Mục lục

LỜI CÁM ƠN

LỜI MỞ ĐẦU

1. GIỚI THIỆU ĐỀ TÀI
   1. Mục đích và mục tiêu của luận văn
   2. Đối tượng và phạm vi nghiên cứu
      1. Đối tượng nghiên cứu
      2. Phạm vi nghiên cứu
   3. Cấu trúc của luận văn
2. TỔNG QUAN VỀ ROBOT VÀ HỆ THỐNG SCADA
   1. Khái niệm về robot công nghiệp
   2. Khái niệm về hệ thống SCADA
   3. Ứng dụng của IIoT cho robot
   4. Động học robot
      1. Một số khái niệm
      2. Biểu diễn Denavit – Hartenberg của bài toán động học robot
      3. Phương trình động học thuận của cánh tay robot 3 bậc tự do
      4. Phương trình động học nghịch của cánh tay robot 3 bậc tự do
      5. Hoạch định quỹ đạo
3. TỔNG QUAN VỀ CẤU TRÚC HỆ THỐNG
4. PHẦN CỨNG VÀ MẠCH ĐIỀU KHIỂN
   1. Cánh tay robot 3 bậc tự do
   2. Mạch điều khiển
   3. Thiết kế tủ điện
   4. Phần mềm trên board điều khiển
5. PHẦN MỀM MÔ PHỎNG VÀ GIÁM SÁT ROBOT
   1. Phần mềm OPC UA Server và OPC UA Client
      1. Giới thiệu về thư viện OPC UA Client & Server SDK
      2. Cấu trúc OPC UA Server đã thiết kế
      3. Lưu trữ dữ liệu vào cơ sở dữ liệu SQL theo mô hình điện toán đám mây
   2. Phần mềm mô phỏng và giám sát robot
      1. Giới thiệu về OpenGL và thư viện Assimp
      2. Thiết kế phần mềm mô phỏng
      3. Chức năng giám sát và mô phỏng
6. WEB APP GIÁM SÁT VÀ LẬP LỊCH ĐIỀU KHIỂN ROBOT
   1. Giới thiệu về Azure Cloud
   2. Thiết kế phần mềm web giám sát và lập lịch điều khiển
      1. Giới thiệu về Node.js và Express.js
      2. Thiết kế giao diện người dùng
      3. Cấu trúc truyền nhận dữ liệu
7. KẾT QUẢ, ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN
   1. Kết quả thực hiện
   2. Đánh giá
   3. Hướng phát triển

Danh mục hình vẽ

LỜI CẢM ƠN

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, Internet vạn vật công nghiệp (IIoT) 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 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 SCADA 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.

Chương 1

GIỚI THIỆU ĐỀ TÀI

* 1. Mục đích và mục tiêu của luận văn

Mục đích: Thiết kế một **hệ thống giám sát, mô phỏng và lập lịch điều khiển robot** hoàn chỉnh đến từ sự kết hợp giữa **phần cứng cánh tay robot có sẵn** và phần mềm, giúp người dùng có thể tương tác với máy tính trong các ứng dụng với robot. Ngoài ra, các phần mềm được thiết kế với **chức năng giám sát và lập lịch điều khiển cho robot từ xa trên các thiết bị có kết nối Internet**, giúp nâng cao năng suất làm việc của robot trong môi trường công nghiệp.

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

* Hệ thống có khả năng giám sát, mô phỏng và lập lịch điều khiển cho robot từ xa một cách hoàn chỉnh và ổn định.
* Điều khiển được cánh tay robot có sẵn hoạt động theo mong muốn một cách ổn định
* Thiết kế cơ sở dữ liệu và phần mềm với tính năng giám sát, mô phỏng thực tế.
* Lập trình robot cho một ứng dụng cụ thể trong công nghiệp.
  1. Đối tượng và phạm vi nghiên cứu
     1. Đối tượng nghiên cứu
* Cánh tay robot 3 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. 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 thức 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.
* 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. Cấu trúc của luận văn

Cấu trúc luận văn được trình bày như sau:

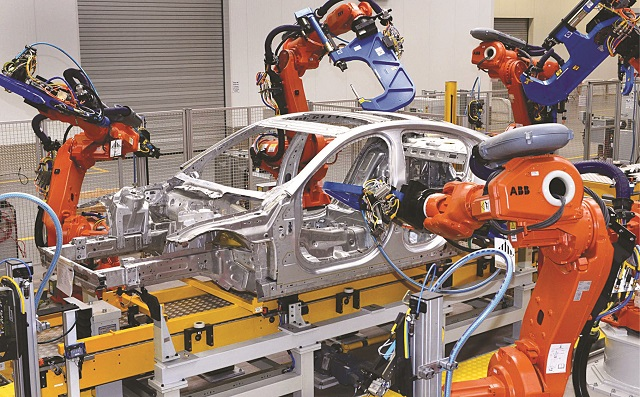
* Chương 1: Giới thiệu đề tài, trình bày khái quát về mục đích, mục tiêu của luận văn, và giới thiệu đối tượng nghiên cứu, phạm vi nghiên cứu của luận văn.
* Chương 2: Tổng quan về robot công nghiệp, hệ thống SCADA cho robot công nghiệp và ứng dụng của IIoT trong các hệ thống robot, các khái niệm robot liên quan và các công nghệ đang được sử dụng.
* 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

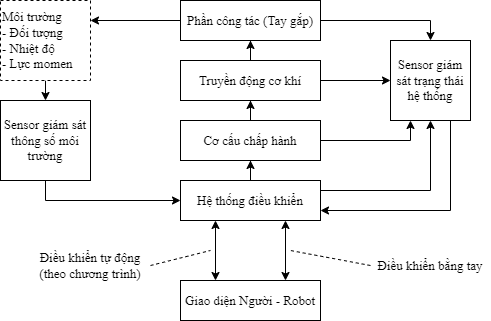
2.1 Khái niệm về robot công nghiệp

Robot công nghiệp (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.



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

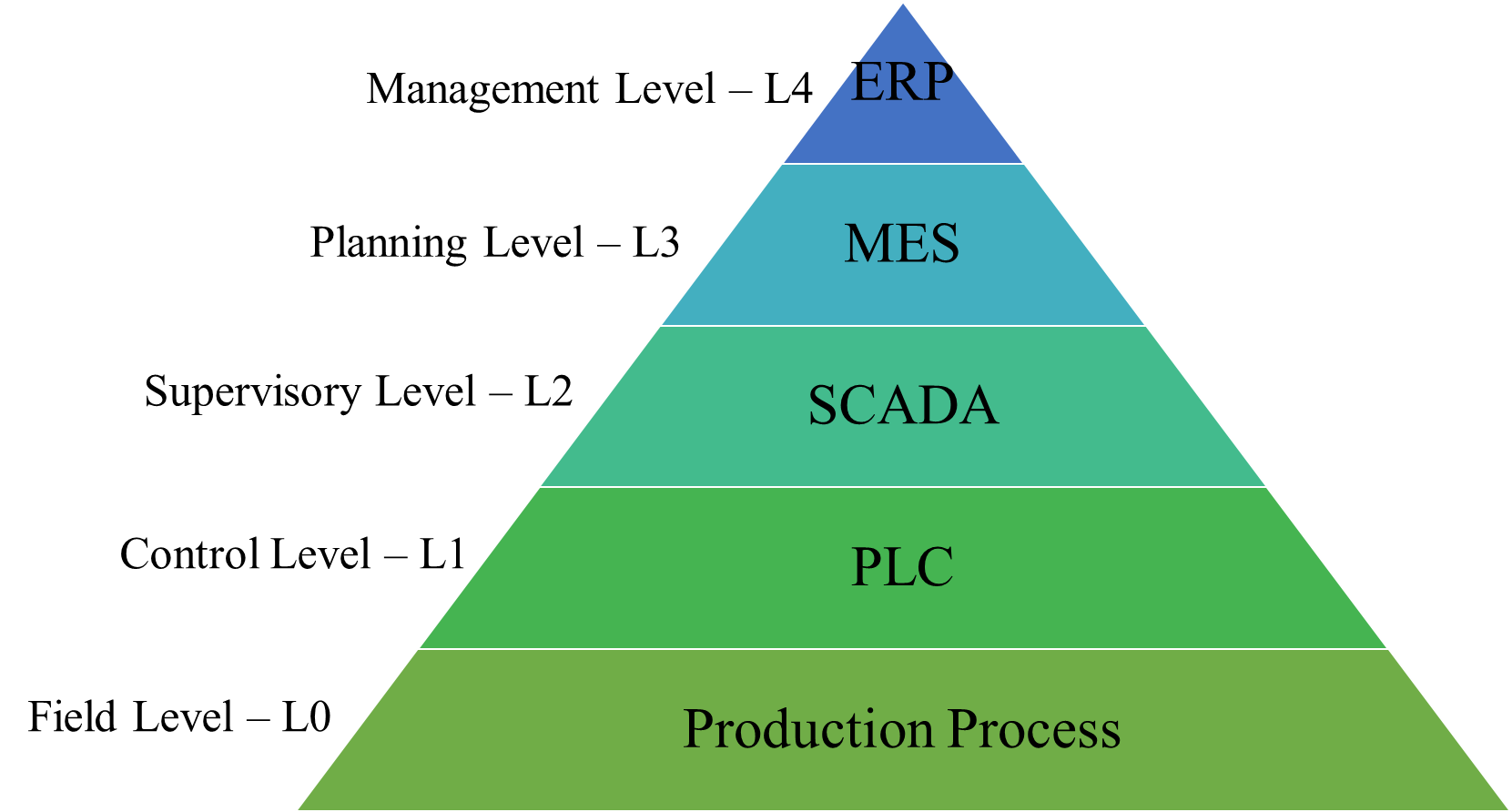
* Cơ khí: Bao gồm các khớp xoay, khớp nối, các khâu linh hoạt và tay gắp kết nối để thực hiện các tác vụ cụ thể. Đây là có thể xem là bộ khung xương của robot.
* Động cơ: Các động cơ điện hoặc thủy lực được sử dụng để giúp các khớp của robot chuyển dộng được. Động cơ được điều khiển bằng các hệ thống điện và truyền động để đạt được độ chính xác và hiệu suất cao.
* Hệ thống điều khiển: Hay là tủ 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.



2.2 Khái niệm về hệ thống SCADA

SCADA là từ viết tắt cho cụm từ Supervisory Control and Data Acquisition trong tiếng Anh, hay còn được gọi là hệ thống điều khiển giám sát và thu thập thông tin.

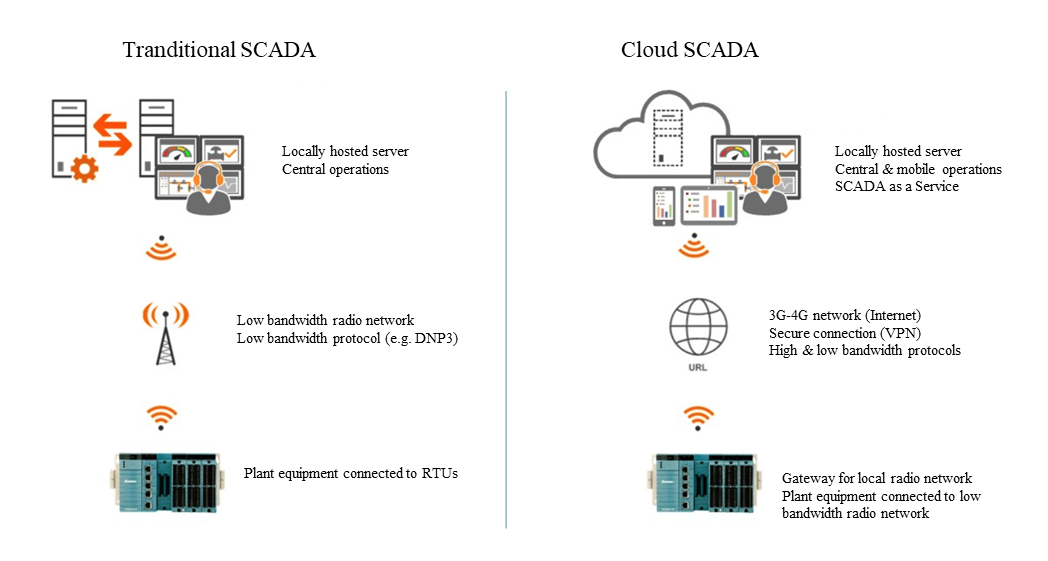
SCADA đóng vai trò quan trọng và có vị trí trung tâm trong hệ thống kim tự tháp tự động hóa (tiếng Anh là Automation Pyramid), chịu trách nhiệm giám sát và điều khiển toàn bộ hệ thống theo mô hình ANSI/ISA-95 (ISO 62264). Hệ thống SCADA là một sự kết hợp giữa các thành phần phần cứng và phần mềm, cho 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.



