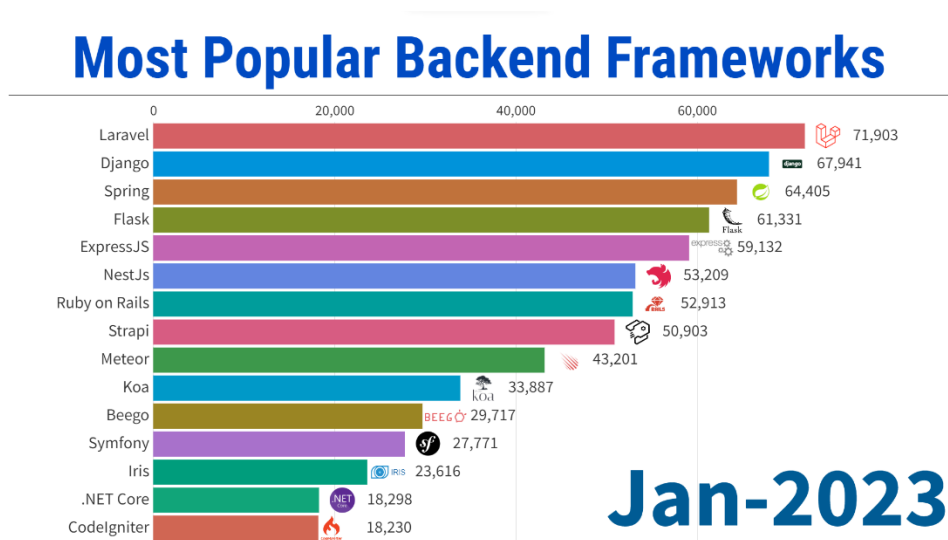


Hình 6.2: Mã nguồn mở Node.js

Node.js là một lựa chọn phù hợp để phát triển các ứng dụng yêu cầu sự mở rộng, xử lý dữ liệu thời gian thực một cách hiệu quả. Thực tế, Node được sử dụng bởi một loạt các công ty lớn như PayPal, Uber, Netflix, v.v. Một trong những ưu điểm của

Node.js là sự hỗ trợ mô hình lập trình không đồng bộ (tiếng Anh là Asynchronous programming), giúp xử lý hàng ngàn kết nối đồng thời mà không gây tắc nghẽn.

Express.js là một framework được phát hành lần đầu tiên vào năm 2010, được thiết kế để làm việc tốt với Node.js. Express.js có thể xem là một framework được bổ sung cho Node.js giúp người dùng đơn giản hóa các ứng dụng web của server với nhiều module hơn và công cụ tiện ích hơn. Trong thực tế, Express.js được sử dụng nhiều thứ 5 trong danh sách những framework dùng để phát triển lớp dưới theo thống kê của trang statisticsanddata.org vào tháng 1 năm 2023.

