

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH



ĐỒ ÁN TỐT NGHIỆP
NGÀNH CNKT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

**ĐIỀU KHIỂN VÀ GIÁM SÁT HỆ THỐNG PHÂN LOẠI
GÓI SẢN PHẨM DÙNG DELTA ROBOT**

**GVHD: TS TRẦN VI ĐÔ
SVTH: LÊ THÀNH TRUNG
TRẦN HOÀNG DŨNG**



Tp.Hồ Chí Minh, tháng 8/2022

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



HCMUTE

ĐỒ ÁN TỐT NGHIỆP

**ĐIỀU KHIỂN VÀ GIÁM SÁT HỆ THỐNG PHÂN
LOẠI GÓI SẢN PHẨM DÙNG DELTA ROBOT**

SVTH1: LÊ THÀNH TRUNG

MSSV: 18151137

SVTH2: TRẦN HOÀNG DŨNG

MSSV: 18151060

Khoá: 2018-2022

Ngành: CNKT Điều khiển và Tự động hóa

GVHD: TS. TRẦN VI ĐÔ

TP. HỒ CHÍ MINH – 08/2022



NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Lê Thành Trung

MSSV: 18151137

Trần Hoàng Dũng

MSSV: 18151060

I. TÊN ĐỀ TÀI : **Điều khiển và giám sát hệ thống phân loại gói sản phẩm dùng Delta Robot.**

II. NHIỆM VỤ

1. Các số liệu ban đầu

PLC S7-1200, Module mở rộng SM 1222, camera 1080p, động cơ băng tải, động cơ bước, bơm hút chân không, công tắc hành trình.

2. Nội dung thực hiện

- Tìm hiểu động học Robot Delta.
- Tìm hiểu các công nghệ xử lý ảnh giúp phát hiện sản phẩm.
- Tìm hiểu, nghiên cứu lập trình xử lý ảnh trên C# và liên kết dữ liệu với TIA Portal.
- Thiết kế, thi công và lập trình phần mềm điều khiển Robot Delta phân loại sản phẩm.
- Chạy thử nghiệm và hiệu chỉnh mô hình.
- Đánh giá kết quả thực hiện và viết luận văn.

III. NGÀY GIAO NHIỆM VỤ: 22/02/2022

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 08/2022

V. CÁN BỘ HƯỚNG DẪN: TS. TRẦN VI ĐÔ

TRƯỞNG NGÀNH

GIẢNG VIÊN HƯỚNG DẪN

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên: Lê Thành Trung
Trần Hoàng Dũng

MSSV: 18151137
MSSV: 18151060

Ngành: Công nghệ Kỹ thuật Điều khiển và Tự động hóa.

Tên đề tài: **Điều khiển và giám sát hệ thống phân loại gói sản phẩm dùng Delta Robot.**

Họ và tên Giáo viên hướng dẫn: TS Trần Vi Đô.

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....
.....

2. Ưu điểm:

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

3. Khuyết điểm:

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

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

6. Điểm: (Bằng chữ:)

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2022
Giảng viên hướng dẫn

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên: Lê Thành Trung
Trần Hoàng Dũng

MSSV: 18151137
MSSV: 18151060

Ngành: Công nghệ Kỹ thuật Điều khiển và Tự động hóa.

Tên đề tài: **Điều khiển và giám sát hệ thống phân loại gói sản phẩm dùng Delta Robot.**

Họ và tên Giáo viên phản biện:

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....

2. Ưu điểm:

.....

.....

3. Khuyết điểm:

.....

.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

6. Điểm: (Bằng chữ:)

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2022
Giảng viên phản biện



LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên 1: Lê Thành Trung

Lớp: 18151CL3A

MSSV: 18151137

Họ tên sinh viên 2: Trần Hoàng Dũng

Lớp: 18151CL1B

MSSV: 18151060

Tên đề tài: **Điều khiển và giám sát hệ thống phân loại gói sản phẩm dùng Robot**

Delta

Tuần	Nội dung	Xác nhận GVHD
1-2	Nhận đề tài, tìm hiểu tổng quan nguyên lý hoạt động Robot Delta.	
3-6	Tìm hiểu các công nghệ xử lý ảnh, động học thuận/nghịch Robot Delta.	
7-9	Lập trình C# điều khiển robot gấp sản phẩm đến vị trí xác định từ trước.	
10-11	Lập trình PLC và liên kết với C# qua OPC.	
12-13	Chỉnh sửa và cải tiến phần cứng.	
14-15	Chạy thử nghiệm và hiệu chỉnh	
16	Viết báo cáo	

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)



LỜI CAM ĐOAN

Nhóm chúng em xin cam kết đề tài này là do nhóm tự nghiên cứu và thực hiện dựa vào kham khảo một số tài liệu trước đó và không sao chép từ tài liệu hay mô hình đã có trước đó.

Sinh Viên thực hiện đề tài

Lê Thành Trung

Trần Hoàng Dũng

LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em chân thành cảm ơn quý thầy cô Trường Đại Học Sư Phạm Kỹ Thuật Tp.HCM nói chung, các thầy cô trong khoa Đào Tạo Chất Lượng Cao Tiếng Việt nói riêng trong xuyên suốt 4 năm học vừa qua đã truyền đạt và trang bị cho nhóm chúng em kiến thức không những về chuyên môn, chuyên ngành mà còn những kinh nghiệm và kỹ năng rất cần thiết để nhóm chúng em có thể hoàn thành đề tài đồ án tốt nghiệp.

Đặc biệt nhóm chúng em xin gửi lời cảm ơn chân thành đến thầy TS. Trần Vi Đô, người đã dành ra thời gian để tận tình giúp đỡ, hướng dẫn và theo sát nhóm chúng em trong suốt quá trình thực hiện đề tài thời gian qua.

Chúng em xin gửi lời cảm ơn đến các thầy cô phản biện và các thầy cô trong hội đồng bảo vệ luận văn đã dành thời gian để nhận xét và giúp đỡ chúng em trong quá trình bảo vệ luận văn.

Một lần nữa xin gửi lời cảm ơn đến toàn thể các bạn trong trong khóa 18151CLC đã giúp đỡ cũng như cho những góp ý trong quá trình làm đồ án tốt nghiệp.

Cuối lời, nhóm chúng em xin gửi đến các thầy cô, các bạn sinh viên, các cán bộ nhân viên Trường Đại Học Sư Phạm Kỹ Thuật Tp.HCM lời chúc sức khỏe, kính chúc toàn thể Nhà trường đạt được nhiều ngày càng phát triển và gặt hái được nhiều thành tựu to lớn.

Một lần nữa nhóm chúng em xin chân thành cảm ơn!

Người thực hiện đề tài

Lê Thành Trung

Trần Hoàng Dũng

MỤC LỤC

DANH MỤC CÁC HÌNH	i
DANH MỤC CÁC BẢNG	iii
DANH MỤC CÁC TỪ VIẾT TẮT	iv
LỜI MỞ ĐẦU	1
CHƯƠNG 1: TỔNG QUAN	2
1.1. Đặt vấn đề	2
1.2. Mục tiêu đề tài	4
1.3. Đối tượng nghiên cứu	4
1.4. Phạm vi nghiên cứu đề tài	4
1.5. Giới thiệu nội dung	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	6
2.1. Yêu cầu của hệ thống	6
2.1.1. <i>Phân loại sản phẩm theo màu sắc</i>	6
2.1.2. <i>Tìm toạ độ của vật và tiến hành phân loại</i>	6
2.2. Phương trình động lực học Robot Delta	6
2.2.1. <i>Cấu tạo tổng quan Robot Delta</i>	7
2.2.2. <i>Động học thuận Robot Delta</i>	7
2.2.3. <i>Động học nghịch Robot Delta</i>	10
2.3. Tổng quan lý thuyết về các công nghệ xử lý ảnh	13
2.3.1. <i>Các khái niệm trong xử lý ảnh</i>	13
2.3.2. <i>Xử lý hình thái học trong xử lý ảnh</i>	18
2.3.3. <i>Đường bao và các ứng dụng (Contours)</i>	21
CHƯƠNG 3. THIẾT KẾ PHẦN CỨNG	24
3.1. Yêu cầu cơ bản của hệ thống	24
3.1.1. <i>Nhận diện sản phẩm</i>	24
3.1.2. <i>Tìm toạ độ chính xác của vật và tiến hành gấp sản phẩm</i>	24
3.2. Yêu cầu thiết kế	24
3.2.1. <i>Xác định mục tiêu thiết kế</i>	24
3.2.2. <i>Lập phương án thiết kế</i>	25
3.3. Lựa chọn thiết bị	26
3.3.1. <i>Lựa chọn các thiết bị và tiến hành thi công mô hình</i>	26

3.3.2. <i>Lựa chọn bộ điều khiển</i>	32
3.3.3. <i>Các thiết bị linh kiện khác</i>	33
3.4. Sơ đồ kết nối phần cứng	35
CHƯƠNG 4. THUẬT TOÁN ĐIỀU KHIỂN	39
4.1. Thuật toán điều khiển PLC	39
4.2. Thuật toán xử lý ảnh	39
4.3. Thuật toán điều khiển robot Delta gấp vật	41
CHƯƠNG 5: KẾT QUẢ	43
5.1. Phần cứng	43
5.2. Phần mềm	44
5.2.1. <i>Phần mềm điều khiển và giám sát robot Delta</i>	44
5.2.2. <i>Robot hoạt động ở chế độ auto</i>	50
5.2.3. <i>Robot hoạt động ở chế độ manual</i>	52
5.3. Nhận xét kết quả đạt được	53
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	54
6.1. Kết luận	54
6.2. Hướng phát triển	54
TÀI LIỆU THAM KHẢO	55
PHỤ LỤC	56

DANH MỤC CÁC HÌNH

Hình 1.1 Robot Delta trong dây chuyền sản xuất công nghiệp	3
Hình 1.2 Ứng dụng Robot Delta trong nền y học	3
Hình 2.1 Cấu tạo cơ bản của Robot Delta	7
Hình 2.2 Mô hình động học nghịch robot Delta.....	8
Hình 2.3 Minh họa động học của cánh tay đòn.....	11
Hình 2.4 Mặt phẳng YZ của mô hình động học nghịch robot Delta.....	12
Hình 2.5 Quy đổi hệ toạ độ robot Delta	13
Hình 2.6 Minh họa về sự thay đổi mức xám	14
Hình 2.7 Không gian màu RGB	15
Hình 2.8 Các dạng ảnh số: (a) Ảnh màu; (b) Ảnh xám; (c) Ảnh nhị phân	16
Hình 2.9 Không gian màu HSV	16
Hình 2.10 Không gian màu HSV	17
Hình 2.11 Minh họa về phép giãn nở ảnh	19
Hình 2.12 Phép co ảnh.....	20
Hình 2.13 Minh họa về phép co ảnh.....	21
Hình 2.14 Phép đóng ảnh (Closing)	22
Hình 2.15 Minh họa tìm về vẽ đường bao của vật	23
Hình 2.16 Minh họa thuật toán xấp xỉ đường bao	23
Hình 2.17 Minh họa xác định tâm của vật bằng hàm Moments.....	26
Hình 3.1 Hình dạng của sản phẩm.....	25
Hình 3.2 Kết cấu của robot Delta 3 bậc tự do	26
Hình 3.3 Bản vẽ của robot Delta	26
Hình 3.4 Động cơ bước điều khiển cánh tay robot	27
Hình 3.5 Driver điều khiển động cơ bước TB6600.....	27
Hình 3.6 Rotary encoder 400.....	28
Hình 3.7 Công tắc hành trình KW11	28
Hình 3.8 Bơm hút chân không Micro 24V	28
Hình 3.9 Động cơ giảm tốc ZGB37R-530 24V	28
Hình 3.10 PLC S7-1200 với CPU 1212 DC/DC/DC.....	28

Hình 3.11 Mô đun mở rộng SM-1222 RLY	27
Hình 3.12 CB LS 40A	27
Hình 3.13 Nguồn tần số	28
Hình 3.14 Bộ chuyển đổi 220AC/12DC.....	28
Hình 3.15 Công tắc xoay	28
Hình 3.16 Nút nhấn xoay khẩn cấp	28
Hình 3.17 Đèn báo nguồn.....	29
Hình 3.18 Sơ đồ kết nối PLC với các thiết bị.....	30
Hình 3.19 Sơ đồ kết nối driver với step motor	31
Hình 3.20 Sơ đồ kết nối của mô đun mở rộng	31
Hình 4.1 Lưu đồ điều khiển của PLC	34
Hình 4.2 Lưu đồ xử lý ảnh	34
Hình 4.3 Lưu đồ điều khiển robot Delta.....	36
Hình 5.1 Robot Delta hoàn chỉnh	37
Hình 5.2 Giao diện đăng nhập hệ thống	38
Hình 5.3 Giao diện trang connection.....	39
Hình 5.4 Giao diện trang manual	42
Hình 5.5 Giao diện trang auto	42
Hình 5.6 Giao diện trang setting.....	43
Hình 5.7 Ứng dụng xử lý ảnh tìm được màu sắc, vẽ đường bao và tâm của vật	44
Hình 5.8 Robot gấp vật trên băng tải và thả vật tại vị trí 1 và 2.....	54

DANH MỤC CÁC BẢNG

Bảng 3.1 Danh sách các linh kiện của mô hình.....	26
Bảng 5.1 Phím chức năng trang connection	39
Bảng 5.2 Phím chức năng trang Manual	40
Bảng 5.3 Phím chức năng trang auto.....	42
Bảng 5.4 Phím chức năng trang setting.....	43
Bảng 5.5 Số liệu thực nghiệm khi gấp, thả vật vào vị trí	52

DANH MỤC CÁC TỪ VIẾT TẮT

PLC	Programmable Logic Controller
RGB	Red, Green và Blue
HSV	Hue, Saturation, Value
CB	Circuit breaker
RLY	Relay
OPC	OLE for Process Control
PC	Personal Computer
IoT	Internet of things
Man	Manual
Auto	Automatic

LỜI MỞ ĐẦU

Cuộc sống con người hiện nay cùng với sự phát triển mạnh của khoa học và kĩ thuật, chúng ta ngày càng đạt được nhiều thành tựu to lớn nhất định ở trong nhiều lĩnh vực khác như nền y học, kĩ thuật, công nghiệp hóa, công nghệ thông tin...

Với sự phát triển mạnh mẽ của các loại Robot công nghiệp hóa thì không thể không kể đến một loại cánh tay Robot đã được được ứng dụng rất nhiều trong lĩnh vực phân loại sản phẩm với nhiều đặc tính nổi trội so với các loại robot cổ điển đó là Robot Delta hay còn được gọi với tên gọi là Robot song song, chính vì thế nhóm chúng em đã chọn đề tài: “Điều khiển và giám sát hệ thống phân loại gói sản phẩm dùng Robot Delta”.

Mục tiêu, nhiệm vụ của đề tài là ứng dụng PLC S7-1200 và công nghệ xử lý ảnh để xây dựng thuật toán điều khiển và thiết kế mô hình Robot Delta phân loại gói sản phẩm theo màu sắc.

Robot Delta là một robot song song, tức là nó bao gồm nhiều chuỗi động học kết nối bộ cố định phận đầu cuối, cũng có thể coi là một dạng khái quát không gian của liên kết bốn thanh. Trong đó có ba nhánh, với mỗi nhánh, một đầu được nối vào giá di động thông qua cơ cấu hình bình hành. Ý tưởng chính của Robot Delta tận dụng cấu trúc hình học để tạo ra chuyển động cho đầu cuối mà vẫn giữ nguyên hướng, tức là chỉ chuyển động theo hướng X, Y hoặc Z mà không bị quay hướng. Đối tượng cần phân loại được nhận diện bằng camera sau đó gửi về PC các giá trị tọa độ, màu sắc hoặc mã vạch QR của đối tượng. Sau đó PC nhận dữ liệu sẽ tính toán qua phương trình động lực học đồng thời kết hợp với ứng dụng xử lý ảnh để tìm tọa độ của vật, gửi thông tin tới PLC để điều khiển robot tới gấp và thả sản phẩm vào ô phân loại được định hình sẵn. Hệ thống đã được lắp đặt và chạy thử nghiệm thực tế. Tỉ lệ xác định và gấp vật chính xác lên tới gần 99%. Phần mềm điều khiển Robot Delta được lập trình với cả 2 chế độ AUTO và MANUAL để có thể đa dạng linh hoạt hơn về nhiều hình thức hoạt động, mô phỏng quá trình phân loại sản phẩm trong công nghiệp.

CHƯƠNG 1: TỔNG QUAN

Robot Delta (Robot cánh tay song song) được phát minh vào đầu những năm 1980 bởi một nhóm nghiên cứu do giáo sư Reymond Clavel dẫn đầu tại EPFL (Thụy Sĩ). Ngày nay, Delta Robot là một loại robot được con người ứng dụng nhiều vô công nghiệp hóa hiện đại trên toàn thế giới, đặc biệt là trong khâu phân loại và đóng gói sản phẩm nhờ vào đặc tính có tốc độ và độ chính xác cao được đề ra trong nền công nghiệp hóa.

1.1. Đặt vấn đề

Hiện nay, với việc con người ứng dụng tự động hóa vào trong cuộc sống thì bên cạnh đó Robot nắm giữ một tầm quan trọng không thể thiếu trong nhiều ngành công nghiệp hóa và cuộc sống con người hiện nay.

Đặc biệt trong tình hình toàn thế giới đang đối diện với tình hình kinh tế sau khi đại dịch Covid -19 diễn ra trầm trọng trong suốt hơn 2-3 năm qua và hiện giờ cũng đang dần xuất hiện lại biến chủng mới của Covid đang dần lây lan nhanh và nguy hiểm hơn tới tính mạng con người làm cho nền kinh tế toàn bộ các nước bị biến động suy sụp 1 cách nặng nề. Chính vì thế nhu cầu sử dụng kỹ thuật robot sẽ là một trong những công nghệ tự động hóa sẽ lại càng được chú ý nhắm tới nhằm đảm bảo an toàn cho nhân công giúp công việc diễn ra một cách liên tục, mang lại độ chính xác cao, có thể làm việc 24/24 với 100% năng suất và làm việc tự động, không cần hoặc cần rất ít sự can thiệp của con người.

Các loại robot công nghiệp sử dụng trong những công việc nặng nhọc (Robot cắt – Cutting robot, Robot gấp sản phẩm – Handling Robot, Robot sơn – Panting Robot...) đến những công việc yêu cầu tính chính xác và liên tục (Robot Articulated, Robot Delta, robot SCARA...). Với các ưu điểm như trọng lượng nhẹ, kích thước nhỏ gọn, tốc độ nhanh, định vị chính xác, chi phí sản xuất thấp và hiệu quả cao, Robot Delta ngày càng được sử dụng rộng rãi trong các dây chuyền sản xuất như gấp và lắp ráp các sản phẩm điện tử, phân loại, đóng gói thực phẩm và đặc biệt hơn được ưu tiên sử dụng trong nền y học hiện đại như minh họa **Hình 1.1** và **Hình 1.2**.



Hình 1.1 Robot Delta ứng dụng trong dây chuyền sản xuất công nghiệp [2]



Hình 1.2 Ứng dụng Robot Delta trong nền y học [2]

1.2. Mục tiêu đề tài

Thiết kế, thi công, tính toán động học và lập trình điều khiển Robot Delta kết hợp ứng dụng xử lý ảnh xác định vị trí để gấp phân loại gói sản phẩm.

Xây dựng phần mềm điều khiển Robot Delta với 2 chế độ auto và manual. Mô hình có khả năng phát hiện chính xác vị trí tọa độ của sản phẩm trên băng tải, tiến hành gấp và đặt sản phẩm vào hộp định sẵn. Toàn bộ quy trình thao tác sẽ dễ dàng hơn khi được hiển thị giao diện giám sát trên máy tính.

1.3. Đối tượng nghiên cứu

Trong đề tài này, các gói sản phẩm sẽ là dạng hình chữ nhật và có kích thước giống nhau. Sản phẩm được đưa lên băng tải và được nhận dạng xác định tọa độ bằng camera và gửi về máy tính (PC). Sau đó PC nhận được dữ liệu sẽ tính toán qua phương trình động lực học để gửi thông tin cho PLC và tiến hành điều khiển robot tới vị trí tọa độ đó và gấp gói sản phẩm và thả vào hộp phân loại.

1.4. Phạm vi nghiên cứu đề tài

Dựa trên nền tảng mô hình và ý tưởng ứng dụng xử lý ảnh cho Robot Delta, nhóm đã kế thừa và hướng đến ứng dụng Robot Delta vào khâu phân loại gói sản phẩm theo màu sắc đồng thời cải tiến thêm phần cứng để tối ưu hóa hơn về nhiễu xạ ánh sáng trong khâu xử lý ảnh. Nghiên cứu về động học Robot Delta để lập trình tối ưu hóa thuật toán giúp cho hệ thống phát hiện sản phẩm, thực thi thao tác gấp – thả ở cường độ chính xác và tốc độ ổn định hơn.