(hình này dựa theo mô hình ANSI/ISA-95 (ISO 62264))

Hệ thống SCADA có thể được chia ra thành các kiến trúc: 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.



<vẽ lại hình>

* 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.

<bảng so sánh>

|  |  |  |
| --- | --- | --- |
| 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 đâu 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 Ứng dụng của IIoT cho robot

2.4 Động học robot

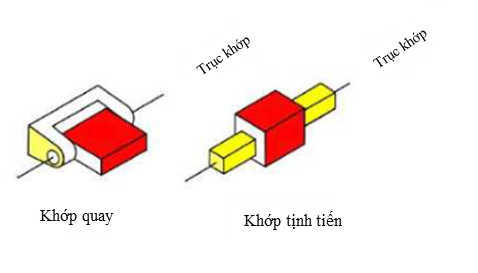
2.4.1 Một số khái niệm

* 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.

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

<Bảng: Các khớp thường gặp trong robot>

* 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: Khớp quay và khớp tịnh tiến>

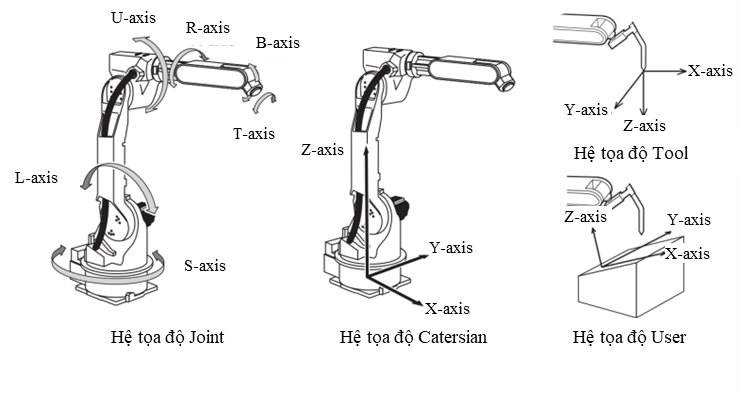
* Bậc tự do (tiếng Anh gọi là Degree of Freedom – DoF) là số lượng tham số tự do hay khả năng di chuyển của cơ cấu. Số bậc tự do của robot chính là khả năng chuyển động của robot trong hệ tọa độ gắn liền với điểm tham chiếu. Vì robot hoạt động trong không gian nên số bậc tự do được tính theo công thức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



Trong đó:

F: bậc tự do của robot,  
 : 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  
*fi*: số chuyển dộng cho phép của khớp i,  
*fp*: số bậc tự do thừa trong robot.  
<minh họa>

* 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: Các hệ tọa độ hoạt động của robot>

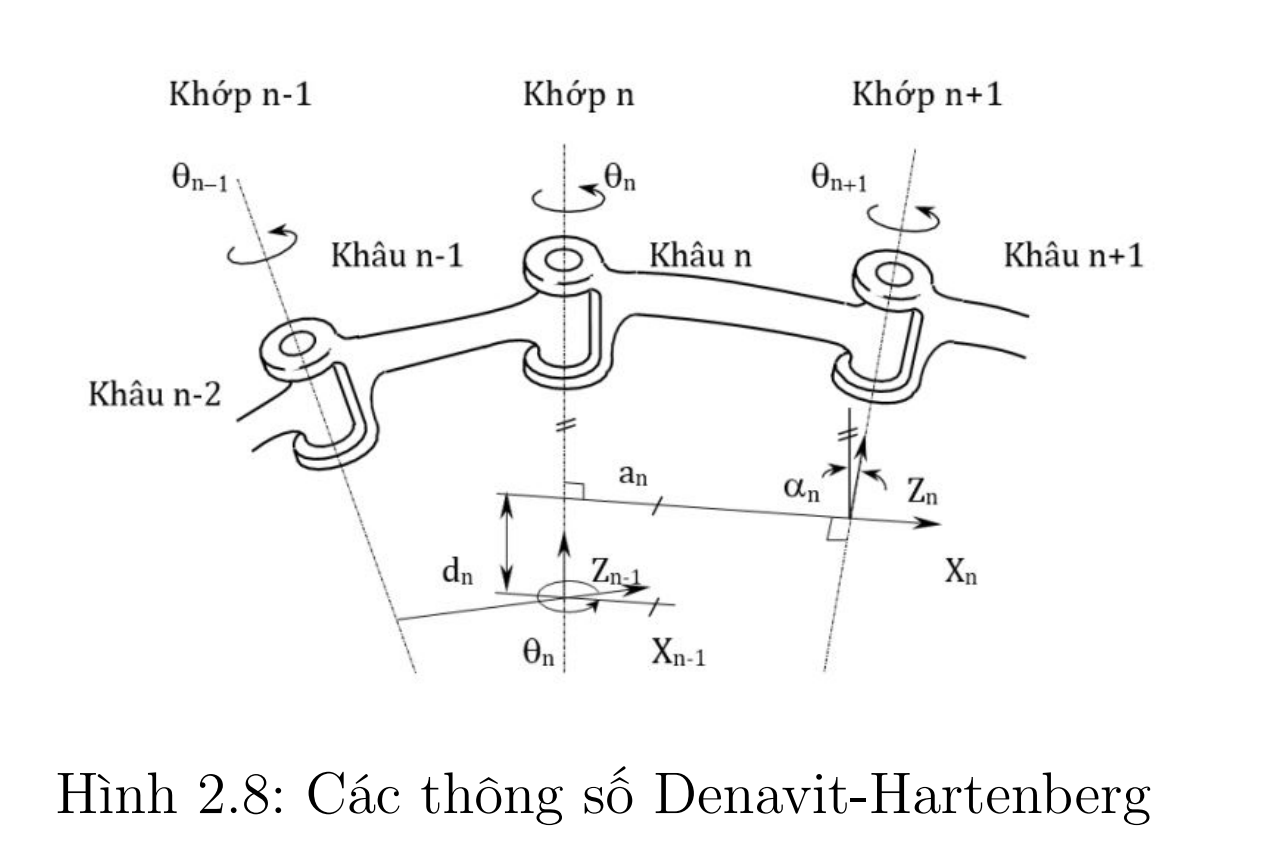
2.4.2 Biểu diễn Denavit – Hartenberg của bài toán động học robot

Mô hình hóa Denavit – Hartenberg (viết tắt là phương pháp D-H) là cách biểu diễn sử dụng các phép biến đổi để mô tạ vị trí và hướng tương đối của các khớp giữa các hệ thọa 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 *zi* nằm dọc theo trục của khớp (i+1).
* Trục *xi* nằm dọc theo đường vuông góc chung hướng từ *zi-1*­ đến *zi*. Trong trường hợp 2 trục giao nhau thì chọn , trục .

Xác định bộ thông số Denavit-Hartenberg:

* Gọi *ai* là khoảng cách giữa trục *zi-1*­ và trục *zi* dọc theo trục *xi*.
* Gọi là góc xoay *zi-1* quanh trục *xi* đến song song với *zi*.
* Gọi *di* là khoảng cách dọc theo trục *zi-1* giữa *xi-1* và *xi*.
* Gọi là góc xoay *xi-1* quanh trục *zi* đến song song với trục *xi*.



Xác định biến khớp:

* Nếu khớp (i) là khớp quay thì là biến khớp.
* Nếu khớp (i) là khớp tịnh tiến thì *di* 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 *zi-1* một góc .
* Tịnh tiến dọc theo *zi-1* một đoạn *di*.
* Tịnh tiến dọc theo *xi-1* một góc .

Ma trận chuyển đổi giữa hai hệ tọa độ:

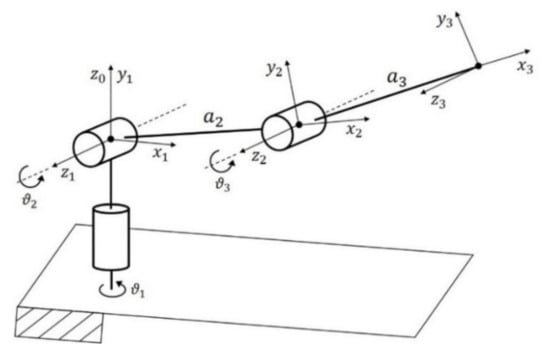


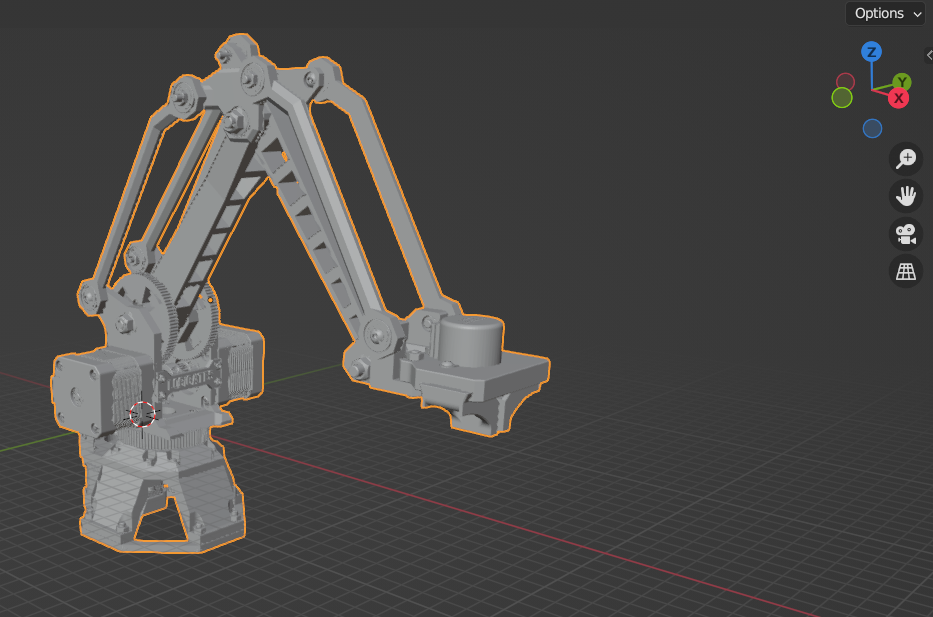
2.4.3 Phương trình động học thuận của cánh tay robot 3 bậc tự do

Bài toán động học thuận của robot mô tả hướng và vị trí của phần tay gắp 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 gắp so với hệ tọa độ gốc được mô tả bằng vector định vị *p0* (biểu diễn tọa độ của tay gắp) 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:

Trong đó, *q* là vector n phần tử, bao gồm các biến khớp, *p* là vector định vị tọa độ của tay gắp 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:





Bảng biểu diễn D – H với các thông số thực tế , ,  
:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Link |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

Các ma trận chuyển đổi được tính như sau:



.

Như vậy tọa độ cuối của tay gắp robot đã được xác định tại vị trí () và hướng được định bởi ba vector từ giá trị các biến khớp .

2.4.4 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 , từ ma trận chuyển đổi

Ta cần tìm giá trị các biến khớp từ các phương trình sau:

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

* Biến đổi 1:
* Biến đổi 2:

với

Lời giải bài toán động học nghịch:

Nếu

Nếu

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

Nếu

hoặc

Nếu

.

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

Nếu

.

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

Nếu

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

2.4.5 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 . Vận tốc gồm hai đoạn dốc ở đầu và cuối, vận tốc ở đoạn giữa trung gian có giá trị là hằng số , thời gian tăng tốc là *t1* bằng với thời gian giảm tốc.