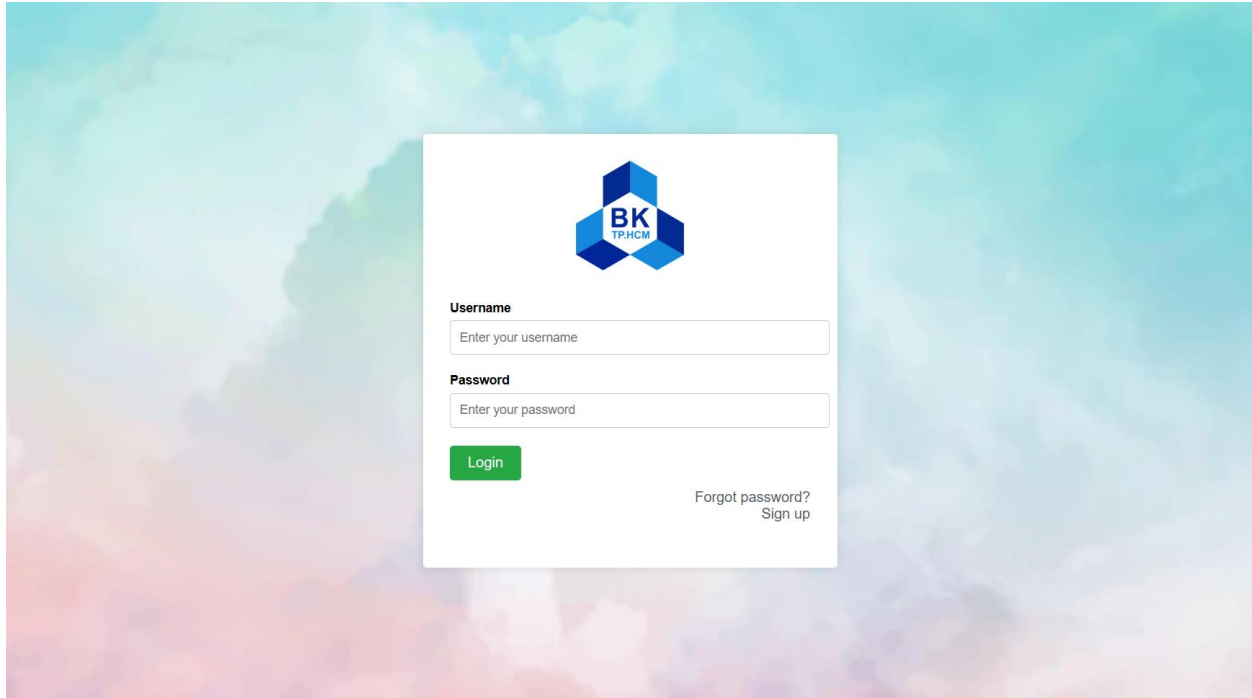


Hình 6.3: Thống kê những framework backend được sử dụng phổ biến nhất vào tháng 1 năm 2023

6.2.2 Thiết kế giao diện người dùng

- Trang đăng nhập:

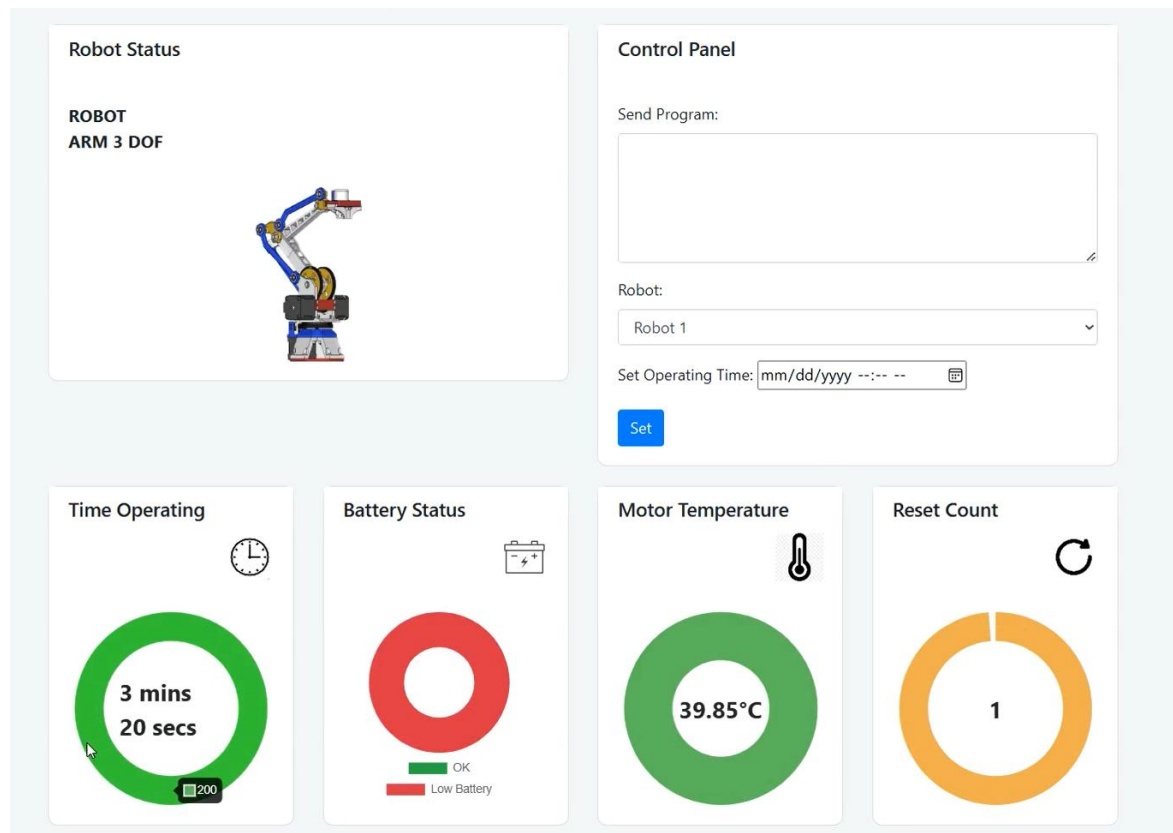
Trang đăng nhập là trang đầu tiên hiển thị cho người dùng khi truy cập đến website. Sau khi nhấn nút đăng nhập, web sẽ điều hướng đến trang màn hình chính dashboard. Trong trang còn có các phần Quên mật khẩu (Forgot password) và Đăng ký (Sign up) nhưng chưa được thực hiện.