1.5. Giới thiệu nội dung

Phần còn lại của đề tài có nội dung như sau:

Chương 2: Cơ sở lý thuyết

Trình bày về lý thuyết tính toán của phương trình động lực học thuận/nghịch robot và có cái nhìn tổng quan hơn về công nghệ xử lý ảnh được ứng dụng trong đề tài.

Chương 3: Thiết kế phần cứng

Trình bày yêu cầu thiết kế của hệ thống, từ đó thiết kế phần cứng cụ thể thỏa yêu cầu, lựa chọn các thiết bị phù hợp với hệ thống và trình bày sơ đồ kết nối các thiết bị.

Chương 1: Tổng Quan

Chương 4: Thuật toán điều khiển

Trình bày thuật toán điều khiển robot và thiết kế giao diện giám sát quy trình trên màn hình máy tính, điều khiển robot Delta.

Chương 5: Kết quả

Trình bày về những kết quả nhất định đã đạt được về phần cứng nói chung và phần mềm nói riêng.

Chương 6: Kết luận và hướng phát triển

Trình bày những vấn đề đã giải quyết trong đề tài được dựa trên các tiêu chí yêu cầu đã đặt ra trước đó và nêu một số hướng phát triển tối ưu giúp đề tài hoàn thiện thêm nhiều hơn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Chương này sẽ trình bày về lý thuyết tính toán của phương trình động lực học thuận/nghịch robot và có cái nhìn tổng quan hơn về công nghệ xử lý ảnh được ứng dụng trong đề tài.

2.1. Yêu cầu của hệ thống

2.1.1. Phân loại sản phẩm theo màu sắc

Đối với chế độ phân loại sản phẩm theo màu sắc, vật cần phân loại ở đây gồm những vật thực tế là gói thuốc Hapacol dễ dàng bắt gặp có các màu sắc là đỏ, xanh lá, xanh dương.

Đề tài sẽ hướng được tới phân loại sản phẩm theo 2 màu chính chủ đạo : cam và xanh dương.

Hệ thống cần phân loại một cách chính xác màu sắc của vật, do đó cần phải ứng dụng các phương pháp xử lý ảnh vào hệ thống từ đó tính toán và gửi tín hiệu về robot.

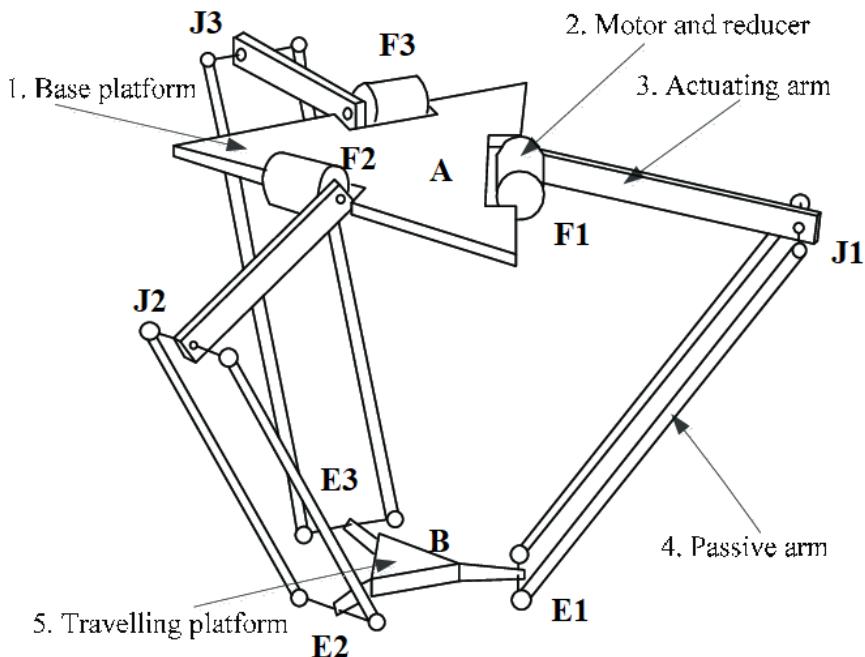
Với trên thị trường rất nhiều linh kiện được sử dụng trong giáo dục lẫn công nghiệp để phân loại sản phẩm ví dụ như cần gạt ngang, cảm biến kết hợp cùng cần đẩy, máy gấp,... Để đạt được đúng yêu cầu tiêu chí đề tài đề ra hệ thống phải gấp được sản phẩm một cách chắc chắn và nhanh nhẹn với tần suất liên tục và phù hợp với kinh tế của sinh viên thì nhóm chúng em đã lựa chọn linh kiện bơm hút chân không để gấp sản phẩm và đưa vào vị trí được xác định trước đó.

2.1.2. Tìm toạ độ của vật và tiến hành phân loại

Hệ thống được ứng dụng thuật toán xử lý ảnh kết hợp với camera, sau khi sản phẩm cần phân loại được đặt trên băng tải và camera phát hiện được, bước tiếp theo là hệ thống phải xác định được vị trí chính xác của vật và gửi các giá trị tọa độ về để robot tiến hành gấp vật. Do yêu cầu về điều khiển vị trí một cách chính xác của hệ thống, vì vậy nên cần ứng dụng phương pháp điều khiển vị trí (motion control) một cách tự động.

2.2. Phương trình động lực học Robot Delta

2.2.1. Cấu tạo tổng quan Robot Delta



Hình 2.1 Cấu tạo cơ bản của Robot Delta [3]

Cấu tạo của Robot Delta **Hình 2.1** bao gồm:

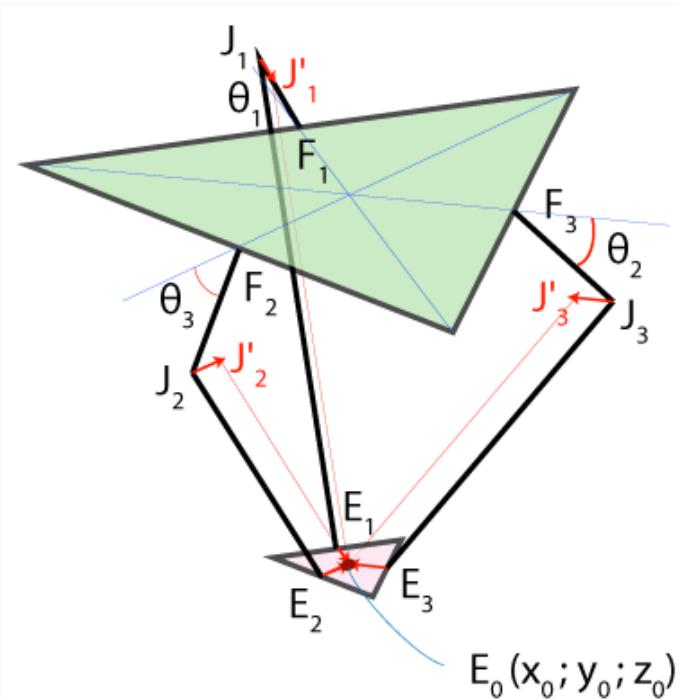
- Bàn máy động E ($E_{1,2,3}$) và bàn máy cố định F ($F_{1,2,3}$).
- Ba khớp chủ động F_1J_1 , F_2J_2 , F_3J_3 liên kết với bàn máy cố định (Base platform) bằng các khớp quay và được dẫn động bởi 3 động cơ, các động cơ này được gắn chặn với bàn máy cố định F.
- Ba khớp bị động J_1E_1 , J_2E_2 , J_3E_3 mỗi khớp là một cấu trúc hình bình hành.

Tính chất hình học của các khớp hình bình hành nên bàn máy động là một vật rắn chuyển động tịnh tiến trong không gian. Như vậy robot có 3 bậc tự do xác định bởi 3 tọa độ $\theta_1, \theta_2, \theta_3$ trong không gian.

2.2.2. Động học thuận Robot Delta

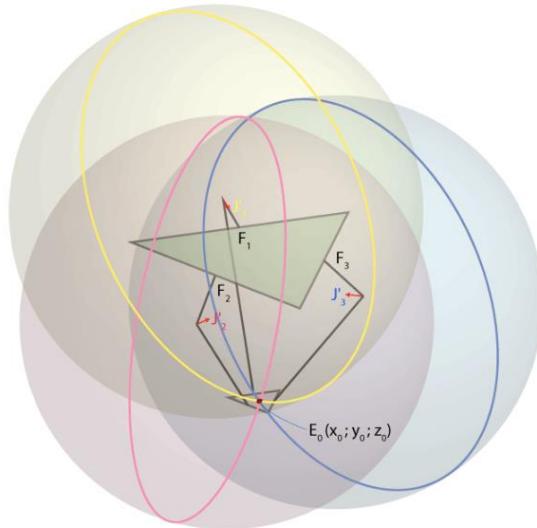
Từ các góc $\theta_1, \theta_2, \theta_3 \rightarrow$ Tọa độ (x_0, y_0, z_0) của điểm E_0 .

Nếu ta đã biết góc θ_1 , chúng ta có thể dễ dàng tìm thấy tọa độ của các điểm J_1, J_2 và J_3 . Các khớp nối J_1E_1, J_2E_2 và J_3E_3 có thể tự do xoay quanh các điểm J_1, J_2 và J_3 tương ứng, tạo thành ba hình cầu có bán kính r_e .



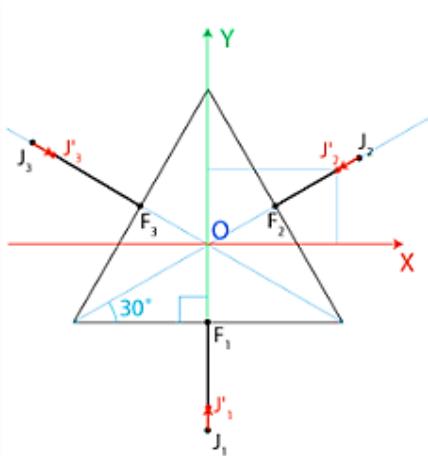
Hình 2.2 Mô hình động học Robot Delta [3]

Ta di chuyển các tâm của các hình cầu từ các điểm J_1, J_2 và J_3 đến các điểm J'_1, J'_2 và J'_3 bằng cách sử dụng các vector chuyển tiếp E_1E_0, E_2E_0 và E_3E_0 tương ứng. Sau quá trình chuyển đổi này, tất cả ba hình cầu sẽ giao nhau tại một điểm: E_0 như được thể hiện trong **Hình 2.3** phía dưới:



Hình 2.3 Mô hình động học Robot Delta [3]

Vì vậy, để tìm tọa độ (x_0, y_0, z_0) của điểm E_0 , chúng ta cần giải quyết tập hợp ba phương trình như $(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_e^2$, nơi tọa độ các tâm hình cầu (x_i, y_i, z_i) và bán kính r_e đã biết.



Hình 2.4 Tọa độ các điểm J'_1, J'_2, J'_3 trên mặt phẳng Oxy [3]

$$J'_1 \left(0; -\frac{f-e}{2\sqrt{3}} - r_f \cos \theta_1; -r_f \sin \theta_1 \right) \quad (2.1)$$

$$J'_2 \left(\left(\frac{f-e}{2\sqrt{3}} + r_f \cos \theta_2 \right) \cos 30^\circ; \left(\frac{f-e}{2\sqrt{3}} + r_f \cos \theta_2 \right) \sin 30^\circ; -r_f \sin \theta_2 \right) \quad (2.2)$$

$$J'_3 \left(\left(\frac{f-e}{2\sqrt{3}} + r_f \cos \theta_3 \right) \cos 30^\circ; \left(\frac{f-e}{2\sqrt{3}} + r_f \cos(\theta_3) \right) \sin 30^\circ; -r_f \sin \theta_3 \right) \quad (2.3)$$

Trong các phương trình sau đây ta sẽ chỉ định tọa độ của các điểm J_1, J_2, J_3 là (x_1, y_1, z_1) , (x_2, y_2, z_2) và (x_3, y_3, z_3) . Lưu ý rằng $x_1 = 0$. Dưới đây là phương trình của ba hình cầu:

$$\left\{ \begin{array}{l} x^2 + y^2 + z^2 - 2y_1y - 2z_1z = r_e^2 - y_1^2 - z_1^2 \\ x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z = r_e^2 - x_2^2 - y_2^2 - z_2^2 \end{array} \right. \quad (2.4)$$

$$\left\{ \begin{array}{l} x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z = r_e^2 - x_3^2 - y_3^2 - z_3^2 \end{array} \right. \quad (2.5)$$

$$\left\{ \begin{array}{l} x^2 + y^2 + z^2 - 2y_1y - 2z_1z = r_e^2 - y_1^2 - z_1^2 \\ x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z = r_e^2 - x_2^2 - y_2^2 - z_2^2 \\ x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z = r_e^2 - x_3^2 - y_3^2 - z_3^2 \end{array} \right. \quad (2.6)$$

Đặt: $w_i = x_i^2 + y_i^2 + z_i^2$, ta được:

$$\left\{ \begin{array}{l} x_2x + (y_1 - y_2)y + (z_1 - z_2)z = \frac{(w_1 - w_2)}{2} \\ x_3x + (y_1 - y_3)y + (z_1 - z_3)z = \frac{(w_1 - w_3)}{2} \end{array} \right. \quad (2.7)$$

$$\left\{ \begin{array}{l} x_2x + (y_2 - y_3)y + (z_2 - z_3)z = \frac{(w_2 - w_3)}{2} \end{array} \right. \quad (2.8)$$

$$\left\{ \begin{array}{l} (x_2 - x_3)x + (y_2 - y_3)y + (z_2 - z_3)z = \frac{(w_2 - w_3)}{2} \end{array} \right. \quad (2.9)$$

Giải hệ 3 phương trình trên ta được:

$$(a_1^2 + a_2^2 + 1)z^2 + 2(a_1 + a_2 - z_1)z + (b_1^2 + (b_2 - y_1)^2 + z_1^2 - r_e^2) = 0 \quad (2.10)$$

Cuối cùng, ta cần giải phương trình này và tìm z_0 (nên chọn gốc phương trình âm nhỏ nhất) và sau đó tính x_0 và y_0 từ (2.7), (2.8) và (2.9).

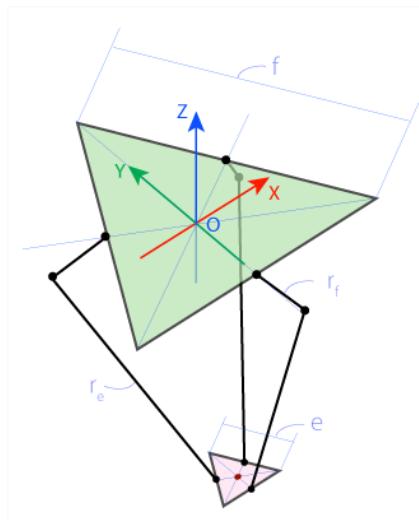
2.2.3. Động học nghịch Robot Delta

Phương trình động học nghịch của Robot Delta là một phương trình quan trọng để xác định góc quay của robot khi biết được tọa độ chính xác của vật, mô hình động học nghịch này được biểu diễn như **Hình 2.5** bên dưới.

Ta có các thông số đã được xác định khi thiết kế:

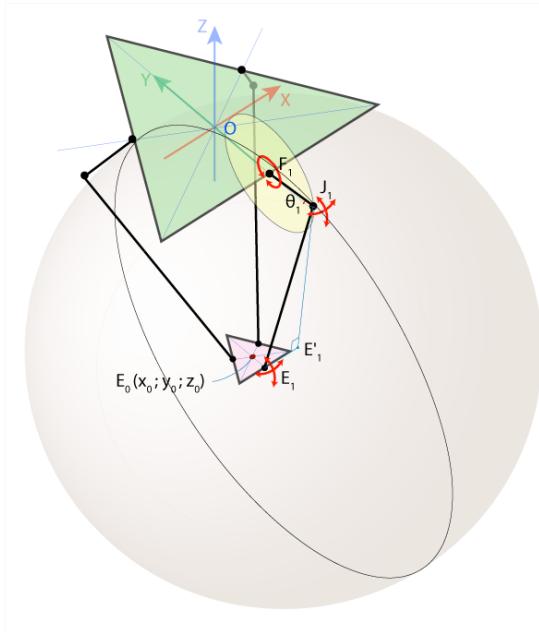
- + f : chiều dài cạnh của bệ cố định hình tam giác phía trên.
- + e : chiều dài cạnh của bệ di động hình tam giác ở phía dưới.
- + r_f, r_e : chiều dài phần trên và dưới cánh tay robot.

Do khớp nối của robot F_1J_1 chỉ có thể xoay trong mặt phẳng YZ, tạo thành hình tròn với tâm tại điểm F_1 và bán kính r_f .



Hình 2.5 Mô hình động học nghịch Robot Delta [4]

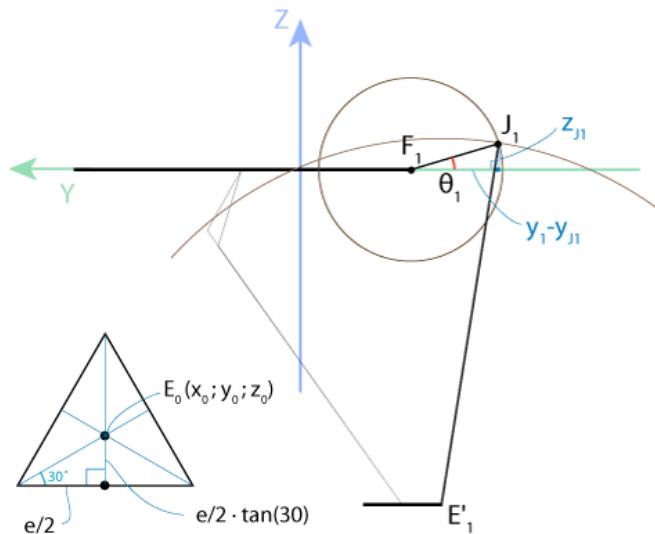
J_1 và E_1 được gọi là khớp nối phẳng quát, có nghĩa là E_1J_1 có thể xoay tự do tương đối với E_1 , tạo thành hình cầu với tâm tại điểm E_1 và bán kính r_e như **Hình 2.6**.



Hình 2.6 Minh họa động học của cánh tay đòn robot [4]

Giao điểm của hình cầu này và mặt phẳng YZ là một đường tròn có tâm tại điểm E'_1 và bán kính E'_1J_1 (trong đó E'_1 là phép chiếu của điểm E_1 trên mặt phẳng YZ). Điểm J_1 có thể được tìm thấy bây giờ như là giao điểm của các đường tròn bán kính đã biết với tâm là E'_1 và F_1 và nếu biết được J_1 thì ta có thể tính được góc θ_1 .

Xét mặt phẳng YZ được biểu diễn như **Hình 2.7** bên dưới:



Hình 2.7 Mặt phẳng YZ của mô hình động học nghịch Robot Delta [4]

Phương trình toán học được suy ra từ hình 2.7:

$$E(x_0, y_0, z_0)$$

$$EE_1 = \frac{e}{2} \tan 30^\circ \quad (2.11)$$

$$E_1(x_0, y_0 - \frac{e}{2\sqrt{3}}, z_0) \rightarrow E_1'(0, y_0 - \frac{e}{2\sqrt{3}}, z_0) \quad (2.12)$$

$$EE_1' = x_0 \Rightarrow E_1'J_1 = \sqrt{(E_1J_1^2 - E_1E_1'^2)} = \sqrt{(r_e^2 - x_0^2)} \quad (2.13)$$

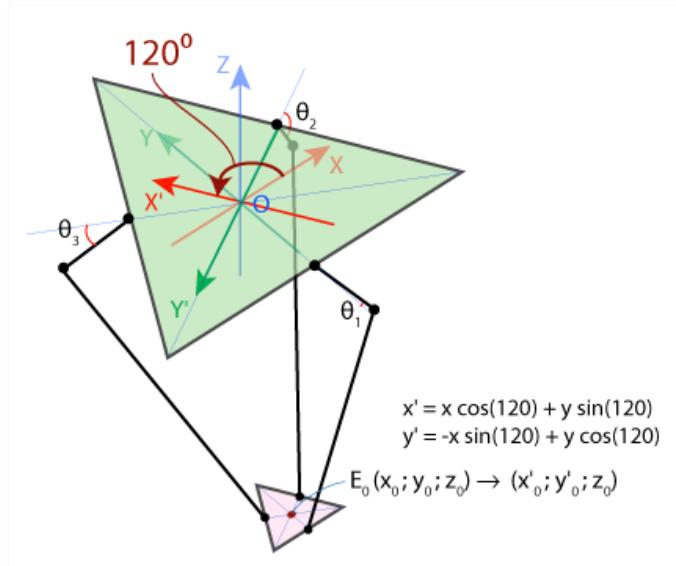
$$F_1(0, \frac{-f}{2\sqrt{3}}, 0)$$

$$\begin{cases} (y_{J_1} - y_{F_1})^2 + (z_{J_1} - z_{F_1})^2 = r_f^2 \\ (y_{J_1} - y_{E_1'})^2 + (z_{J_1} - z_{E_1'})^2 = r_e^2 - x_0^2 \end{cases} \quad (2.14)$$

$$\Rightarrow \begin{cases} \left(y_{J_1} + \frac{f}{2\sqrt{3}}\right)^2 + z_{J_1}^2 = r_f^2 \\ \left(y_{J_1} - y_0 + \frac{e}{2\sqrt{3}}\right)^2 + (z_{J_1} - z_0)^2 = r_e^2 - x_0^2 \end{cases} \Rightarrow J_1(0, y_{J_1}, z_{J_1}) \quad (2.15)$$

$$\theta_1 = \arctan\left(\frac{z_{J_1}}{y_{F_1} - y_{J_1}}\right) \quad (2.16)$$

Do khớp F_1J_1 chỉ di chuyển trong mặt phẳng YZ, do đó chúng ta có thể hoàn toàn bỏ qua toạ độ X. Tận dụng tính chất này cho việc tính toán các góc còn lại θ_2 , θ_3 và θ_3 , chúng ta có thể sử dụng tính đối xứng của Robot Delta. Đầu tiên, hãy xoay hệ toạ độ trong mặt phẳng XY xung quanh trục Z một góc 120° theo hướng ngược chiều kim đồng hồ như **Hình 2.8** dưới ta sẽ có khung tham chiếu mới X'Y'Z'.