<Hình đồ thị quỹ đạo LSPB>

Phương trình của quỹ đạo:

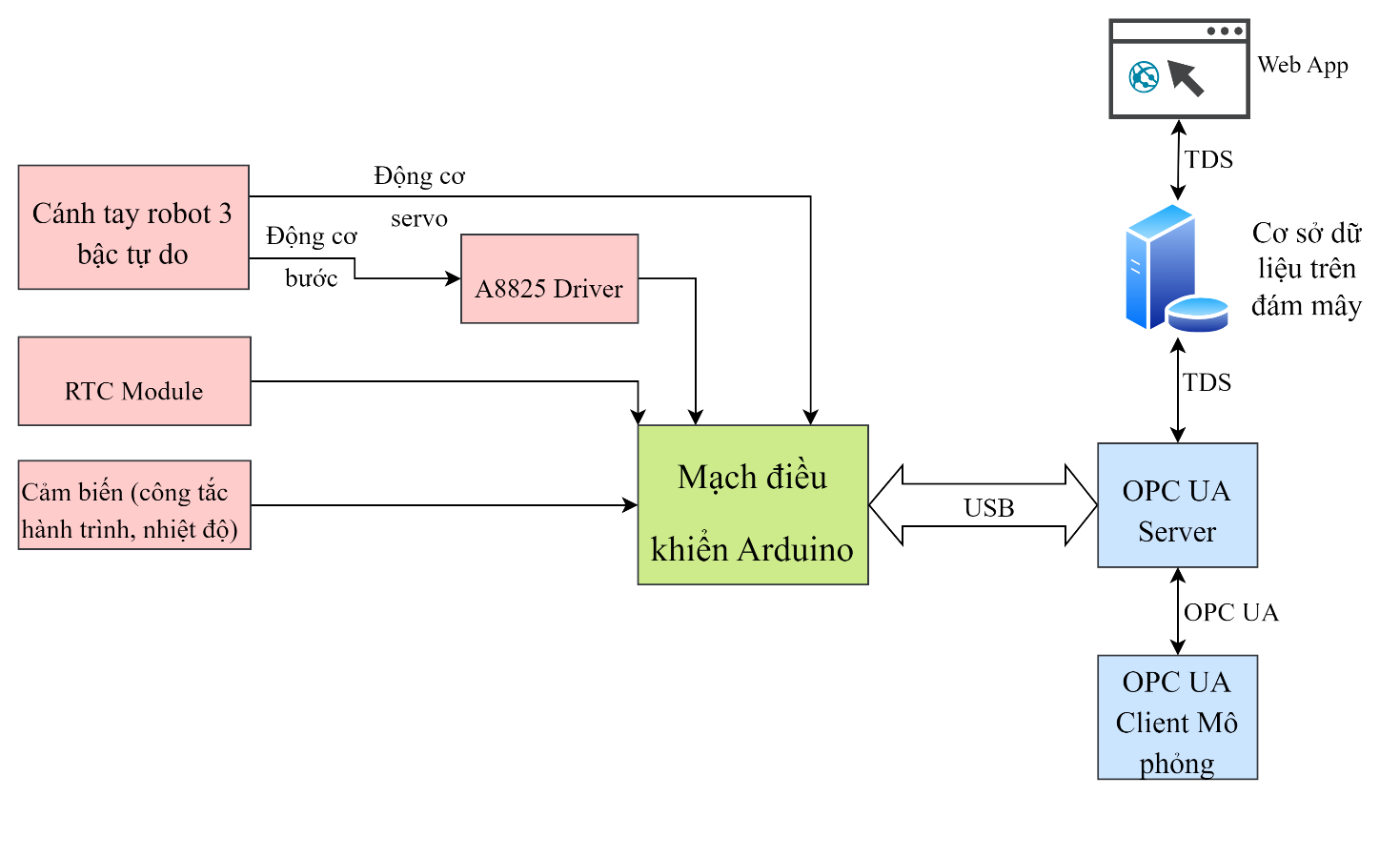
Phương trình vận tốc:

Phương trình gia tốc:

Chương 3

TỔNG QUAN VỀ CẤU TRÚC HỆ THỐNG

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



Hệ thống được thiết kế bao gồm:

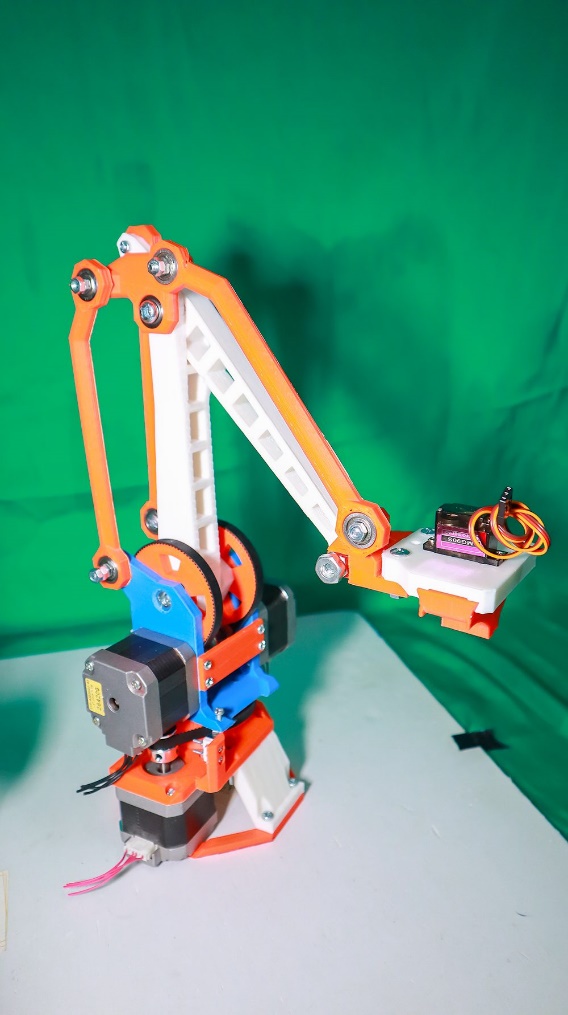
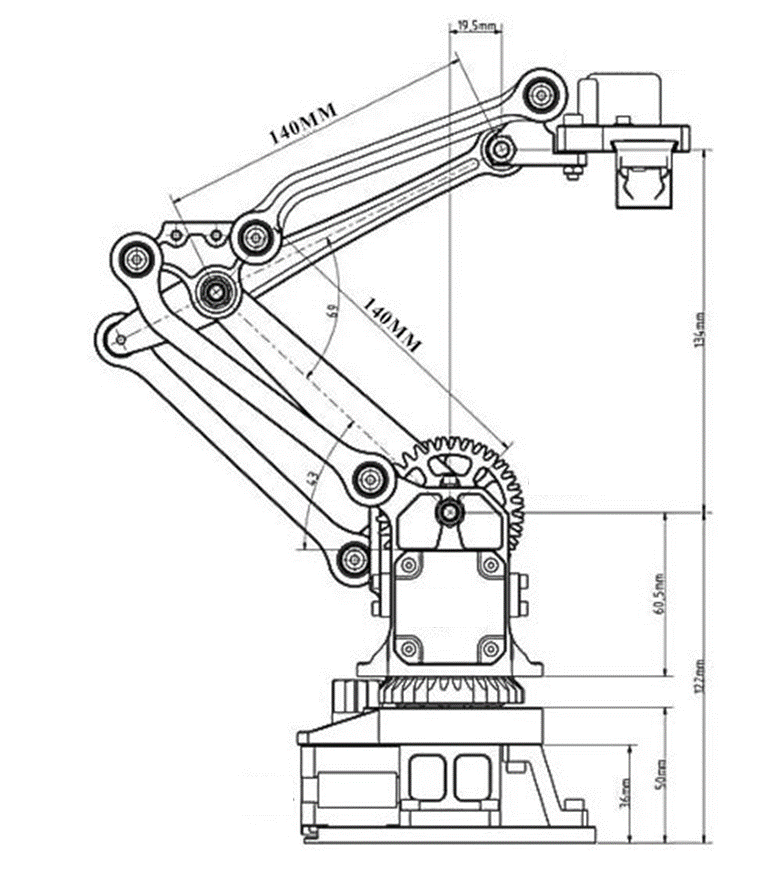
* Phần cứng:
* Cánh tay robot ba bậc tự do: Là kết cấu cơ khí, thực hiện các nhiệm vụ gắp, thả vật và di chuyển trong không gian.
* Mạch điều khiển xuất xung điều khiển đến các driver để điều khiển các động cơ bước của robot, điều khiển động cơ servo để đóng mở tay kẹp của robot, thu thập các giá trị 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.



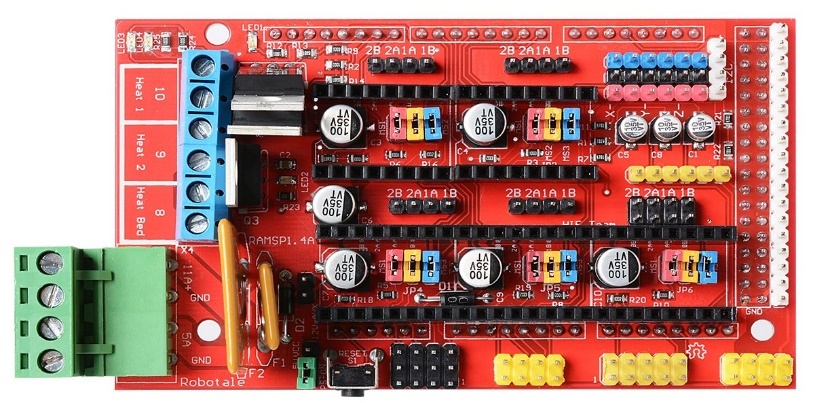
<sửa lại hình thực tế>

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).



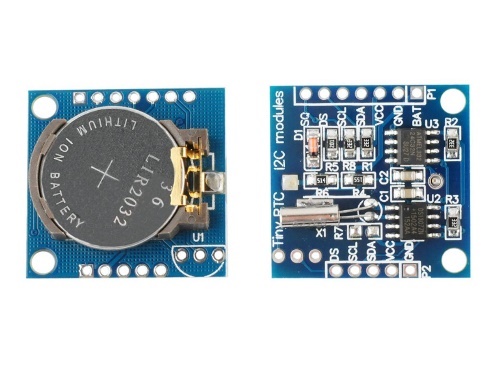
* Kết hợp với Arduino Mega 2560, em sử dụng thêm mạch Shield Ramps 1.4 (RepRap Arduino Mega Pololu Shield) để hỗ trợ Arduino Mega 2560 điều khiển các động cơ bước, các cảm biến được kết nối đơn giản hơn.



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



* 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: 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.



* 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.