Hình 6.4: Trang đăng nhập

- Trang Dashboard: Sau khi đăng nhập thành công, web sẽ chuyển hướng về trang Dashboard, là giao diện chính của Web App. Trang Dashboard có chứa các thông số vận hành tổng quát của robot. Cụ thể các thông số vận hành của robot và phần lập lịch như:
 - Trạng thái vận hành của Robot (Robot Status): Có bốn trạng thái vận hành của một robot bao gồm Vận hành (Run), Tạm dừng (Pause), Dừng (Stop) và Cảnh báo(Alarm).
 - Trạng thái nguồn cấp cho robot (Battery Status): Giám sát trạng thái nguồn điện cấp cho robot hoạt động đang ở mức ổn định (OK ~ 12V) hoặc mức điện yếu (Low battery < 5V).
 - Nhiệt độ động cơ (Motor Temperature): Giám sát nhiệt độ hoạt động của động cơ.
 - Đếm số lần Reset robot (Reset Count): Có thể phục vụ quá trình sửa lỗi cho robot, đo thời gian robot hoạt động ổn định.

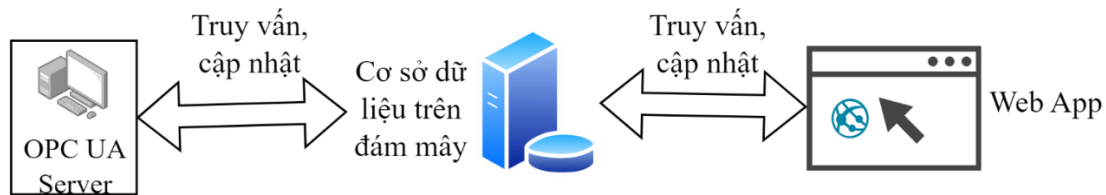
- Gửi lệnh để lập lịch điều khiển robot từ xa (Send program và Set Operating Time): Lựa chọn các nhiệm vụ để lập lịch cho robot hoạt động từ thời điểm bắt đầu đến thời điểm kết thúc.



Hình 6.5: Trang Dashboard

6.2.3 Cấu trúc truyền nhận dữ liệu

Cấu trúc truyền nhận dữ liệu của Web App với OPC UA Server được tổ chức như sau:



Hình 6.6: Cấu trúc truyền nhận dữ liệu giữa Web App và OPC UA Server

Em sử dụng cơ sở dữ liệu đã được triển khai trước đó trên đám mây của Azure Microsoft như một kết nối trung gian giữa Web App và phần mềm OPC UA Server ở mạng cục bộ. Cụ thể trong Web App, Web App sẽ gửi yêu cầu đến bên Node.js server trong Web App, từ đó truy xuất đến cơ sở dữ liệu trên đám mây. Trong thực tế, sử dụng cơ sở dữ liệu trên đám mây để lưu trữ và truyền đạt dữ liệu là một phương pháp phổ biến thường được sử dụng trong ứng dụng IoT và giám sát công nghiệp.

- Ưu điểm: Đơn giản để triển khai, nền tảng Azure Microsoft mạnh mẽ và linh hoạt, dễ dàng điều chỉnh tài nguyên theo yêu cầu của ứng dụng, đảm bảo hệ thống có khả năng chịu tải mà không gặp vấn đề về hiệu suất, khi sử dụng dịch vụ của Azure, quá trình bảo mật và xác thực sẽ đảm bảo tính đáng tin cậy và toàn vẹn của dữ liệu.
- Nhược điểm: Gặp vấn đề với độ trễ trong quá trình truyền nhận dữ liệu cho đến giai đoạn hiển thị trên trình duyệt cho người dùng vì phải trải qua các bước truy vấn, yêu cầu, nhận phản hồi. Cụ thể độ trễ phụ thuộc vào rất nhiều yếu tố như tốc độ kết nối Internet, thời gian xử lý từ OPC UA Server, quy mô của hệ thống, v.v.

CHƯƠNG 7. KẾT QUẢ, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

7.1 Kết quả thực hiện

7.1.1 Điều khiển cánh tay robot 3 bậc tự do

- Điều khiển được robot thực hiện được một số nhiệm vụ cơ bản: di chuyển tự do đến các vị trí cho trước, đóng mở cánh tay gấp.



Hình 7.1 Điều khiển robot di chuyển tự do đến các điểm cho trước

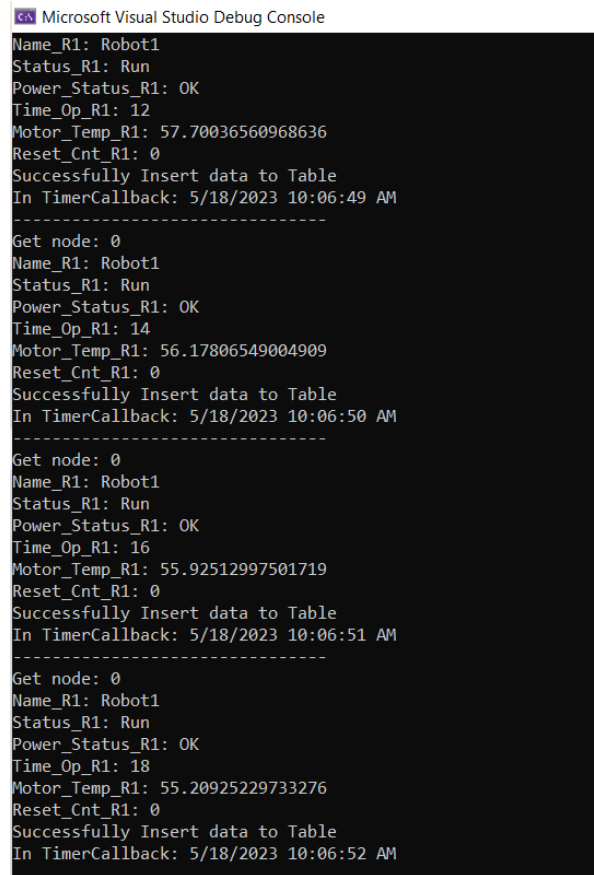
7.1.2 Giám sát, quản lý robot từ xa bằng Web App trên đám mây