Hình 2.8 Quy đổi hệ tọa độ robot Delta [4]

Với khung tham chiếu này chúng ta có thể tìm thấy góc θ_2 bằng cách sử dụng cùng một thuật toán mà ta sử dụng tìm góc θ_1 .

Sự thay đổi duy nhất là ta cần phải xác định tọa độ x'_0 và y'_0 cho điểm E_0 , có thể dễ dàng thực hiện bằng cách sử dụng “ma trận xoay” tương ứng. Ta có:

$$x'_0 = x \cos(120) + y \sin(120) \quad (2.17)$$

$$y'_0 = -x \sin(120) + y \cos(120) \quad (2.18)$$

Cuối cùng còn góc θ_3 có thể được tìm tương tự như θ_2 bằng cách xoay hệ trục tọa độ theo hướng chiều kim đồng hồ. Ta có:

$$x'_0 = x \cos(120) - y \sin(120) \quad (2.19)$$

$$y'_0 = x \sin(120) + y \cos(120) \quad (2.20)$$

2.3. Tổng quan lý thuyết về các công nghệ xử lý ảnh

2.3.1. Các khái niệm trong xử lý ảnh

Cũng như xử lý dữ liệu bằng đồ họa, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa để cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hóa các ảnh tự nhiên.

a. Ảnh đen trắng và ảnh màu

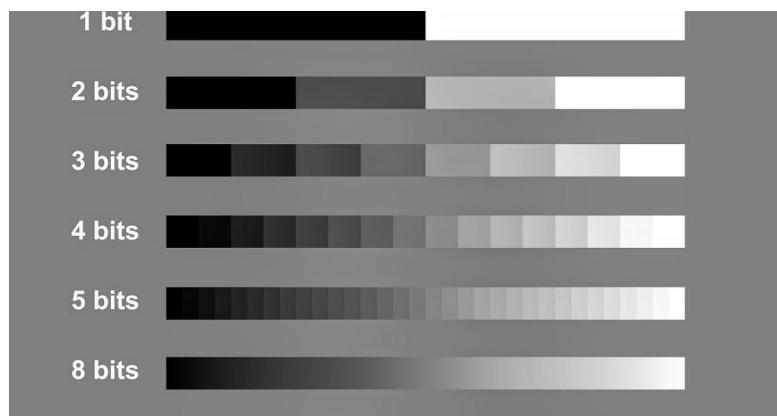
Ảnh đen trắng chỉ bao gồm 2 màu: màu đen và màu trắng. Người ta phân mức đen trắng đó thành L mức. Nếu sử dụng số bit $B=8$ bit để mã hóa mức đen trắng (hay mức xám) thì L được xác định:

$$L=2^B \text{ (trong ví dụ của ta } L=2^8= 256 \text{ mức)}$$

Nếu L bằng 2, $B=1$, nghĩa là chỉ có 2 mức: mức 0 và mức 1, còn gọi là ảnh nhị phân. Mức 1 ứng với màu sáng, còn mức 0 ứng với màu tối. Nếu L lớn hơn 2 ta có ảnh đa cấp xám.

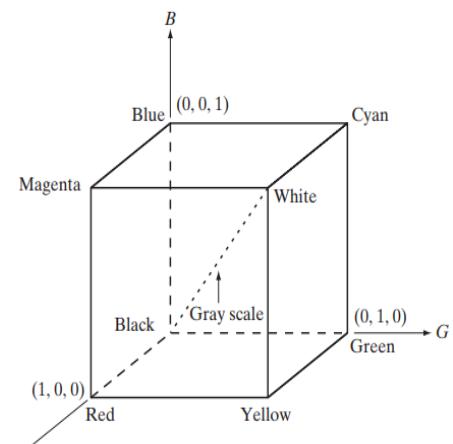
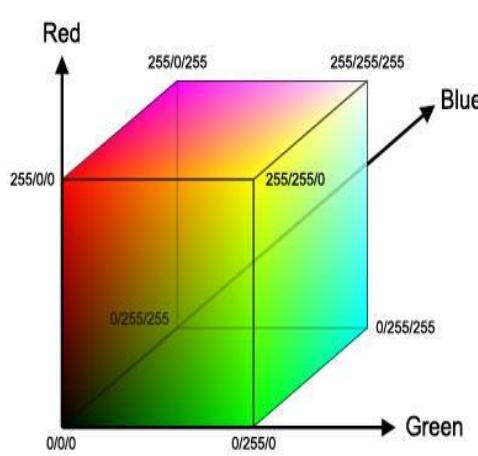
Nói cách khác, với ảnh nhị phân mỗi điểm ảnh được mã hóa trên 1 bit, còn với ảnh 256 mức, mỗi điểm ảnh được mã hóa trên 8 bit. Như vậy, với ảnh đen trắng: nếu dùng 8 bit (1 byte) để biểu diễn mức xám, số các mức xám có thể biểu diễn được là 256. Mỗi mức xám được biểu diễn dưới dạng là một số nguyên nằm trong khoảng từ 0 đến 255, với mức 0 biểu diễn cho mức cường độ đen nhất và 255 biểu diễn cho mức cường độ sáng nhất.

Ví dụ minh họa về sự thay đổi mức xám được thể hiện trong **Hình 2.9**. [1]

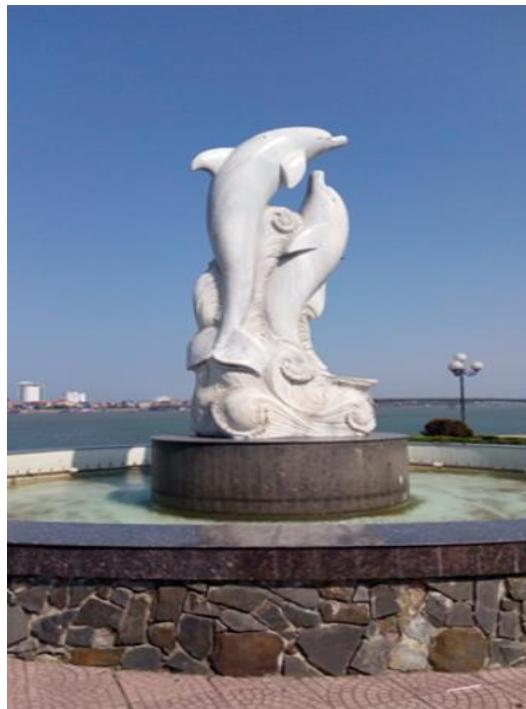


Hình 2.9 Minh họa về sự thay đổi mức xám

Ảnh màu: Theo lý thuyết của Thomas là ảnh tổ hợp 3 màu cơ bản: đỏ (Red), lục (Green), xanh lơ (Blue) và thường thu nhận trên các dải băng tần khác nhau. Các kênh màu Red, Green và Blue (trong không gian màu RGB) có thể tạo ra các màu khác nhau bằng phương pháp pha trộn. Với ảnh màu, cách biểu diễn cũng tương tự như với ảnh đen trắng, chỉ khác là các số tại mỗi phần tử của ma trận biểu diễn cho ba màu riêng rẽ. Để thuận tiện biểu diễn, giả thuyết các giá trị màu được chuẩn hóa nằm trong phạm vi $[0,1]$ như **Hình 2.10**. Với việc chuẩn hóa 256 (2^8) mức cho từng kênh màu chính, từ đó có thể thấy một pixel màu có thể biểu diễn được một trong $(2^8)^3 = 16,777$ triệu màu khác nhau. [1]



Hình 2.10 Không gian màu RGB



(a) Ảnh màu



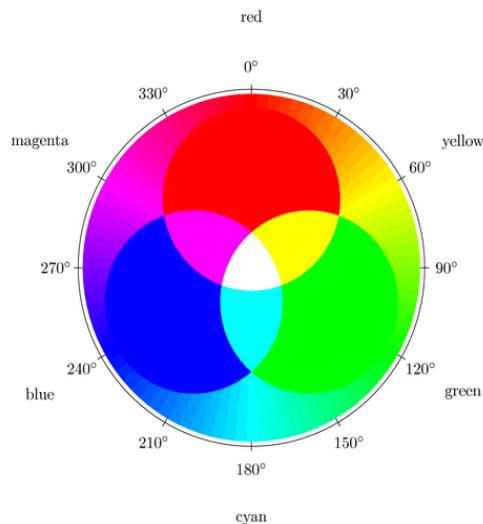
(b) Ảnh xám



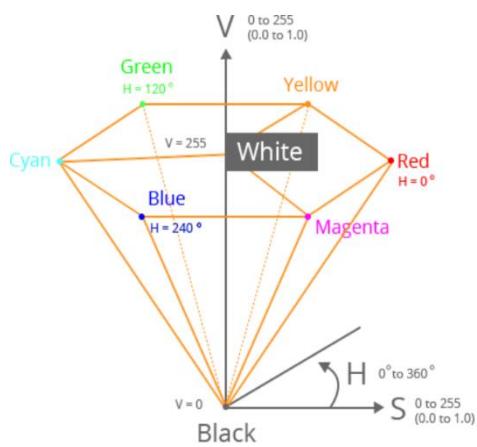
(c) Ảnh nhị phân

Hình 2.11 Các dạng ảnh số: (a) Ảnh màu; (b) Ảnh xám; (c) Ảnh nhị phân

b. Không gian màu HSV



Hình tròn HSV



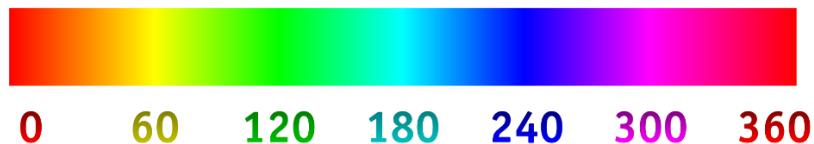
Hình nón HSV

Hình 2.12 Không gian màu HSV

Không gian màu HSV hay còn gọi là không gian màu HSB, là một không gian màu dựa trên ba số liệu:

Chương 2: Cơ sở lý thuyết

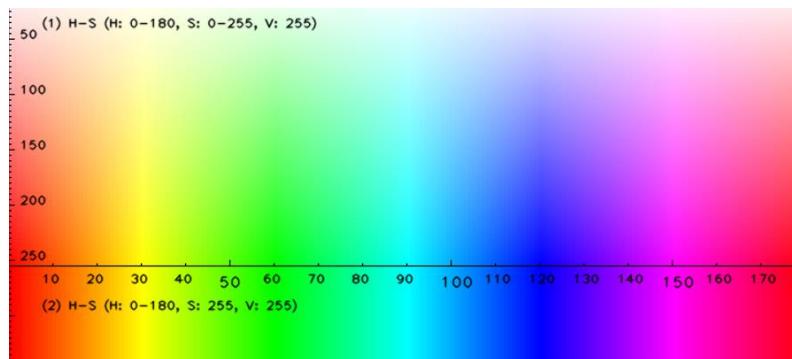
- H (Hue): là vùng màu, thể hiện sự thay đổi màu sắc từ 0 đến 360 độ với: góc 0° cho màu đỏ, 120° cho màu xanh lục và 240° cho màu xanh lam.
- S (Saturation): có nghĩa là độ bão hòa, là lượng màu xám trong màu, từ 0 đến 100 phần trăm. Một hiệu ứng mờ nhạt có thể có được từ việc giảm độ bão hòa về không để có được màu xám hơn. Tuy nhiên, độ bão hòa đổi khi được xem trên phạm vi từ 0-1, trong đó 0 là màu xám và 1 là màu chính.
- V (hay B) (Value hay Bright): là giá trị hoặc độ sáng kết hợp với độ bão hòa và mô tả độ sáng hoặc cường độ của màu sắc, từ 0-100%, trong đó 0 là hoàn toàn đen và 100 là sáng nhất và cho thấy màu sắc nhất.



Hình 2.13 Dải màu HSV

c. Xử lý ảnh với không gian màu HSV trong thư viện EmguCV

Trong thư viện EmguCV, giá trị của kênh Hue có phạm vi [0,179], Saturation và Value có phạm vi [0,255].



Hình 2.14 Không gian màu HSV

Thông thường ảnh được biểu diễn trong không gian màu BGR, nhưng không gian màu HSV lại dễ dàng xử lý hơn. Dãy màu của kênh Hue là “bảy sắc cầu vồng” từ đỏ đến tím rồi lại về đỏ, nên có thể xác định được ngay nó là màu gì. Do đó trong hầu hết bài toán đều phải tiến hành chuyển ảnh ban đầu sang không gian màu HSV trước khi xử lý những bước tiếp theo.

Trong bài toán khoanh vùng màu của sản phẩm cần gấp, trường hợp sản phẩm có màu vàng, nhóm chúng em sẽ lọc ra những điểm ảnh có màu “gần” vàng qua các bước sau:

- Chuyển ảnh gốc từ BGR sang HSV
- Giới hạn vùng mã của màu vàng, hay những giá trị trong không gian màu HSV “gần” vàng.
- Lọc ra những pixel có giá trị nằm trong giới hạn mã màu tìm được.

Sở dĩ phải tìm các pixel có màu gần “vàng” vì trên thực tế màu vàng hay bất kì màu gì mà ta nói đến đều không có giá trị chính xác cả. Ví dụ các màu đỏ cam, đỏ tươi, đỏ thắm, đỏ tươi hay đỏ hồng, ... luôn có sự pha trộn với các màu sắc khác nhưng ta vẫn gọi chung đó là màu đỏ. Và việc lựa chọn ngưỡng khoanh vùng màu sắc cũng thay đổi tùy thuộc màu sắc của ảnh mà ta thu được. Những yếu tố như ánh sáng và chất lượng ảnh của camera cũng có thể làm sai lệch đi ngưỡng khoanh vùng.

EmguCV cung cấp hàm `inRange()` để lọc ra những pixel có giá trị nằm trong vùng ta chọn.

2.3.2. Xử lý hình thái học trong xử lý ảnh

a. Phép giãn nở ảnh (Dilation)

Phép toán này có tác dụng làm cho đối tượng ban đầu trong ảnh tăng lên về kích thước (giãn nở ra). Việc làm giãn hình ảnh dẫn đến các đối tượng trở nên lớn hơn, các lỗ trống nhỏ được lấp đầy, loại bỏ các phần tử nhiễu và hợp nhất các đối tượng với nhau.

Hình ảnh gốc được giãn nở bằng cách sử dụng phần tử cấu trúc được chỉ định để xác định hình dạng lân cận của các pixel. Mỗi vị trí phần tử cấu trúc quét qua sẽ được chọn giá trị lớn nhất và trả về mốc tương ứng trên ảnh sau khi thực hiện phép giãn nở. [5]

Ví dụ: Ta có ma trận điểm ảnh I_{src} , phần tử cấu trúc B với $(0,0)$ là điểm gốc, ma trận sau phép giãn nở I_{dst} . Nếu ta lần lượt đặt phần tử cấu trúc vào các phần tử có giá trị 1 của ma trận điểm ảnh I_{src} . Kết quả thu được là ma trận điểm ảnh I_{dst} .

$$I_{src} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad I_{dst} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Hình 2.15 Phép giãn nở ảnh [1]



Hình 2.16 Minh họa về phép giãn nở ảnh [5]

b. Phép co ảnh (Erosion)

Ngược lại với phép giãn nở, phép co ảnh làm cho các đối tượng trong hình ảnh nhị phân bị giảm kích thước. Sự co ảnh của một hình ảnh dẫn đến các đối tượng trở nên nhỏ hơn, các đối tượng nhỏ biến mất và các đối tượng lớn hơn bị tách thành các đối tượng nhỏ hơn. [5]

Phép co của ảnh gốc bởi phần tử cấu trúc, mỗi vị trí phần tử cấu trúc quét qua sẽ được chọn giá trị nhỏ nhất và trả về giá trị tương ứng trên ảnh sau khi thực hiện phép co ảnh.

Chương 2: Cơ sở lý thuyết

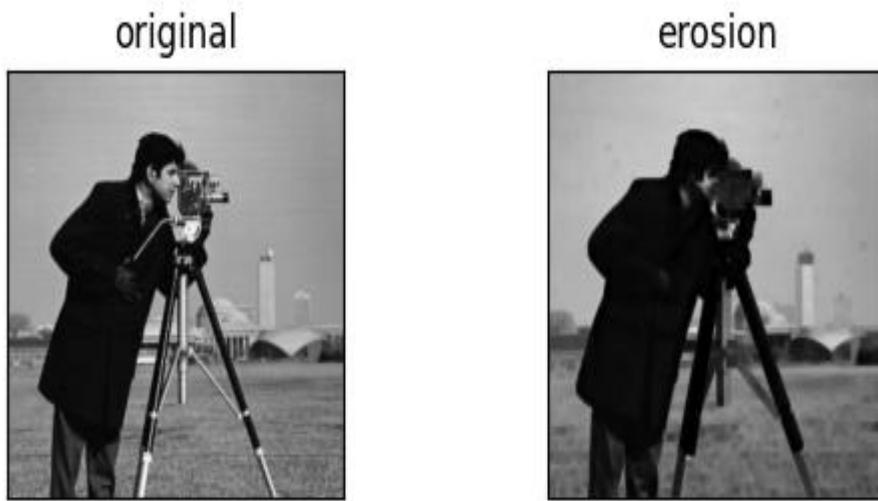
Ví dụ: Ta có ma trận điểm ảnh I_{src} , phần tử cấu trúc B với $(0,0)$ là điểm gốc, ma trận sau phép giãn nở I_{dst} . Nếu ta lần lượt đặt phần tử cấu trúc vào các phần tử có giá trị 1 của ma trận điểm ảnh I_{src} . Kết quả thu được là ma trận điểm ảnh I_{dst} .

$$I_{src} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad I_{dst} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Hình 2.17 Phép co ảnh [5]

Giải thích:

Ở đây không cần quan tâm toàn bộ các điểm đen (giá trị 1) ở I_{src} mà chỉ quan tâm tới các điểm khi di chuyển phần tử cấu trúc B trên đối tượng ảnh I_{src} thì gốc của B trùng với một điểm ảnh và các điểm mang giá trị 1 của B cũng trùng với điểm đen (mang giá trị 1) của I_{src} . Theo đó, ta có 6 điểm đen trên I_{src} phù hợp với điều kiện trên: $I_{src}(2,2)$, $I_{src}(3,2)$, $I_{src}(3,3)$, $I_{src}(3,4)$, $I_{src}(5,2)$, $I_{src}(2,4)$.



Hình 2.18 Minh họa về phép co ảnh [5]

c. Phép đóng ảnh (Closing)

Khi phép giãn nở ảnh (Dilation) được thực hiện, kích thước của ảnh sẽ được tăng lên đồng thời các lỗ trống trên ảnh sẽ được lấp đầy. Sau đó phép co ảnh (Erosion) sẽ được thực hiện để thu nhỏ ảnh về lại kích thước ban đầu. Quá trình này gọi là phép đóng ảnh (Closing).

Phép đóng ảnh (Closing) được thực hiện trong các ứng dụng làm trơn đường bao các đối tượng, lấp đầy các khoảng trống biên - làm mượt đường biên, loại bỏ các lỗ trống nhỏ gây nhiễu.