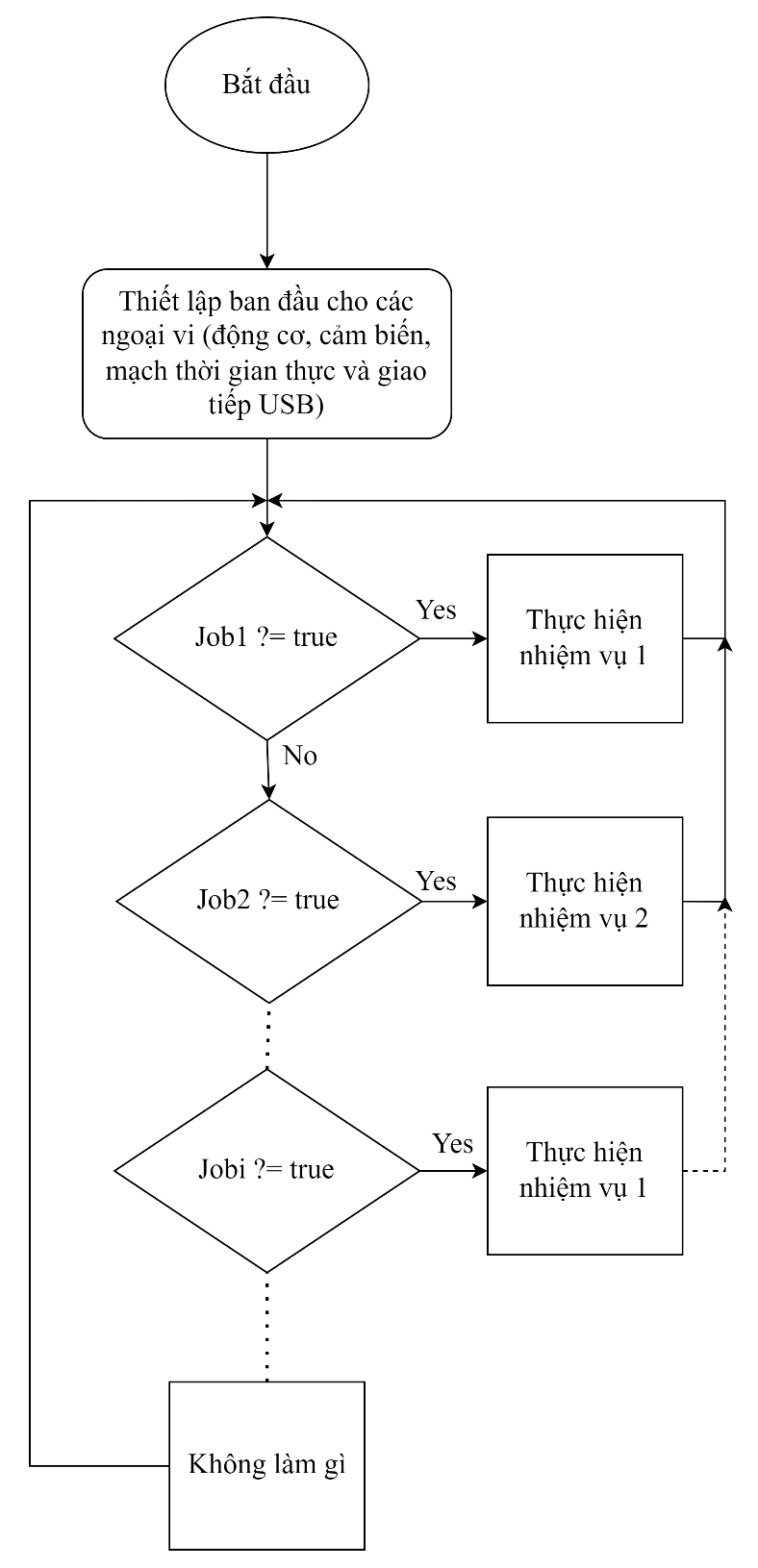
4.3 Thiết kế mạch điện

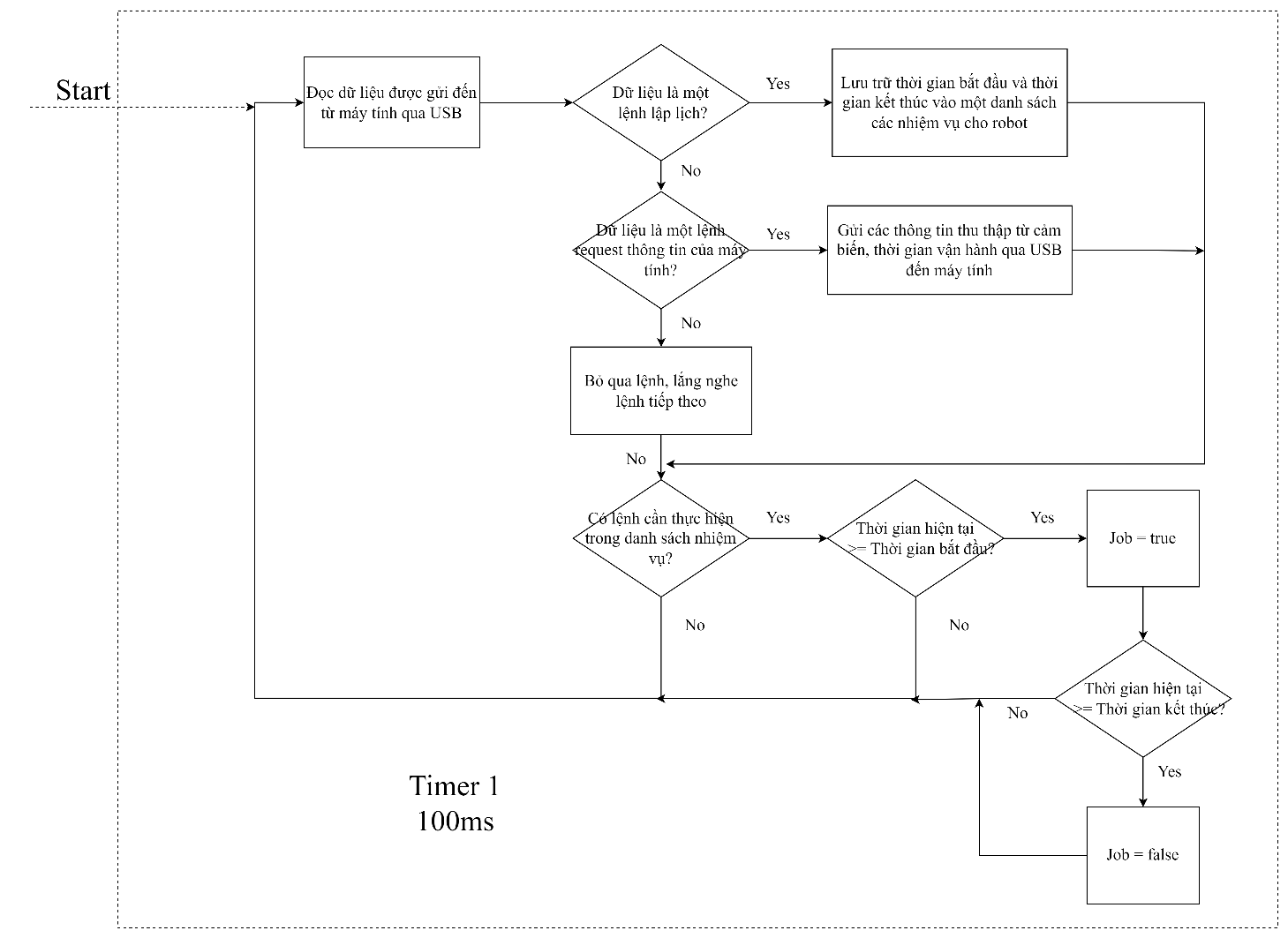
<hình thể hiện kết nối>

4.4 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



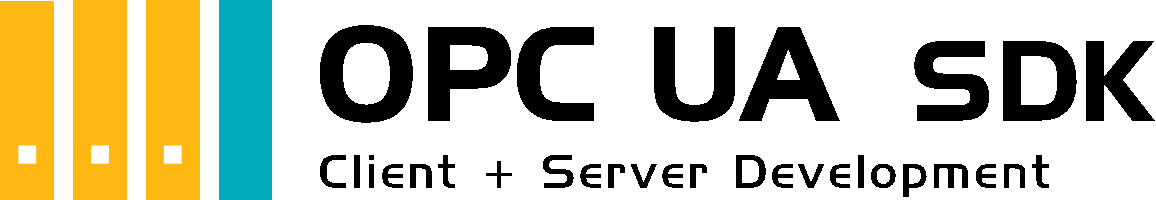


Chương 5

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

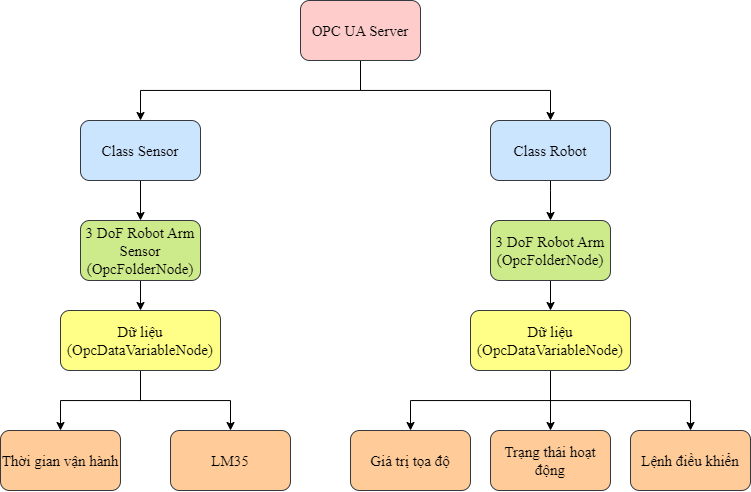


Thư viện OPC UA Client & Server SDK là một tập hợp các công cụ, thư viện và API (Application Programming Interface) hỗ trợ để phát triển các ứng dụng với chuẩn giao thức OPC UA (Open Platform Communications Unified Architecture). Đây là một giao thức truyền thông đa nền tảng trong lĩnh vực tự động hóa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

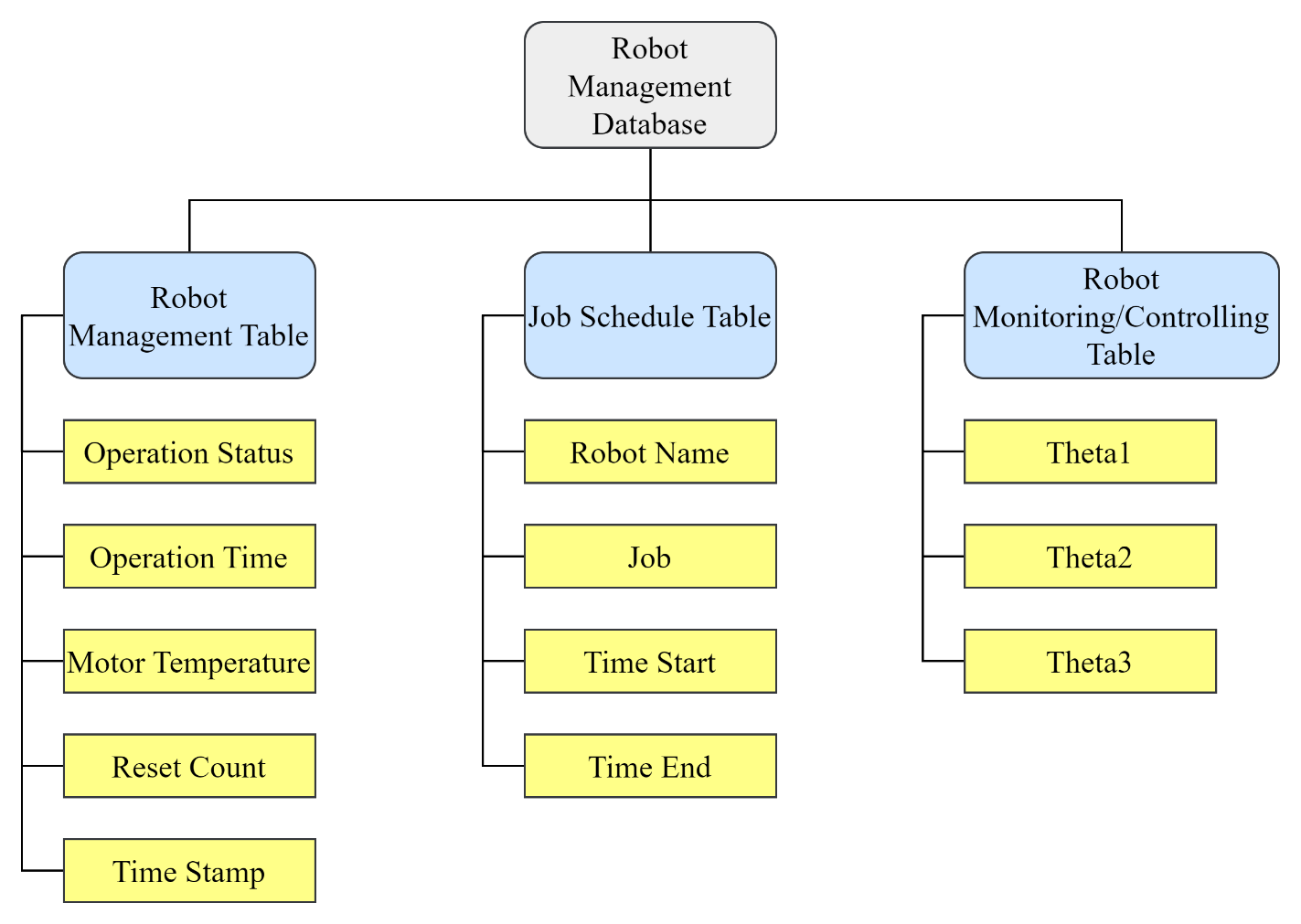
Dữ liệu trong OPC UA Server được tổ chức như sau:  