- Khi khởi động phần mềm OPC UA Server, OPC UA Server khởi tạo giá trị các node dữ liệu ban đầu về giá trị mặc định.

```
Microsoft Visual Studio Debug Console
OPC UA Server is ready...
Starting server...In TimerCallback: 5/18/2023 10:00:43 AM
-----
Get node: 0
Name_R1: Robot1
Status_R1:
Power_Status_R1:
Time_Op_R1: 0
Motor_Temp_R1: 0
Reset_Cnt_R1: 0
Successfully Insert data to Table
In TimerCallback: 5/18/2023 10:00:44 AM
-----
Get node: 0
Name_R1: Robot1
Status_R1:
Power_Status_R1:
Time_Op_R1: 0
Motor_Temp_R1: 0
Reset_Cnt_R1: 0
Successfully Insert data to Table
In TimerCallback: 5/18/2023 10:00:45 AM
-----
Get node: 0
Name_R1: Robot1
Status_R1:
Power_Status_R1:
Time_Op_R1: 0
Motor_Temp_R1: 0
Reset_Cnt_R1: 0
Successfully Insert data to Table
done.
Press any key to continue...In TimerCallback: 5/18/2023 10:00:46 AM
-----
```

Hình 7.2 Phần mềm OPC UA Server khởi tạo các node

- Khi kết nối đến cánh tay robot, các giá trị node sẽ được cập nhật tức thời với chu kỳ ngắt timer 1s.



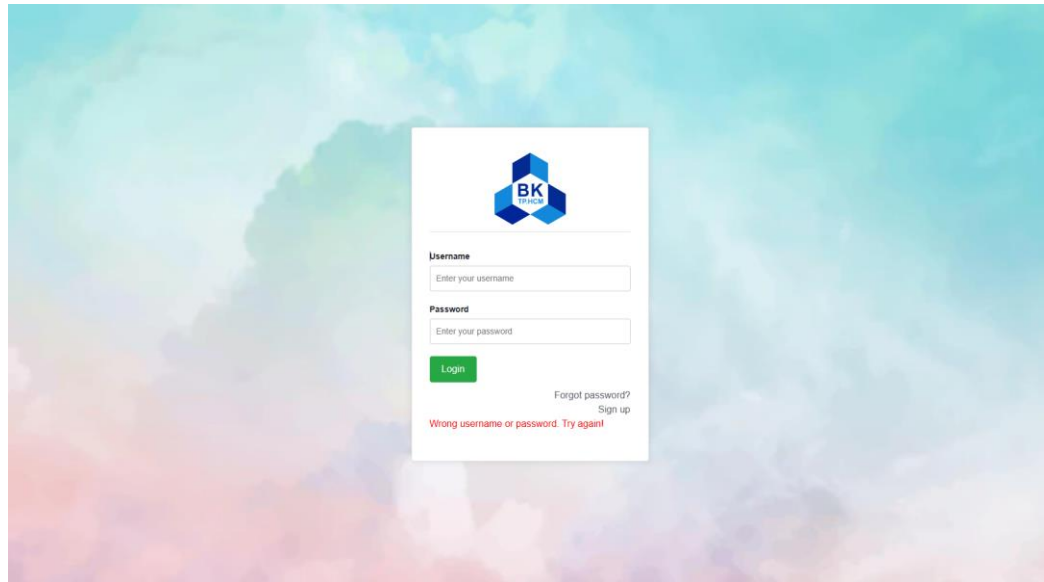
```

Microsoft Visual Studio Debug Console
Name_R1: Robot1
Status_R1: Run
Power_Status_R1: OK
Time_Op_R1: 12
Motor_Temp_R1: 57.70036560968636
Reset_Cnt_R1: 0
Successfully Insert data to Table
In TimerCallback: 5/18/2023 10:06:49 AM
-----
Get node: 0
Name_R1: Robot1
Status_R1: Run
Power_Status_R1: OK
Time_Op_R1: 14
Motor_Temp_R1: 56.17806549004909
Reset_Cnt_R1: 0
Successfully Insert data to Table
In TimerCallback: 5/18/2023 10:06:50 AM
-----
Get node: 0
Name_R1: Robot1
Status_R1: Run
Power_Status_R1: OK
Time_Op_R1: 16
Motor_Temp_R1: 55.92512997501719
Reset_Cnt_R1: 0
Successfully Insert data to Table
In TimerCallback: 5/18/2023 10:06:51 AM
-----
Get node: 0
Name_R1: Robot1
Status_R1: Run
Power_Status_R1: OK
Time_Op_R1: 18
Motor_Temp_R1: 55.20925229733276
Reset_Cnt_R1: 0
Successfully Insert data to Table
In TimerCallback: 5/18/2023 10:06:52 AM
-----

