

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN



ĐỒ ÁN TỐT NGHIỆP
CAPSTONE PROJECT

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

ĐỀ TÀI:

**CẢI TIẾN HỆ THỐNG DÁN TEM CỦA PHÂN
XUỞNG SẢN XUẤT DÂY SẠC**

Giảng viên hướng dẫn: TS. TRƯƠNG THỊ BÍCH THANH

Cán bộ hướng dẫn: TT. TRẦN VĂN THƯƠNG

Sinh viên thực hiện:

- 1. PHAN NGỌC NAM – 105200417 – LỚP: 20TDHCLC2**
- 2. LÊ ĐỨC LONG – 105200458 – LỚP: 20TDHCLC3**
- 3. NGUYỄN HUY NGÂN – 105200503 – LỚP: 20TDHCLC4**

Đà Nẵng, tháng 06 năm 2025

TÓM TẮT

Tên đề tài: Cải tiến hệ thống dán tem của phân xưởng sản xuất dây sặc.

Sinh viên thực hiện:

1. LÊ ĐỨC LONG – 105200458 – LỚP: 20TDHCLC3
2. NGUYỄN HUY NGÂN – 105200503 – LỚP: 20TDHCLC4
3. PHAN NGỌC NAM – 105200417 – LỚP: 20TDHCLC2

Mô hình dán tem tự động được xây dựng nhằm thay thế công đoạn dán tem thủ công trong quy trình đóng gói sản phẩm dây sặc. Hệ thống sử dụng cơ cấu cơ khí (xy lanh khí nén và bộ cấp tem) kết hợp với công nghệ xử lý ảnh (Vision – OpenCV) để kiểm tra hình dạng hộp trước khi dán và xác minh vị trí tem sau khi dán.

Cụ thể, cảm biến hồng ngoại thứ nhất sẽ quét bề mặt và phân tích hình dạng hộp nhằm loại bỏ các sản phẩm bị biến dạng như móp, thủng,... Trường hợp hộp hợp lệ, hệ thống sẽ cấp tem và điều khiển cơ cấu dán để thực hiện dán tem. Sau đó, camera sẽ kiểm tra lại vị trí tem đã dán nhằm phát hiện các lỗi như lệch tem, bong tróc, dán sai vị trí,... Thông tin về sản phẩm đạt, lỗi hình dạng và lỗi dán tem sẽ được hiển thị trên giao diện phần mềm.

Mô hình giúp tăng độ chính xác, tiết kiệm thời gian và công sức so với thao tác thủ công, đồng thời góp phần hướng đến dây chuyền sản xuất thông minh, hiện đại hơn trong công nghiệp.

Kết quả thực nghiệm cho thấy mô hình có khả năng dán tem tự động với độ chính xác cao, thời gian xử lý nhanh và khả năng phát hiện lỗi hiệu quả. Đây là cơ sở để triển khai trong thực tế tại các dây chuyền sản xuất, góp phần nâng cao tính tự động hóa và hiện đại hóa nhà máy.

KHOA ĐIỆN

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Ngành
1	Lê Đức Long	105200458	20TDHCLC3	Kỹ thuật điều khiển & tự động hóa
2	Nguyễn Huy Ngân	105200503	20TDHCLC4	Kỹ thuật điều khiển & tự động hóa
3	Phan Ngọc Nam	105200417	20TDHCLC2	Kỹ thuật điều khiển & tự động hóa

1. Tên đề tài đồ án:

Cải tiến hệ thống dán tem của phân xưởng sản xuất dây sặc.

2. Nội dung các phần thuyết minh và tính toán:

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Lê Đức Long	Viết báo cáo và thuyết minh
2	Nguyễn Huy Ngân	Nghiên cứu giải pháp thiết kế và điều khiển giám sát hệ thống
3	Phan Ngọc Nam	

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Lê Đức Long	Nghiên cứu và đề xuất sơ đồ cấu trúc chi tiết các thiết bị phần cứng của hệ thống. Nghiên cứu lập trình xử lý tín hiệu và xây dựng chương trình điều khiển bằng phần mềm Tia Portal
2	Nguyễn Huy Ngân	Nghiên cứu, lập trình xử lý ảnh trên phần mềm Pycharm với thư viện OpenCV Đề xuất báo cáo kết quả và phương pháp đánh giá cho đề tài
3	Phan Ngọc Nam	Nghiên cứu, thiết kế giao diện điều khiển giám sát trên Pycharm Nghiên cứu các giải pháp điều khiển và cơ cấu dán tem

3. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Lê Đức Long	Nghiên cứu các thông số, kích thước để thiết kế các bản vẽ, mô hình
2	Nguyễn Huy Ngân	
3	Phan Ngọc Nam	

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Lê Đức Long	Sơ đồ tổng quát, sơ đồ cấu trúc thiết bị của hệ thống Bản vẽ tổng quát mô hình trên Solidworks

2	Nguyễn Huy Ngân	Sơ đồ thiết kế tủ điện điều khiển bao gồm vỏ tủ và phân bố trí thiết bị Lưu đồ thuật toán xử lý ảnh trên phần mềm Pycharm
3	Phan Ngọc Nam	Lưu đồ thuật toán tổng quan của hệ thống trên phần mềm TIA Portal

4. <i>Họ tên người hướng dẫn</i>	<i>Phần/Nội dung:</i>
TS. Trương Thị Bích Thanh	Hướng dẫn tư vấn giải pháp, lựa chọn công nghệ cho dự án Hướng dẫn làm thuyết trình báo cáo dự án Theo dõi tiến độ của dự án
TT. Trần Văn Thương	Nêu ra yêu cầu của dự án Thảo luận, tư vấn về giải pháp công nghệ Theo dõi tiến độ dự án Đánh giá sản phẩm của dự án

5. Ngày giao nhiệm vụ đồ án: 10/02/2025

6. Ngày hoàn thành đồ án: 10/06/2025

Đà Nẵng, ngày ...tháng... năm 2025

Trưởng Bộ môn

Người hướng dẫn

TS. Giáp Quang Huy

TS. Trương Thị Bích Thanh

TT. Trần Văn Thương

PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP

(Phiếu dành cho người hướng dẫn/sinh viên)

Họ và tên sinh viên:

1. LÊ ĐỨC LONG – số thẻ SV: 105200458
2. NGUYỄN HUY NGÂN – số thẻ SV: 105200503
3. PHAN NGỌC NAM – số thẻ SV: 105200417

Tên đề tài ĐATN: Cải tiến hệ thống dán tem của phân xưởng sản xuất dây sặc.

Họ và tên người HD1: TS. Trương Thị Bích Thanh – Đơn vị: Trường Đại học Bách Khoa – Đại học Đà Nẵng.

Họ và tên người HD2: TT. Trần Văn Thương – Đơn vị: Công ty Luxshare ICT Nghệ An.

Tuần	Ngày	Khối lượng		GVHD ký tên
		đã thực hiện (%)	tiếp tục thực hiện (%)	
1	10/02/2025	Nhận đề tài	Gặp người hướng dẫn định hướng đồ án	
2	17/02/2025	Khảo sát và nghiên cứu về các phương pháp dán tem trong phân xưởng	Xác định và thống nhất các phần cứng sử dụng trong dự án	
3	25/02/2025	Xác định những yêu và bài toán đặt ra của hệ thống	Báo cáo hoạt động của nhóm với giảng viên hướng dẫn	
4	10/03/2025	Duyệt lần 1: Đánh giá khối lượng hoàn thành ____ %: Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐		
5	20/03/2025	Thiết kế tổng quan mô hình của toàn hệ thống		
6	25/03/2025	Thiết kế phần cơ khí và điện	Tham khảo ý kiến người hướng dẫn, từ đó đầu nối các thiết bị điện, cảm biến,...	
7	05/04/2025	Lập trình, code chương trình trên phần mềm Tia Portal	Tiến hành chạy mô phỏng chương trình trên phần mềm	

8	12/04/2025	Duyệt lần 2: Đánh giá khối lượng hoàn thành ____ %: Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐		
9	20/04/2025	Lắp ráp hoàn chỉnh mô hình Cài đặt các thông số cho các cảm biến	Tinh chỉnh mô hình và thông số, nạp chương trình code PLC vào chạy trên mô hình thực tế	
10	25/04/2025	Code chương trình xử lý ảnh cho hệ thống	Chạy thử để lấy dữ liệu hình ảnh sản phẩm để đưa vào xử lý	
11	10/05/2025	Thiết kế giao diện điều khiển giám sát trên phần mềm Pycharm.	Thực hiện chạy test giao diện đã thiết kế Báo cáo kết quả với giảng viên hướng dẫn	
12	20/05/2025	Duyệt lần 3: Đánh giá khối lượng hoàn thành ____ %: Được tiếp tục làm ĐATN ☐ Không tiếp tục thực hiện ĐATN ☐		
13	30/05/2025	Tiến hành test, thực nghiệm toàn bộ hệ thống	Báo cáo các kết quả đạt được với giảng viên. Quay video mô phỏng thực nghiệm trong quá trình thực hiện	
14	05/06/2025	Tổng hợp tài liệu, hoàn thành các phần trong báo cáo		
15	15/06/2025	Chuẩn bị cho phần thuyết trình		

LỜI NÓI ĐẦU VÀ CẢM ƠN

Trong bối cảnh nền công nghiệp ngày càng phát triển, việc ứng dụng các hệ thống tự động hóa vào sản xuất không chỉ giúp tăng năng suất mà còn đảm bảo chất lượng sản phẩm và giảm thiểu sự phụ thuộc vào lao động thủ công. Đặc biệt trong các dây chuyền đóng gói, công đoạn dán tem – tương chừng đơn giản – lại đóng vai trò quan trọng trong việc xác định thông tin sản phẩm, truy xuất nguồn gốc và đảm bảo yếu tố thẩm mỹ.

Từ thực tế đó, nhóm chúng em đã thực hiện đề tài: **“Cải tiến hệ thống dán tem của phân xưởng sản xuất dây sặc”** nhằm nghiên cứu và đề xuất một giải pháp kỹ thuật phù hợp để thay thế cho công đoạn dán tem bằng tay trong xưởng sản xuất dây sặc.

Báo cáo này trình bày quá trình tìm hiểu, thiết kế và xây dựng mô hình thực tế, bao gồm: quy trình công nghệ, lựa chọn thiết bị, thiết kế cơ khí – điện – phần mềm, tích hợp hệ thống xử lý ảnh và điều khiển tự động. Thông qua đề tài, nhóm mong muốn thể hiện khả năng vận dụng kiến thức chuyên ngành để giải quyết một bài toán thực tiễn mang tính ứng dụng cao.

Tuy nhiên, do giới hạn về thời gian, kinh nghiệm và điều kiện thực nghiệm, chắc chắn báo cáo không tránh khỏi những thiếu sót. Nhóm rất mong nhận được những ý kiến đóng góp từ thầy cô và các anh/chị để hoàn thiện đề tài tốt hơn.

Nhóm sinh viên thực hiện đề tài xin chân thành cảm ơn:

- **Ban giám hiệu Nhà trường, Khoa Điện**, đã tạo điều kiện thuận lợi để nhóm có thể thực hiện đề tài này trong khuôn khổ học phần.
- **Cô Trương Thị Bích Thanh** đã tận tình hướng dẫn, định hướng chuyên môn và hỗ trợ nhóm trong suốt quá trình thực hiện, từ bước khảo sát thực tế đến hoàn thiện báo cáo.
- **Các cán bộ kỹ thuật và nhân viên xưởng sản xuất** đã tạo điều kiện cho nhóm khảo sát, thu thập số liệu thực tế và chia sẻ kinh nghiệm thực hành quý báu.

Cuối cùng, nhóm xin gửi lời cảm ơn đến các thành viên trong nhóm vì tinh thần làm việc nhóm tích cực, sự hỗ trợ lẫn nhau và nỗ lực không ngừng để hoàn thành đề tài đúng tiến độ.

Xin chân thành cảm ơn!

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Chúng em xin cam đoan rằng toàn bộ nội dung trong báo cáo này là kết quả của quá trình nghiên cứu, học tập và thực hiện nghiêm túc của chính nhóm chúng em. Các thông tin, số liệu, hình ảnh, sơ đồ được trích dẫn từ các tài liệu tham khảo đều được ghi rõ nguồn.