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:



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:

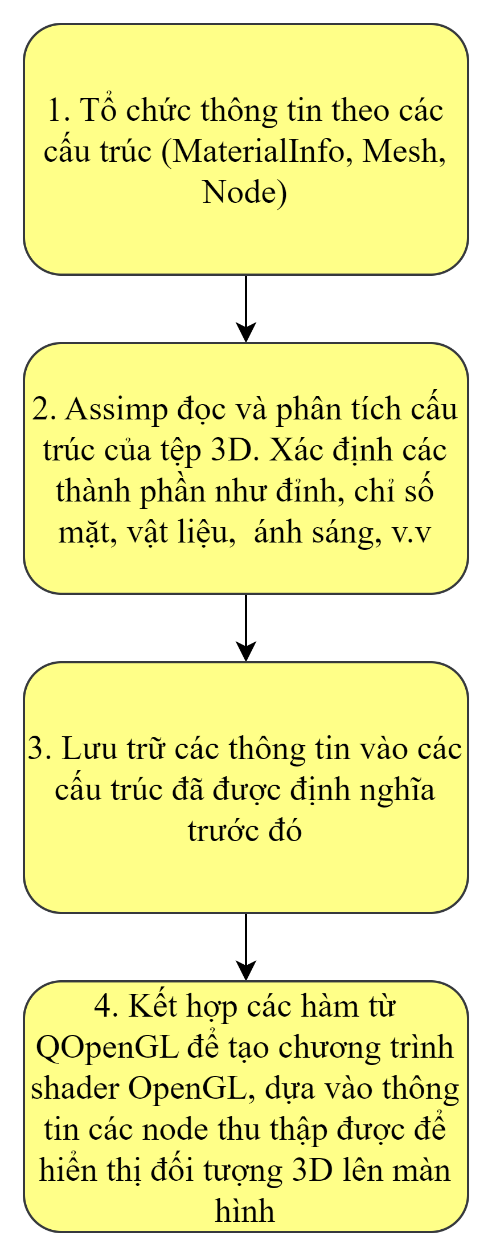
OpenGL (được gọi đầy đủ là Open Graphic Library), chủ yếu được dùng như một API (được gọi đầy đủ là Application Programming Interface) cho phép người dùng các hàm và quy tắc cho phép người dùng phát triển các ứng dụng đồ họa chất lượng cao trên nhiều nền tảng khác nhau, được phát triển bởi Khronos Group.



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 tỏng 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:

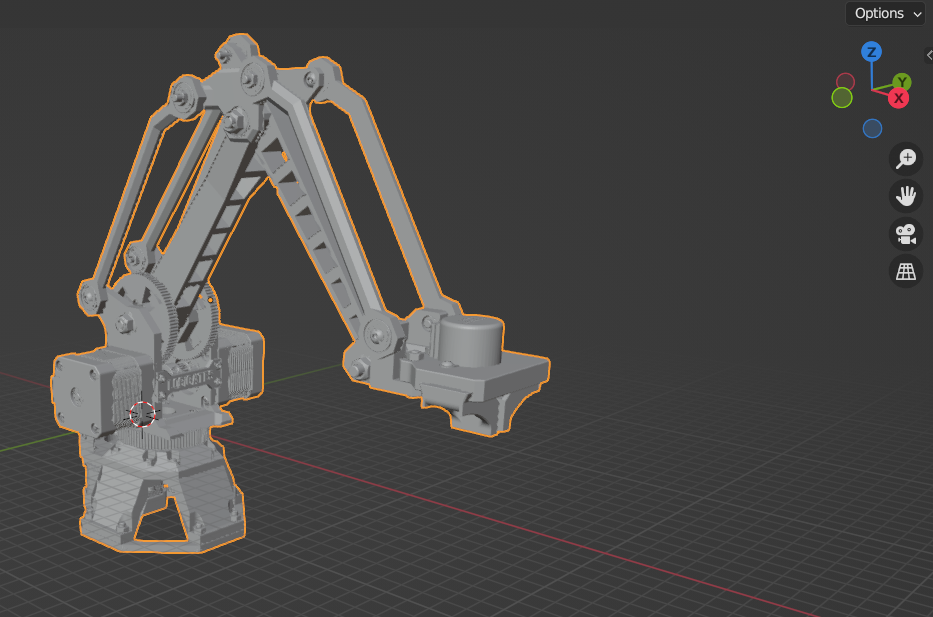
Thư viện Assimp (tên đầy đủ là Open Asset Import Library) là một thư viện mã nguồn mở được sử dụng để tải các tệp có định dạng 3D vào trong ứng dụng đồ họa. Thư viện hỗ trợ nhiều định dạng tệp phổ biến như OBJ, STL, v.v. Cơ chế hoạt động của thư viện Assimp:

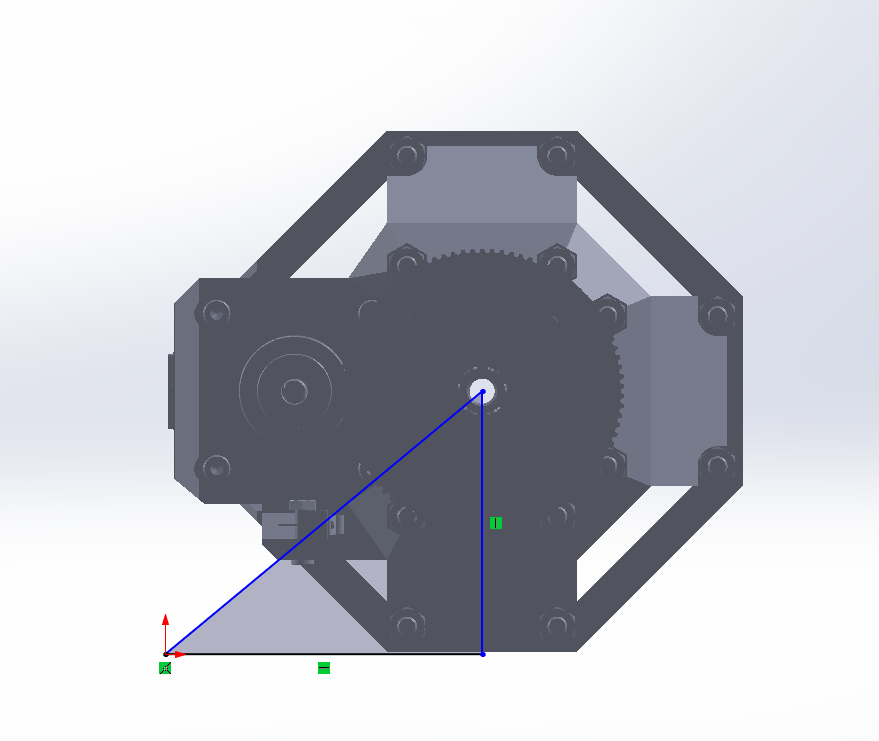


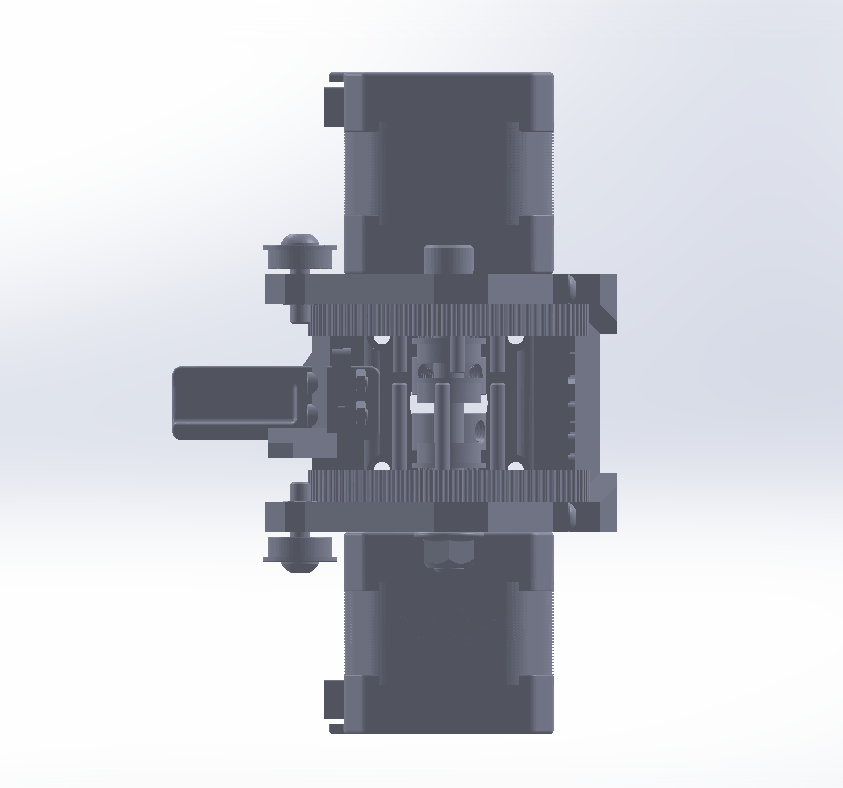
* + 1. Thiết kế phần mềm mô phỏng

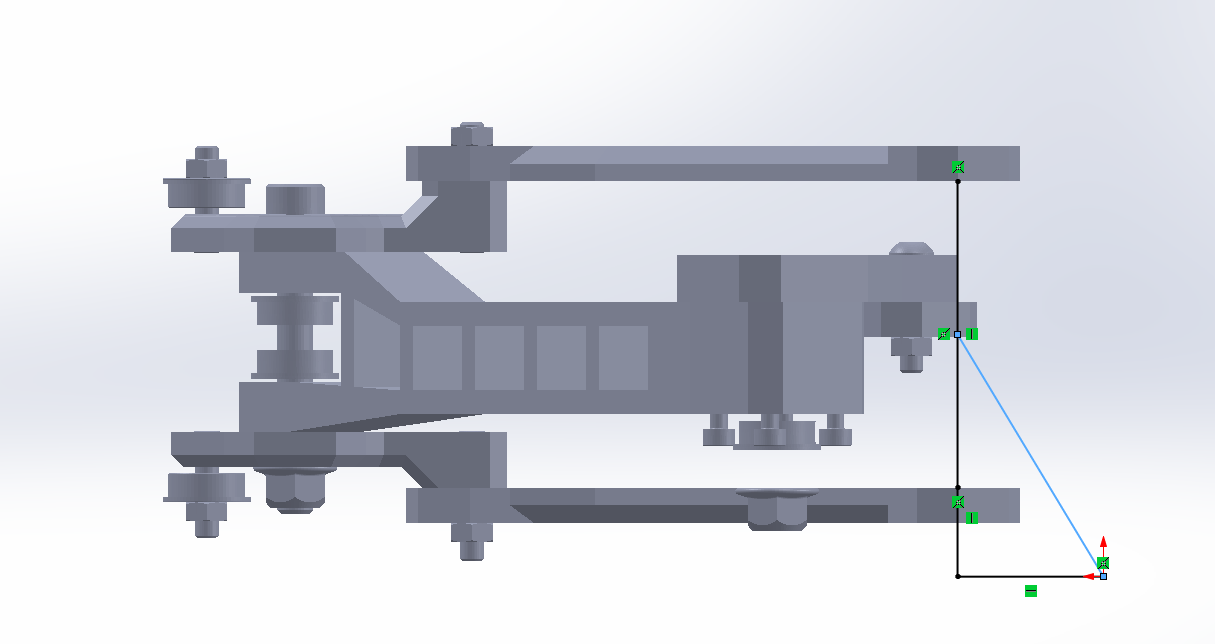
Phần mềm được phát triển có đầy đủ những tính năng cơ bản trong việc mô phỏng trực tuyến và ngoại tuyến cho robot hoạt động. Quá trình xây dựng phần mềm mô phỏng cần trải qua các bước: Đo đạc robot, đo đạc file 3D của robot; Tính toán động học thuận và nghịch cho robot; Phát triển phần động học của robot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