```

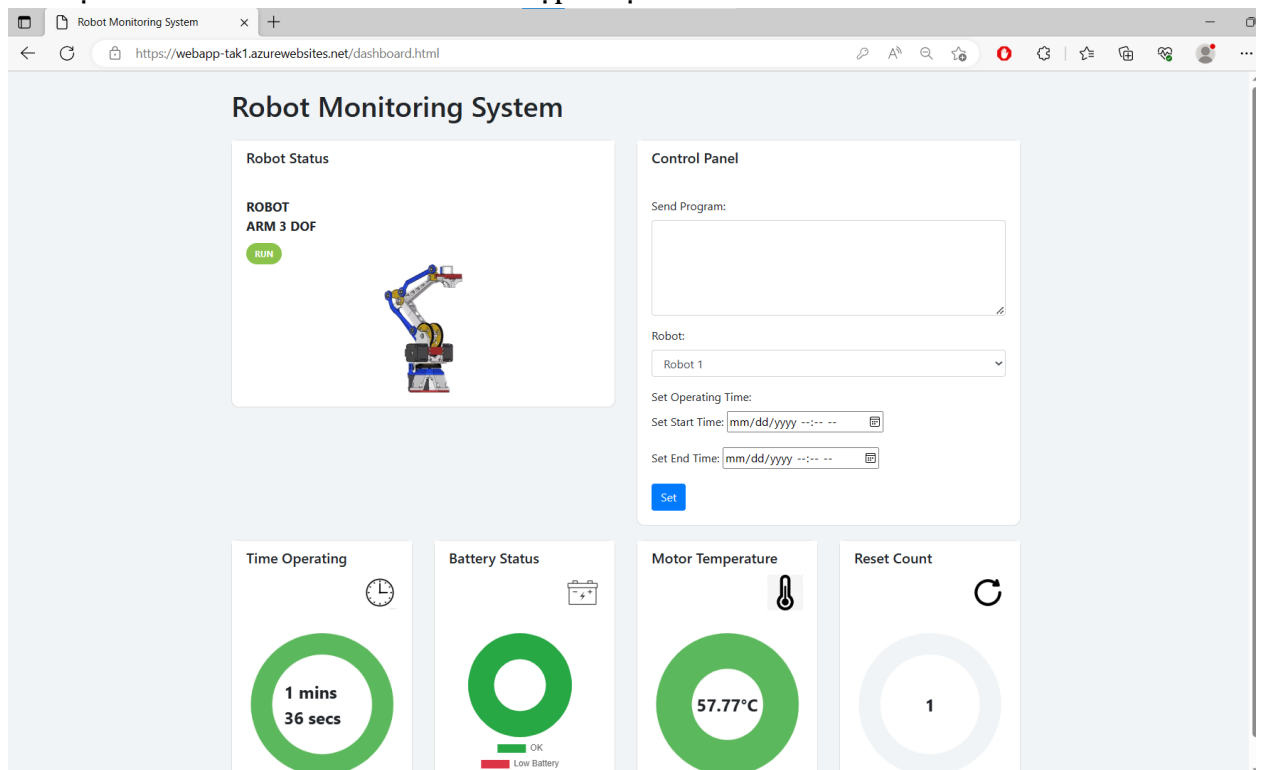
Hình 7.3: Phần mềm OPC UA Server chạy ổn định khi thu thập trạng thái hoạt động của robot

- Trên phần mềm Web App, ban đầu đăng nhập vào hệ thống, nếu đăng nhập sai hệ thống sẽ báo lỗi và yêu cầu đăng nhập lại.



Hình 7.4: Giao diện đăng nhập của Web App

- Nếu đăng nhập thành công, Web App sẽ tự động chuyển hướng đến trang dashboard, và các thông số vận hành của robot cũng được hiển thị, đúng với dữ liệu của OPC UA Server đã thu thập được từ robot.