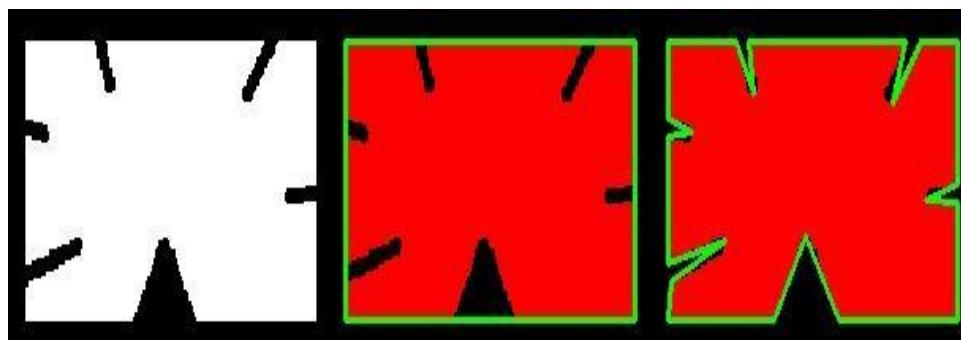
Hình 2.19 Phép đóng ảnh (Closing) [5]

2.3.3. Đường bao và các ứng dụng (Contours)

a. Khái niệm đường bao

Đường bao là một đường cong nối tất cả các điểm liên tục (dọc theo đường biên) có cùng màu hoặc cường độ. Các đường bao là một công cụ hữu ích để phân tích hình dạng, phát hiện và nhận dạng đối tượng.

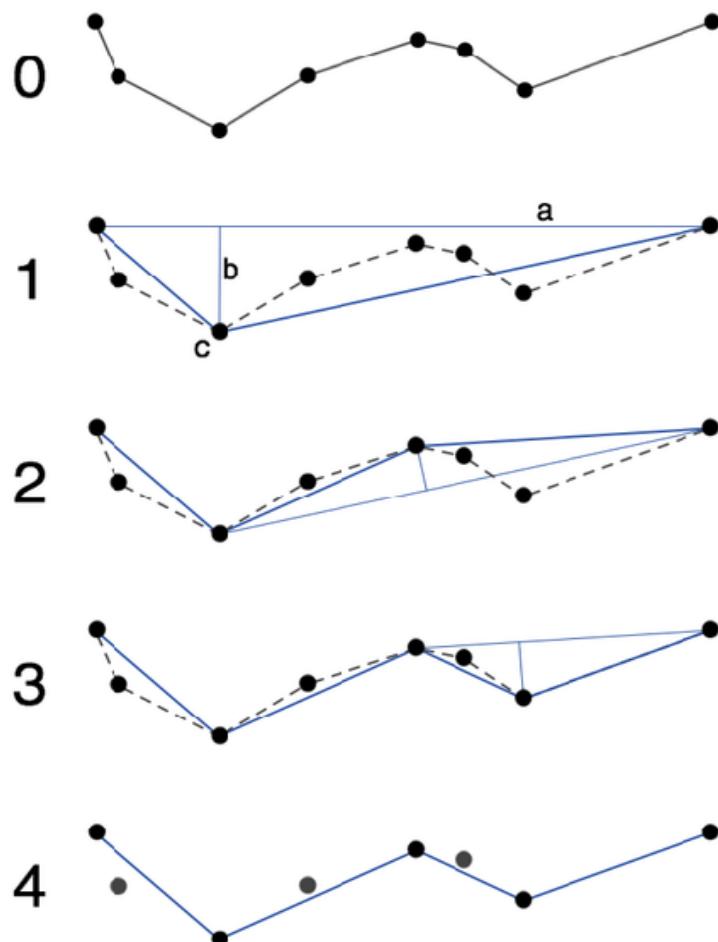
Đường bao là một dữ liệu tương đối quan trọng để từ đó có thể xác định được hình dạng và tìm chính xác tâm của vật thể đó.



Hình 2.20 Minh họa tìm và vẽ đường bao của vật [6]

b. Thuật toán xấp xỉ đường bao (Contour approximation)

Thuật toán này nhằm mục đích đơn giản hóa đa đường bằng cách giảm các đỉnh của nó cho một giá trị ngưỡng. Nói cách khác, chúng ta lấy một đường cong và giảm số đỉnh của nó trong khi vẫn giữ nguyên hình dạng của nó.



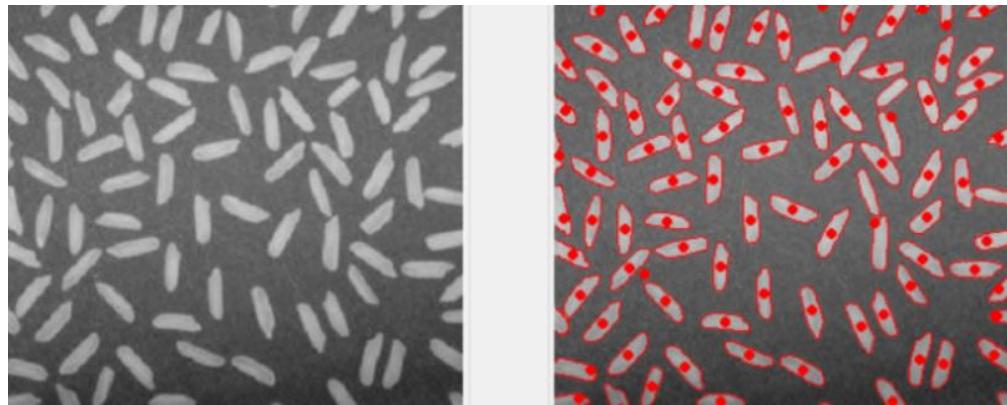
Hình 2.21 Minh họa thuật toán xấp xỉ đường bao [6]

Thuật toán này đầu tiên sẽ lưu toạ độ điểm đầu và điểm cuối của đường cong và sau đó nó vẽ đường ngắn nhất nối hai điểm này lại. Sau đó thuật toán xác định điểm xa nhất từ đoạn thẳng là điểm đầu mới, điểm đầu và cuối cũ bây giờ trở thành điểm cuối mới. Bất kỳ điểm nào trong nằm trong khoảng epsilon từ đường thẳng này sẽ bị xóa, quá trình lặp lại liên tục cho đến khi không còn điểm xa nhất nữa.

c. Image Moment

Chương 2: Cơ sở lý thuyết

Image Moment là giá trị trung bình có trọng số cụ thể của cường độ pixel hình ảnh. Các thuộc tính đơn giản của hình ảnh như diện tích, tâm và thông tin về hướng của ảnh có thể được xác định thông qua hàm Moment .



Hình 2.22 Minh họa xác định tâm của vật bằng hàm Moment [6]

CHƯƠNG 3. THIẾT KẾ PHẦN CỨNG

Chương này trình bày yêu cầu cơ bản của hệ thống, từ đó thiết kế phần cứng cụ thể thỏa yêu cầu, lựa chọn các thiết bị phù hợp với hệ thống và trình bày sơ đồ kết nối các thiết bị.

3.1. Yêu cầu cơ bản của hệ thống

3.1.1. Nhận diện sản phẩm

Sản phẩm được đưa vào hệ thống thông qua băng tải di chuyển sẽ được cánh tay Robot gấp và thả ở vị trí chính xác được thiết kế sẵn. Chính vì vậy, cần phải được nhận diện hình ảnh và tọa độ sản phẩm một cách chính xác thông qua các ứng dụng, thuận toán và phương pháp xử lý ảnh.

Để tài sử dụng sản phẩm theo hướng thực tế có hình dạng và mẫu mã gói thuốc Hapacol như **Hình 3.1**. Sản phẩm có kích thước 85mm x 55mm và 2 màu cam và xanh dương.



Hình 3.1 Hình dạng, màu sắc của sản phẩm

3.1.2. Tìm tọa độ chính xác của vật và tiến hành gấp sản phẩm

Với phương pháp xử lý ảnh sản phẩm sau khi được nhận diện qua camera, sau đó sản phẩm phải xác định được vị trí chính xác, vẽ đường bao, tiếp đến xác định chính xác tọa độ tâm của sản phẩm và tiến hành gửi các giá trị tọa độ về PC để robot thu nhập dữ liệu và robot tiến hành gấp sản phẩm theo đúng giá trị tọa độ nhận được.

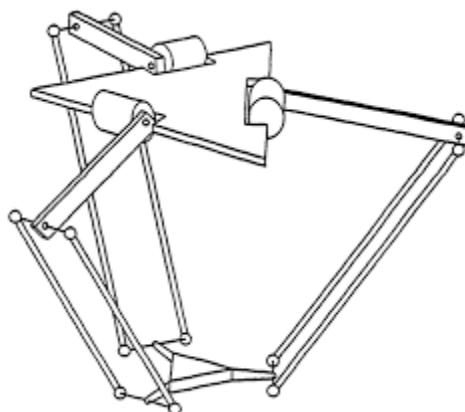
3.2. Yêu cầu thiết kế

3.2.1. Xác định mục tiêu thiết kế

Mục tiêu của đề tài đề ra là nhận diện và gấp các sản phẩm đang được di chuyển trên băng chuyền nên yêu cầu đề chuyển động dưới của Robot Delta phải chuyển động song phẳng với đề có định nên phương án thiết kế phải đáp ứng được số bậc tự do của cơ cầu là 3 bậc. Ngoài ra việc robot có chuyển động song phẳng của đề dưới, robot còn được gắn thêm một đầu hút vào chính giữa tâm đề dưới để gấp vật từ băng tải và thả vào vị trí theo mong muốn.

3.2.2. *Lập phương án thiết kế*

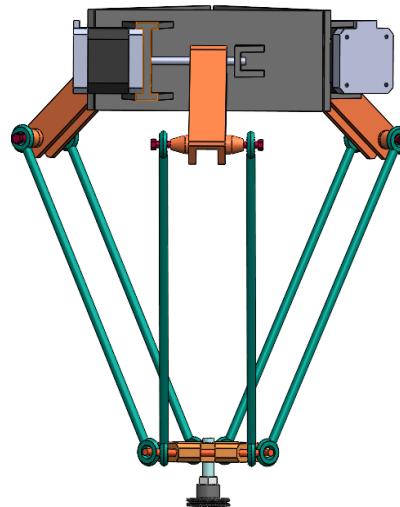
Đề tài chọn kết cấu sơ đồ động học như **Hình 3.2** để thiết kế. Đây là loại sơ đồ động học giống với nguyên bản ban đầu của loại Robot Delta nhất với các động cơ quay gắn trực tiếp với các cơ cầu hình bình hành. Với những ưu điểm vượt trội như độ cứng vững cao, khả năng thay đổi vị trí và định hướng linh hoạt, độ chính xác và ổn định cao nên cấu hình của Robot Delta này phù hợp với việc gấp sản phẩm trên băng chuyền và sắp xếp vào hộp với tọa độ được thiết kế sẵn. Với cấu trúc robot đã được lựa chọn như trên thì cần tới ba động cơ để tạo ra chuyển động quay cho các cánh tay robot phía trên cùng.



Hình 3.2 Kết cấu của Robot Delta 3 bậc tự do

Cánh tay của robot được xếp thành 3 nhánh đôi và được truyền động từ 3 động cơ riêng biệt như ở **Hình 3.3**. Mỗi tay gồm có 2 phần: phần trên và phần dưới. Cánh tay được nối với giá đỡ băng hộp số được gá chặt trên khung. Phần trên và phần dưới cánh tay cũng như phần dưới cánh tay và bệ chuyển động được liên kết với nhau bằng khớp cầu (mắt trâu). Các cánh tay robot được gia công bằng vật liệu nhựa và sợi carbon với ưu điểm khối lượng nhỏ, dễ gia công chính xác, độ thẩm mỹ cao do không bị ăn mòn hay oxy hóa do tác

động môi trường làm việc, không có từ tính nên không gây nhiễu việc hoạt động của các thiết bị điện.



Hình 3.3 Bản vẽ của Robot Delta

3.3. Lựa chọn thiết bị

3.3.1. Lựa chọn các thiết bị và tiến hành thi công mô hình

Bảng 3.1 Danh sách các linh kiện thi công mô hình

STT	Tên linh kiện	Thông số kỹ thuật	Số lượng	Ghi chú
1	Bộ in 3D robot Delta + 3 thanh carbon măt trâu	Trình bày bên dưới	1	
2	Step motor	Điện áp: 2.75VDC Đòng điện: 2A	3	
3	Driver động cơ TB6600	Điện áp: 12VDC Đòng điện: 2A	3	
4	Bơm chân không	Điện áp: 24VDC	1	Có kèm van
5	Nút hút chân không	Đường kính: 2cm	1	Nút hút chân không
6	Công tắc hành trình	Áp định mức 24VDC	3	

Chương 3: Thiết kế phần cứng

	KW11	Dòng định mức 2A		
7	Băng tải	Chiều dài 60cm Chiều rộng 10cm	1	Mặt băng tải màu trắng
8	Động cơ quay băng tải ZGB37R-530	Trọng lượng 226g Tại 12V tốc độ 11.5rpm	1	

a. Thông số cánh tay robot

Các thành phần chính của Robot Delta được thiết kế với các thông số sau:

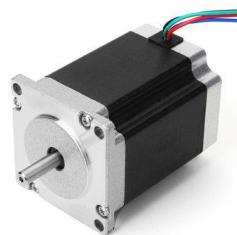
- Chiều dài bệ cố định hình tam giác phía trên: $f = 75\text{mm}$
- Khoảng cách từ trục động cơ đến cánh tay đòn: $r_f = 80\text{mm}$
- Độ dài cánh tay đòn (phần cánh tay dưới): $r_e = 235\text{mm}$
- Chiều dài cạnh bệ di động hình tam giác phía dưới chuyển động: $e = 20\text{mm}$

b. Động cơ bước và driver điều khiển động cơ bước

Robot Delta cần có tới ba động cơ để tạo chuyển động quay cho các cánh tay và các chuyển động này phải tuyệt đối chính xác theo góc quay đã được đặt trước. Chọn hai loại động cơ đảm bảo được yêu cầu thi công này là động cơ bước (step motor) và động cơ servo.

Cơ bản, cả hai loại động cơ này đều là động cơ điều khiển chuyển động vị trí chính xác. Ưu điểm của động cơ bước là giá thành rẻ, điều khiển góc quay chính xác, có khả năng cung cấp mô men xoắn lớn ở dải vận tốc trung bình và thấp; nhược điểm của động cơ này là nếu bị quá tải động cơ sẽ bị trượt bước gây sai lệch trong điều khiển, gây ra nhiều nhiễu và rung động hơn động cơ servo. Đối với động cơ servo, ưu điểm có nó là có tích hợp bộ encoder phản hồi tín hiệu về giúp chống trượt bước trong điều khiển nên độ chính xác gần như tuyệt đối, có thể hoạt động ở tốc độ cao hơn so với động cơ bước; nhược điểm của nó là giá thành cao hơn rất nhiều so với động cơ bước.

Với yêu cầu hệ thống là tải nhỏ, tốc độ phân loại không cần quá nhanh, chi phí đầu tư vừa phải nên động cơ bước là sự lựa chọn phù hợp có thể đáp ứng các yêu cầu trên của hệ thống.



Hình 3.4 Động cơ bước điều khiển cánh tay robot

Chọn động cơ để điều khiển cánh tay robot là động cơ bước mã hiệu: ZB17BBK-200 như **Hình 3.4** gồm các thông số:

- Điện áp pha: 2.75 VDC
- Dòng điện pha: 2.00 A
- Mô men xoắn: 0.25 Nm
- Góc bước: 1.8^0
- Mô men quán tính rotor J_o : 36.10^{-7} Kg.m 2

Để PLC phát xung điều khiển động cơ bước thì cần sử dụng thêm driver điều khiển động cơ bước TB6600 như minh họa ở **Hình 3.5**. Driver này sử dụng IC TB6600HQ/HG, dùng cho các loại động cơ bước 2 pha hoặc 4 dây có dòng tải là 4A/42VDC.



Hình 3.5 Driver điều khiển động cơ bước TB6600

- Thông số kỹ thuật:
 - Nguồn đầu vào: 9V – 42V
 - Dòng cấp tối đa: 4A
- Cài đặt và ghép nối:
 - DC + : Nối với nguồn điện từ 9 – 40VDC
 - DC - : Điện áp (-) âm của nguồn
 - A + và A - : Nối vào cặp cuộn dây của động cơ bước
 - B + và B - : Nối với cặp cuộn dây còn lại của động cơ
 - PUL + : Tín hiệu cấp xung điều khiển tốc độ cho M6600

- PUL - : Tín hiệu cáp xung điều khiển tốc độ cho M6600
- DIR + : Tín hiệu cáp xung đảo chiều cho M6600
- DIR - : Tín hiệu cáp xung đảo chiều (-) cho M6600
- ENA + và ENA - : khi cấp tín hiệu cho cặp này động cơ sẽ không có lực momen giữ và quay nữa.
- Có thể đấu tín hiệu dương (+) chung hoặc tín hiệu âm (-) chung

c. Đọc xung tốc độ cao rotary encoder 400

Do khi robot phát hiện vật, nó phải mất một khoảng thời gian nhất định để di chuyển từ vị trí bắt đầu cho đến vị trí vật đang nằm trên băng tải, mà băng tải thì luôn chuyển động với một vận tốc nhất định. Do đó cần tính được tốc độ băng tải kết hợp với thời gian di chuyển của cánh tay robot đến vị trí vật thì sẽ suy ra được vị trí lệch của vật sau khoảng thời gian đó và đây cũng là vị trí robot cần di chuyển đến chứ không phải vị trí ngay lúc phát hiện vật. Để tính được tốc độ băng tải này thì encoder là một lựa chọn phù hợp với giá thành rẻ và đáp ứng được yêu cầu trên của hệ thống.

Encoder được sử dụng trong đề tài có kí hiệu là encoder tương đối 400 xung như hình 3.8, nó có 2 pha AB/ 5-24V có kích thước nhỏ, trọng lượng nhẹ, cài đặt dễ dàng, lợi thế chi phí thấp.



Hình 3.6 Rotary encoder 400

Thông số kỹ thuật:

- Số xung/vòng: 400 xung/vòng
- Điện áp hoạt động: DC5-24V

Kết nối: dây xanh lục = Pha A, trắng = pha B, màu đỏ = VCC, đen = GND.

d. Công tắc hành trình

Khi hoạt động, Robot Delta cần xác định một vị trí để làm toạ độ Home, nghĩa là tại vị trí này góc quay của động cơ bước sẽ được đặt bằng 0° . Do đó công tắc hành trình trong

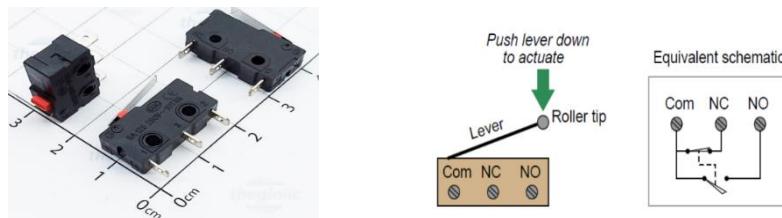
Chương 3: Thiết kế phần cứng

đè tài này được sử dụng với mục đích để khi khởi động hoặc reset robot, các cánh tay của nó thu về cho đến khi nào động công tắc hành trình thì sẽ dừng lại và đặt vị trí đó là Home.

Với nhiều ưu điểm như giá thành hợp lý với sinh viên, độ bền cao, có nhiều kích thước đa dạng thuận lợi cho nhu cầu khác nhau của đối tượng sử dụng nên công tắc hành trình được sử dụng khá phổ biến trong các dự án vừa và nhỏ như minh họa **Hình 3.7**.

Nguyên lý hoạt động của công tắc hành trình:

Công tắc hành trình dùng để đóng cắt mạch điện. Nó có công dụng tương đương với việc nhấn 1 nút nhấn bằng tay, chỉ khác là thay thế việc nhấn bằng tay bằng va chạm của các bộ phận cơ khí, từ đó tín hiệu va chạm cơ khí thành tín hiệu điện.



Hình 3.7 Công tắc hành trình KW11

e. Động cơ băng tải và bơm hút vật

Khi Robot Delta di chuyển tiến đến tọa độ vị trí của vật để gấp vật thì cần một lực hút để bám chặt lấy vật. Vì vậy đè tài được sử dụng máy bơm hút như **Hình 3.8** dưới đây là loại máy bơm hút chân không Micro 12V – 24V và nó tương đối yên tĩnh. Tuy nhiên có một chút nhược điểm của máy bơm hút là khối lượng hơi nặng và kích cỡ hơi cồng kềnh nên phải thiết kế thêm hộp động máy bơm và các dây nối.



Hình 3.8 Bơm hút chân không Micro 24V

Thông số của máy bơm hút:

- Điện áp định mức: DC24V
- Dòng định mức: 900 mA
- Lực hút tối đa: ~50KPa

Chương 3: Thiết kế phần cứng

- Lưu lượng: > 13,0 l/phút, tiếng ồn: <60 dB

Động cơ để điều khiển băng tải sử dụng động cơ giảm tốc ZGB37R-530 24V 23rpm-DCGT_ZGB37R-530 như **Hình 3.9**.