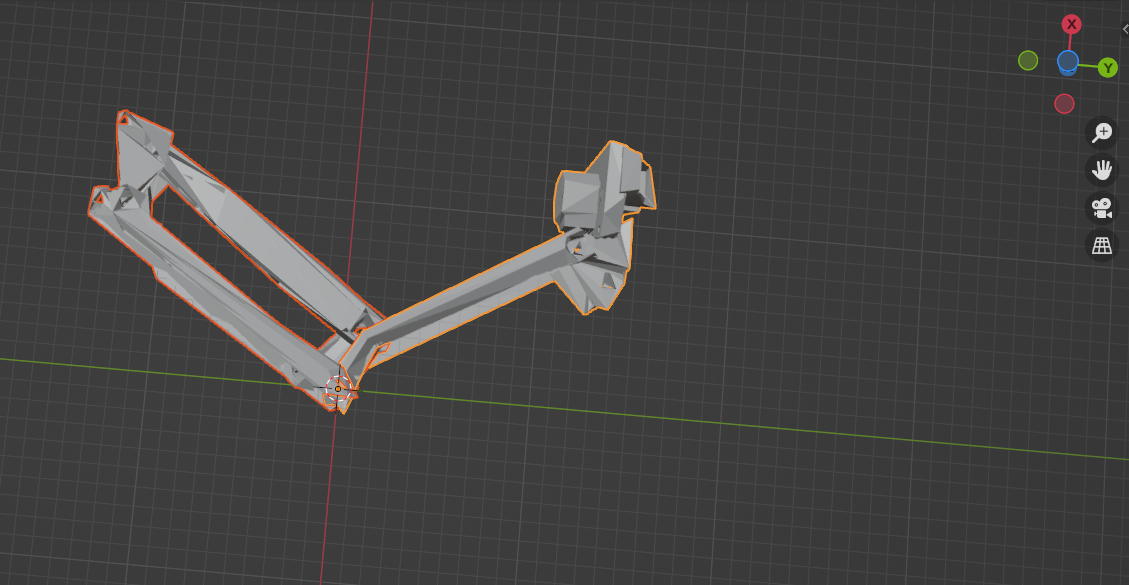
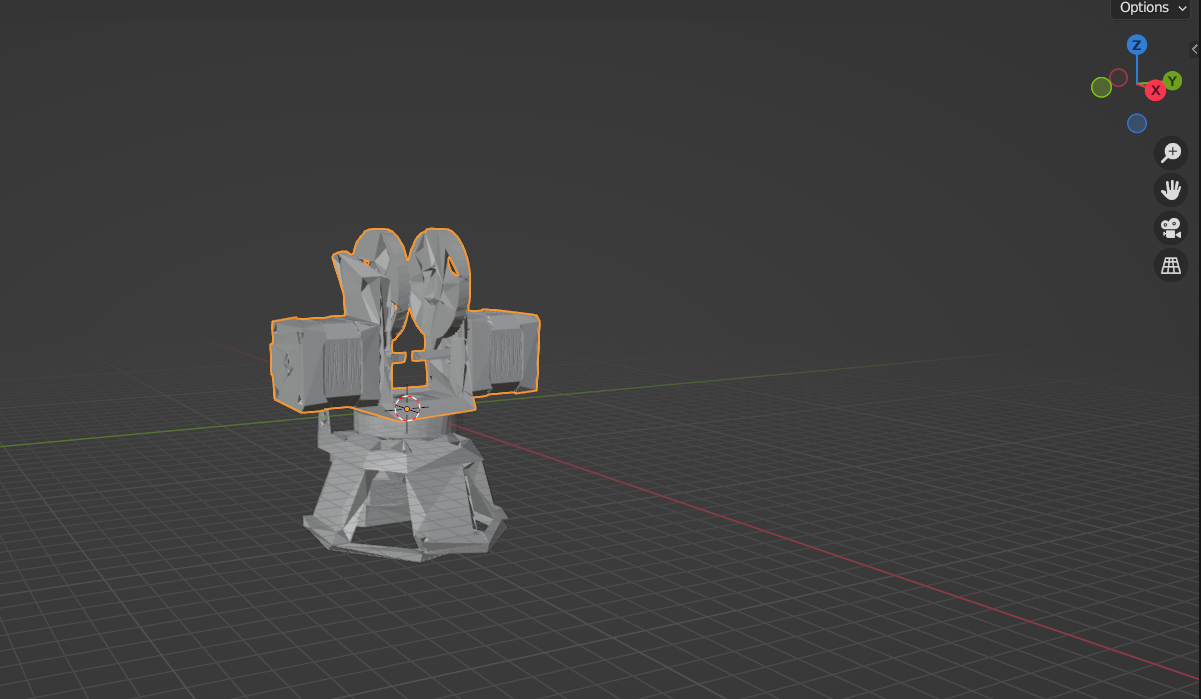
<hình robot>

<đo kích thước của base>

<đo kích thước của link1>

<đo kích thước của link2>

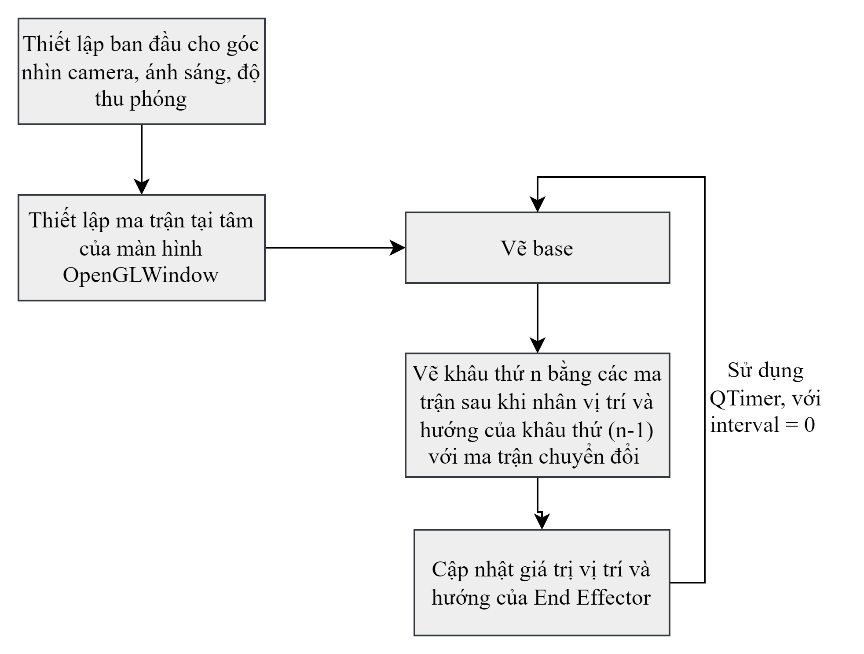
<đo kích thước của link3>



<định lại tọa độ các link>

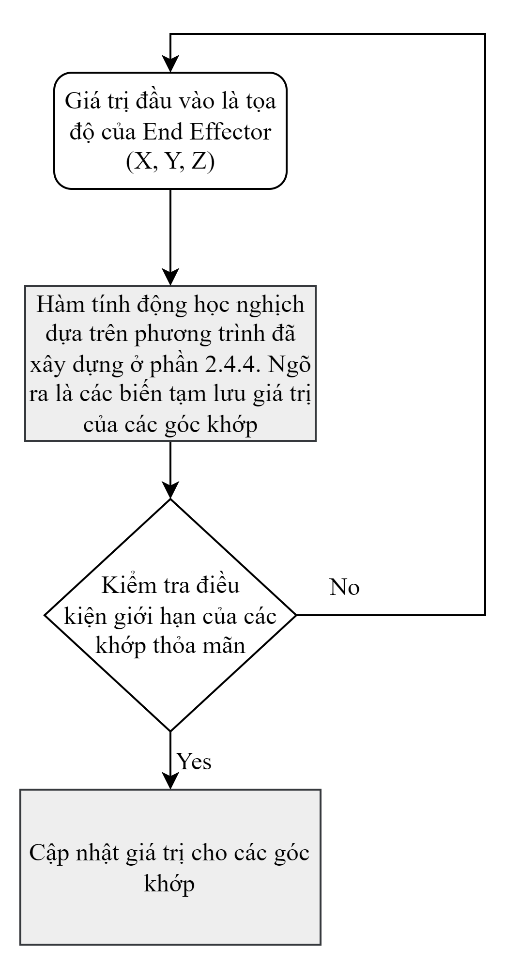
<sơ đồ khối của phần mềm>

* Tính toán động học thuận và nghịch của robot, hoạch định quỹ đao 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 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().