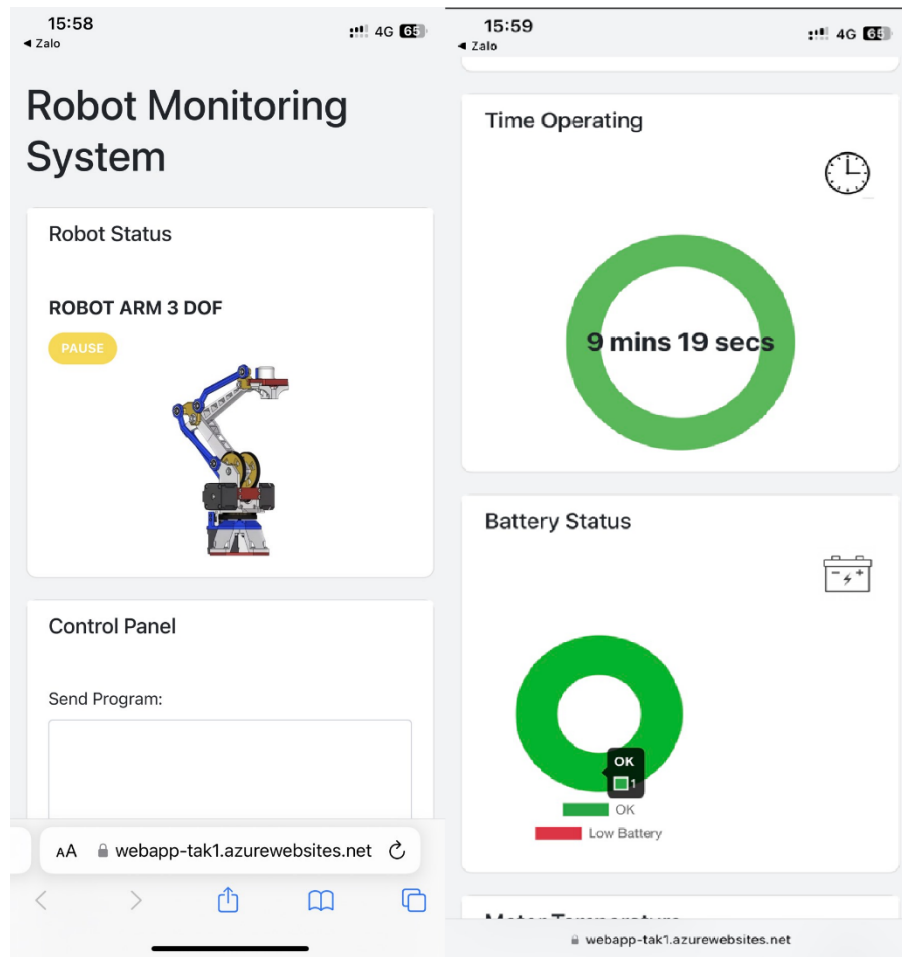
Hình 7.5: Giao diện Dashboard của Web App

- Tại trang dashboard, người dùng có thể thiết lập thời gian hoạt động theo các nhiệm vụ cho robot, khi thiết lập thành công, danh sách công việc sẽ được lưu trữ vào bảng Job_Schedule trong cơ sở dữ liệu.

Results		Messages		
	Robot	Program	Time_Start	Time_End
1	Robot 1	Task1	2023-05-04 10:18:00.000	2023-05-04 11:18:00.000
2	Robot 1	Task2	2023-05-05 16:20:00.000	2023-05-05 17:30:00.000
3	Robot 1	Task3	2023-05-05 17:30:00.000	2023-05-05 17:40:00.000

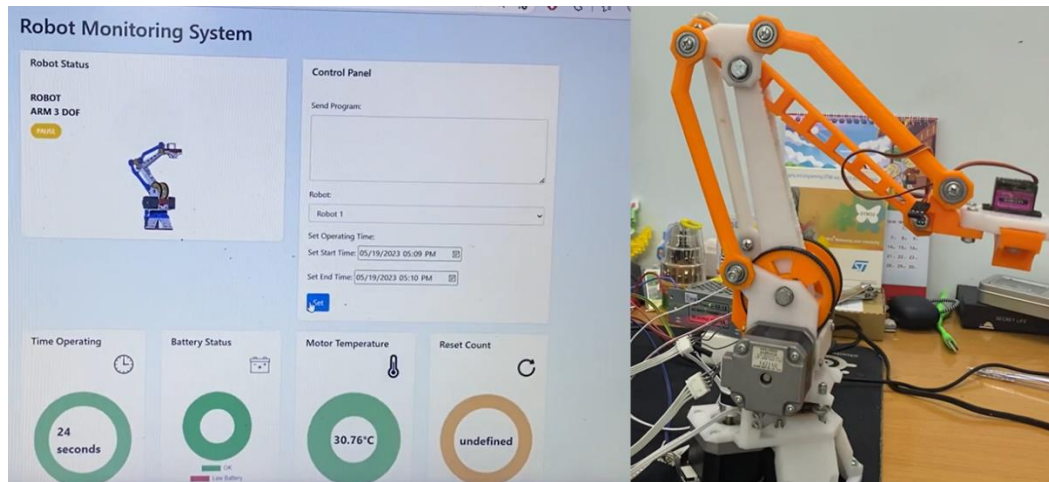
Hình 7.6: Danh sách các nhiệm vụ của robot được lưu trong cơ sở dữ liệu

- Trên các thiết bị di động có Internet cũng có thể truy cập được miễn là có kết nối Internet.