Hình 3.9 Động cơ giảm tốc ZGB37R-530 24V

Thông số kỹ thuật:

- Trọng lượng 226g
- Tại 12v dòng không tải 35mA tốc độ 11.5rpm
- Tại 24v dòng không tải 45mA tốc độ 23rpm

f. Băng tải

Băng tải PVC được sử dụng rộng rãi trong các dây chuyền lắp ráp trong các lĩnh vực chế biến thực phẩm, dược phẩm, điện tử,

Ưu điểm:

- + Độ bền kéo tốt, tuổi thọ dài, không bị dãn trong quá trình hoạt động.
- + Bề mặt dây PVC chống nước chịu nhiệt từ (-10 ° C – 80 ° C) có thể chịu được nhiệt độ 110° C trong một thời gian nhất định.
- + Khả năng chống thấm hơi ẩm, axit, dầu, khí, ánh sáng mặt trời.
- + Khả năng chống tĩnh điện, chống rách, bong tróc và tác động trọng lượng dây PVC nhẹ, linh hoạt ...

Để tài phân loại sản phẩm theo màu sắc (xanh dương, cam,...) do đó sử dụng băng tải dải băng chuyền màu trắng như **Hình 3.10** để dễ dàng nhận diện sản phẩm di chuyền trên băng tải mà không bị nhiễu màu sắc khi sử dụng ứng dụng xử lý ảnh.



Hình 3.10 Băng tải PVC

3.3.2. Lựa chọn bộ điều khiển

Hiện nay Robot Delta chủ yếu được điều khiển bằng PLC hoặc vi điều khiển với các ưu và nhược điểm của từng loại. Tùy vào mục đích sử dụng cũng như yêu cầu kinh tế thì có thể lựa chọn sao cho phù hợp.

Cả PLC và vi điều khiển đều có những điểm giống nhau cơ bản là có khả năng lập trình được để thực hiện một số tác vụ điều khiển các thiết bị trong thực tế dựa trên nền tảng sử dụng tín hiệu điện. Để điều khiển được máy móc hoạt động thì chúng phải có hệ thống các ngõ vào bao gồm cảm biến, nút nhấn, công tắc và đi kèm với đó là các thiết bị cơ cấu chấp hành kết nối với ngõ ra như relay, contactor, động cơ. Tuy nhiên, sự khác nhau nổi bật nhất giữa PLC và vi điều khiển chính là sự thiết kế cơ bản của hai thiết bị này hoàn toàn khác nhau nhằm mục đích phục vụ cho những mảng ứng dụng khác nhau.

Đối với PLC thuộc dạng thiết bị dùng trong công nghiệp với nhiều đặc điểm thiết kế để hạn chế nhiều từ nguồn và từ trường để có thể hoạt động tốt trong nhà xưởng, bên cạnh đó PLC còn được thiết kế theo dạng từng module đảm nhiệm các chức năng khác nhau để có thể dễ dàng thay thế khi bị hư hỏng.

Còn mạch vi xử lý, vi điều khiển chủ yếu chỉ bao gồm CPU nên chưa được xử lý liên quan tới phần chống nhiễu nên chỉ ứng dụng trong một số mô hình hoạt động trong gia dụng hoặc nghiên cứu giảng dạy.

Với yêu cầu vận hành trong môi trường công nghiệp với các điều kiện như: độ nhiễu cao, bụi bặm, cần thay thế dễ dàng khi bị sự cố thì PLC là một lựa chọn phù hợp có thể đáp ứng tất cả các yêu cầu trên. Bên cạnh đó với yêu cầu cần phải phát xung tốc độ cao điều khiển động cơ bước, đọc xung tốc độ cao từ encoder về PLC nên cần chọn PLC có loại ngõ ra transistor và ngõ vào có khả năng đọc xung tốc độ cao. Ngõ ra Transistor đóng ngắt bằng linh kiện bán dẫn nên độ bền cao, chịu được số lần đóng ngắt nhiều lần phù hợp để phát xung điều khiển động cơ bước (Step motor), động cơ Servo hay một ứng dụng gì đó cần xung.... Với các yêu cầu trên, lựa chọn PLC S7-1200 (1212 DC/DC/DC) như **Hình 3.11**.



Hình 3.11 PLC S7-1200 với CPU 1212 DC/DC/DC

Vì số lượng ngõ vào và ngõ ra của PLC không đủ để đáp ứng yêu cầu của hệ thống nên lựa chọn thêm một mô đun mở rộng SM-1222 RLY như **Hình 3.12**.

Ngõ ra RELAY đóng ngắt bằng tiếp điểm cơ khí, độ bền không cao nên thích hợp với nhiều loại tải có tần số đóng ngắt không nhiều, tiết kiệm chi phí. Chính vì vậy, ta nên chọn mô đun mở rộng ngõ ra Relay để điều khiển bằng tải và bơm hút chân không.



Hình 3.12 Mô đun mở rộng SM-1222 RLY

3.3.3. Các thiết bị linh kiện khác

a. Aptomat (CB LS 2P 40A)

Aptomat là tên thường gọi của thiết bị đóng cắt tự động (cầu dao tự động). Trong tiếng Anh thiết bị đóng cắt là Circuit Breaker (viết tắt là CB). Aptomat có chức năng bảo vệ quá tải và ngắn mạch trong hệ thống điện. Một số dòng Aptomat có thêm chức năng bảo vệ chống dòng rò được gọi là aptomat chống rò hay aptomat chống giật.



Hình 3.13 Aptomat dạng tép MCB của hãng LS

b. Nguồn tổ ong

Chương 3: Thiết kế phần cứng

Nguồn tổ ong dùng để biến đổi điện áp xoay chiều 220VAC sang điện áp một chiều 24VDC nhằm mục đích cung cấp nguồn điện cho các thiết bị như là: PLC, bơm hút chân không.



Hình 3.14 Nguồn tổ ong

c. Bộ chuyển đổi

Bộ chuyển đổi 220AC/12DC có nhiệm vụ chuyển điện áp 220VAC sang điện áp 12VDC cấp điện cho động cơ quay của băng tải.



Hình 3.15 Bộ chuyển đổi 220AC/12DC

d. Công tắc xoay

Công tắc xoay có nhiệm vụ cấp nguồn điện chính cho hệ thống để khởi động nguồn tủ điện lên.



Hình 3.16 Công tắc xoay

e. Nút nhấn Emergency

Emergency hay còn được gọi là nút nhấn dừng khẩn cấp. Khi có sự cố gì trên hệ thống khi đang hoạt động, chúng ta có thể sử dụng nút nhấn dừng khẩn cấp để dừng toàn bộ hệ thống một cách nhanh chóng để tránh những trường hợp không mong muốn xảy ra.



Hình 3.17 Nút dừng khẩn cấp

f. Đèn báo nguồn

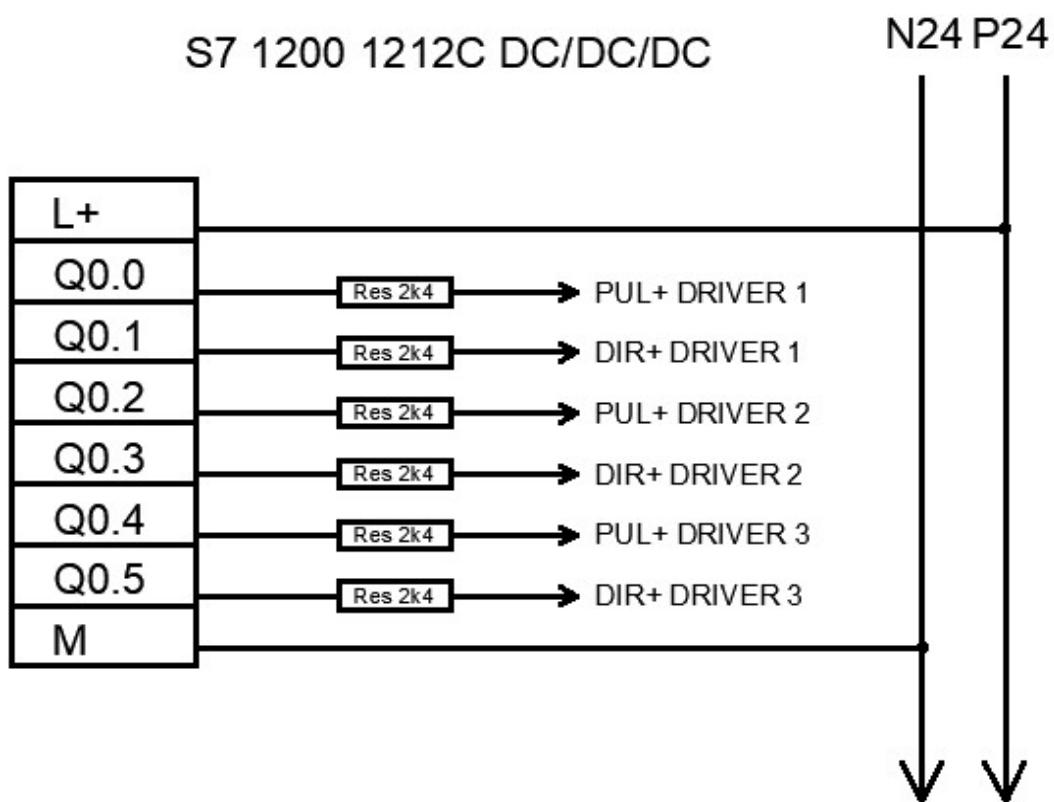
Khi chúng ta cấp nguồn điện cho hệ thống thì đèn báo nguồn sẽ sáng đèn để báo hiệu cho chúng ta biết hệ thống đã được cấp nguồn khởi động. Ngược lại khi tắt nguồn hệ thống thì đèn báo nguồn cũng sẽ tắt đi.



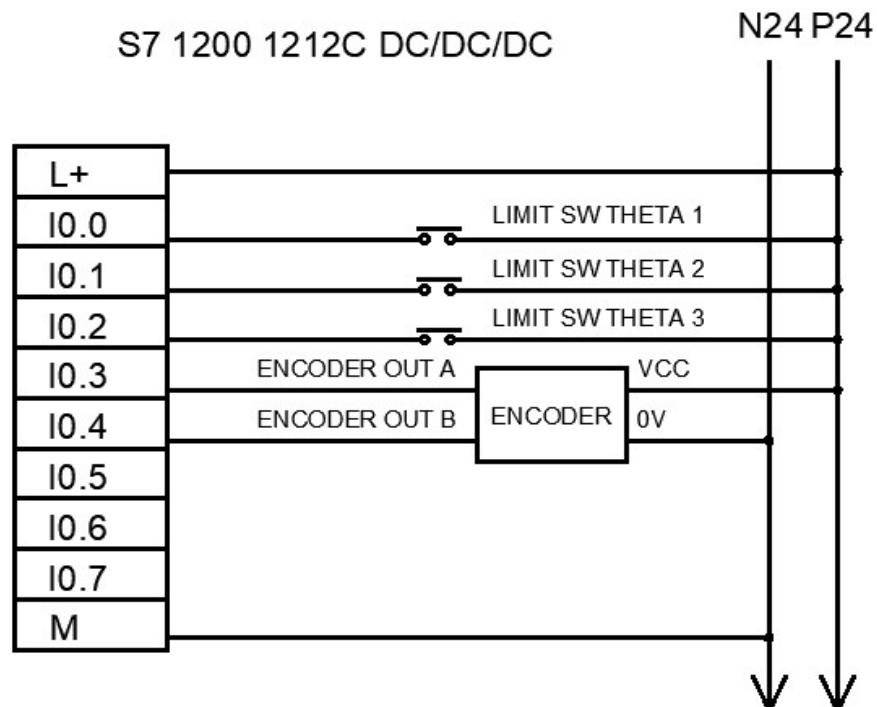
Hình 3.18 Đèn báo nguồn

3.4. Sơ đồ kết nối phần cứng

Sơ đồ kết nối phần cứng của PLC S7-1200 (CPU 1212C DC/DC/DC) được kết nối như **Hình 3.19** với 3 ngõ vào là 3 công tắc hành trình giúp đặt cánh tay robot ở trạng thái chạm vị trí công tắc hành trình. Encoder được nối với 2 ngõ vào để đọc tốc độ băng tải và tính toán tọa độ gấp vật. Ngõ ra của PLC có 6 ngõ ra đều có khả năng phát xung điều khiển 3 step motor.

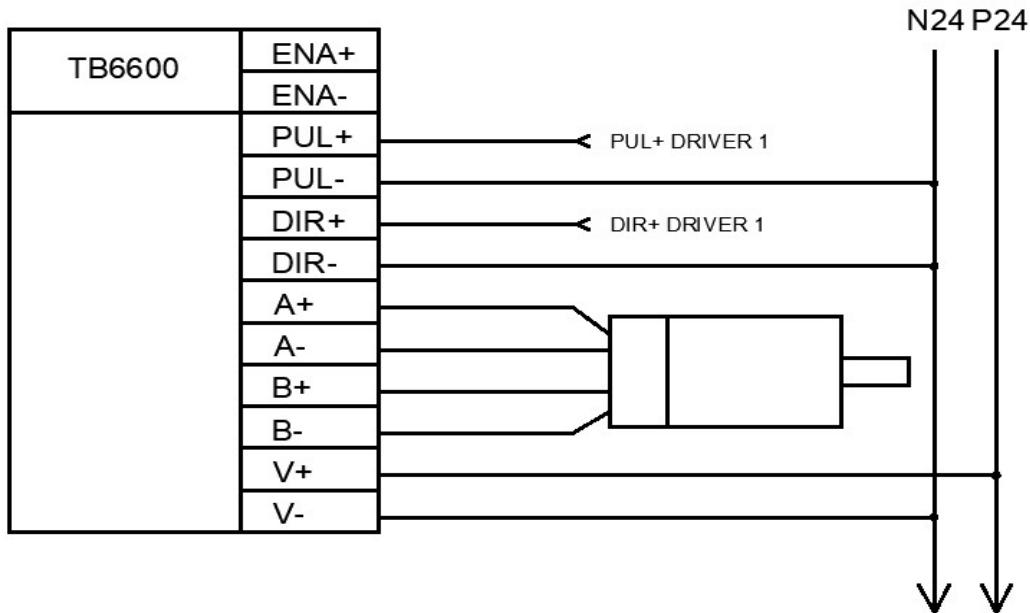


Hình 3.19 Sơ đồ kết nối ngõ ra PLC với các thiết bị



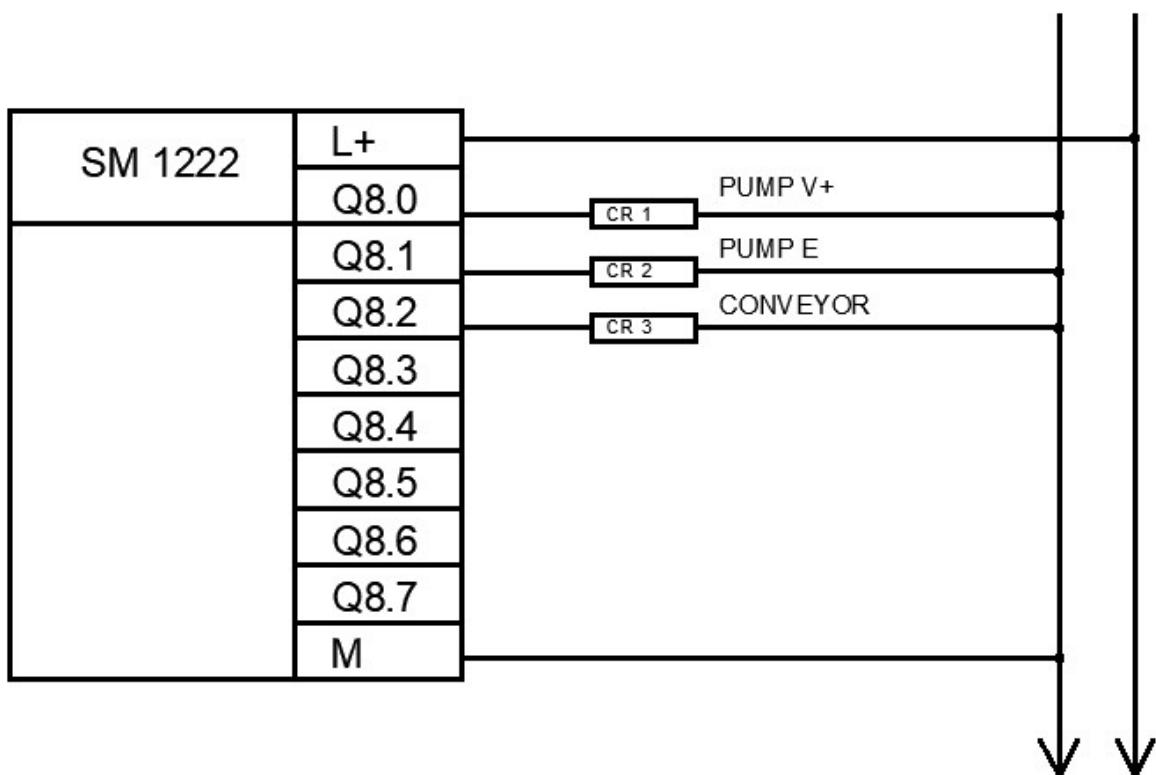
Hình 3.20 Sơ đồ kết nối ngõ vào PLC với các thiết bị

Sơ đồ nối dây giữa động cơ bước và driver TB6600 được trình bày như **Hình 3.21**. Ở đây các chân A+, A-, B+, B- sẽ được nối với bốn đầu dây ngõ ra của các pha động cơ bước, chân PUL+ và DIR+ được nối chung lại với nhau và nối xuống GND, còn chân ENA+ và ENA- không nối để luôn cho phép driver hoạt động khi được cấp điện.



Hình 3.21 Sơ đồ kết nối driver với step motor

Vì PLC S7-1200 (CPU 1212C DC/DC/DC) chỉ có 6 ngõ ra để phát xung cho 3 step motor nên không đủ ngõ ra để điều khiển các thiết bị nên cần thêm một mô đun mở rộng SM 1222 RLY để điều khiển bơm hút với 2 ngõ ra của mô đun mở rộng và 1 ngõ ra để điều khiển băng tải như **Hình 3.22**.



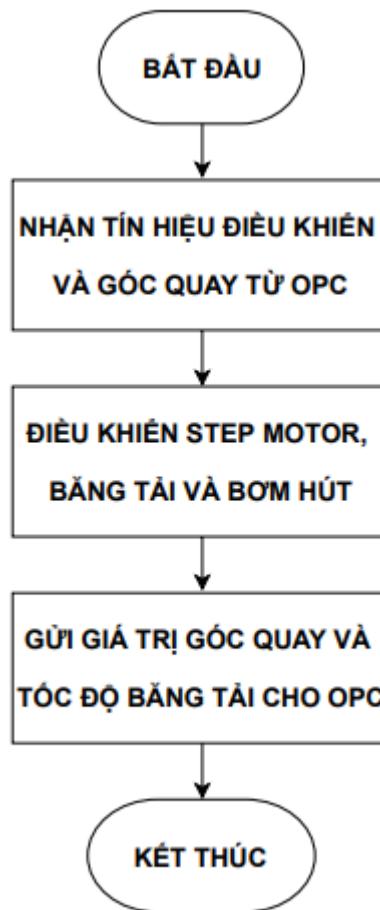
Hình 3.22 Sơ đồ kết nối của mô đun mở rộng

CHƯƠNG 4. THUẬT TOÁN ĐIỀU KHIỂN

Chương này trình bày thuật toán điều khiển bao gồm: thuật toán PLC, thuật toán xử lý ảnh và thuật toán điều khiển Robot Delta gắp vật.

4.1. Thuật toán điều khiển PLC

Thuật toán điều khiển PLC được minh họa như **Hình 4.1**. Khi bật nguồn cho hệ thống, PLC sẽ nhận tín hiệu điều khiển từ OPC các tín hiệu điều khiển và góc quay theta. Sau khi nhận dữ liệu từ OPC thì PLC sẽ cấp tín hiệu cho driver điều khiển step motor giúp cánh tay robot hoạt động. Đồng thời PLC cũng sẽ gửi giá trị góc quay theta của 3 step motor và tốc độ băng tải về OPC để cho C# nhận dữ liệu để tính toán và hiển thị giá trị góc quay thực tế.