Chúng em không sao chép, sử dụng nội dung của người khác dưới bất kỳ hình thức nào mà không trích dẫn đầy đủ. Báo cáo không vi phạm các quy định về đạo đức học thuật và bản quyền. Trong trường hợp phát hiện có hành vi gian lận, sao chép không đúng quy định, chúng em xin hoàn toàn chịu trách nhiệm trước Nhà trường và giảng viên hướng dẫn.

Chúng em xin chân thành cảm ơn.

Sinh viên thực hiện 1

Sinh viên thực hiện 2

Sinh viên thực hiện 3

Phan Ngọc Nam

Lê Đức Long

Nguyễn Huy Ngân

MỤC LỤC

TÓM TẮT	I
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	II
PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP.....	IV
LỜI NÓI ĐẦU VÀ CẢM ƠN	VI
LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT	VII
MỤC LỤC	VIII
DANH MỤC HÌNH ẢNH.....	XI
DANH MỤC BẢNG BIỂU	XI
DANH MỤC CÁC CUM TỪ VIẾT TẮT.....	XIV
MỞ ĐẦU	1
CHƯƠNG 1. TỔNG QUAN CỦA ĐỀ TÀI.....	3
1.1 Tính cấp thiết của đề tài.....	3
1.1.1 Các giải pháp hiện có	4
1.2 Giải pháp sơ bộ	7
1.2.1 Sơ đồ khối mô tả hệ thống	7
1.2.2 Giải pháp kỹ thuật	7
1.2.3 Quy trình thiết kế.....	9
1.2.4 Công cụ	10
1.2.5 Nguyên vật liệu cần để triển khai.....	10
1.3 Dự kiến kết quả đạt được.....	10
1.3.1 Sản phẩm dự kiến.....	10
1.3.2 Phương pháp thực hiện	10
1.3.3 Quy trình dự kiến	11
1.4 Đề xuất phương pháp đánh giá và tiêu chí kiểm thử.....	11
1.4.1 Phương pháp đánh giá.....	11
1.4.2 Tiêu chí kiểm thử.....	12

1.5	Kết hoạch thực hiện và phân công công việc	12
1.5.1	Kế hoạch thực hiện	12
1.6	Kết luận.....	14
1.7	Mục tiêu đề tài.....	14
CHƯƠNG 2. GIẢI PHÁP THỰC HIỆN CƠ CẤU DÁN TEM.....		16
2.1	Đề xuất giải pháp cải tiến máy dán tem.....	16
2.2	Quy trình công nghệ của hệ thống	17
2.3	Lựa chọn thiết bị trong mô hình	19
2.3.1	Tổng quan về các thiết bị	19
2.3.2	Thiết bị trong mô hình dán tem	20
2.4	Chức năng và thông số của các thiết bị trong mô hình	21
2.4.1	PLC Siemens S7-1200	21
2.4.2	Webcam 1080P.....	25
2.4.3	Mạch giảm áp XL4016E1	27
2.4.4	Động cơ giảm tốc GB37-550 /12V DC.....	30
2.4.5	Moter giảm tốc.....	32
2.4.6	Cảm biến khoảng cách vật cản E3F-DS10C4	33
2.4.7	E3Z-D61 Cảm Biến Quang Khuếch Tán Khoảng Dò 5-100mm NPN 12~24VDC.....	35
2.4.8	Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F.....	37
2.4.9	Van điện từ (solenoid) khí nén LAIZE 4V210-08.....	39
2.4.10	Xi lanh TN10-100.....	43
2.5	Thiết kế mô hình dán tem	44
2.5.1	Thiết kế tủ điện điều khiển.....	44
2.5.2	Phần cơ khí.....	46
2.5.3	Phần điều khiển – khí nén	47
2.6	Kết luận.....	48
CHƯƠNG 3. THIẾT LẬP CẤU HÌNH CÁC THIẾT BỊ TRÊN PHẦN MỀM VÀ XÂY DỰNG THUẬT TOÁN ĐIỀU KHIỂN		50

3.1	Xây dựng thuật toán điều khiển hệ thống trên phần mềm TIA Portal	50
3.1.1	<i>Giới thiệu phần mềm TIA Portal.</i>	50
3.1.2	<i>Một số chức năng chính.</i>	51
3.1.3	<i>Lưu đồ thuật toán</i>	52
3.1.4	<i>Bảng phân kênh</i>	57
3.2	Xây dựng thuật toán xử lý ảnh trên phần mềm Pycharm	57
3.2.1	<i>Giới thiệu về phần mềm</i>	57
3.2.2	<i>Giới thiệu về lý thuyết xử lý ảnh</i>	58
3.2.3	<i>Các bước để xử lý ảnh</i>	60
3.2.4	<i>Điều khiển PLC và hệ thống phần cứng</i>	63
3.2.5	<i>Thiết kế giao diện GUI trên Python</i>	65
3.2.6	<i>Màn hình chính của hệ thống</i>	66
3.3	Kết luận.....	67
CHƯƠNG 4. KIỂM NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ ĐẠT ĐƯỢC ..		68
4.1	Mô hình hệ thống	68
4.1.1	<i>Chi tiết các bộ phận của hệ thống</i>	68
4.1.2	<i>Video hoạt động chi tiết</i>	69
4.2	Đánh giá hiệu quả vận hành thực tế của sản phẩm.....	70
4.2.1	<i>Phân loại sản phẩm</i>	70
4.2.2	<i>Kết quả vận hành thực tế</i>	72
4.3	Đánh giá và cải tiến.....	74
4.3.1	<i>Đánh giá độ chính xác</i>	74
4.3.2	<i>Đánh giá tính ổn định.</i>	74
4.3.3	<i>Đánh giá kỹ thuật</i>	74
4.3.4	<i>Đánh giá tổng quát và cải tiến</i>	75
KẾT LUẬN		76
TÀI LIỆU THAM KHẢO		I
PHỤ LỤC		II

DANH MỤC HÌNH ẢNH

Hình 1.1 Máy dán tem cuộn Model HERMA 400	5
Hình 1.2 Máy dán tem LT – 100	6
Hình 1.3 Sơ đồ khối hệ thống	7
Hình 2.1 Quy trình công nghệ của hệ thống	17
Hình 2.2 Cơ cấu phát hiện biến dạng bề mặt dán tem	17
Hình 2.3 Cơ cấu dán tem	18
Hình 2.4 Cơ cấu xử lý ảnh	19
Hình 2.5 Sơ đồ kết nối thiết bị trong hệ thống.....	20
Hình 2.6 Bộ điều khiển PLC-S1200 CPU 1212 DC	23
Hình 2.7 Đầu nối dây bộ điều khiển PLC-S1200 CPU 1212C.....	24
Hình 2.8 Webcam 1080P	25
Hình 2.9 Mạch giảm áp XL4016E1	28
Hình 2.10 cấu hình chân cắm của Mạch giảm áp XL4016E1	28
Hình 2.11 Sơ đồ khối chức năng của Mạch giảm áp XL4016E1	29
Hình 2.12 Động cơ giảm tốc GB37-550	30
Hình 2.13 sơ đồ kích thước động cơ giảm tốc GB37-550	31
Hình 2.14 Motor giảm tốc 24V DC	32
Hình 2.15 Bản vẽ của motor giảm tốc	33
Hình 2.16 Cảm Biến Khoảng Cách Vật Cản E3F-DS10C4 3-10cm, NPN-NO	34
Hình 2.17 Sơ đồ nối dây cảm biến khoảng cách vật cản E3F-DS10C4 3-10cm, NPN-NO	35
Hình 2.18 E3Z-D61 Cảm Biến Quang Khuếch Tán Khoảng Dò 5-100mm NPN 12~24VDC.....	35
Hình 2.19 Ngõ ra NPN.....	36
Hình 2.20 Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F.....	37
Hình 2.21 Sơ đồ cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F	38

Hình 2.22 Van điện từ (solenoid) khí nén LAIZE 4V210-08	40
Hình 2.23 Cấp nguồn cho van điện từ.	41
Hình 2.24 Sơ đồ nguyên lý hoạt động	42
Hình 2.25 Xi lanh TN10-100	43
Hình 2.26 Kích thước bản vẽ mặt cắt tủ điều khiển.	45
Hình 2.27 Bản vẽ bố trí thiết bị trong tủ điều khiển	45
Hình 2.28 Bản vẽ thiết kế tủ điều khiển 3D.....	46
Hình 2.29 Kích thước thiết kế khung cơ khí.....	47
Hình 2.30 Bản vẽ thiết kế mô hình dán tem	48
Hình 3.1 Phần mềm TIA protal.....	50
Hình 3.2 Giao diện phần mềm TIA portal	52
Hình 3.3 Lưu đồ thuật toán khâu I.....	52
Hình 3.4 Lưu đồ thuật toán khâu II.....	54
Hình 3.5 Lưu đồ thuật toán khâu III	56
Hình 3.6 Giao diện phần mềm Pycharm.....	57
Hình 3.7 Mô tả quá trình xử lý ảnh.....	58
Hình 3.8 Màn hình điều khiển hệ thống	66
Hình 4.1 Ảnh tổng thể của mô hình.....	68
Hình 4.2 Cơ cấu xử lý hộp biến dạng	68
Hình 4.3 Cơ cấu dán tem	69
Hình 4.4 Cơ cấu xử lý ảnh	69
Hình 4.5 Tủ điện PLC	69
Hình 4.6 Sản phẩm đạt yêu cầu	70
Hình 4.7 Sản phẩm biến dạng bề mặt dán tem	71
Hình 4.8 Sản phẩm lỗi dán tem.....	71
Hình 4.9 Lỗi bề mặt dán tem	72
Hình 4.10 Xi lanh ép thực hiện dán tem	72
Hình 4.11 Xi lanh đẩy hộp dán tem bị lỗi.....	73

Hình 4.12 Màn hình giám sát cập nhật số lượng sản phẩm.....	73
---	----

DANH MỤC BẢNG BIỂU

Bảng 1.1 Các loại máy dán tem bán tự động	4
Bảng 1.2 Giải pháp cơ cấu dán tem	7
Bảng 1.3 Phương pháp phân loại sản phẩm lỗi.....	9
Bảng 1.4 Bảng đánh giá tiêu chí kiểm thử.....	12
Bảng 1.5 Kế hoạch thực hiện dự án	12
Bảng 1.6 Phân công công việc	13
Bảng 2.1 Các thiết bị trong mô hình	20
Bảng 2.2 Phân tích, đánh giá S7-1200 cấp nhỏ (Hãng Siemens):.....	21
Bảng 2.3 Lý do chọn CPU 1212C	22
Bảng 2.4 Chức năng bộ điều khiển PLC-S1200 CPU 1212 DC.....	23
Bảng 2.5 Thông số kỹ thuật PLC – S1200 CPU1212.....	23
Bảng 2.6 Thông số kỹ thuật webcam 1080P	26
Bảng 2.7 Thông số kỹ thuật của mạch giảm áp XL4016E1	29
Bảng 2.8 Thông số kỹ thuật Động cơ giảm tốc GB37-550	31
Bảng 2.9 Thông số kỹ thuật Moter giảm tốc	33
Bảng 2.10 Bảng thông số kỹ thuật Cảm Biến Khoảng Cách Vật Cản E3F-DS10C4 3-10cm, NPN-NO	34
Bảng 2.11 Thông số kỹ thuật E3Z-D61 Cảm Biến Quang Khuếch Tán Khoảng Dò 5-100mm NPN 12~24VDC	36
Bảng 2.12 Thông số kỹ thuật của cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F	38
Bảng 2.13 Thông số kỹ thuật Van điện từ (solenoid) khí nén LAIZE 4V210-08.....	40
Bảng 2.14 Thông số kỹ thuật Van điện từ (solenoid) khí nén LAIZE 4V210-08.....	44
Bảng 3.1 Bảng phân kênh Input/Output	57
Bảng 3.2 Các thành phần trong giao diện BUI	65
Bảng 3.3 Các chức năng có trên màn hình hệ thống	66
Bảng 4.1 Bảng đánh giá độ chính xác.....	74

DANH MỤC CÁC CUM TỪ VIẾT TẮT

STT	Cum từ viết tắt	Giải thích ý nghĩa
1	PLC	Programmable Logic Controller: bộ điều khiển logic lập trình
2	CBKC	Cảm biến khoảng cách vật cản E3F-DS10C4
3	CBHN	Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F
4	CBKT	Cảm biến quang khuếch tán E3Z-D61
5	VDT	Van điện từ
6	MTGT	Mô tơ giảm tốc

MỞ ĐẦU

Trong xu hướng phát triển mạnh mẽ của công nghiệp 4.0, việc ứng dụng các giải pháp tự động hóa và công nghệ xử lý ảnh vào quá trình sản xuất đang trở thành một yêu cầu tất yếu để tăng năng suất, giảm thiểu sai sót và nâng cao chất lượng sản phẩm. Trong các nhà máy sản xuất dây sọc, công đoạn dán tem sản phẩm tuy đơn giản nhưng lại giữ vai trò rất quan trọng trong việc đảm bảo khả năng truy xuất nguồn gốc, nhận diện sản phẩm và đáp ứng các tiêu chuẩn về đóng gói. Hiện nay, nhiều cơ sở sản xuất vẫn đang thực hiện công đoạn dán tem bằng tay, dẫn đến nhiều bất cập như sai lệch vị trí, dán thiếu tem, tem bị bong tróc, hoặc nhầm lẫn tem sản phẩm.