Hình 7.7: Web App hoạt động ổn định trên điện thoại di động, truy cập từ một mạng bất kỳ

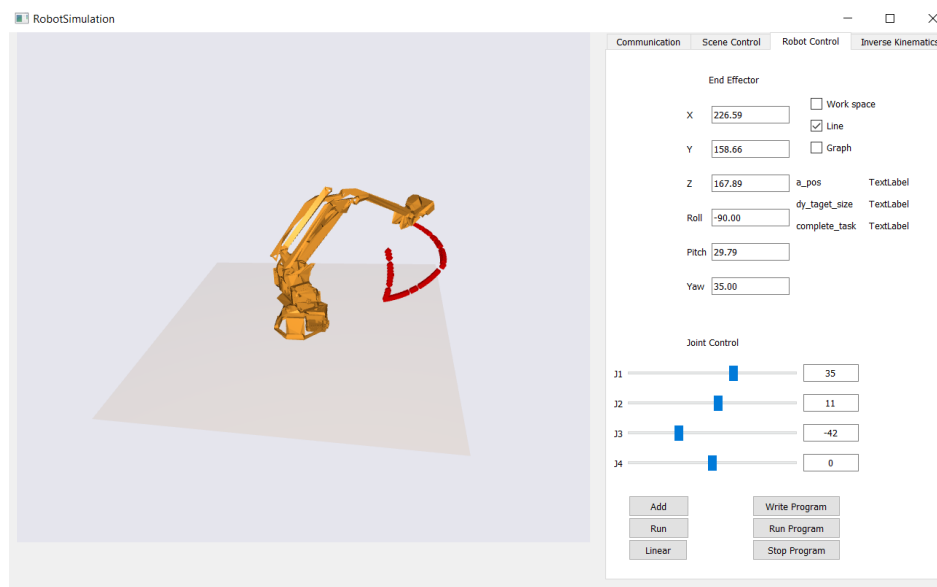
- Nếu đúng thời điểm được thiết lập, cánh tay robot sẽ vận hành với nhiệm vụ tương ứng.



Hình 7.8 Lập thời gian bắt đầu và kết thúc nhiệm vụ cho robot từ Web App

7.1.3 Phần mềm giám sát kết hợp mô phỏng

- Phần mềm giám sát kết hợp mô phỏng được phát triển có thể đóng vai trò như một OPC UA Client, kết nối đến OPC UA Server để trao đổi dữ liệu.
- Phần mềm có thể thu phòng, thay đổi góc nhìn bằng chuột giúp người dùng làm việc trực quan hơn.
- Phần mềm có tab riêng cho phần động học thuận của robot và phần động học nghịch của robot.



Hình 7.9: Giao diện của phần mềm giám sát kết hợp mô phỏng

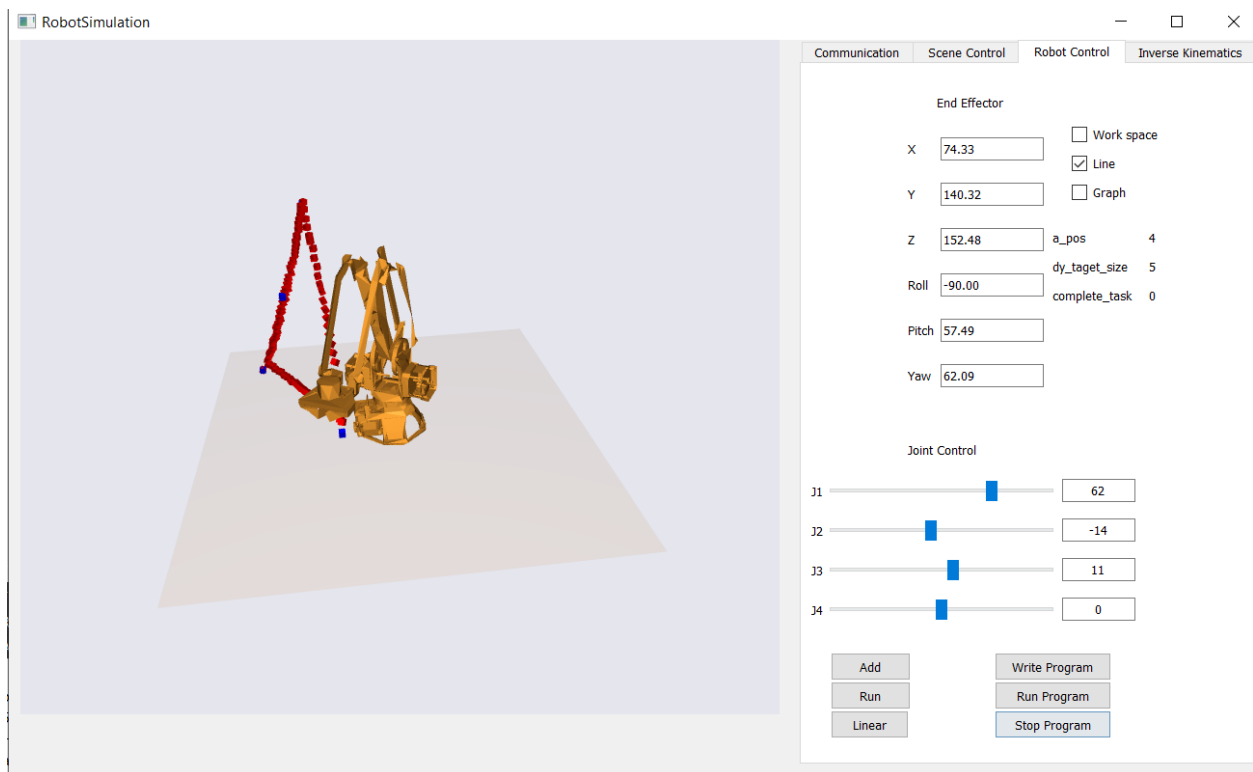
- Người dùng có thể tự lập trình mô phỏng offline cho robot di chuyển theo tập lệnh được định nghĩa ở bảng 5.1.
- Khi được lập trình với chương trình ở hình 7.8, kết quả hoạt động của phần mềm được trình bày ở hình 7.9.

```

MAIN M0
WHILE 1
LMOVE P[1]
LMOVE P[0]
LMOVE P[2]
LMOVE P[3]
ENDWL
EOP