Hình 4.1 Lưu đồ điều khiển của PLC

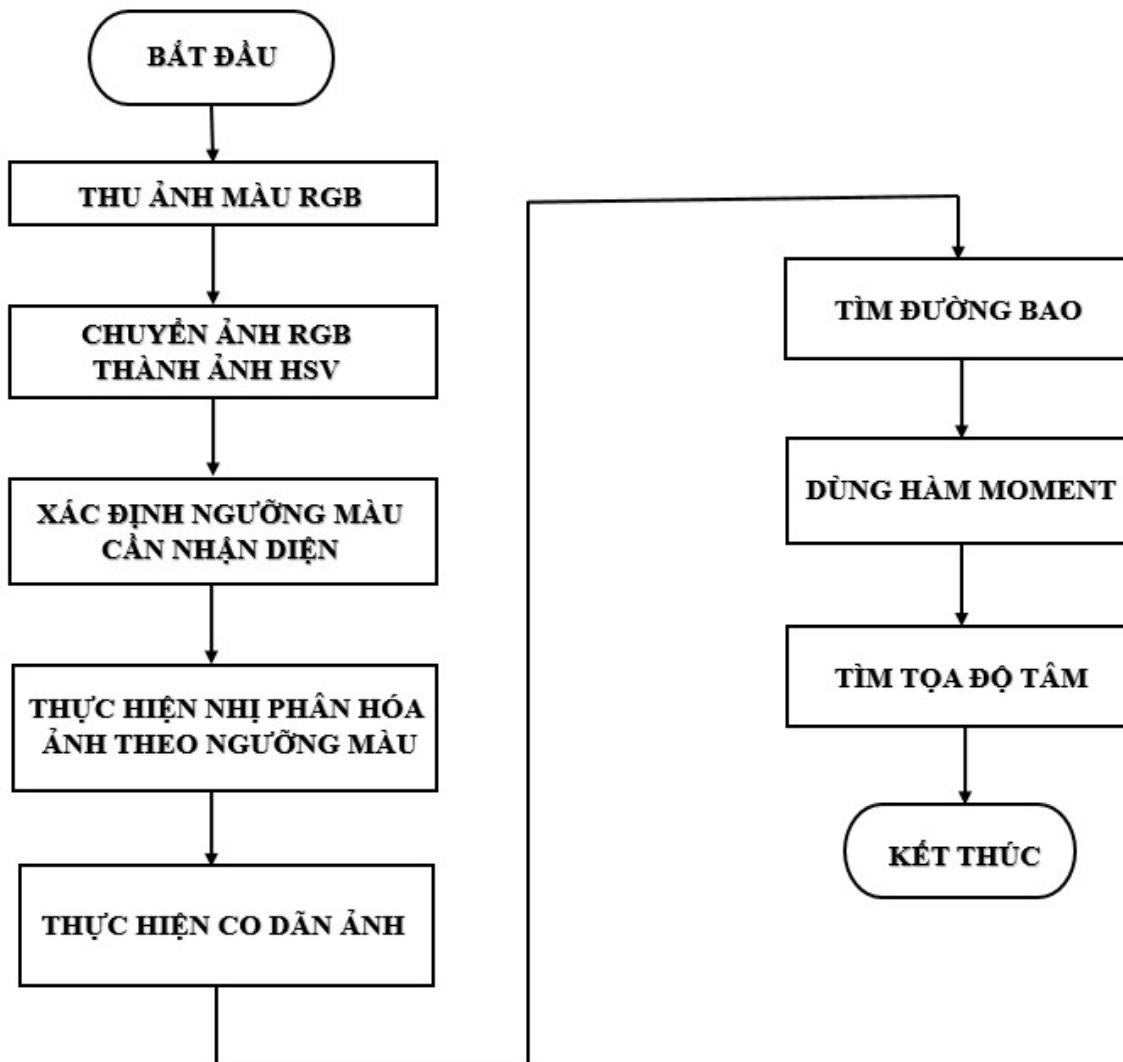
4.2. Thuật toán xử lý lý ảnh

Thuật toán xử lý lý ảnh phân loại sản phẩm theo màu sắc được trình bày như **Hình 4.2**. Ảnh gốc được thu vào từ camera với kích thước khung hình 480x640 sẽ được chuyển đổi

Chương 4: Thuật toán điều khiển

từ hệ ảnh màu RGB sang hệ màu HSV để giúp quá trình xử lý màu sắc thuận tiện hơn. Sau đó ảnh màu HSV sẽ được nhị phân hóa theo ngưỡng tương ứng với màu sắc của sản phẩm cần phân loại gồm 2 màu: cam và xanh dương. Sau bước xử lý ảnh này thì hình ảnh sản phẩm sẽ trở thành ảnh nhị phân, phần sản phẩm nằm trong ngưỡng phát hiện sẽ chuyển thành màu trắng, các phần không thuộc ngưỡng phát hiện sẽ chuyển thành màu đen.

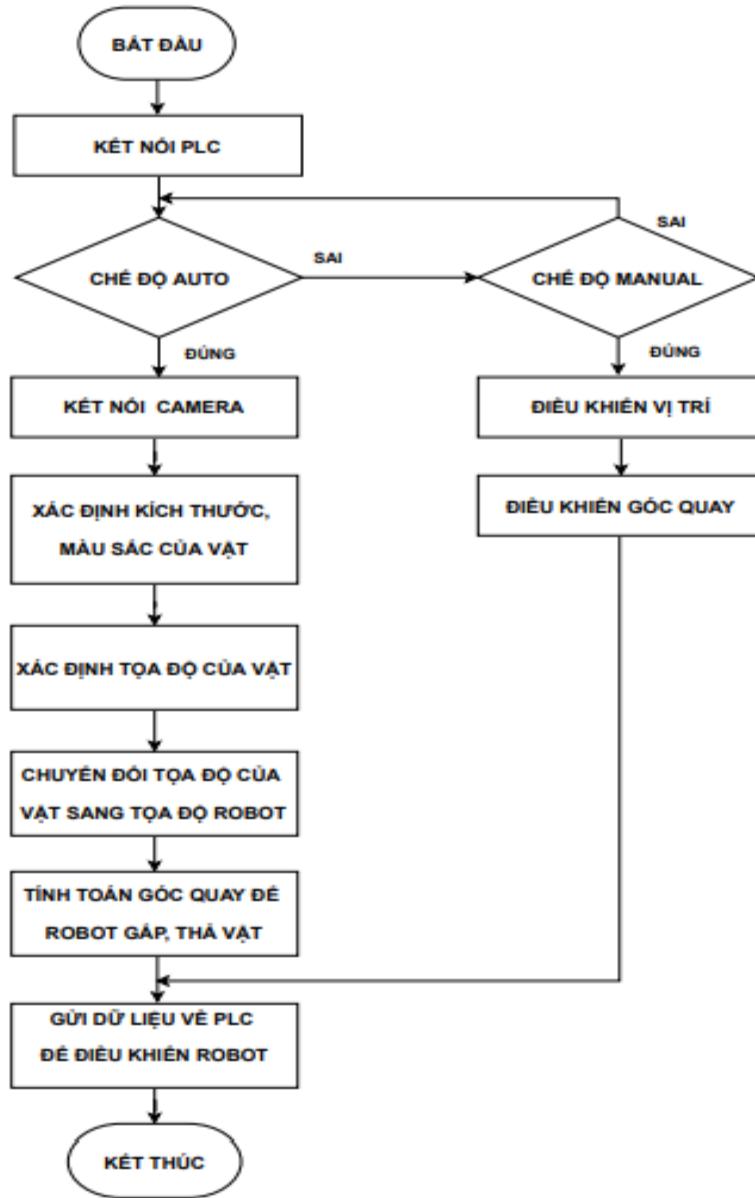
Tiếp đó ảnh nhị phân của sản phẩm được nhận diện sẽ trải qua 2 lần co giãn ảnh giúp loại bỏ các đối tượng nhiễu, nối liền các phần gãy khúc và đứt đoạn, tiến hành tìm đường bao trên ảnh này. Sau khi tìm được đường bao thì thuật toán xấp xỉ và hàm moment được áp dụng để tìm được chính xác tọa độ của vật trên băng tải.



Hình 4.2 Lưu đồ xử lý ảnh

4.3. Thuật toán điều khiển robot Delta gấp vật

Thuật toán điều khiển Robot Delta gấp vật được trình bày như **Hình 4.3**. Ban đầu khi nhấn nút Start ở trang kết nối sẽ giúp liên kết Application C# với PLC thông qua OPC KEPserver.



Hình 4.3 Lưu đồ điều khiển robot Delta

Chế độ điều khiển bằng tay (**MANUAL**) sẽ cho phép tùy chỉnh các góc theta của từng động cơ bước, điều khiển robot đến vị trí tọa độ được chỉ định và hiển thị các giá trị góc quay thực tế được gửi về từ PLC.

Chương 4: Thuật toán điều khiển

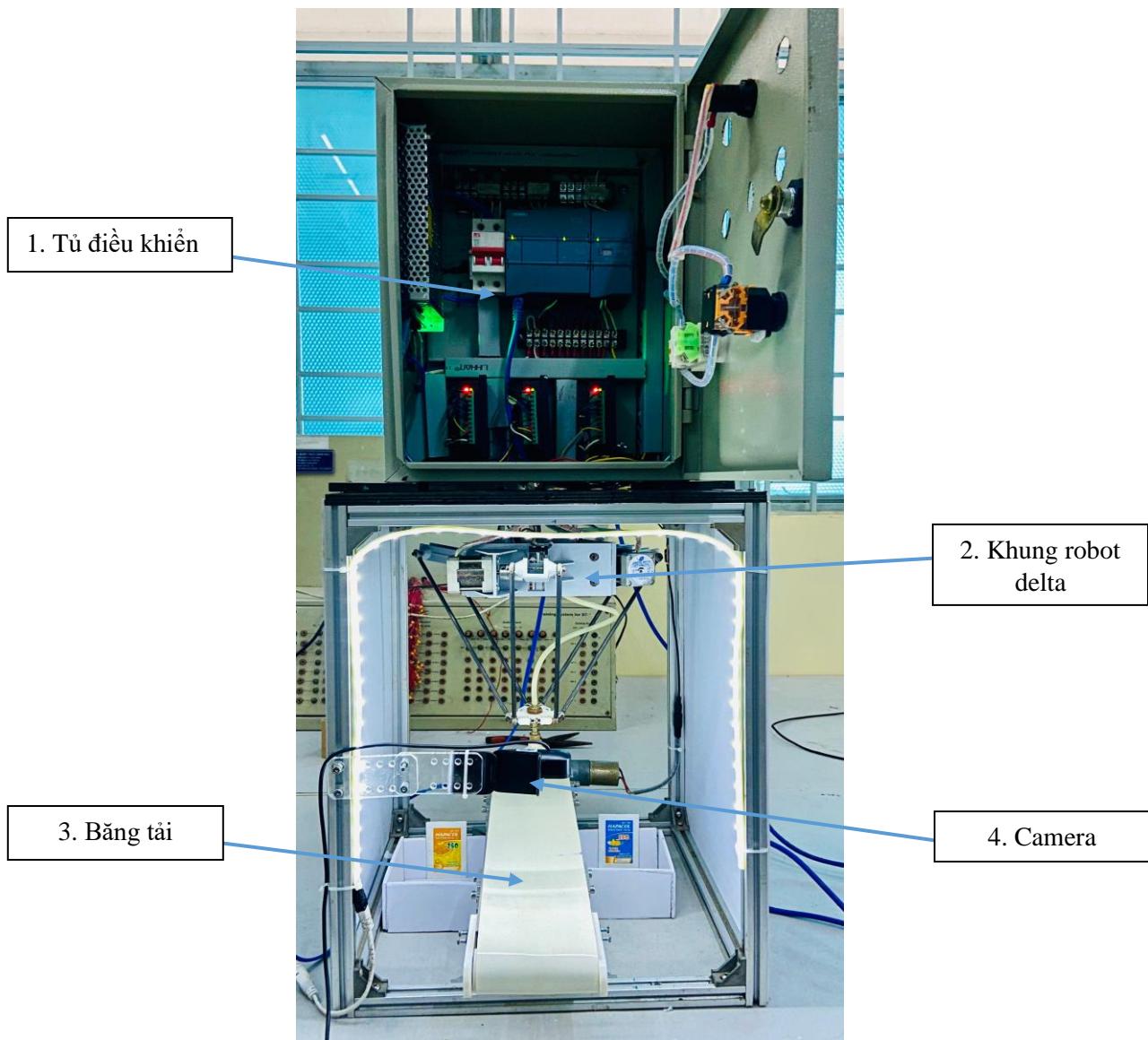
Chế độ tự động (**AUTO**) sẽ cho phép robot chạy tự động theo chế độ phân loại sản phẩm khi phát hiện có sản phẩm. Sau khi xác định được tọa độ của vật, chuyển đổi tọa độ này (tọa độ vật trong hệ camera) sang hệ tọa độ robot từ đó tính toán các giá trị góc quay và gửi dữ liệu về PLC điều khiển robot Delta gấp và thả sản phẩm vào vị trí định trước.

CHƯƠNG 5: KẾT QUẢ

Chương này trình bày về những kết quả đạt được về phần cứng và phần mềm điều khiển, giám sát của đề tài.

5.1. Phần cứng

Hoàn tất việc cài tiến mô hình Robot Delta như **Hình 5.1** bên dưới, với việc lắp thêm tấm chắn 2 bên khung robot để hạn chế được ánh sáng không mong muốn ở bên ngoài đánh vô và lắp thêm đèn trợ sáng giúp cải thiện ánh sáng khi ở trong môi trường thiếu sáng cũng như chỉ số hoàn màu của vật, từ đó giúp quá trình giám sát và phân loại sản phẩm được diễn ra một cách trọn tru và tỉ lệ chính xác khi phân loại sản phẩm cao hơn.



Hình 5.1 Robot Delta hoàn chỉnh

5.2. Phần mềm

5.2.1. Phần mềm điều khiển và giám sát robot Delta

Với yêu cầu của đề tài, nhóm chúng em đã thiết kế giao diện để giám sát và điều khiển hoạt động của Robot Delta thông qua việc chọn các chế độ hoạt động của hệ thống. Giao diện cho phép điều khiển đồng thời giám sát các thông số về toạ độ, góc quay, số lượng sản phẩm và hoạt động của robot thông qua camera, từ đó lập trình Winform được sử dụng để thiết kế tạo giao diện thỏa các yêu cầu trên.

Giao diện giám sát và điều khiển Robot Delta được thiết kế gồm bốn trang chính để thuận tiện vận hành cũng như giám sát robot một cách chặt chẽ gồm: trang connection, trang manual, trang auto, trang setting với các chức năng hoạt động trong từng trang được liệt kê cụ thể như sau.

a. Trang Connection



Hình 5.2 Giao diện đăng nhập hệ thống

Chương 5: Kết quả

Trước khi kết nối với hệ thống, chúng ta phải trải qua công việc cực kỳ quan trọng, đó là đăng nhập vào hệ thống. Nếu không thực hiện việc đăng nhập chúng ta không thể kết nối được với hệ thống. Ở mục **Username**: nhập vào **Robot** và mục **Password**: nhập vào **123** giao diện sẽ chuyển sang như **Hình 5.3**. Nếu nhập sai **Username** hay **Password** sẽ có báo lỗi ở phần nội dung đó.

Để kết nối được với hệ thống, chúng ta phải chọn vào nút CONNECT để có thể kết nối vào hệ thống và bắt đầu quy trình điều khiển và giám sát.



Hình 5.3 Giao diện trang Connection

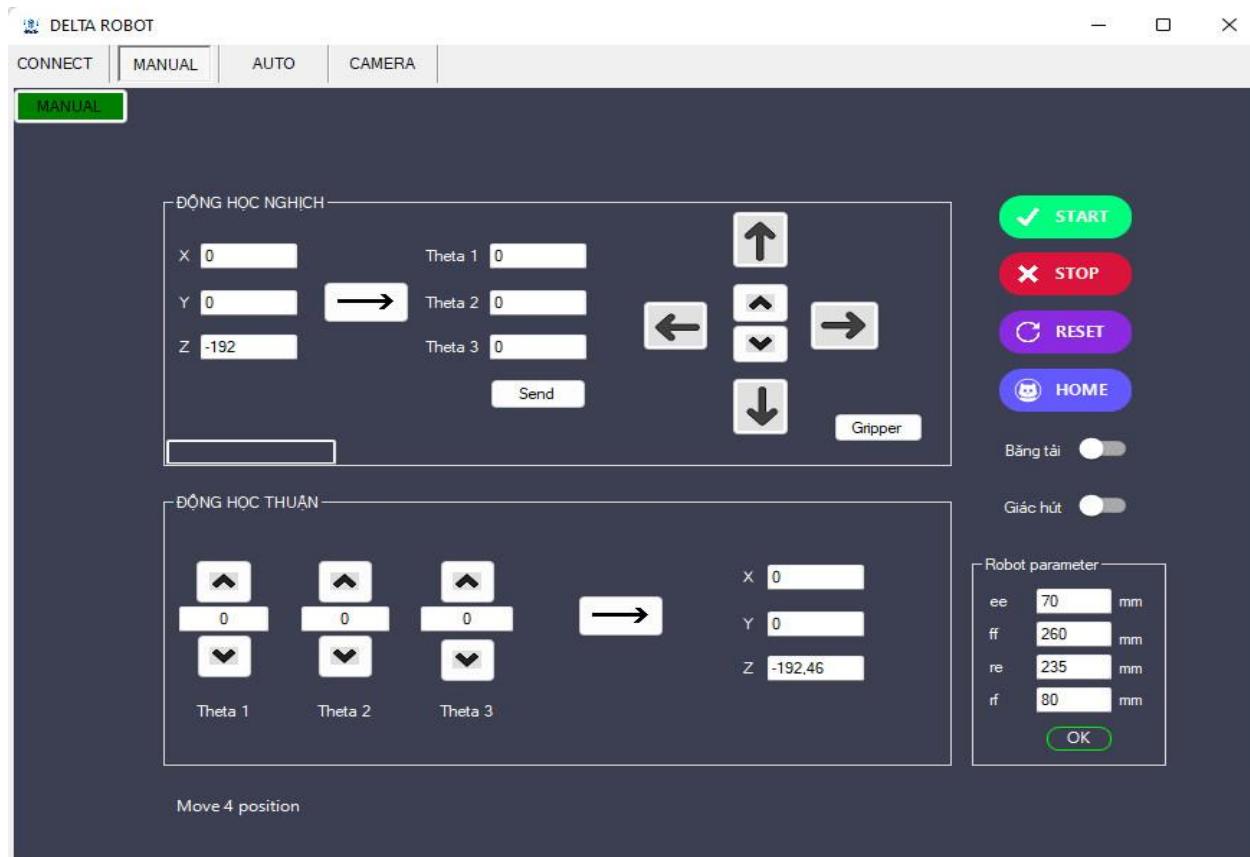
Trang Connection được thiết kế như **Hình 5.3**, giúp kết nối C# và PLC với nhau và cấp điện cho toàn bộ hệ thống hoạt động. Các phím chức năng cụ thể được liệt kê ở **Bảng 5.1** bên dưới.

Chương 5: Kết quả

Bảng 5.1 Phím chức năng trang Connection

STT	Tên nút	Chức năng
1	PLC S7 1200	Chọn loại PLC kết nối
2	IP 192.168.0.4	Chọn địa chỉ IP PLC
3	CONNECT	Kết nối C# và PLC
4	DISCONNECT	Huỷ kết nối C# và PLC
5	START	Cấp nguồn cho hệ thống hoạt động
6	STOP	Ngừng hệ thống

b. Trang Manual



Hình 5.4 Giao diện trang Manual

Chương 5: Kết quả

Trang Manual được thiết kế như **Hình 5.4**, giúp điều khiển robot di chuyển bằng các nút nhấn trên giao diện và đồng thời hiển thị các giá trị tọa độ, góc quay của robot đọc về từ PLC. Các phím chức năng cụ thể được liệt kê ở **Bảng 5.2** bên dưới.