Trước thực trạng đó, nhóm sinh viên thực hiện đề tài:

“Cải tiến hệ thống dán tem của phân xưởng sản xuất dây sọc”. Nhằm nghiên cứu, thiết kế và đề xuất một hệ thống tích hợp tự động hóa với xử lý ảnh để thay thế công đoạn dán tem thủ công, hướng tới một mô hình hiệu quả, chính xác, và có khả năng ứng dụng thực tế.

Mục tiêu của đề tài

- Xây dựng một mô hình mô phỏng hệ thống dán tem tự động có tích hợp kiểm tra bằng cảm biến trước và camera sau khi dán tem.
- Thiết kế phần mềm điều khiển giao diện thân thiện với người dùng, hiển thị kết quả kiểm tra hình ảnh và điều khiển thiết bị.
- Đánh giá hiệu quả hoạt động của hệ thống thông qua các tiêu chí: độ chính xác dán tem, khả năng phát hiện lỗi, tính ổn định và chi phí triển khai.

Phạm vi và đối tượng nghiên cứu

- Phạm vi: Đề tài tập trung xây dựng mô hình thu nhỏ phục vụ mục đích nghiên cứu và thử nghiệm trong môi trường trường học. Không triển khai trên quy mô dây chuyền thực tế.
- Đối tượng nghiên cứu: Quá trình dán tem hộp sản phẩm dây sọc (loại hộp nhỏ, hình hộp chữ nhật), hệ thống xử lý ảnh, các thiết bị cơ khí – điện điều khiển.

Phương pháp nghiên cứu

- Phương pháp khảo sát thực tế: Tìm hiểu quy trình dán tem hiện tại trong xưởng sản xuất dây sọc.
- Phương pháp thiết kế hệ thống: Gồm thiết kế cơ khí, thiết kế tủ điện điều khiển, lập trình PLC và phần mềm xử lý ảnh.

- Phương pháp thực nghiệm và mô phỏng: Xây dựng mô hình, chạy thử, thu thập dữ liệu và đánh giá.
- Phương pháp xử lý ảnh số: Áp dụng các thuật toán trong thư viện OpenCV như phát hiện cạnh, xác định vị trí, kiểm tra lỗi tem.

Cấu trúc của báo cáo

Báo cáo được chia thành các chương như sau:

- **Chương 1:** Tổng quan về đề tài, tính cấp thiết, khảo sát công nghệ và đề xuất giải pháp thiết kế
- **Chương 2:** Giải pháp thực hiện cơ cấu dán tem
- **Chương 3:** Thiết lập cấu hình các thiết bị trên phần mềm và xây dựng thuật toán điều khiển
- **Chương 4:** Kiểm nghiệm và đánh giá kết quả đạt được
- **Kết luận và hướng phát triển:** Tổng kết kết quả đạt được, các điểm còn hạn chế và định hướng phát triển sau này.

CHƯƠNG 1. TỔNG QUAN CỦA ĐỀ TÀI

1.1 Tính cấp thiết của đề tài

Trong bối cảnh công nghiệp hóa và tự động hóa ngày càng phát triển mạnh mẽ tại Việt Nam, các doanh nghiệp sản xuất đang không ngừng tìm kiếm các giải pháp để tối ưu hóa quy trình sản xuất, nâng cao năng suất và giảm thiểu chi phí. Một trong những công đoạn vẫn còn tồn tại nhiều phương pháp thủ công, đặc biệt ở các doanh nghiệp vừa và nhỏ, là quy trình dán tem sản phẩm.

Việc dán tem nhãn lên sản phẩm thường bao gồm các thao tác như lấy tem, định vị và dán lên bề mặt sản phẩm. Khi thực hiện thủ công, quy trình này tiềm ẩn nhiều hạn chế như:

- **Năng suất thấp:** Tốc độ dán tem phụ thuộc vào kỹ năng và sự tập trung của người lao động, khó đạt được năng suất cao khi số lượng sản phẩm lớn.
- **Sai sót:** Dễ xảy ra các lỗi như dán lệch, nhăn, hoặc không đúng vị trí, ảnh hưởng đến thẩm mỹ và thông tin sản phẩm.
- **Chi phí nhân công cao:** Tốn nhiều nhân lực cho một công đoạn đơn giản, làm tăng chi phí sản xuất.
- **Tính ổn định không cao:** Chất lượng dán tem có thể không đồng đều giữa các sản phẩm và giữa các ca làm việc khác nhau.

Xuất phát từ thực tiễn qua quá trình thực tập và làm việc tại doanh nghiệp, nhóm đã có cơ hội tiếp cận và quan sát trực tiếp quy trình sản xuất tại xưởng. Qua quá trình tìm hiểu, nhóm nhận thấy rằng trong dây chuyền sản xuất hiện tại, công đoạn dán tem sản phẩm vẫn đang được thực hiện hoàn toàn thủ công. Đây là một khâu tương chừng đơn giản nhưng lại tiêu tốn khá nhiều thời gian và nhân công, đồng thời tiềm ẩn nguy cơ sai sót như dán lệch, dán sai vị trí hoặc thiếu tem, ảnh hưởng đến tính chuyên nghiệp, chất lượng thẩm mỹ và độ tin cậy của sản phẩm.

Thực tế này đặt ra yêu cầu cấp thiết về việc cải tiến hoặc tự động hóa công đoạn dán tem nhằm tăng năng suất, giảm phụ thuộc vào lao động thủ công, đảm bảo tính đồng nhất và chính xác trong quá trình sản xuất. Đây cũng chính là động lực để nhóm lựa chọn và triển khai đề tài nghiên cứu nhằm góp phần nâng cao hiệu quả hoạt động chung của doanh nghiệp.

1.1.1 Các giải pháp hiện có

Hiện nay, các giải pháp dán tem tự động được chia thành 4 nhóm chính, tùy theo công nghệ và mức độ tự động hóa. Dưới đây là tổng hợp chi tiết:

1. Máy Dán Tem Bán Tự Động (Semi-Automatic Labeling Machines)

Đặc điểm:

- Cần người vận hành đặt sản phẩm vào vị trí, máy tự động dán tem.
- Phù hợp cho doanh nghiệp nhỏ, sản lượng vừa phải.

Các loại phổ biến:

Bảng 1.1 Các loại máy dán tem bán tự động

Loại máy	Nguyên lý hoạt động	Ưu điểm	Nhược điểm
Máy dán tem cầm tay	Người dùng vận hành máy, tem được phun hoặc ép lên sản phẩm	Giá thành rẻ từ 3 – 8 triệu và dễ sử dụng	Tốc độ dán chậm
Máy dán tem quay	Dùng cơ cấu quay để dán tem lên chai hoặc lọ tròn	Phù hợp chai nhựa, lon nước	Khó áp dụng cho sản phẩm phẳng

Ứng dụng: Dán tem date, tem bảo hành, tem QR code trên hộp giấy, chai lọ.

2. Máy Dán Tem Tự Động (Automatic Labeling Machines)

❖ Máy dán tem dạng cuộn (Roll label applicator)

Máy dán tem dạng cuộn (Roll Label Applicator) là một loại máy dán tem tự động sử dụng tem được cung cấp dưới dạng cuộn (roll label). Máy này được ứng dụng rộng rãi trong các ngành công nghiệp như thực phẩm, dược phẩm, mỹ phẩm, đồ uống, điện tử và logistics nhờ khả năng dán tem nhanh, chính xác và tiết kiệm nhân công.

- Một máy dán tem dạng cuộn thường bao gồm các bộ phận sau:
 - Hệ thống cấp tem (Label Unwinder)

Giữ cuộn tem và nhả tem theo tốc độ dán.

Có thể tích hợp cảm biến để phát hiện hết tem.

- Hệ thống kéo tem (Label Feeder/Peeler)

Tem được kéo qua một lưỡi bóc (peeler blade) để tách lớp đế và dán lên sản phẩm.

- Hệ thống dán tem (Label Applicator)

Sử dụng cơ cấu khí nén (pneumatic), con lăn (roller) hoặc đệm xóp (tamp pad) để ép tem lên sản phẩm.

- Hệ thống điều khiển (PLC & HMI)

Điều chỉnh tốc độ, vị trí dán, thông số kỹ thuật.

Một số máy cao cấp tích hợp camera kiểm tra tem (vision system).

- Băng tải tự động (Conveyor System)

Vận chuyển sản phẩm qua trạm dán tem đồng bộ với tốc độ máy.

- Nguyên lý hoạt động.

Cuộn tem được lắp vào trục cấp và kéo qua lưỡi bóc, tách lớp đế silicon để lộ lớp keo dính. Khi sản phẩm di chuyển trên băng tải đến vị trí dán, cơ cấu dán (khí nén, con lăn hoặc đệm xóp) sẽ ép tem lên bề mặt sản phẩm. Quá trình được kiểm soát chính xác nhờ cảm biến quang học hoặc encoder, đảm bảo tem dán đúng vị trí với tốc độ cao.

- Các loại máy hiện có trên thị trường
 - Model HERMA 400

Tốc độ: 200 – 300 tem/phút.

Bề mặt dán: Chai tròn, hộp phẳng, túi mềm, ống/tuýp.

Phù hợp cho dây chuyền vừa và lớn, hỗ trợ tem nhãn phức tạp.



Hình 1.1 Máy dán tem cuộn Model HERMA 400

– Máy dán tem LT – 100

Tốc độ dán tem: 100 – 150tem/phút.

Hỗ trợ đa dạng bề mặt: chai tròn, hộp, túi, ống.



Hình 1.2 Máy dán tem LT – 100

❖ Hạn chế của các máy dán tem hiện tại

Các máy dán tem tự động hiện có trên thị trường đều tồn tại những nhược điểm quan trọng khiến chúng không phù hợp với nhu cầu thực tế của nhiều doanh nghiệp, đặc biệt là các đơn vị vừa và nhỏ:

- Không có hệ thống kiểm tra chất lượng: Tem rách, mờ hoặc sai thông tin vẫn được dán lên sản phẩm.
- Dễ kẹt tem: Gây gián đoạn sản xuất, tốn thời gian xử lý sự cố.
- Chi phí quá cao vượt ngân sách của hầu hết doanh nghiệp nhỏ.
- Phức tạp trong vận hành: Yêu cầu kỹ thuật viên trình độ cao.

❖ Đề xuất cải tiến các vấn đề kỹ thuật.

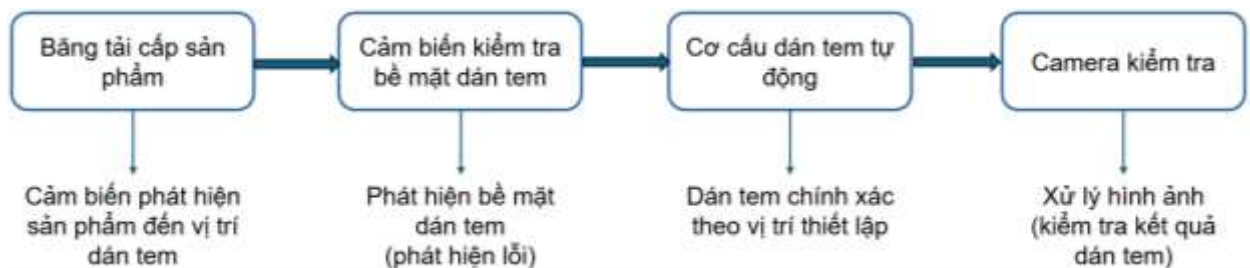
Nhóm nghiên cứu đề xuất giải pháp tích hợp hai cải tiến kỹ thuật quan trọng nhằm nâng cao chất lượng và hiệu quả hệ thống dán tem tự động, đồng thời tối ưu hóa chi phí đầu tư:

- Cơ Cấu Xử Lý Biến Dạng Bề Mặt Thông Minh
 - Sử dụng cảm biến quang học chi phí thấp kết hợp thuật toán xử lý ảnh để phát hiện và hiệu chỉnh bề mặt sản phẩm trước khi dán tem.
 - Tiết kiệm chi phí bằng cách tận dụng hệ thống điều khiển hiện có (PLC hoặc vi điều khiển nhúng).