```

Hình 7.10: Chương trình mẫu đơn giản



Hình 7.11: Kết quả chạy mô phỏng

7.2 Đánh giá

Kết quả thực hiện đề tài đã đáp ứng được các mục tiêu:

- Điều khiển được cánh tay robot thực hiện một số nhiệm vụ (di chuyển đến vị trí, gắp thả vật).
- Phát triển được một phần mềm máy tính có khả năng giám sát hoạt động của robot và có tích hợp tính năng mô phỏng.
- Thiết kế cơ sở dữ liệu và phát triển được ứng dụng Web chạy được trên nhiều nền tảng, có chức năng giám sát và lập lịch hoạt động cho cánh tay robot từ xa.

Hạn chế:

- Thời gian cập nhật dữ liệu lên Web App trải qua nhiều bước truy vấn và phản hồi nên có trễ.
- Chưa đưa ra được cảnh báo cho người sử dụng khi robot vận hành lỗi.
- Hệ thống chỉ có thể lập lịch hoạt động cho robot những nhiệm vụ được lập trình sẵn, chứ không có khả năng điều khiển và viết chương trình cho robot hoạt động.
- Chưa hoạch định được quỹ đạo LSPB cho cánh tay robot ba bậc mô hình thật, hiện tại chỉ hoạch định được cho mô hình mô phỏng.

7.3 Hướng phát triển

- Kết nối và giám sát đồng thời được nhiều robot hơn.
- Hoạch định quỹ đạo LSPB cho cánh tay robot mô hình thật, phát triển tính năng điều.
- Tích hợp khả năng lập trình và điều khiển cho robot hoạt động theo ý muốn bằng phần mềm.

TÀI LIỆU THAM KHẢO

- [1] Introduction to Node.js. [Introduction to Node.js \(nodejs.dev\)](https://nodejs.dev/en/about/getting-started/)
- [2] Learn Full Stack Development – HTML, CSS, JavaScript, Node.js, MongoDB
[Learn Full Stack Development – HTML, CSS, JavaScript, Node.js, MongoDB \(freecodecamp.org\)](https://www.freecodecamp.org/learn/full-stack-development/)
- [3] WebSockets and Node.js - testing WS and SockJS by building a web app
[WebSockets and Node.js - testing WS and SockJS by building a web app | Ably Blog: Data in Motion](https://ably.com/blog/websockets-and-nodejs-testing-ws-and-sockjs-by-building-a-web-app/)
- [4] Create a Node.js web app in Azure with Visual Studio Code and App Service
[Create a Node.js web app in Azure with Visual Studio Code and App Service - YouTube](https://www.youtube.com/watch?v=753...)
- [5] OPC UA SDK for .NET, Development Guides.
<https://docs.traeger.de/en/software/sdk/opc-ua/net/start#opc-ua-sdkfor-net>
- [6] Open62541, open62541 release 1.3 Documentation.
<https://www.open62541.org/doc/open62541-1.3.pdf>
- [7] OpenGL API Documentation Overview
[OpenGL News Archives](https://www.khronos.org/opengl/news/)
- [8] Qt and OpenGL: Loading a 3D Model with Open Asset Import Library (Assimp)
[Loading a 3D Model with Open Asset Import Library \(Assimp\) | ICS](https://www.ics.uni-stuttgart.de/~mayer/teaching/2019/qt-and-opengl/)