Bảng 5.2 Phím chức năng trang Manual

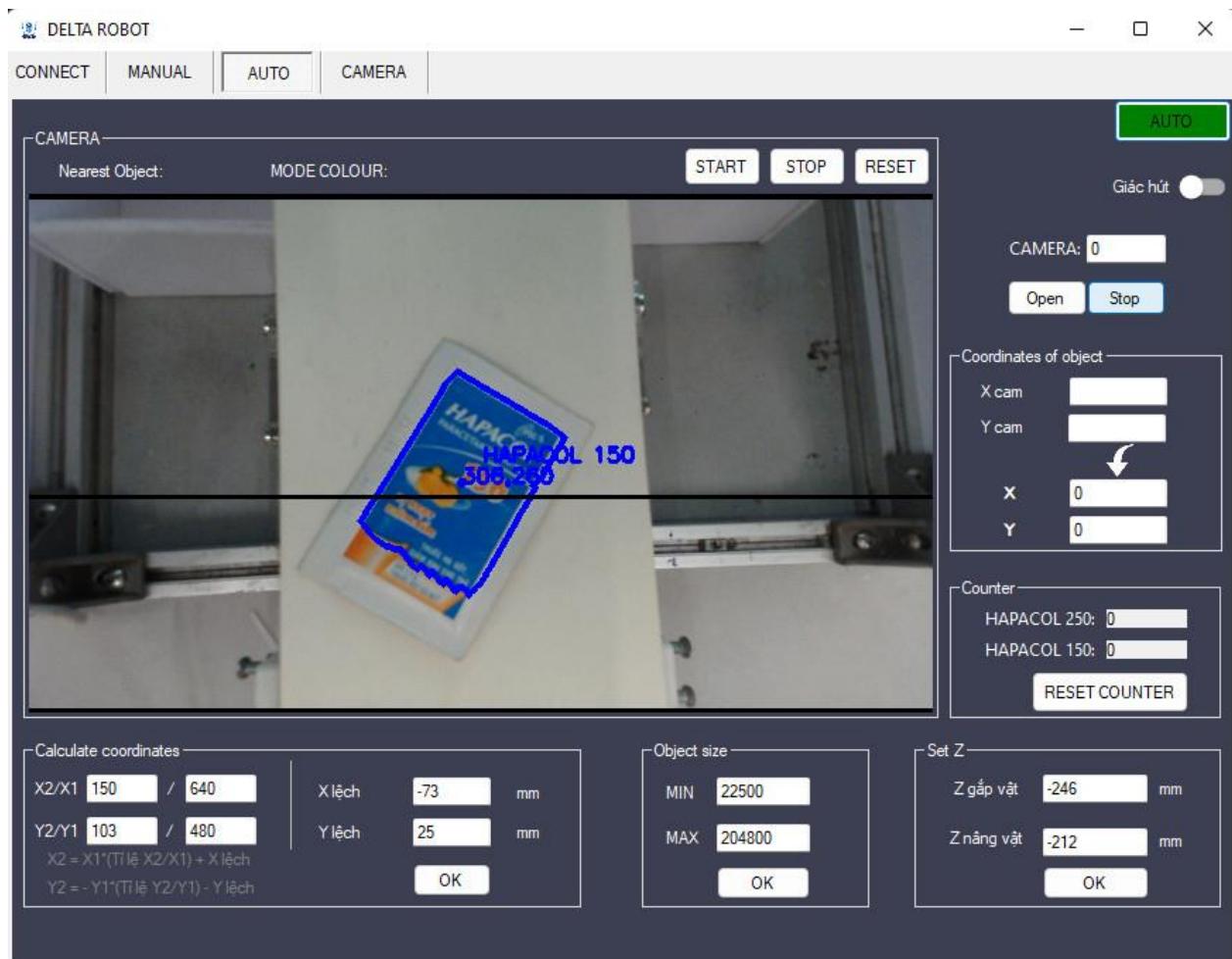
STT	Tên nút	Chức năng
1		Cấp nguồn cho hệ thống hoạt động
2		Ngừng hệ thống
3		Khởi tạo cánh tay Robot về vị trí công tắc hành trình
4	Giác hút	Điều khiển trạng thái giác hút vật
5	Băng tải	Điều khiển băng tải chạy/dừng
6		Trạng thái ban đầu
7		Gửi giá trị tọa độ của vật cho robot
8		Tọa độ của vật
9		Góc quay của vật
10		Dịch từ tọa độ sang góc quay của vật
11		Di chuyển tới, lui, trái, phải

Chương 5: Kết quả

12		Di chuyển lên, xuống
13		Tăng, giảm 3 góc theta cho 3 động cơ bước

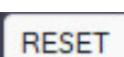
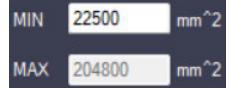
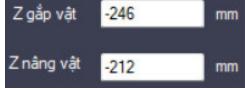
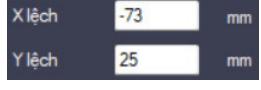
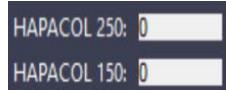
c. Trang Auto

Trang Auto được thiết như **Hình 5.5**, điều khiển và giám sát robot gấp và thả vật vào vị trí xác định từ trước, đồng thời hiển thị màn hình của camera để dễ dàng giám sát tiến trình nhận diện sản phẩm của hệ thống. Các phím chức năng cụ thể được liệt kê ở **Bảng 5.3** bên dưới.



Hình 5.5 Giao diện trang Auto

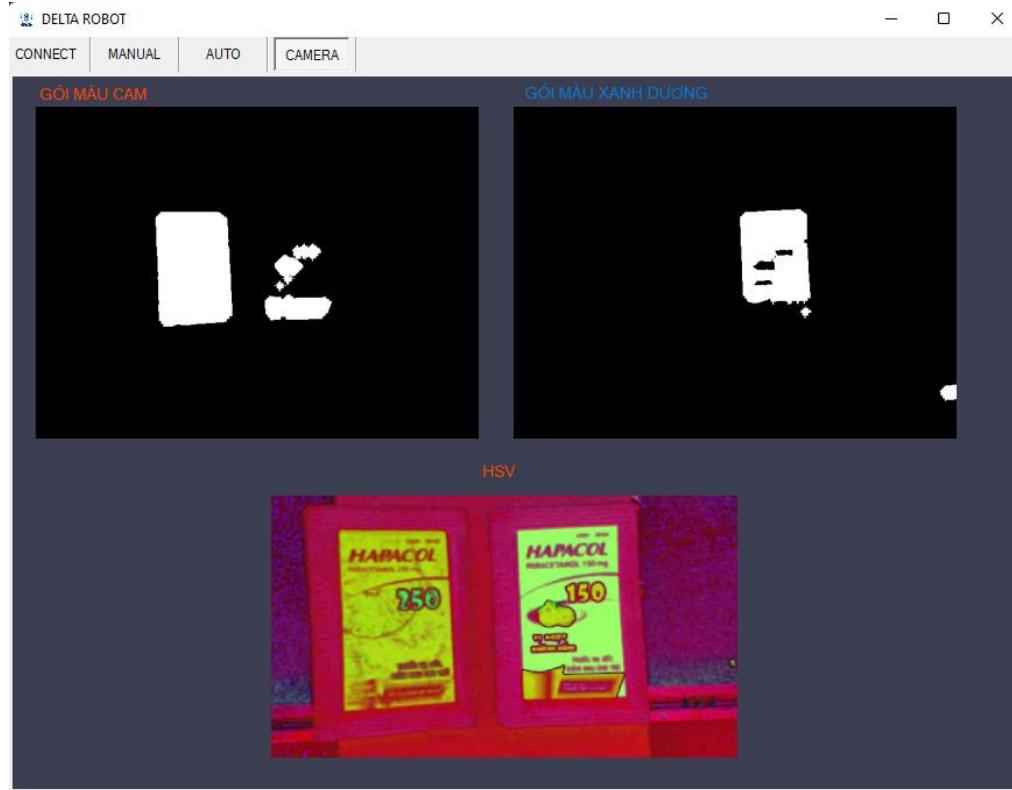
Bảng 5.3 Phím chức năng trang Auto

STT	Tên nút	Chức năng
1		Cáp nguồn cho hệ thống hoạt động
2		Ngừng hệ thống
3		Khởi tạo cánh tay Robot về vị trí công tắc hành trình
4		Lựa chọn chế độ điều khiển tự động
5		Lựa chọn kết nối camera
6		Mở camera
7		Ngừng thu hình từ camera
8		Xóa bộ nhớ sản phẩm
9		Giới hạn kích thước của vật
10		Chiều cao gấp/nâng vật
11		Độ lệch tâm của hệ robot và hệ camera theo X/Y
12		Tỉ lệ quy đổi từ Pixel sang mm của tọa độ X/Y
13		Bộ nhớ số lượng sản phẩm

d. Trang camera

Trang camera được thiết kế như **Hình 5.6** bên dưới, giúp chúng ta dễ dàng quan sát thuật toán xử lý ảnh khi chỉnh ảnh đó sang HSV và tìm ngưỡng màu của sản phẩm. Những

màu nằm trong ngưỡng sản phẩm thì chuyển thành màu trắng, còn ngược lại những màu không nằm trong ngưỡng thì chuyển thành màu đen. Từ đó giúp robot dễ dàng nhận biết sản phẩm một cách dễ dàng hơn.

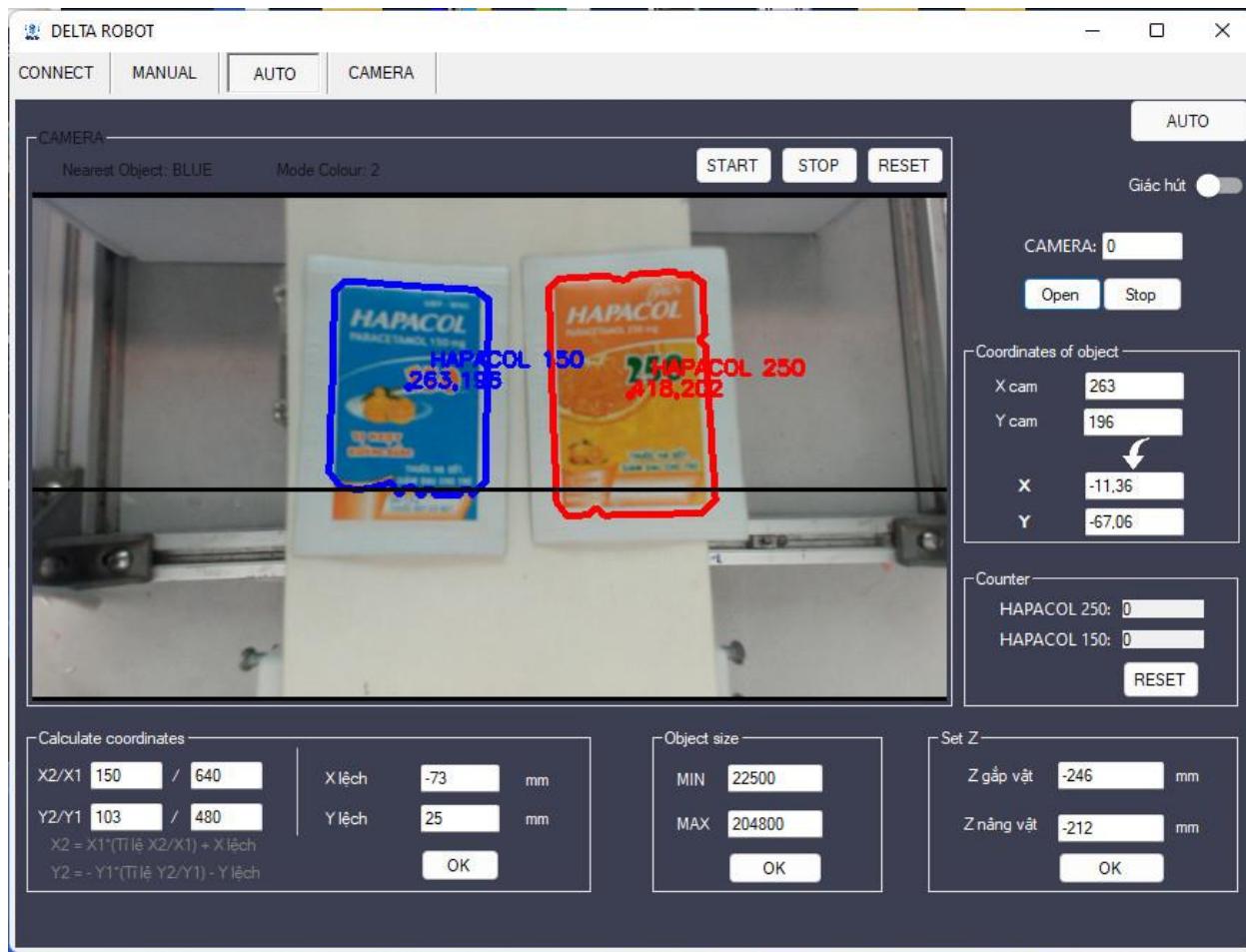


Hình 5.6 Giao diện trang camera

5.2.2. Robot hoạt động ở chế độ auto

Ở chế độ **auto**, sản phẩm qua băng tải đi vào hệ thống, ảnh của sản phẩm lấy từ camera sau khi được ứng dụng các thuật toán xử lý ảnh sẽ tiến hành vẽ đường bao cho vật, đồng thời xác định chính xác tọa độ tâm của vật.

Chương 5: Kết quả



Hình 5.7 Ứng dụng xử lý ảnh tìm được màu sắc, vẽ đường bao và tâm của vật

Sau khi tìm được tâm của vật, sẽ tiến hành tính toán độ lệch của sản phẩm khi dịch chuyển trên băng tải để trùng khớp với chuyển động gấp vật của cánh tay Robot tránh trường hợp gấp vật không chính xác.



Hình 5.8 Robot gấp sản phẩm màu cam trên băng tải và thả vật tại vị trí 1



Hình 5.9 Robot gấp sản phẩm màu xanh trên băng tải và thả vật tại vị trí 2

Robot tiến hành thực hiện gấp và thả vật vào vị trí đã xác định trước là VT1 và VT2, số liệu số liệu được ghi lại vào bảng **Bảng 5.4** dưới đây.

Bảng 5.4 Số liệu thực nghiệm khi gấp, thả sản phẩm vào vị trí

Sản phẩm	GÓI CAM	GÓI XANH	Tổng
	50	50	100
Thực hiện gấp	49/50	48/50	97/100
Tỉ lệ chính xác (%)	98%	96%	97%

5.2.3. Robot hoạt động ở chế độ manual

Ở chế độ **manual**, khi nhấn các phím mũi tên , thì robot dịch chuyển tới lui, qua trái, sang phải, hay lên, xuống chính xác theo chức năng của từng phím nhấn đã được lập trình sẵn trong chương trình, đặc biệt nếu nhấn giữ các phím trên thì robot sẽ dịch chuyển liên tục đến khi nào buông phím nhấn ra thì mới dừng lại.

Khi nhập các giá trị tọa độ của vật vào các ô sau đó nhấn nút để dịch tọa độ sang góc quay của 3 động cơ. Nếu tọa độ vị trí thỏa mãn thì màn hình sẽ hiện 3 giá trị góc quay, tiếp đến ta nhấn robot di chuyển đến đúng vị trí đó trong không gian. Nếu nhấn nút mà 3 góc theta đều ra ký tự “Ø” có nghĩa là không tìm được vị trí

Robot có thể di chuyển tới. Khi có bất kì lỗi nào xảy ra làm góc quay sai lệch, robot bị mất điểm **Home** thì có thể set lại vị trí **Home** bằng cách nhấn vào nút  thì khi đó các cánh tay sẽ thu lại đến khi chạm công tắc hành trình và nhấn  set lại điểm Home cho hệ thống.

Ngoài ra, chúng ta cũng có điều khiển Robot bằng cách điều chỉnh góc quay của 3 động cơ step , mỗi động cơ step có 2 nút tăng và giảm góc quay, mỗi lần tăng giảm động cơ góc quay sẽ thay đổi 2° . Chúng ta có thể nhấn liên tục, hệ thống vẫn hoạt động trơn tru liên tục mà không bị lỗi.

5.3. Nhận xét kết quả đạt được

Từ kết quả gấp và thả sản phẩm cho thấy nhóm chúng em đã giải quyết được các mục tiêu đã đề ra ban đầu là gấp sản phẩm trên băng chuyền và thả sản phẩm vào vị trí đã xác định. Kết quả gấp chính xác gần 97% và thả vào vị trí chính xác 97%. Bên cạnh đó vẫn còn sai số, đó là vị trí gấp không chính xác trọng tâm của vật 100% bởi nhiều yếu tố khác nhau. Nếu đặt các vật quá sát nhau, tốc độ đáp ứng của robot không kịp nhanh dẫn đến việc gấp các vật không được chính xác tại tâm của vật. Mặc khác, do tầm nhìn camera có giới hạn và cánh tay robot Delta chỉ hoạt động trong khoảng không gian cố định nên một số sản phẩm nằm ngoài tầm hoạt động của robot sẽ bị bỏ qua.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

Đề tài “**Điều khiển và giám sát hệ thống phân loại gói sản phẩm dùng Robot Delta**” thực hiện ở tốc độ cao nhưng vẫn đảm bảo sự chính xác và ổn định, phù hợp với công việc gấp và thả sản phẩm vào vị trí mong muốn một cách tự động.

Robot Delta có thể hoạt động ở 2 chế độ bằng tay (**MANUAL**) và tự động (**AUTO**). Ở chế độ **AUTO** có thể gấp sản phẩm trên băng chuyền và thả vào các vị trí đã xác định từ trước một cách chính xác. Đối với chế độ **MANUAL**, chúng ta có thể điều khiển vị trí của cánh tay Robot bằng cách thay đổi các vị trí tới, lui, sang trái, qua phải, lên, xuống hay thay đổi bằng cách thay đổi góc quay của các động cơ step, hệ thống hoạt động một cách mượt mà mà không có lỗi. Với những tính năng này thì đề tài đã thỏa mãn những mục tiêu ban đầu đã đề ra.

Khó khăn chưa được giải quyết ở sự thay đổi môi trường tác động xung quanh có thể dẫn tới sự thay đổi ánh sáng làm ảnh hưởng một phần nhỏ không đáng kể đến độ chính xác của tính toán trong chương trình xử lý ảnh.

6.2. Hướng phát triển

- Hệ thống có thể thay động cơ step motor thành động cơ servo để cánh tay hoạt động chính xác, ổn định và mượt hơn.
- Nâng cấp camera với độ phân giải cao và nét hơn để robot nhận diện được nhanh với độ chính xác cao hơn.
- Cải thiện thuật toán để robot di chuyển chính xác hơn.
- Ứng dụng của IoT giám sát điều khiển hệ thống từ xa thông qua điện thoại hoặc máy tính có kết nối Ethernet.
- Thiết kế thêm cơ cấu để có thể xoay chuyển hướng sau khi gấp vật.
- Có thể in cánh tay robot bằng CNC để cánh tay có thêm được độ bền và tăng thêm tính thẩm mỹ hơn.

TÀI LIỆU THAM KHẢO

Tiếng việt

- [1] Nguyễn Thanh Hải, *Giáo trình xử lý ảnh*, Đại học sư phạm kỹ thuật TP.HCM, 2014.

Tiếng anh

- [2] L.M. Zhang, *Integrated optimal design of delta robot using dynamic performance indices*, Tianjin, Tianjin University, 2011.
- [3] I.A. Bonev, *Deltas parallel robot – The story of success*, <http://www.Parallelmic.org/Reviews/Reviews002.html>, 2014.
- [4] Mzavatsky, Delta robot kinematics, 2009.
- [5] Bradski, Gary, and Adrian Kaehler, *Learning EmguCV: Computer vision with the EmguCV library*, O'Reilly Media, Inc., 2008.
- [6] Thomas B. Moeslund, *Introduction to Video and Image Processing*, Undergraduate Topics in Computer Science, Springer-Verlag London, 2012.
- [7] Ayan Bag, *Ramer–Douglas–Peucker Algorithm*, <https://ayanbag.com/ramer-douglas-peucker-algorithm>, 2020.
- [8] Các tác giả, *Image moment*, https://en.wikipedia.org/wiki/Image_moment, 2021.
- [9] Kang & Atul, *Image moments EmguCV python*, <https://theailearner.com/tag/image-moments-opencv-python/>, 2020.