- Hệ Thống Kiểm Tra Chất Lượng Sau Dán Tự Động
 - Triển khai camera công nghiệp giá rẻ (kết hợp OpenCV) để kiểm tra vị trí, góc dán và phát hiện lỗi (sai lệch tem khi dán).
 - Tự động phân loại sản phẩm lỗi thông qua cơ cấu đẩy khí nén hoặc băng tải rẽ nhánh, giảm thiểu can thiệp thủ công.
 - Giảm chi phí vận hành nhờ khả năng tự động hóa cao và dễ bảo trì.

1.2 Giải pháp sơ bộ

1.2.1 Sơ đồ khối mô tả hệ thống



Hình 1.3 Sơ đồ khối hệ thống

→ Nếu cảm biến kiểm tra bề mặt dán tem phát hiện sản phẩm lỗi → hệ thống loại bỏ sản phẩm trước khi đi đến cơ cấu dán tem.

→ Nếu camera kiểm tra phát hiện tem dán lỗi → ghi nhận lỗi và loại sản phẩm ra khỏi dây chuyền. Khi camera kiểm tra tem được dán đúng tiêu chuẩn thì cho sản phẩm đi đến công đoạn tiếp theo

1.2.2 Giải pháp kỹ thuật

- Phần cơ khí (đảm nhận việc dán tem).
- Phần tự động hóa (xử lý ảnh để định vị tem).
- Hệ thống kiểm tra & đẩy sản phẩm lỗi (loại bỏ sản phẩm không đạt chuẩn).

1. Phần cơ cấu dán tem

Bảng 1.2 Giải pháp cơ cấu dán tem

Phương pháp	Nguyên lý hoạt động	Ưu điểm	Nhược điểm
Dán tem bằng cơ cấu cuộn (Label Applicator)	Tem được cuộn vào rulo, máy tự bóc và dán lên sản phẩm.	Tốc độ nhanh (~100-200 tem/phút), phù hợp dây chuyền nhanh.	Khó dán trên bề mặt cong, chi phí rulo tem cao.

❖ Cơ cấu định vị sản phẩm

Băng tải tự động: Đưa sản phẩm đến vị trí dán tem.

Cảm biến quang học/phản xạ: Xác định vị trí sản phẩm để kích hoạt cơ cấu dán.

Hệ thống kẹp giữ (clamp): Cố định sản phẩm trong quá trình dán (tránh rung lắc).

2. Phần Tự Động (Xử Lý Ảnh & Điều Khiển)

❖ Hệ thống xử lý ảnh (Vision System)

- Phần cứng:
 - Sử dụng camera độ phân giải cao (1080P trở lên)
 - Hệ thống đèn LED trợ sáng đa hướng
- Phần mềm xử lý ảnh:
 - Ngôn ngữ Python với thư viện OpenCV
 - Quy trình xử lý:

1. Thu nhận ảnh (Image Acquisition):

Đọc ảnh từ nguồn (camera, video, hoặc file ảnh).

Chuẩn hóa kích thước ảnh để xử lý đồng nhất.

2. Chuyển đổi không gian màu (Color Space Conversion):

Chuyển ảnh từ hệ màu BGR (OpenCV mặc định) sang HSV để dễ tách đối tượng dựa trên màu sắc.

HSV giúp phân biệt màu tốt hơn trong điều kiện ánh sáng thay đổi.

3. Tạo Mask (Thresholding):

Áp dụng ngưỡng màu để tạo mask nhị phân (ảnh trắng-đen), trong đó:

Vùng trắng: Pixel thỏa mãn khoảng màu đích (hộp/tem).

Vùng đen: Pixel nền hoặc nhiễu.

4. Lọc nhiễu (Noise Removal):

Sử dụng phép toán hình thái học (morphology) hoặc lọc Gaussian để loại bỏ nhiễu từ mask.

Tách biệt các vùng liên thông (connected components).

5. Phát hiện biên (Edge Detection):

Dùng thuật toán Canny để phát hiện đường biên của đối tượng từ mask.

Kết quả là ảnh nhị phân chỉ chứa các đường viền.

6. Tìm Contour (Contour Detection):

Xác định các đường bao (contour) từ ảnh biên.

Lọc contour theo diện tích hoặc hình dạng để loại bỏ nhiễu (ví dụ: chỉ giữ contour có diện tích lớn hơn ngưỡng).

7. Phân tích đối tượng (Object Analysis):

Tính toán đặc trưng của contour (diện tích, tâm, hình dạng).

So sánh vị trí tâm hoặc kích thước để đưa ra quyết định (ví dụ: tem có lệch khỏi vị trí chuẩn hay không).

Vẽ contour và đánh dấu lên ảnh gốc để hiển thị kết quả.

❖ **Giải pháp AI kiểm tra chất lượng tem**

Hệ thống loại bỏ sản phẩm lỗi

Bảng 1.3 Phương pháp phân loại sản phẩm lỗi

Phương pháp	Các hoạt động	Ưu điểm	Nhược điểm
Xi lanh khí nén	Dùng xi-lanh khí nén đẩy sản phẩm ra khỏi băng tải.	Tốc độ nhanh, chi phí thấp.	Lực đẩy không ổn định.

1.2.3 Quy trình thiết kế

1. **Khảo sát yêu cầu:** Xác định tốc độ dán, loại tem, môi trường làm việc.
2. **Thiết kế cơ khí:** Cơ cấu cấp tem (tem cuộn).
3. **Hệ thống điều khiển:**
 - PLC kết nối cảm biến, động cơ.
 - Lập trình logic điều khiển.
4. **Xử lý ảnh:** Camera và thư viện OpenCV
5. **Tích hợp & tối ưu:**
 - Đồng bộ băng tải, cơ cấu dán.
 - Kiểm tra độ chính xác, tốc độ.

1.2.4 Công cụ

- Phần mềm lập trình PLC: TIA Portal
- Phần mềm vẽ 3D: SolidWorks
- Phần mềm lập trình xử lý ảnh: Pycharm (Ngôn ngữ Python)

1.2.5 Nguyên vật liệu cần để triển khai

- Băng tải
- PLC: Siemens S7 – 1200
- Xi lanh
- Động cơ
- Cảm biến
- Các thiết bị điện khác như: role, van điện từ, CB, ...

1.3 Dự kiến kết quả đạt được

1.3.1 Sản phẩm dự kiến

Mô hình hệ thống dán tem tự động tích hợp Vision hoàn chỉnh ở dạng mẫu thử (prototype), hoạt động theo quy trình kiểm tra trước và sau dán tem.

Báo cáo thiết kế chi tiết bao gồm:

- Sơ đồ nguyên lý, sơ đồ khối, sơ đồ lắp ráp cơ khí.
- Tài liệu lập trình xử lý ảnh từ Camera.
- Chương trình điều khiển hệ thống PLC.

Báo cáo đánh giá hiệu quả hoạt động:

- Tỷ lệ chính xác dán tem.
- Tỷ lệ phát hiện lỗi hộp trước khi dán.
- Tỷ lệ phát hiện lỗi dán tem.
- Năng suất trung bình (sản phẩm/giờ).

1.3.2 Phương pháp thực hiện

Nghiên cứu tổng quan các công nghệ dán tem và hệ thống kiểm tra hình ảnh (Vision).

Thiết kế phần cứng: khung máy, cấp tem, hệ thống cấp sản phẩm, cơ cấu loại sản phẩm lỗi.

Thiết kế phần mềm:

- Phát triển thuật toán kiểm tra hình ảnh bằng phần mềm OpenCV.
- Lập trình điều khiển hệ thống trên PLC Siemens S7-1200.

Lắp ráp và tích hợp hệ thống: cơ khí, điện, điều khiển, giao diện điều khiển.

Thử nghiệm và hiệu chỉnh:

- Đánh giá độ chính xác, tốc độ dán tem.
- Điều chỉnh các ngưỡng phát hiện lỗi (threshold).

1.3.3 Quy trình dự kiến

1. Xây dựng mô hình cơ khí sơ bộ (khung đỡ sản phẩm, gắn camera, bộ dán tem).
2. Lắp đặt các thiết bị điều khiển (cảm biến, động cơ, PLC).
3. Lập trình xử lý ảnh cho cảm biến và Camera (phát hiện lỗi hình dáng và lỗi dán tem).
4. Kết nối tín hiệu điều khiển toàn hệ thống.
5. Chạy thử nghiệm nội bộ, tinh chỉnh các thông số vận hành.
6. Đo lường, đánh giá kết quả hoạt động: tốc độ, tỷ lệ lỗi, độ chính xác vị trí dán.
7. Hoàn thiện báo cáo tổng kết.

1.4 Đề xuất phương pháp đánh giá và tiêu chí kiểm thử

1.4.1 Phương pháp đánh giá

Đánh giá hiệu quả vận hành:

- Thực hiện kiểm thử hệ thống với nhiều sản phẩm có tình trạng khác nhau (sản phẩm đạt và lỗi).
- Ghi nhận các chỉ số: tốc độ dán tem, tỷ lệ lỗi phát hiện đúng, tỷ lệ dán tem chính xác.

Đánh giá độ chính xác:

- So sánh vị trí tem dán thực tế với vị trí tiêu chuẩn cho phép (sai số cho phép $\pm 1\sim 2\text{mm}$).
- Kiểm tra khả năng phát hiện sản phẩm lỗi (tỷ lệ phát hiện lỗi đúng và sai).

Đánh giá chi phí:

- Tổng hợp chi phí đầu tư phần cứng, phần mềm và chi phí lắp đặt.
- Phân tích chi phí vận hành (bảo trì, tiêu hao năng lượng).

Đánh giá độ phức tạp hệ thống:

- Phân tích số lượng thiết bị, mức độ phức tạp trong lắp đặt, vận hành, lập trình.
- Đánh giá tính dễ bảo trì, dễ mở rộng hệ thống.

Đánh giá tính ổn định:

- Vận hành liên tục trong một khoảng thời gian dài (khoảng 8 giờ/ngày trong 5 ngày liên tiếp) để kiểm tra sự ổn định và độ bền.