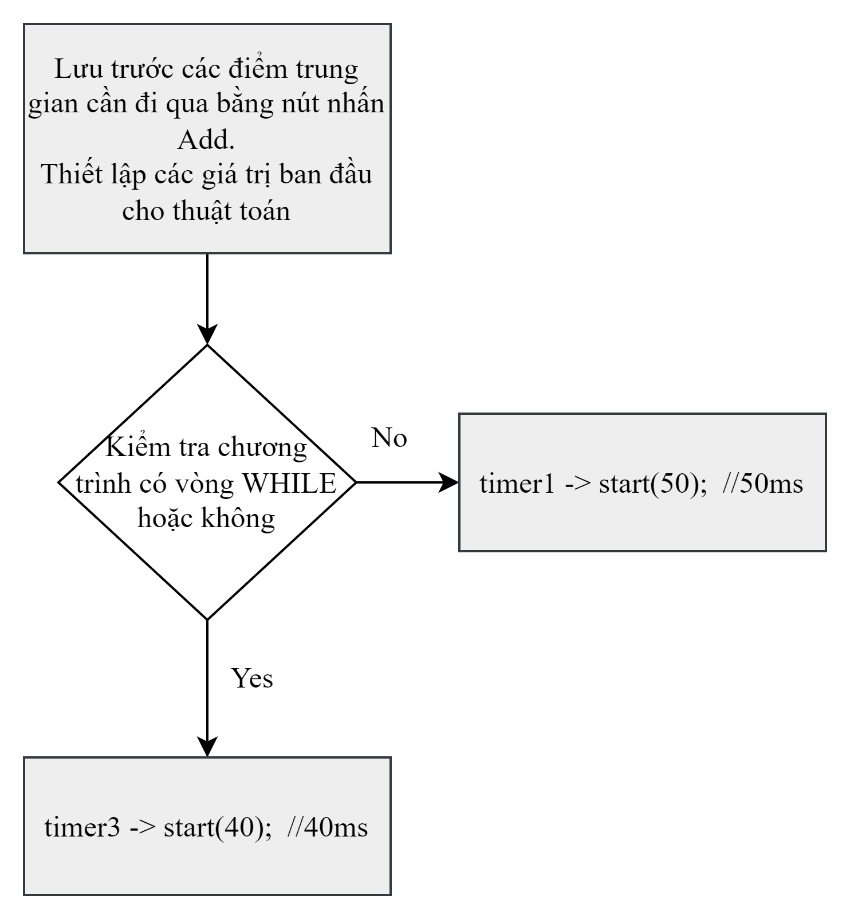


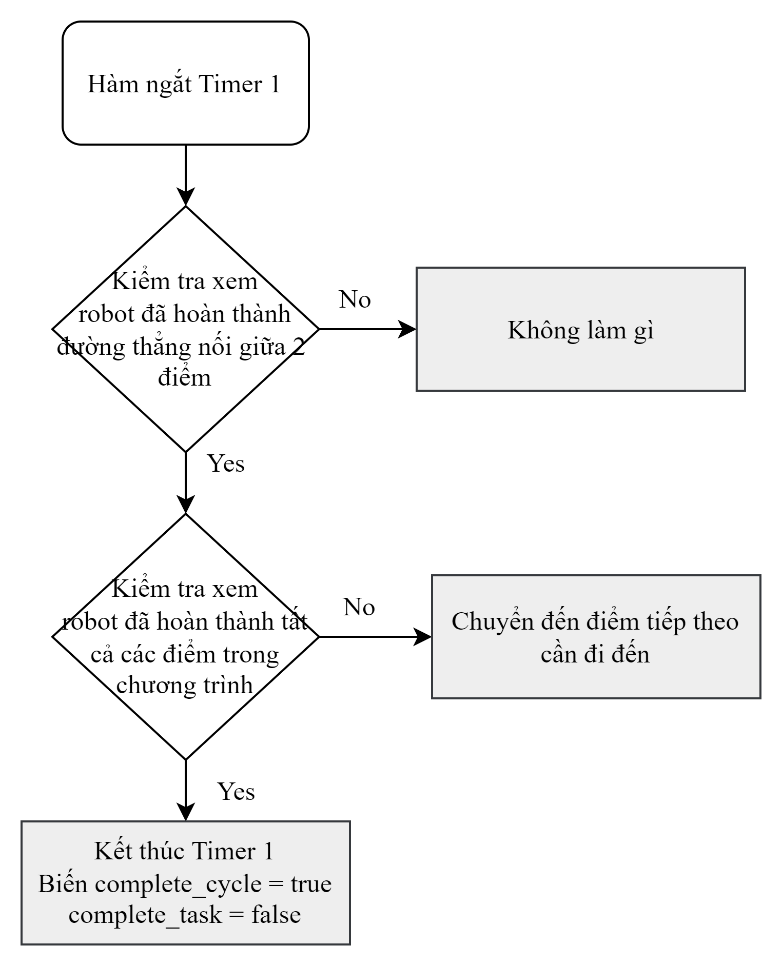
<hình động học nghịch>

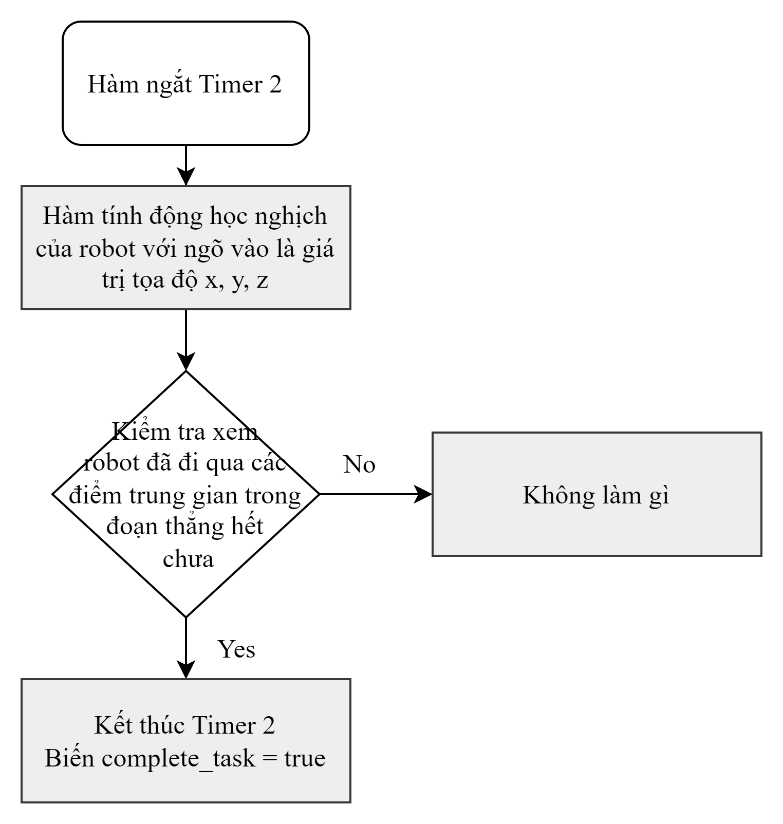
Phần quy hoạch đường thẳng LSPB, người dùng có thể thêm các điểm robot đi qua bằng nút nhấn *`Add`*. Giải thuật quy hoạch sẽ dựa trên công thức được xây dựng ở phần 2.4.5. Mặc định chương trình sẽ quy hoạch đường thẳng qua hai điểm được chọn. Trong trường hợp mô phỏng mong muốn robot di chuyển liên tục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

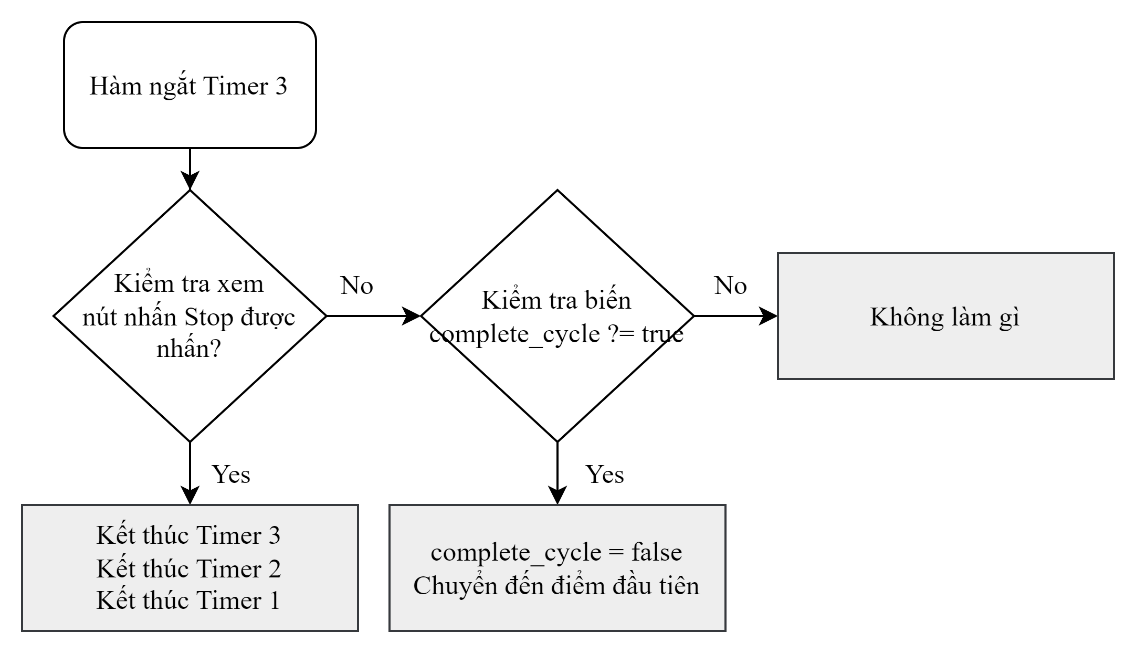
|  |  |
| --- | --- |
| Câu lệnh | Chức năng |
| WHILE 1 | Bắt đầu vòng lặp |
| ENDWL | Kết thúc vòng lặp |
| LMOVE P[i] | Quy hoạch robot theo đường thẳng từ vị trí hiện tại đến vị trí điểm P[i] |
| EOP | Kết thúc chương trình |

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.









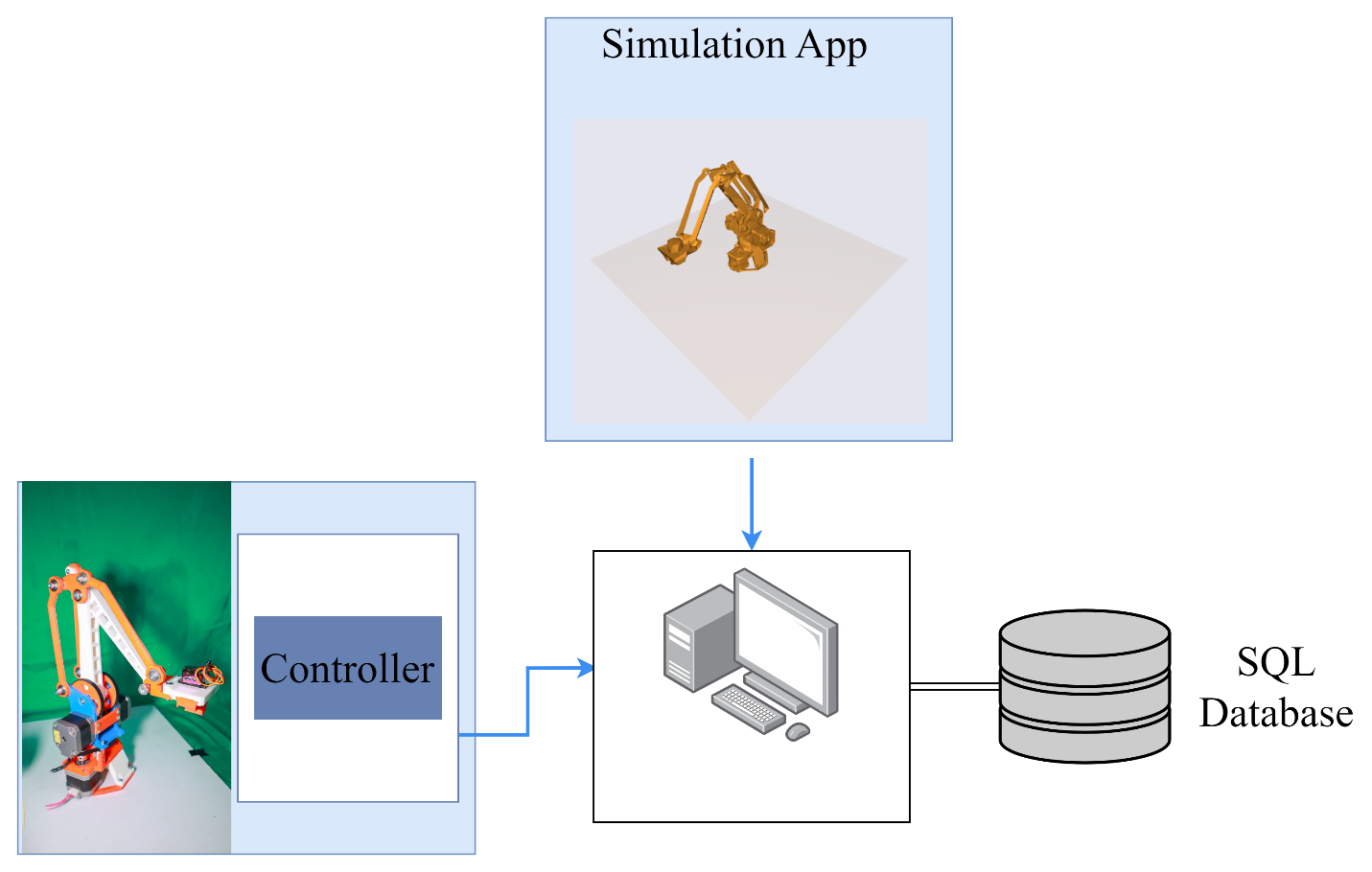
* Phát triển các tính năng cơ bản khác của một phần mềm (có thể thay đổi góc nhìn, vẽ đồ thị, trực quan hóa đường đi của End Effector, hoạch định quỹ đạo offline) Em đã phát triển một số tính năng cơ bản như trên để phục vụ người dùng có thể sử dụng phần mềm dễ dàng hơn bằng các hàm và lớp được cung 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.



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.