PHỤ LỤC

Code C#:

- Xử lý ảnh:

#region XỬ LÝ ẢNH...

```
private void btn_numbercamera_Click(object sender, EventArgs e)
{
    cam = Convert.ToInt16(txb_cameranumber.Text);
}

private void btn_connectcamera_Click(object sender, EventArgs e)
{
    capture = new VideoCapture(cam);
    capture.SetCaptureProperty(Emgu.CV.CvEnum.CapProp.FrameWidth, 640);
    capture.SetCaptureProperty(Emgu.CV.CvEnum.CapProp.FrameHeight, 480);
    Mat m = new Mat();
    capture.Read(m);
    lbTrangThai.Text = "CONNECTED";
}

private async void btn_opencamera_Click(object sender, EventArgs e)
{
    if (capture == null){return; }
    try
    {
        while (!Pause)
        {
            Mat m = new Mat();
            capture.Read(m);
            lbTrangThai.Text = "CAMERA ON";
            if (!m.IsEmpty)          // khi có khung hình
            {
                contourss(m.ToImage<Bgr, byte>());
                SAPXEP();
            }
        }
    }
}
```

```
        double fps = capture.GetCaptureProperty(Emgu.CV.CvEnum.CapProp.Fps);
        await Task.Delay(500 / Convert.ToInt32(fps));
    }
    else {break;}
}
}
catch (Exception ex){MessageBox.Show(ex.Message);}
}

private void contourss(Image<Bgr, byte> img)
{
    y2[0] = y4[0] = y6[0] = 0;
    //chuyen sang HSV
    Image<Hsv, byte> HSV = new Image<Hsv, byte>(img.Width, img.Height);
    CvInvoke.CvtColor(img, HSV, Emgu.CV.CvEnum.ColorConversion.Bgr2Hsv);
    // chuyen img thuộc không gian màu bgr sang HSV
    Hsv Limitduoi = new Hsv(11, 55, 80);
    Hsv Limittren = new Hsv(60, 255, 255);
    Image<Gray, byte> temp = HSV.InRange(Limitduoi, Limittren);
    Mat oMat1 =
    CvInvoke.GetStructuringElement(Emgu.CV.CvEnum.ElementShape.Rectangle, new
    System.Drawing.Size(5, 5), new System.Drawing.Point(-1, -1));
    CvInvoke.MorphologyEx(temp, temp, Emgu.CV.CvEnum.MorphOp.Close,
    oMat1, new System.Drawing.Point(-1, -1), 4, Emgu.CV.CvEnum.BorderType.Default,
    new MCvScalar(255, 0, 0, 255));
    // mở rộng và ăn mòn
    Mat oMat2 =
    CvInvoke.GetStructuringElement(Emgu.CV.CvEnum.ElementShape.Rectangle, new
    System.Drawing.Size(5, 5), new System.Drawing.Point(-1, -1));
    CvInvoke.Erode(temp, temp, oMat2, new System.Drawing.Point(-1, -1), 2,
    Emgu.CV.CvEnum.BorderType.Default, new MCvScalar(255, 0, 0, 255));
```

```
CvInvoke.Dilate(temp, temp, oMat2, new System.Drawing.Point(-1, -1), 2,
Emgu.CV.CvEnum.BorderType.Default, new MCvScalar(255, 0, 0, 255));

pictureBox2.Image = temp.Bitmap;
#region for COLOR

VectorOfVectorOfPoint contours = new VectorOfVectorOfPoint();
// khởi tạo biến đường bao

Mat n = new Mat();

CvInvoke.FindContours(temp, contours, n, Emgu.CV.CvEnum.RetrType.Ccomp,
Emgu.CV.CvEnum.ChainApproxMethod.ChainApproxSimple);

// tìm đường bao

for (int i = 0; i < contours.Size; i++) // kiểm tra xem có bao nhiêu đường bao
{
    double pemir = CvInvoke.ArcLength(contours[i], true);
    // tính độ dài đường bao

    VectorOfPoint approx = new VectorOfPoint();
    CvInvoke.ApproxPolyDP(contours[i], approx, 0.03 * pemir, true);

    //lay trong tam

    var moments = CvInvoke.Moments(contours[i]);
    double area = moments.M00; // diện tích hình

    if (area > MIN && area < MAX && (approx.Size >= 4) )
    {

        int x = (int)(moments.M10 / moments.M00);
        int y = (int)(moments.M01 / moments.M00);
        if (x != 0 && x > 200)

        {
            if (y < 100) { y2[i] = y; x2[i] = x; }

            x1 = x.ToString("D2");
            y1 = y.ToString("D2");
        }
    }
}

CvInvoke.DrawContours(img, contours, i, new MCvScalar(0, 0, 255), 3);
```

```
CvInvoke.Circle(img, new Point(x, y), 15, new MCvScalar(0, 0, 255), 2);  
}  
}  
//sap xep lai theo chieu den noi truoc  
for (int l = 0; l < 10; l++)  
{  
    for (int k = l + 1; k < 10; k++)  
    {  
        if (y2[l] < y2[k])  
        {  
            int tam;  
            tam = x2[l];  
            x2[l] = x2[k];  
            x2[k] = tam;  
            int tam2;  
            tam2 = y2[l];  
            y2[l] = y2[k];  
            y2[k] = tam2;  
        }  
    }  
}  
  
CvInvoke.Line(img, new Point(90, 0), new Point(90, 480), new  
Bgr(Color.DarkGray).MCvScalar, 200); //hien thi  
pictureBox1.Image = img.Bitmap;  
}
```

- Chương trình gửi tín hiệu về PLC điều khiển robot Delta gấp vật:

```
private void timer10_Tick(object sender, EventArgs e)
{
    if (mode == 0)
    {
        if (counter == 0) { dang_gap = true; toichovat(); zz = 20; }
        if (counter == 10) { gapvat(); }
        if (counter == 10 + 3) { nhacvatlen();
            if (T1 == TG1) { zz = 17; }
            if (T1 == TR1) { zz = 16; } }
        if (counter == 10 + 3 + 5) { toichothatav(); }
        if (counter == 10 + 3 + 5 + zz) { thavat(); }
        if (counter == 10 + 3 + 5 + zz + 6) { CloseValve(); }
        if (counter == 10 + 3 + 5 + zz + 6 + 1)
        {
            if (((VT1 + VT2) % 2) == 0) { vitri2_nanglen(); }
            else { vitri1_nanglen(); }
        }
        if (counter == 10 + 3 + 5 + zz + 6 + 1 + 4)
        {
            dang_gap = false; a = true;
            timer10.Stop();
            counter = 0;
        }
        counter++;
    }

    private void toichovat()
    {
        txtZ.Text = Zchogap;
        XYZtoTheta();
    }
}
```

```
try
{
    WriteItems.SetValue(txTheta1.Text, 1);
    WriteItems.SetValue(txTheta2.Text, 2);
    WriteItems.SetValue(txTheta3.Text, 3);
    ConnectedGroup.SyncWrite(ItemCount, ref ItemServerHandles, ref WriteItems,
out ItemServerErrors);
}
catch (Exception ex) { MessageBox.Show(ex.ToString()); }

Enable_move();
Disable_move();
}

private void gapvat()
{
    txtZ.Text = Zgapvat;
    XYZtoTheta();
    try
    {
        WriteItems.SetValue(txTheta1.Text, 1);
        WriteItems.SetValue(txTheta2.Text, 2);
        WriteItems.SetValue(txTheta3.Text, 3);
        ConnectedGroup.SyncWrite(ItemCount, ref ItemServerHandles, ref WriteItems,
out ItemServerErrors);
    }
    catch (Exception ex) { MessageBox.Show(ex.ToString()); }

    Enable_move();
    Disable_move();
}

private void nhacvatlen()
{
```

```
        double zn = Convert.ToDouble(Znangvat);
        txtZ.Text = Convert.ToString(zn);
        XYZtoTheta();
        while (Convert.ToDouble(txbTheta1.Text) < ho1 |
Convert.ToDouble(txbTheta2.Text) < ho2 | Convert.ToDouble(txbTheta3.Text) < ho3)
{
    zn--;
    txtZ.Text = Convert.ToString(zn);
    XYZtoTheta();
}
try
{
    WriteItems.SetValue(txbTheta1.Text, 1);
    WriteItems.SetValue(txbTheta2.Text, 2);
    WriteItems.SetValue(txbTheta3.Text, 3);
    ConnectedGroup.SyncWrite(ItemCount, ref ItemServerHandles, ref WriteItems,
out ItemServerErrors);
}
catch (Exception ex) { MessageBox.Show(ex.ToString()); }
Enable_move();
Disable_move();
}
private void toichothatavat()
{
try
{
    WriteItems.SetValue(T1, 1);      // tới ô thả vật
    WriteItems.SetValue(T2, 2);
    WriteItems.SetValue(T3, 3);
```

```
ConnectedGroup.SyncWrite(ItemCount, ref ItemServerHandles, ref WriteItems,
out ItemServerErrors);

}

catch (Exception ex) { MessageBox.Show(ex.ToString()); }

Enable_move();

Disable_move();

}

private void thavat()

{

    OpenValve();           //thả vật

    if (T1 == TR1)

    { VT1 = VT1 + 1; }

    if (T1 == TG1)

    { VT2 = VT2 + 1; }

    try

    {

        WriteItems.SetValue(VT1, 19);

        WriteItems.SetValue(VT2, 20);

        ConnectedGroup.SyncWrite(ItemCount, ref ItemServerHandles, ref WriteItems,
out ItemServerErrors);

    }

    catch (Exception ex) { MessageBox.Show(ex.ToString()); }

}

private void thavatxong()

{ CloseValve(); }

private void vitri1_nanglen()

{

    txtX.Text = "-70";

    txtY.Text = "-42";

    txtZ.Text = "-210";
```

```
XYZtoTheta();

try
{
    WriteItems.SetValue(txTheta1.Text, 1);
    WriteItems.SetValue(txTheta2.Text, 2);
    WriteItems.SetValue(txTheta3.Text, 3);
    ConnectedGroup.SyncWrite(ItemCount, ref ItemServerHandles, ref WriteItems,
out ItemServerErrors);
}

catch (Exception ex) { MessageBox.Show(ex.ToString()); }

Enable_move();
Disable_move();
}

private void vitri2_nanglen()
{
    txtX.Text = "-72";
    txtY.Text = "48";
    txtZ.Text = "-210";
    XYZtoTheta();

    try
    {
        WriteItems.SetValue(txTheta1.Text, 1);
        WriteItems.SetValue(txTheta2.Text, 2);
        WriteItems.SetValue(txTheta3.Text, 3);
        ConnectedGroup.SyncWrite(ItemCount, ref ItemServerHandles, ref WriteItems,
out ItemServerErrors);
    }

    catch (Exception ex) { MessageBox.Show(ex.ToString()); }

    Enable_move();
    Disable_move();
}
```

```
}

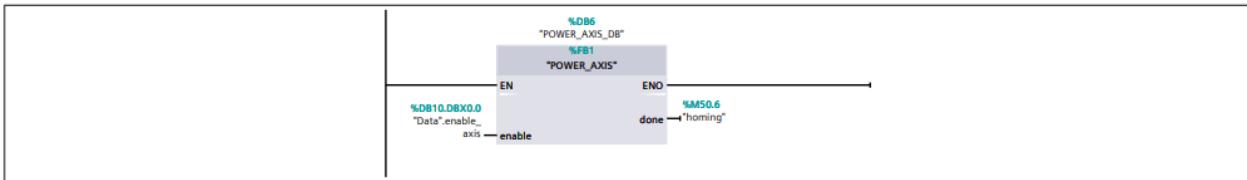
//Tinh goc theta1 (mp YZ)

int delta_calcAngleYZ(double x0, double y0, double z0, ref double theta, ref double
YJ, ref double ZJ)

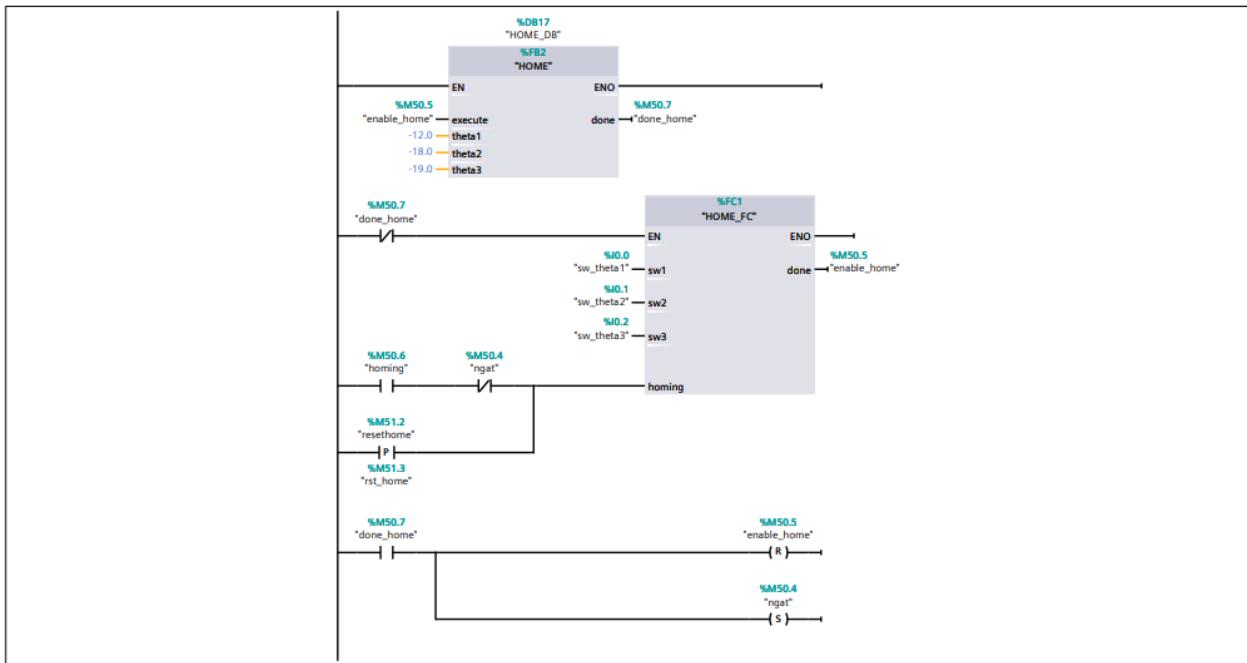
{
    double y1 = -0.5 * 0.57735 * ff; // f/2 * tg 30
    double y3 = y0 - 0.5 * 0.57735 * ee; // e/2 * tg30
    double a = (x0 * x0 + y3 * y3 + z0 * z0 + rf * rf - re * re - y1 * y1) / (2 * z0);
    double b = (y1 - y3) / z0; // discriminant
    double d = -(a + b * y1) * (a + b * y1) + rf * (b * b * rf + rf); //delta
    if (d < 0) return -1; // non-existing point
    // suy ra tọa độ J(0,y,zj)
    double yj = (y1 - a * b - Math.Sqrt(d)) / (b * b + 1); // choosing outer point
    double zj = a + b * yj;
    theta = Math.Round((180.0 * Math.Atan(-zj / (y1 - yj))) / pi + ((yj > y1) ? 180.0 :
0.0)), 2);
    YJ = Math.Round(yj, 2);
    ZJ = Math.Round(zj, 2);
    return 0;
}
```

Code PLC trong TIA Portal:

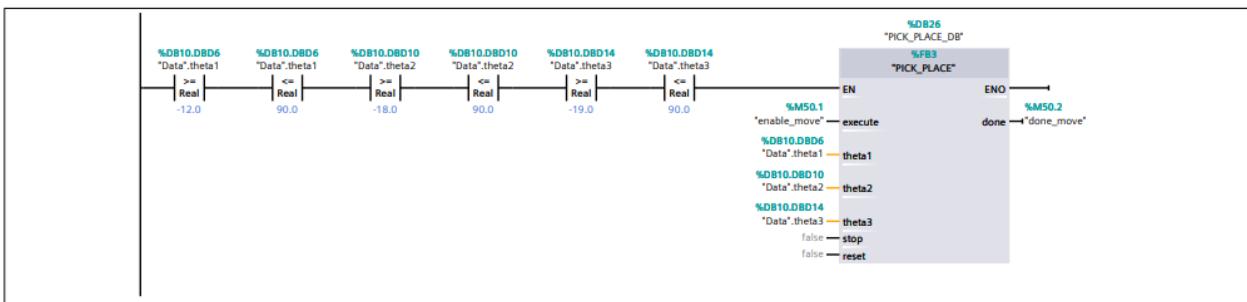
Network 1: NHAN START CAP NGUON CHO HE THONG



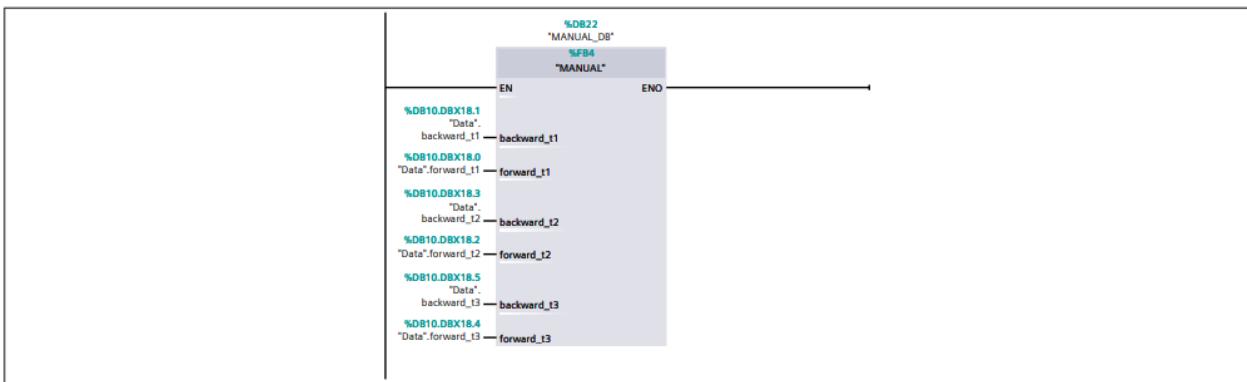
Network 2: NHAN RESET ROBOT DELTA VE VI TRI CHAM CONG TAC HANH TRINH



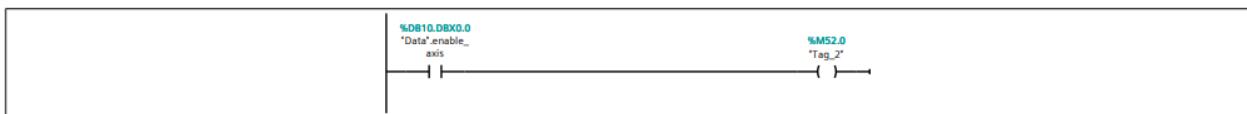
Network 3: KICH 1 XUNG LAM QUAY DONG CO



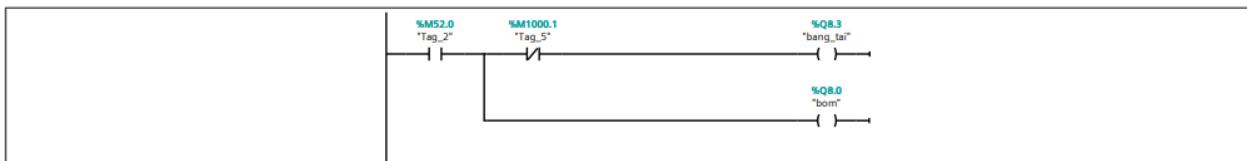
Network 4: CHE DO MANUAL, 1 THI QUAY, 0 THI DUNG



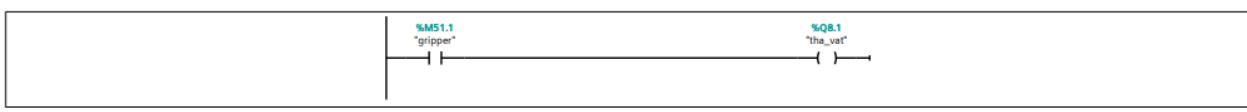
Network 5:



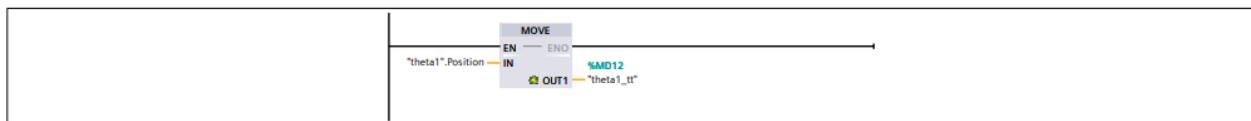
Network 6:



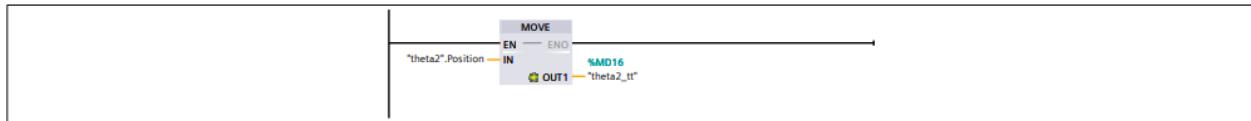
Network 7: MAY BOM



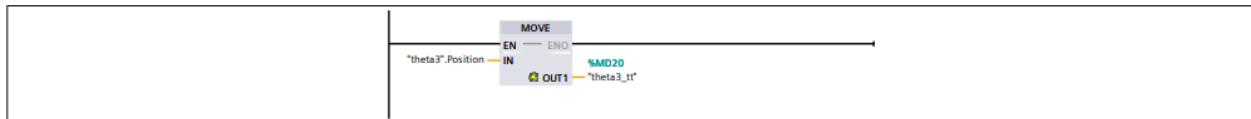
Network 8:



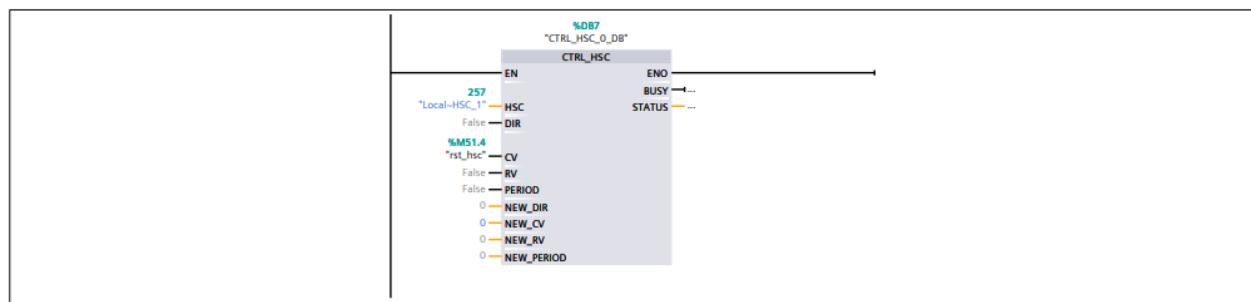
Network 9:



Network 10:



Network 11: KHAI BAO DOC XUNG TOC DO CAO



Network 12: DOC XUNG 1S ROI DOI SANG VONG/PHUT