1.4.2 Tiêu chí kiểm thử

Bảng 1.4 Bảng đánh giá tiêu chí kiểm thử

Tiêu chí	Nội dung kiểm thử	Chỉ số/kết quả mong đợi
Tốc độ dán tem	Số sản phẩm dán tem mỗi phút	≥ 15 sản phẩm/phút
Độ chính xác vị trí dán	Sai số vị trí tem so với tiêu chuẩn	$\leq \pm 2$ mm
Tỷ lệ phát hiện lỗi sản phẩm trước dán	Phát hiện móp, méo, thùng đúng	$\geq 98\%$
Tỷ lệ phát hiện lỗi sau dán	Phát hiện tem bong tróc, lệch đúng	$\geq 95\%$
Tỷ lệ sản phẩm đạt tiêu chuẩn	Tỷ lệ sản phẩm không bị lỗi sau kiểm tra	$\geq 95\%$
Chi phí đầu tư	Tổng chi phí thiết bị lắp đặt	Cạnh tranh so với thuê nhân công dán tem thủ công (giảm ít nhất 20% chi phí sau 1 năm)
Độ ổn định hệ thống	Hoạt động liên tục không lỗi trong thời gian test	$\geq 95\%$ thời gian vận hành
Mức độ tự động hóa	Khả năng tự động vận hành mà không cần can thiệp bằng tay	$\geq 90\%$ quy trình tự động

1.5 Kết hoạch thực hiện và phân công công việc

1.5.1 Kế hoạch thực hiện

Bảng 1.5 Kế hoạch thực hiện dự án

Giai đoạn	Công việc chính	Thời gian dự kiến
Giai đoạn 1: khảo sát và nghiên cứu	- Tìm hiểu công nghệ dán tem tự động, hệ thống Vision. - Xác định yêu cầu kỹ thuật.	Tuần 1-2
Giai đoạn 2: thiết kế hệ thống	- Thiết kế sơ đồ khối hệ thống.	Tuần 3

	- Chọn thiết bị: camera, cảm biến, bộ dán tem, PLC.	
Giai đoạn 3: thiết kế cơ khí và điện	- Thiết kế khung cơ khí. - Thiết kế sơ đồ đấu nối điện. - Thiết kế bố trí camera, cảm biến.	Tuần 4
Giai đoạn 4: lập trình và phát triển phần mềm	- Lập trình xử lý ảnh Vision. - Lập trình điều khiển PLC. - Lập trình HMI.	Tuần 5-7
Giai đoạn 5: lắp ráp mô hình	- Gia công cơ khí - Lắp đặt thiết bị, đấu nối điện.	Tuần 8-9
Giai đoạn 6: kiểm thử và tinh chỉnh	- Chạy thử nghiệm sản phẩm. - Đánh giá hiệu suất. - Tinh chỉnh camera, cảm biến, cơ cấu dán tem.	Tuần 10-11
Giai đoạn 7: tổng kết và hoàn thiện	- Viết báo cáo tổng kết.	Tuần 12-13

Bảng 1.6 Phân công công việc

Thành viên	Công việc phụ trách	Thời gian dự kiến	Ghi chú
Lê Đức Long	- Điều phối dự án, theo dõi tiến độ chung. - Thiết kế hệ thống tổng thể (sơ đồ khối, bố trí thiết bị). - Lập trình điều khiển PLC trên phần mềm Tia Portal.	Toàn thời gian dự án	Kiểm vai trò điều phối
Nguyễn Huy Ngân	- Thiết kế và gia công phần cơ khí (khung máy, giá đỡ camera, băng tải). - Lắp đặt thiết bị và đấu nối điện (motor, cảm biến, camera, bộ dán tem). - Hỗ trợ kiểm thử và hiệu chỉnh.	Tuần 3-9	- Kiểm lắp ráp và bảo trì. - Hỗ trợ về phần thuật toán xử lý ảnh.
Phan Ngọc Nam	- Phát triển thuật toán xử lý ảnh cho hệ thống. - thiết kế giao diện giám sát và điều khiển. - Hiệu chỉnh camera, ánh sáng. - Chuẩn bị sản phẩm test, kiểm thử hệ thống, ghi log dữ liệu.	Tuần 5-13	Kiểm phần thống kê kết quả.

	- Hỗ trợ viết báo cáo tổng kết.		
--	---------------------------------	--	--

1.6 Kết luận

Trong chương này, nhóm đã nêu ra được tính cấp thiết của đề tài, trình bày chi tiết quá trình xây dựng giải pháp thực hiện dự án mô hình dán tem tự động cho xưởng sản xuất dây sọc. Giải pháp đề xuất được xây dựng trên cơ sở đánh giá hiện trạng sản xuất thực tế, nhận diện các bất cập của phương pháp dán tem thủ công, từ đó phát triển một hệ thống tích hợp tự động hóa và xử lý ảnh nhằm nâng cao hiệu quả và chất lượng công đoạn dán tem.

Giải pháp thiết kế bao gồm ba phần chính: cơ cấu cơ khí dán tem, hệ thống xử lý ảnh sử dụng camera kiểm tra trước và sau dán, và phần mềm điều khiển tích hợp giao diện giám sát. Các thiết bị được lựa chọn đảm bảo tính khả thi cao trong điều kiện thực tế sản xuất, đồng thời phần mềm được thiết kế với tiêu chí thân thiện với người dùng, dễ vận hành và bảo trì.

Tổng thể giải pháp mang tính liên kết chặt chẽ với các công đoạn trước và sau trong dây chuyền sản xuất, bảo đảm tính liên tục và không làm gián đoạn chu trình đóng gói. Ngoài ra, khả năng mở rộng và cải tiến của mô hình cũng được tính đến, cho phép tích hợp thêm tính năng truy xuất dữ liệu, kết nối hệ thống SCADA hoặc MES trong tương lai.

Việc hoàn thiện giải pháp không chỉ giúp tối ưu hóa công đoạn dán tem mà còn là bước tiến quan trọng trong lộ trình chuyển đổi sang sản xuất thông minh của doanh nghiệp, góp phần giảm chi phí, tăng năng suất và nâng cao năng lực cạnh tranh.

1.7 Mục tiêu đề tài

Đề tài nghiên cứu "**Cải tiến hệ thống dán tem của phân xưởng sản xuất dây sọc**" nhằm giải quyết các hạn chế của phương pháp dán tem thủ công bằng cách phát triển một hệ thống tự động hóa toàn diện, tích hợp vào dây chuyền sản xuất hiện có. Mục tiêu tổng quát là nâng cao năng suất, giảm sai sót và tiết kiệm chi phí nhân công thông qua việc thiết kế một hệ thống dán tem chính xác, tốc độ cao (≥ 20 tem/phút) và độ tin cậy $\geq 95\%$. Cụ thể, hệ thống sẽ bao gồm cơ cấu cấp tem tự động, bộ định vị sản phẩm và cơ chế dán tem chính xác, đồng thời tích hợp công nghệ xử lý ảnh (Vision) để kiểm tra chất lượng tem, phát hiện lỗi và tự động loại bỏ sản phẩm không đạt yêu cầu. Bên cạnh đó, hệ thống được tối ưu để có chi phí chế tạo dưới 50 triệu đồng, phù hợp với doanh nghiệp vừa và nhỏ tại Việt Nam, với giao diện điều khiển thân thiện, dễ vận hành

và khả năng điều chỉnh linh hoạt cho nhiều loại tem và bề mặt sản phẩm khác nhau. Đối tượng áp dụng chính là các doanh nghiệp sản xuất thực phẩm, dược phẩm, điện tử và hàng tiêu dùng, với phạm vi nghiên cứu tập trung vào tem dạng cuộn trên sản phẩm có bề mặt phẳng. Kết quả dự kiến sẽ giúp giảm 30–50% chi phí nhân công, nâng cao chất lượng sản phẩm và góp phần thúc đẩy xu hướng tự động hóa trong sản xuất công nghiệp tại Việt Nam.

CHƯƠNG 2. GIẢI PHÁP THỰC HIỆN CƠ CẤU DÁN TEM

2.1 Đề xuất giải pháp cải tiến máy dán tem

Trong thực tế sản xuất hiện nay, công đoạn dán tem cho sản phẩm – cụ thể là hộp dây sọc – thường được thực hiện thủ công, dẫn đến nhiều hạn chế về mặt năng suất, độ chính xác và tính ổn định. Việc dán sai vị trí, nhãn bị bong tróc hoặc thiếu tem không chỉ ảnh hưởng đến chất lượng sản phẩm, mà còn làm giảm uy tín thương hiệu và tăng chi phí nhân lực, tái kiểm tra. Trước thực trạng đó, nhóm đề xuất **cải tiến hệ thống dán tem của phân xưởng sản xuất** nhằm thay thế hoàn toàn công đoạn thủ công, hướng tới sản xuất thông minh, ổn định và có khả năng kiểm soát chất lượng theo thời gian thực.

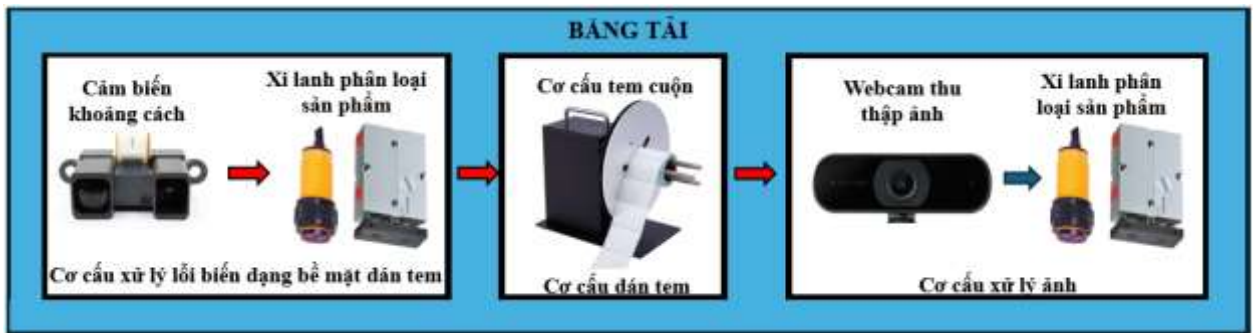
Giải pháp được xây dựng dưới dạng mô hình tích hợp các thành phần cơ điện tử như bộ điều khiển PLC, cơ cấu truyền động, camera công nghiệp và phần mềm xử lý ảnh. Trong đó, hệ thống sẽ đảm nhận việc kiểm tra tình trạng sản phẩm trước khi dán (phát hiện lỗi như móp, méo, sai hình dạng), xác định vị trí chính xác để dán tem, thực hiện thao tác dán và kiểm tra lại kết quả sau khi dán (xem tem có đúng vị trí, bị lệch hay bong tróc không). Toàn bộ quá trình diễn ra một cách tự động, đảm bảo tốc độ đồng bộ với các khâu khác trong dây chuyền.

Giải pháp này không hoạt động riêng lẻ mà được thiết kế **liên kết chặt chẽ với hai công đoạn liền kề trong dây chuyền**:

- **Khâu trước (đóng hộp):** Hệ thống dán tem nhận tín hiệu và sản phẩm đầu vào từ khâu đóng gói, kiểm tra hình dạng hộp và xác nhận tính hợp lệ để tiếp tục.
- **Khâu sau (kiểm tra cuối và đóng thùng):** Dữ liệu kết quả dán tem được truyền sang bước tiếp theo để phục vụ truy xuất nguồn gốc, quản lý chất lượng hoặc loại bỏ những sản phẩm bị lỗi nếu có.

Với giải pháp này, nhóm kỳ vọng không chỉ nâng cao hiệu quả lao động, giảm sai sót và chi phí, mà còn góp phần xây dựng mô hình sản xuất hiện đại, sẵn sàng tích hợp vào hệ thống MES/SCADA và hướng tới chuyển đổi số trong công nghiệp 4.0.

2.2 Quy trình công nghệ của hệ thống



Hình 2.1 Quy trình công nghệ của hệ thống

Quy Trình Vận Hành Chi Tiết Máy Dán Tem Tự Động:

❖ Khâu I xử lý bề mặt dán tem:

Bước 1: Hộp sản phẩm được đặt lên băng chuyền tải để đưa đến khâu I xử lý bề mặt, điều chỉnh tốc độ phù hợp (thường 0.2–0.5 m/s). Băng chuyền tải được thiết kế có gờ chắn đảm bảo hộp sản phẩm không bị lệch/lật trong quá trình di chuyển.

Bước 2: Ở đây sản phẩm sẽ đi qua cảm biến khoảng cách hồng ngoại để thực hiện quá trình quét bề mặt nhằm phát hiện các lỗi biến dạng bề mặt dán tem như móp, méo. Cảm biến được cấu hình với một ngưỡng giá trị khoảng cách cho phép, nếu giá trị đo được khi sản phẩm đi qua nằm trong ngưỡng này, sản phẩm sẽ được đưa đến bước xử lý tiếp theo. Ngược lại, nếu giá trị vượt ngoài khoảng giới hạn, hệ thống sẽ cho sản phẩm tiếp tục di chuyển đến vị trí cảm biến khoảng cách vật cản. Tại đây, cảm biến khoảng cách vật cản phát hiện tín hiệu của sản phẩm lỗi và gửi tín hiệu về PLC. Bộ điều khiển sẽ kích hoạt xi lanh để thực hiện loại bỏ sản phẩm lỗi ra khỏi băng chuyền và tăng bộ đếm “Số lượng hộp biến dạng”



Hình 2.2 Cơ cấu phát hiện biến dạng bề mặt dán tem

❖ **Khâu II dán tem.**

Khi sản phẩm di chuyển đến vị trí cơ cấu dán tem, nó sẽ đi qua cảm biến khoảng cách vật cản. Ngay khi cảm biến phát hiện có sản phẩm, tín hiệu sẽ được gửi về PLC để điều khiển van điện từ kích hoạt xi lanh thực hiện thao tác dán tem. Xi lanh sẽ đẩy tem ép lên bề mặt sản phẩm, và sau khi quá trình dán hoàn tất, xi lanh sẽ trở về vị trí ban đầu. Đồng thời, cảm biến quang khuếch tán E3Z-D61 sẽ kiểm tra xem cuộn tem còn tem hay không. Nếu phát hiện thiếu tem, PLC sẽ kích hoạt động cơ kéo tem để cuộn tem tiến lên cho đến khi cảm biến E3Z-D61 nhận thấy đã có tem mới ở đúng vị trí. Khi đó, một chu trình dán tem được xem là hoàn tất và hệ thống sẵn sàng xử lý sản phẩm tiếp theo.