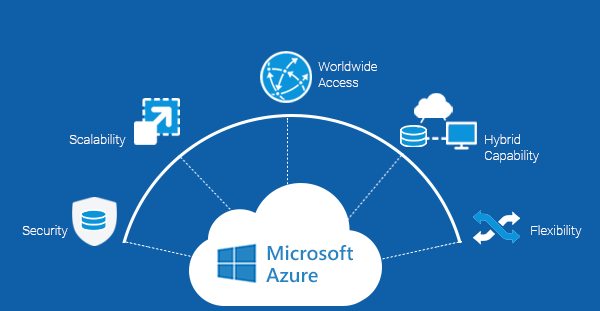


<sửa lại hình thật>

Chương 6

WEB APP GIÁM SÁT VÀ LẬP LỊCH ĐIỀU KHIỂN ROBOT

6.1 Giới thiệu về Azure Cloud



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 khia 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

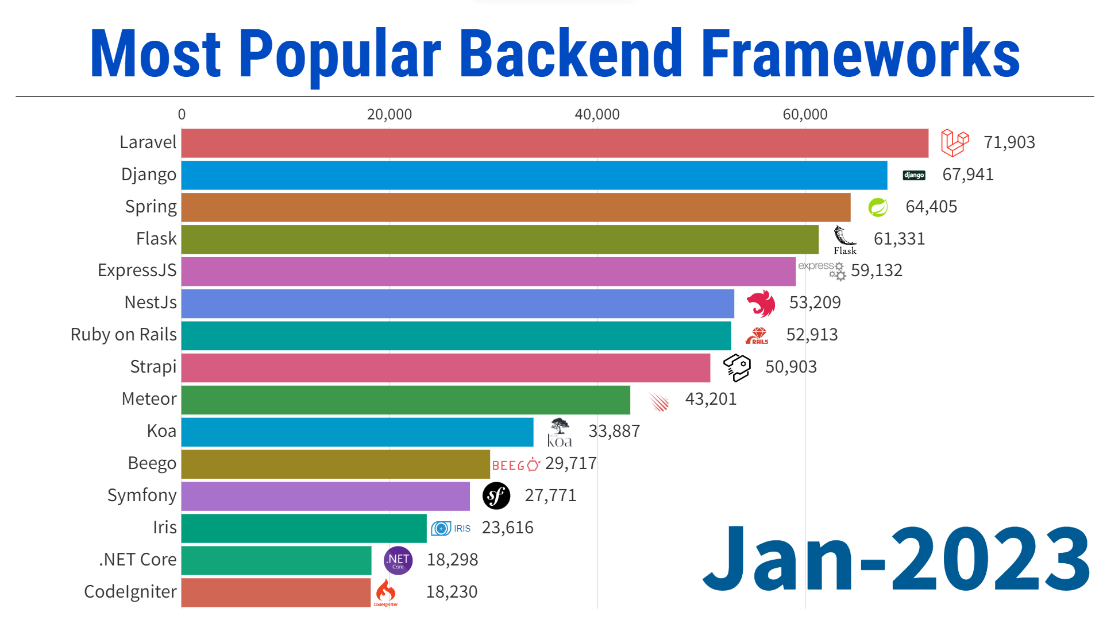
6.2.1 Giới thiệu về Node.js và Express.js

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.



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.

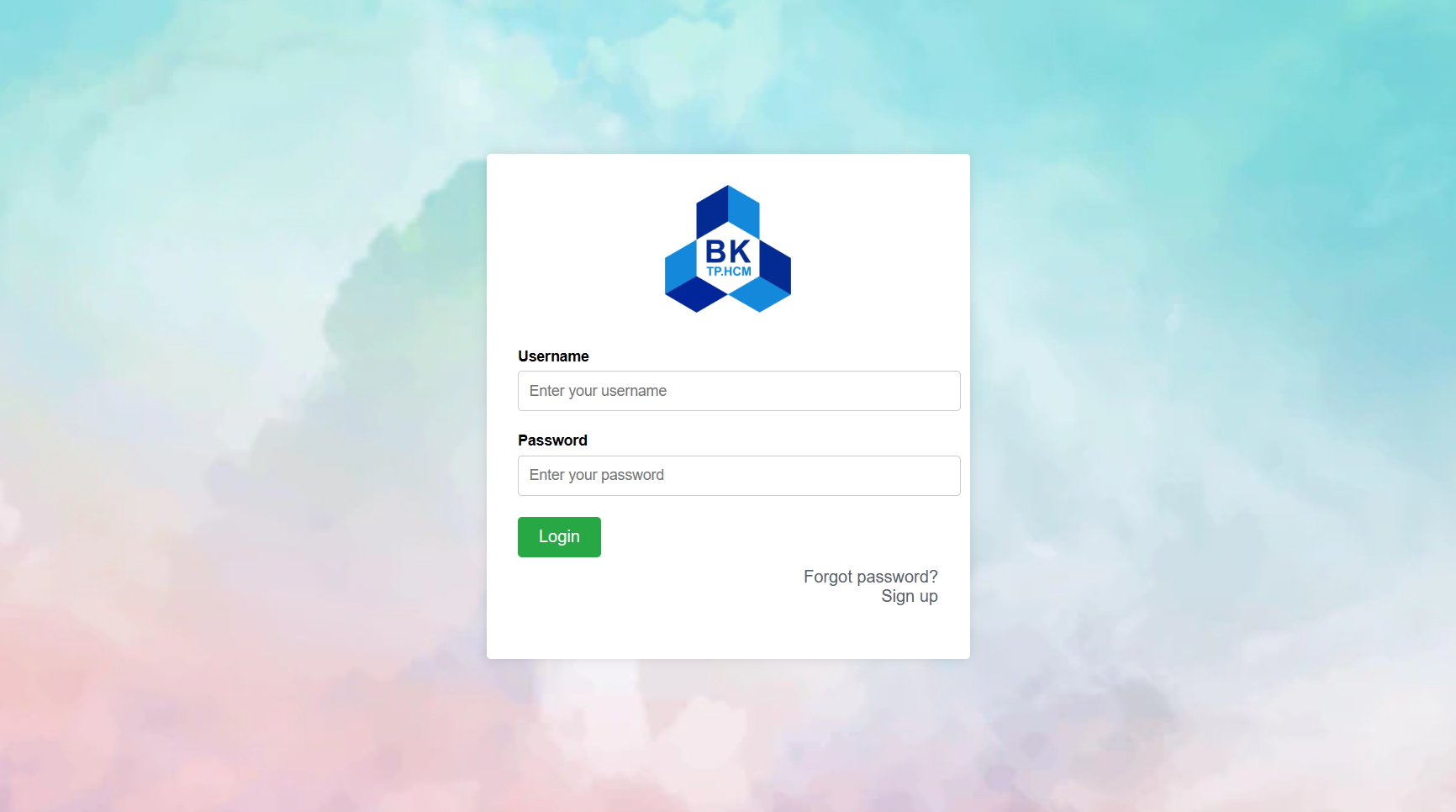
Express.js là một framework được phát hành lần đầu tiên vào năm 2010, được thiết kế để làm việc tốt với Node.js. Express.js có thể xem là một framework được bổ sung cho Node.js giúp người dùng đơn giản hóa các ứng dụng web của server với nhiều module hơn và công cụ tiện ích hơn. Trong thực tế, Express.js 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.



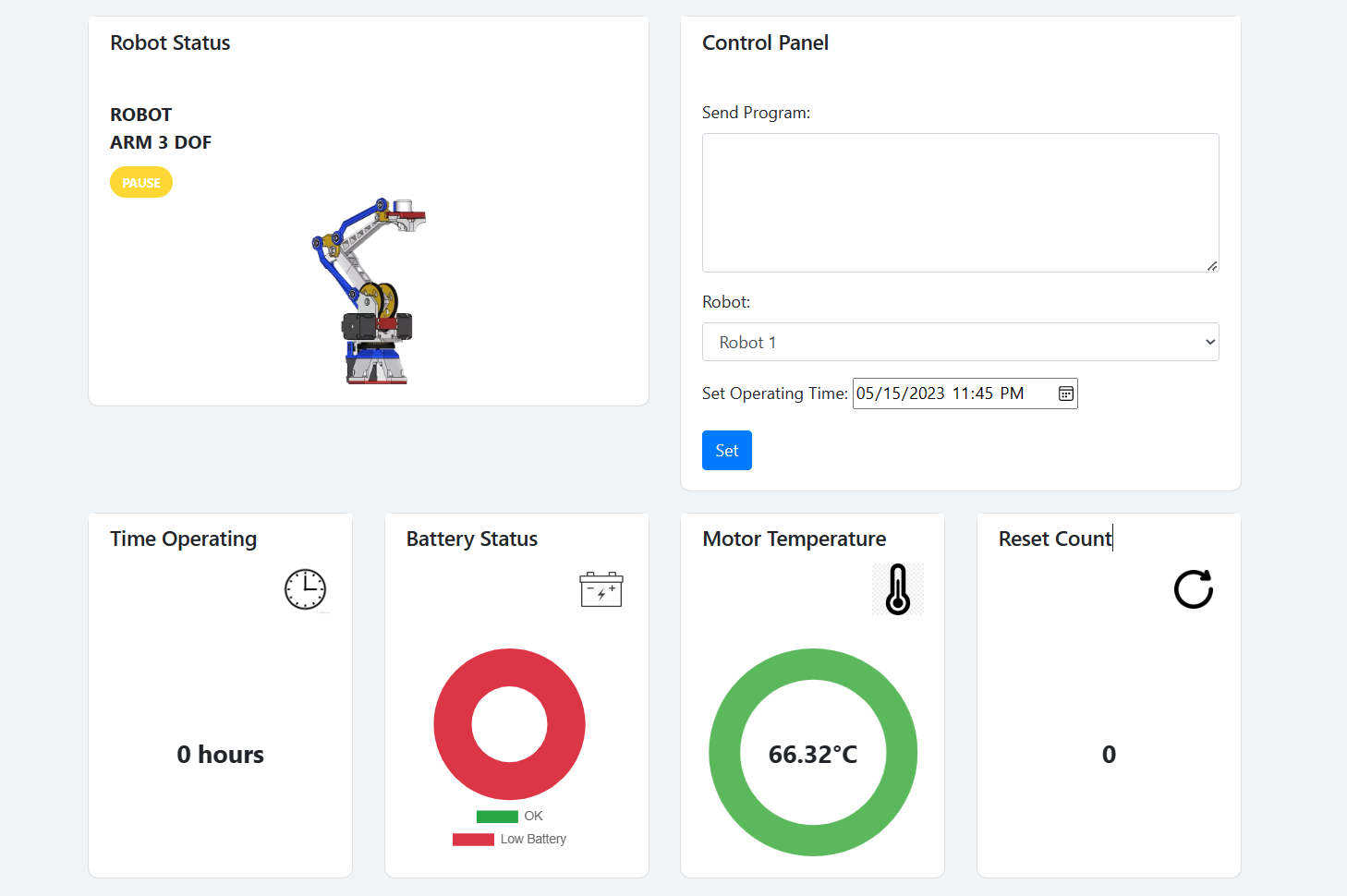
6.2.2 Thiết kế giao diện người dùng

* Trang đăng nhập:

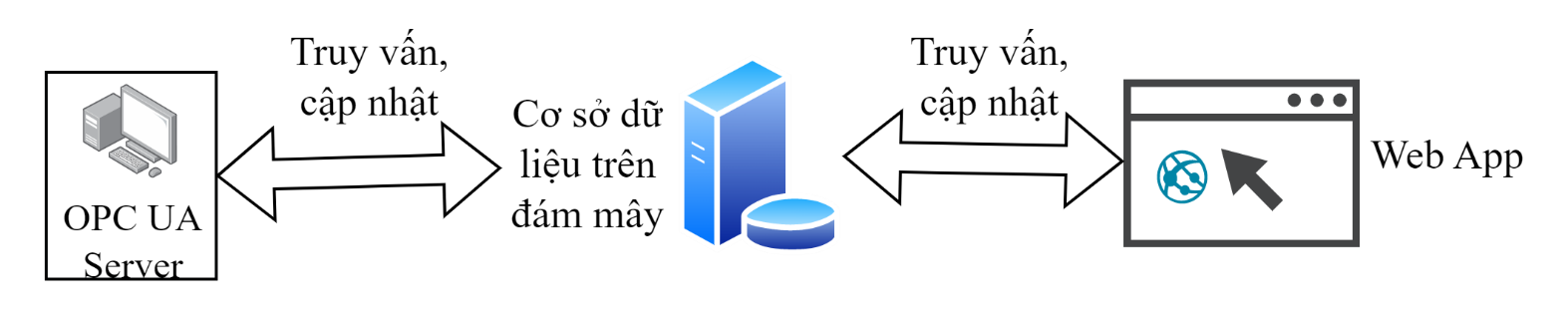
Trang đăng nhập là trang đầu tiên hiển thị cho người dùng khi truy cập đến website. Sau khi nhấn nút đăng nhập, web sẽ điều hướng đến trang màn hình chính dashboard. Trong trang còn có các phần Quên mật khẩu (Forgot password) và Đăng ký (Sign up) nhưng chưa được thực hiện.



* 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.



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:  


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

7.2 Đánh giá

7.3 Hướng phát triển