Cơ cấu tem cuộn



Hình 2.3 Cơ cấu dán tem

❖ **Khâu III xử lý ảnh.**

Ở bước xử lý ảnh, sản phẩm sẽ đi qua camera công nghiệp để thu thập hình ảnh, sau đó dữ liệu ảnh sẽ được truyền về bộ xử lý trung tâm, nơi áp dụng các thuật toán xử lý ảnh (sử dụng thư viện OpenCV) nhằm phân loại giữa sản phẩm lỗi và sản phẩm đạt yêu cầu. Nếu hình ảnh cho thấy sản phẩm bị lỗi, chẳng hạn như tem bị dán lệch vị trí, hệ thống sẽ cho phép sản phẩm tiếp tục di chuyển đến vị trí cảm biến vật cản. Khi cảm biến khoảng cách vật cản phát hiện có hộp sản phẩm, tín hiệu sẽ được gửi về PLC để điều khiển van điện từ kích hoạt xi lanh đẩy sản phẩm lỗi ra khỏi băng chuyền, đồng thời giá trị của bộ đếm “Số lượng hộp sai tem” sẽ được tăng thêm một đơn vị. Ngược lại, nếu kết quả xử lý ảnh cho thấy tem đã được dán chính xác, sản phẩm sẽ tiếp tục đi qua các công đoạn tiếp theo trong dây chuyền, và bộ đếm “Số lượng hộp đạt” cũng được tăng lên một đơn vị và tiếp tục sang các công đoạn phía sau.



Hình 2.4 Cơ cấu xử lý ảnh

Tiêu chí tem đạt chuẩn:

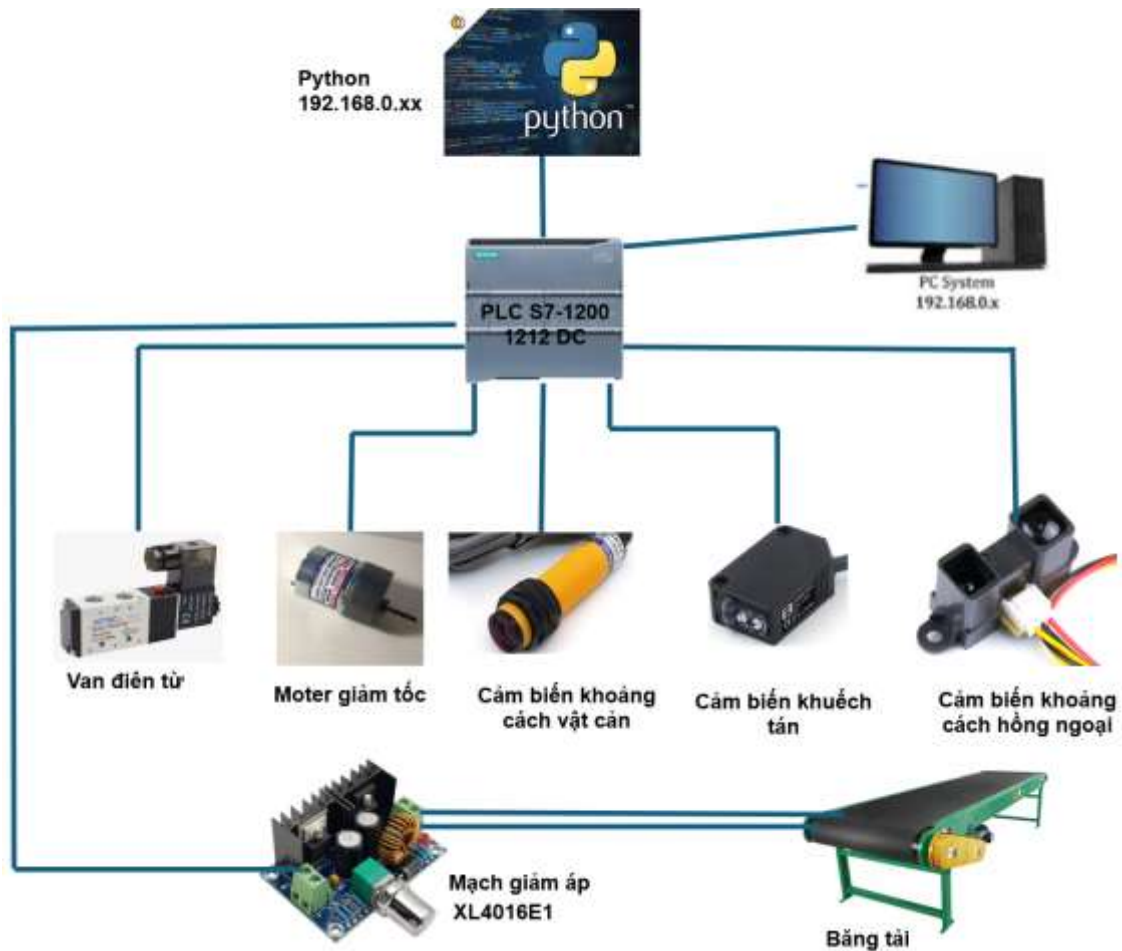
- Tem có đủ 4 cạnh.
- Độ lệch tâm tem so với hộp $\leq 5\%$ kích thước hộp.
- Diện tích tem nằm trong ngưỡng cho phép (90–110% diện tích chuẩn).

2.3 Lựa chọn thiết bị trong mô hình

2.3.1 Tổng quan về các thiết bị

Đề tài mô hình “Cải tiến hệ thống dán tem của phân xưởng sản xuất dây sạc” ra đời nhằm giải quyết nhu cầu trên, với khả năng thực hiện hai chức năng chính: **tự động dán tem** lên sản phẩm (cụ thể là hộp dây sạc trong xưởng sản xuất) và **kiểm tra hình ảnh** bằng cảm biến và camera công nghiệp trước và sau khi dán. Hệ thống này giúp phát hiện lỗi sản phẩm trước khi dán (như móp, thủng, sai hình dạng), đồng thời đảm bảo tem được dán đúng vị trí, không bị bong tróc hay sai lệch sau khi hoàn tất.

Để hiện thực hóa mô hình, hệ thống bao gồm nhiều thành phần thiết bị liên quan đến các lĩnh vực: cơ khí – truyền động, điều khiển lập trình (PLC), xử lý ảnh (cảm biến và camera công nghiệp, đèn chiếu sáng, phần mềm OpenCV), giao diện người – máy, và các thiết bị phụ trợ khác. Mỗi nhóm thiết bị đóng một vai trò quan trọng, góp phần tạo nên một dây chuyền hoạt động khép kín, thông minh và hiệu quả. Sơ đồ kết nối các thiết bị được thể hiện như hình 2.2.



Hình 2.5 Sơ đồ kết nối thiết bị trong hệ thống

2.3.2 Thiết bị trong mô hình dán tem

Mô hình dán tem tự động được thiết kế rất chi tiết và đảm bảo về mặt công nghệ, giúp cho việc thao tác vô cùng dễ dàng. Ứng dụng cụ thể vào xưởng sản xuất trong công nghiệp.

Các thành phần trong mô hình được liệt kê ở bảng sau:

Bảng 2.1 Các thiết bị trong mô hình

STT	Tên thiết bị	Mô tả	SL
1	PLC Seimens S7-1200 (CPU 1212 DC)	Đầu vào 24VDC,	1
2	Webcam 1080P	Nguồn 5VDC-1A, độ phân giải 1920 × 1080, góc quay 130 độ.	1
3	Mạch giảm áp XL4016E1	Đầu vào 4-40VDC, đầu ra 1.25-36VDC.	1

4	Động cơ giảm tốc GB37-550	Đầu vào 12VDC, 53rpm, trục 6mm.	1
5	Moter giảm tốc	Đầu vào 12VDC, không tải trọng hiện tại 6-12r/min	1
6	Cảm biến khoảng cách vật cản E3F-DS10C	Đầu vào 6-36VDC, 3-10cm, NPN-NO	3
7	Cảm biến khuếch tán E3Z-D61	Đầu vào 12-24VDC, khoảng dò 5-100mm NPN	1
8	Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F	Đầu vào 4.5-5.5VDC, đo khoảng cách 2-15 cm	1
9	Van điện từ LAIZE 4V210-08	Đầu vào 24VDC, loại 5/2	3
10	Xi lanh hai Ty TN 10	Xi lanh dạng khí nén	3

2.4 Chức năng và thông số của các thiết bị trong mô hình

2.4.1 PLC Siemens S7-1200

Phân tích, đánh giá S7 - 1200 cấp nhỏ (Hãng Siemen):

Bảng 2.2 Phân tích, đánh giá S7-1200 cấp nhỏ (Hãng Siemen):

Tiêu chí	CPU 1212C	CPU 1214C
Mã sản phẩm tiêu biểu	6ES7 212-1AE40-0XB0	6ES7 214-1AG40-0XB0
Bộ nhớ chương trình	75 KB	100 KB
Bộ nhớ dữ liệu	50 KB	50 KB
Ngõ vào số (DI)	8 DI	14 DI
Ngõ ra số (DO)	6 DO	10 DO
Ngõ vào tương tự (AI)	2 AI	2 AI
Số module mở rộng I/O hỗ trợ	2 module mở rộng	8 module mở rộng
Module truyền thông (CM)	RS485/RS232/Profibus/Profinet (gắn rời)	Tối đa 3 module CM
Tốc độ xử lý	~0.08 μ s/operation	~0.08 μ s/operation
Cổng truyền thông	1 cổng PROFINET (Ethernet RJ45) tích hợp	1 cổng PROFINET (RJ45)
Nguồn cấp	24 VDC	24 VDC hoặc 120/230 VAC
Ứng dụng phù hợp	Hệ thống đơn giản, ít I/O	Hệ thống vừa, nhiều thiết bị vào ra

Giá tham khảo	Thấp hơn 1214C (phù hợp ngân sách trung bình)	Cao hơn, nhưng hiệu năng mạnh hơn
---------------	---	-----------------------------------

Lý do chọn CPU 1212C:

Bảng 2.3 Lý do chọn CPU 1212C

Lý do	Giải thích chi tiết
Chi phí hợp lý	Giá thành cạnh tranh so với các dòng PLC lớn hơn như CPU 1214C hay S7-1500
Đầy đủ chức năng cho ứng dụng nhỏ	Tích hợp sẵn 8 DI, 6 DO, 2 AI – phù hợp điều khiển máy đơn giản mà không cần mở rộng
Lập trình bằng TIA Portal	Tận dụng môi trường phát triển TIA Portal mạnh mẽ, dễ học, dễ bảo trì
Có cổng PROFINET tích hợp	Giao tiếp Ethernet với HMI/SCADA, kết nối mạng công nghiệp mà không cần module phụ
Hỗ trợ PID điều khiển tuyến tính	Có khối hàm PID Compact tích hợp sẵn – điều khiển nhiệt độ, áp suất, mức...
Tích hợp Web Server nội bộ	Cho phép giám sát và điều khiển từ xa qua trình duyệt Web mà không cần phần mềm riêng
Mở rộng module khi cần thiết	Hỗ trợ mở rộng thêm module I/O hoặc module truyền thông khi dự án phát triển
Tính ổn định và độ tin cậy cao	Siemens là thương hiệu hàng đầu trong ngành tự động hóa công nghiệp
Kích thước nhỏ gọn	Phù hợp lắp trong các tủ điều khiển hạn chế về không gian
Hỗ trợ kỹ thuật và tài liệu đầy đủ	Siemens cung cấp tài liệu chi tiết, cộng đồng hỗ trợ rộng lớn, dễ tìm giải pháp



Hình 2.6 Bộ điều khiển PLC-S1200 CPU 1212 DC

Chức năng:

Bảng 2.4 Chức năng bộ điều khiển PLC-S1200 CPU 1212 DC

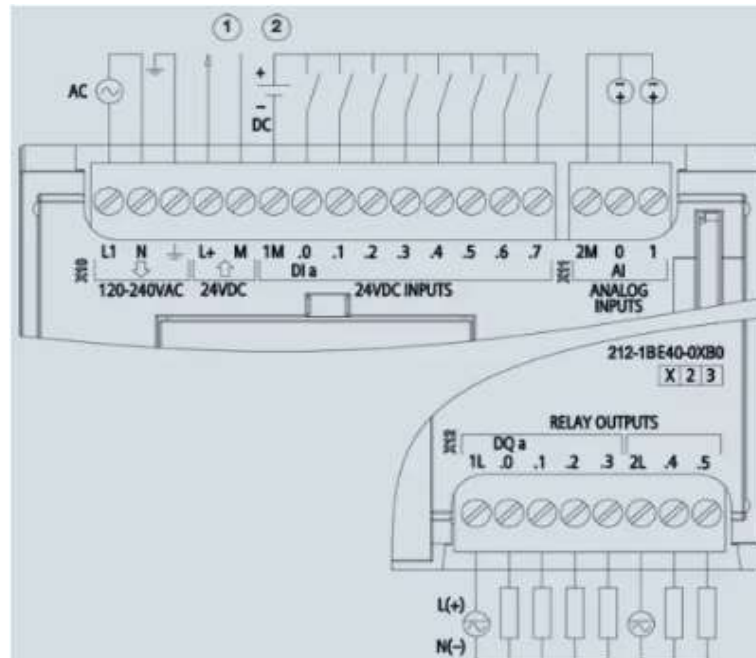
Chức năng	Mô tả
Điều khiển logic	Hỗ trợ mạnh mẽ các lệnh logic, số học, so sánh, điều kiện, vòng lặp, timer, counter...
Lập trình linh hoạt	Lập trình bằng TIA Portal với các ngôn ngữ: LAD, FBD, STL, SCL, Graph
Truyền thông mạng	Tích hợp sẵn cổng PROFINET (Ethernet) – hỗ trợ truyền thông với HMI, SCADA, PLC khác
Web server tích hợp	Truy cập, giám sát và điều khiển PLC qua trình duyệt web.
Tích hợp bảo mật	Có tính năng bảo vệ truy cập (password), phân quyền người dùng, bảo vệ mã chương trình

Thông số kỹ thuật PLC – S1200 CPU1212:

Bảng 2.5 Thông số kỹ thuật PLC – S1200 CPU1212

Thông số kỹ thuật	SIMATIC S7-1200, CPU 1212C, COMPACT CPU, DC/DC/DC, ONBOARD I/O: 6 DI 24V DC; 4 DO 24 V DC; 2 AI 0 – 10V DC, POWER SUPPLY: DC 20.4 – 28.8 V DC, PROGRAM/DATA MEMORY: 50 KB
Kích thước	90 mm (Rộng) x 100 mm (Cao) x 75 mm (Sâu)
Khối lượng	Khoảng 310 g

Đầu nối dây bộ điều khiển PLC-S1200 CPU 1212C:



Hình 2.7 Đầu nối dây bộ điều khiển PLC-S1200 CPU 1212C.

Giải thích sơ đồ đầu nối:

❖ Nguồn cấp điện (Power Supply):

- AC (L1, N, GND): Đây là các cực để kết nối nguồn điện xoay chiều (AC) chính cho PLC.
 - L1: Nối dây pha (Live/Phase) của nguồn điện 120-240VAC.
 - N: Nối dây trung tính (Neutral) của nguồn điện 120-240VAC.
 - GND: Đây là cực nối đất bảo vệ bộ điều khiển PLC.
- DC (L+, M): PLC Siemens S7-1200 CPU 1212C có khả năng cung cấp một nguồn điện một chiều (DC) 24V nội bộ để cấp cho các cảm biến đầu vào.
 - L+: Ngõ ra 24VDC dương.
 - M: Ngõ ra 0VDC âm (Mass/Ground).

❖ Đầu vào số (24VDC Inputs - DI):

- 1M: Đây là điểm chung (Common) cho các đầu vào số. Thông thường, nó được nối với cực 0VDC (M) của nguồn 24VDC nội bộ hoặc nguồn 24VDC bên ngoài.
- 0, 1, 2, 3, 4, 5, 6, 7: Đây là các kênh đầu vào số riêng lẻ (từ DI 0 đến DI 7).

❖ Đầu vào tương tự (AI - Analog Inputs):

- 2M: Đây là điểm chung (Common) cho các đầu vào tương tự. Cực 0V của cảm biến tương tự thường được nối vào đây.

- 0, 1: Đây là hai kênh đầu vào tương tự riêng lẻ (AI 0 và AI 1). Bạn sẽ kết nối dây tín hiệu từ cảm biến tương tự vào các cực này.
- ❖ Đầu ra rơ-le (DQ a - Relay Outputs):
- 1L: Đây là cực chung (Common) cho nhóm các đầu ra rơ-le đầu tiên (.0, .1, .2, .3). Bạn sẽ kết nối một dây nguồn (dây pha AC hoặc cực dương DC) của nguồn cấp cho tải bên ngoài vào đây. (.0, .1, .2, .3) là các tiếp điểm rơ-le riêng lẻ. Khi đầu ra tương ứng của PLC được kích hoạt, rơ-le bên trong sẽ đóng, nối cực 1L với các cực này, từ đó cấp nguồn cho tải mà bạn đã đấu nối.
 - 2L: Đây là cực chung cho nhóm các đầu ra rơ-le thứ hai (.4, .5). Tương tự như 1L, bạn sẽ kết nối nguồn cấp cho tải bên ngoài vào đây. (.4, .5) là các tiếp điểm rơ-le riêng lẻ. Khi kích hoạt, 2L sẽ được nối với các cực này.

2.4.2 Webcam 1080P

Webcam 1080P là một loại camera kỹ thuật số có khả năng ghi hình với độ phân giải Full HD (1920 x 1080 pixels). Thiết bị này thường được kết nối với máy tính hoặc bộ xử lý trung tâm thông qua cổng USB, và được sử dụng rộng rãi trong các ứng dụng như gọi video, giám sát, cũng như xử lý hình ảnh trong công nghiệp và học thuật.

- ❖ Cấu tạo của webcam 1080P:



Hình 2.8 Webcam 1080P

Chú thích:

1	Micro âm thanh nội tích hợp
2	Đèn báo
3	Đế chống trượt đa chức năng có thể điều chỉnh
4	Ống kính AF HD
5	Cổng kết nối USB
6	Giao diện giá ba chân (tripod)

❖ Nguyên lý hoạt động của webcam 1080P:

- Ánh sáng từ cảnh vật đi qua ống kính và tập trung vào cảm biến hình ảnh.
- Cảm biến chuyển đổi ánh sáng thành tín hiệu điện thô.
- Bộ xử lý tín hiệu hình ảnh (ISP) xử lý tín hiệu thô, thực hiện các hiệu chỉnh (cân bằng trắng, phơi sáng, khử nhiễu) và nén dữ liệu video.
- Microphone thu âm thanh, chuyển đổi thành tín hiệu điện và số hóa.
- Dữ liệu video và âm thanh (đã được nén) được truyền đến máy tính thông qua cáp USB.
- Máy tính nhận dữ liệu, giải nén và hiển thị trên màn hình, phát âm thanh qua loa.
- Đèn báo cho biết webcam đang hoạt động.
- Đế kẹp giúp định vị webcam.

Chức năng:

Webcam 1080P được sử dụng để thu nhận hình ảnh sản phẩm trên băng tải nhằm phục vụ cho các tác vụ xử lý ảnh như nhận diện vị trí, kiểm tra hình dạng. Với độ phân giải cao và khả năng hoạt động thời gian thực (real-time), webcam 1080P là lựa chọn phù hợp cho các hệ thống xử lý ảnh đơn giản đến trung bình.

- Ghi nhận hình ảnh sản phẩm đang di chuyển trên băng tải.
- Truyền hình ảnh theo thời gian thực về vi xử lý (ví dụ: máy tính) để phân tích, xử lý hình ảnh như: xác định vị trí tem, phát hiện đối tượng, nhận dạng sản phẩm.
- Hỗ trợ các thao tác kiểm tra chất lượng, vị trí dán nhãn, hoặc làm cơ sở để kích hoạt cơ cấu gắn tem tự động.

Thông số kỹ thuật:

Bảng 2.6 Thông số kỹ thuật webcam 1080P

Thông số	Giá trị
Độ phân giải	Full HD 1080p (1920x1080 pixel)
Tốc độ khung hình	30 fps (khung hình/giây)
Góc nhìn	70–90 độ (FOV)
Cảm biến hình ảnh	CMOS
Giao tiếp	USB 2.0 hoặc 3.0
Tự động lấy nét	Có hoặc không (tùy loại)
Chiều dài cáp	Khoảng 1.5 mét
Tương thích	Windows, Linux, Raspberry Pi, v.v.

2.4.3 Mạch giảm áp XL4016E1

XL4016E1 là mạch chuyển đổi DC / DC với tần số cố định 180 kHz PWM, bộ mạch dễ sử dụng, hoạt động ổn định, dải điện áp đầu vào rộng từ 4 đến 40VDC. Ngoài ra mạch còn có chức năng bảo vệ quá dòng được tích hợp bên trong. Khi chức năng bảo vệ ngắt xảy ra, tần số hoạt động sẽ giảm từ 180KHz xuống 48KHz.

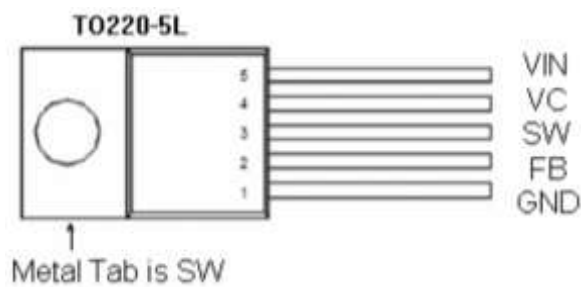
Chức năng:

- ❖ Cung cấp điện áp ổn định và phù hợp cho động cơ GB37-550:
 - Động cơ GB37-550 có điện áp hoạt động lý tưởng khoảng 6V–12V.
 - Hệ thống cấp nguồn tổng thường có điện áp cao hơn (ví dụ: 24V hoặc 36V), nên cần mạch giảm áp DC-DC để đưa điện áp về mức phù hợp với động cơ.
 - XL4016E1 thực hiện chức năng này bằng cách hạ áp đầu ra xuống mức chính xác (12V) giúp động cơ hoạt động ổn định, tránh quá áp và cháy động cơ.
- ❖ Điều chỉnh tốc độ di chuyển của băng tải 90cm:
 - Trong hệ thống dán tem, tốc độ của băng tải phải phù hợp với tốc độ dán tem để đảm bảo độ chính xác.
 - XL4016E1 cho phép tinh chỉnh điện áp đầu ra bằng biến trở, giúp thay đổi tốc độ quay của động cơ GB37-550, từ đó điều chỉnh tốc độ di chuyển của băng tải.
- ❖ Giảm tải cho hệ thống điều khiển:
 - Việc sử dụng mạch XL4016E1 giúp tách riêng phần nguồn cho động cơ, tránh gây nhiễu điện hoặc sụt áp cho các mạch điều khiển khác như PLC.
 - Đảm bảo hệ thống hoạt động ổn định, tránh lỗi hoặc treo mạch do dòng khởi động cao từ động cơ.
- ❖ Tăng tuổi thọ động cơ và hệ thống:
 - Mạch giảm áp giúp duy trì điện áp ổn định, hạn chế quá nhiệt, tiếng ồn, và rung động do điện áp không phù hợp.
 - Qua đó, tăng độ bền của động cơ GB37-550 và giúp hệ thống vận hành liên tục, bền bỉ hơn trong dây chuyền dán tem.



Hình 2.9 Mạch giảm áp XL4016E1

Cấu hình chân cắm:

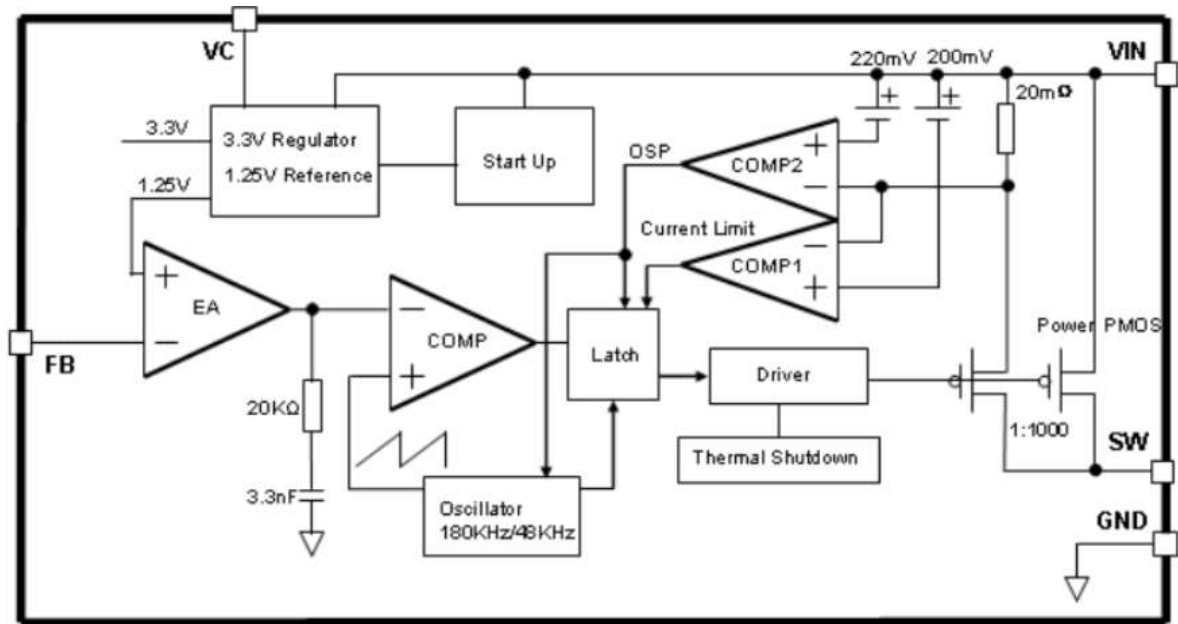


Hình 2.10 cấu hình chân cắm của Mạch giảm áp XL4016E1

Chú thích:

Tên chân	Mô tả
Vin	Đầu vào điện áp cung cấp (Supply Voltage Input)
VC	Tụ điện bỏ qua bộ điều chỉnh điện áp nội bộ (Internal Voltage Regulator Bypass Capacity)
SW	Chân đầu ra công tắc nguồn (Power Switch Output Pin)
FB	Chân phản hồi (Feedback Pin)
GND	Chân nối đất (Ground Pin)

Sơ đồ khối chức năng:



Hình 2.11 Sơ đồ khối chức năng của Mạch giảm áp XL4016E1

❖ Nguyên lý hoạt động sơ đồ khối:

- Cấp nguồn: Khi VIN được cấp nguồn, mạch Start Up sẽ hoạt động, và bộ điều chỉnh 3.3V bắt đầu cung cấp điện cho các mạch nội bộ.
- Khởi tạo dao động: Bộ tạo dao động Oscillator tạo ra tín hiệu răng cưa.
- Vòng phản hồi:
 - Điện áp đầu ra VOUT được chia bởi mạch phản hồi và đưa vào chân FB.
 - EA so sánh FB với 1.25V tham chiếu. Đầu ra của EA (tín hiệu lỗi) tỷ lệ với sự chênh lệch.
 - COMP so sánh tín hiệu lỗi từ EA với tín hiệu răng cưa từ Oscillator để tạo ra tín hiệu PWM.
 - Tín hiệu PWM được đưa qua Latch và Driver để điều khiển Mosfet công suất.
- Chuyển mạch: Mosfet bật/tắt liên tục, điều này cùng với cuộn cảm và tụ điện bên ngoài sẽ làm mịn điện áp đầu ra, tạo ra điện áp DC ổn định ở ngõ ra.
- Bảo vệ: Mạch giới hạn dòng và bảo vệ quá nhiệt liên tục giám sát hoạt động của IC để đảm bảo an toàn và độ tin cậy.

Thông số kỹ thuật:

Bảng 2.7 Thông số kỹ thuật của mạch giảm áp XL4016E1

Thông số	Giá trị tiêu chuẩn
----------	--------------------

IC chính	XL4016E1 – Buck Converter Controller
Điện áp đầu vào (Vin)	5V – 40V DC
Điện áp đầu ra (Vout)	1.25V – 36V DC (có thể điều chỉnh)
Dòng đầu ra tối đa	8A (với tản nhiệt tốt)
Điều chỉnh điện áp	Bằng biến trở (chiết áp)
Kích thước phổ biến	~65mm x 47mm x 25mm (thay đổi theo module)

2.4.4 Động cơ giảm tốc GB37-550 /12V DC

Động cơ giảm tốc GB37-550/12V DC đạt được sự cân bằng tuyệt vời về tốc độ và mô-men xoắn phù hợp, kích thước nhỏ gọn và sự phù hợp cho các ứng dụng tự động hóa quy mô nhỏ, do đó trở thành lựa chọn rất hợp lý và hiệu quả cho băng tải mini mà nhóm chúng em đang thực hiện trong hệ thống giám tem tự động.

Chức năng:

Động cơ giảm tốc GB37-550 (12V) có chức năng điều khiển tốc độ cho hệ thống băng tải. Động cơ đáp ứng cho hệ thống yêu ổn định, mô-men xoắn đủ lớn và tốc độ phù hợp để duy trì hoạt động liên tục, chính xác và đồng bộ với các thiết bị khác như cảm biến, bộ gián tem, v.v.

Động cơ giảm tốc GB37-550 là loại động cơ DC chổi than, được tích hợp sẵn hộp số giảm tốc bằng kim loại để hạ tốc độ quay và tăng mô-men xoắn. Đây là lựa chọn phổ biến trong các ứng dụng robot, mô hình cơ khí, hệ thống tự động hóa nhỏ, xe tự hành và băng tải mini.

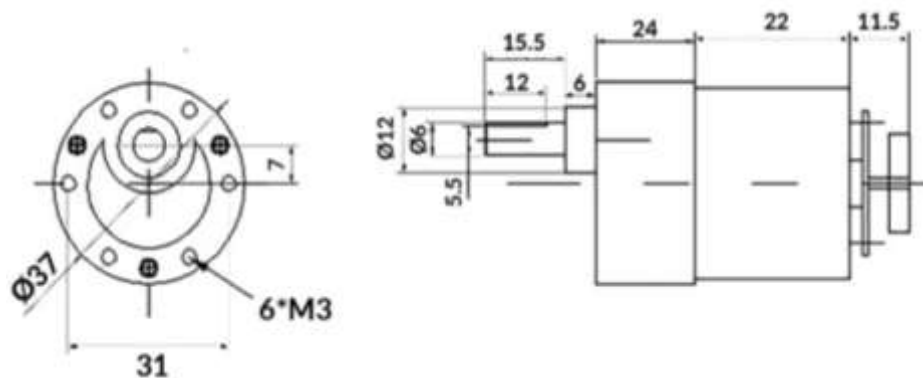


Hình 2.12 Động cơ giảm tốc GB37-550

❖ Cấu tạo cơ bản:

- Động cơ DC 550 (12V):
 - Là loại động cơ điện một chiều chổi than.

- Khi cấp điện 12V DC, roto quay nhanh (thường hàng nghìn vòng/phút).
- Hộp số giảm tốc (gearbox):
 - Gắn liền với trục động cơ.
 - Gồm nhiều bánh răng giúp giảm tốc độ quay nhưng tăng mô-men xoắn (lực quay).
 - Có nhiều tỷ số truyền khác nhau (ví dụ: 1:30, 1:50, 1:100,...).
- ❖ Nguyên lý hoạt động:
 - Khi cấp nguồn 12V DC:
 - Rotor trong động cơ quay rất nhanh.
 - Chuyển động này được truyền qua hộp số giảm tốc.
 - Tại trục đầu ra của hộp số, tốc độ quay giảm nhưng lực xoắn tăng lên đáng kể.
 - Nhờ vậy, động cơ có thể kéo/tải vật nặng, dù quay chậm.
 - Đảo chiều dòng điện → đảo chiều quay của trục động cơ.
- ❖ Sơ đồ kích thước:



Hình 2.13 sơ đồ kích thước động cơ giảm tốc GB37-550

Thông số kỹ thuật:

Bảng 2.8 Thông số kỹ thuật Động cơ giảm tốc GB37-550

Thông số	Giá trị
Điện áp hoạt động:	12V DC
Dòng tối đa: 3A	3A
Đường kính trục:	6mm (hình D)
Đường kính thân động cơ:	37.3 mm
Chiều dài động cơ:	57.5 mm
Chiều dài trục:	15.5 mm
Chất liệu hộp số:	Kim loại

Tỉ số truyền:	1:56
Tốc độ sau hộp số giảm tốc (trục chính):	$10.000 / 56 = 178\text{RPM}$
Momen xoắn định mức:	6.5 Kg.cm
Momen xoắn cực đại:	9.0 Kg.cm
Trọng lượng:	155g

2.4.5 Motor giảm tốc

Motor giảm tốc 24V DC là một loại động cơ điện một chiều (DC) tích hợp với bộ giảm tốc nhằm giảm tốc độ quay và tăng mô-men xoắn. Loại motor này thường được sử dụng trong các hệ thống cơ khí cần chuyển động chính xác và ổn định, như cơ cấu gián tem tự động.

Chức năng:

Motor giảm tốc 24V DC đảm nhiệm vai trò quan trọng trong việc điều khiển chuyển động của cơ cấu hạ tem từ trên xuống sản phẩm. Cụ thể:

- Điều khiển trực lặn: Động cơ quay chậm, ổn định giúp hạ tem đúng vị trí với lực vừa đủ.
- Độ chính xác cao: Giúp đồng bộ với cảm biến vị trí sản phẩm trên băng chuyền.
- Tạo chuyển động mượt: Tránh rung lắc hay sai lệch tem khi dán.

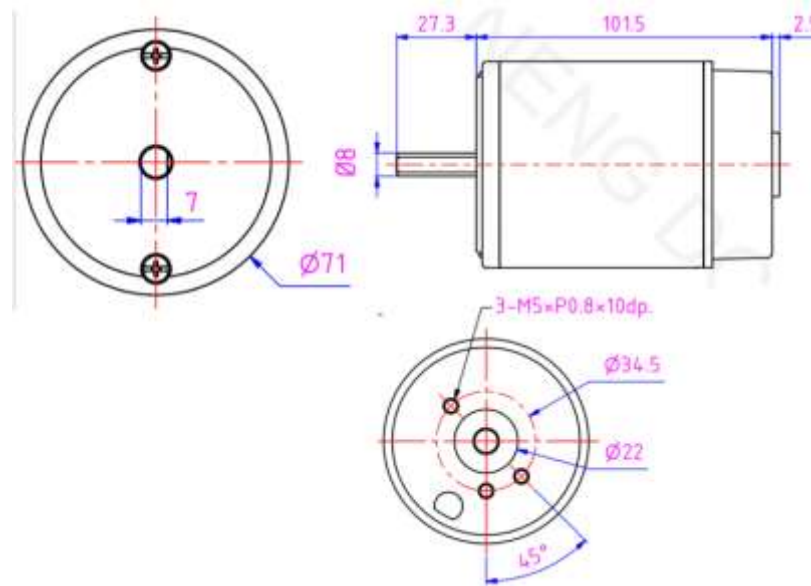


Hình 2.14 Motor giảm tốc 24V DC

Cấu tạo:

- Động cơ DC: Dùng nam châm (stator) và cuộn dây quay (rotor) với chổi than/bộ góp để chuyển điện thành quay.
- Hộp giảm tốc: Hệ thống bánh răng nhận tốc độ cao từ động cơ, giảm xuống tốc độ thấp hơn ở trục ra, đồng thời tăng lực xoắn.

Bản vẽ động cơ:



Hình 2.15 Bản vẽ của motor giảm tốc

Thông số kĩ thuật:

Bảng 2.9 Thông số kĩ thuật Moter giảm tốc

Thông số	Giá trị
Điện áp:	DC12V-24V
Không- tải trọng hiện tại:	100mA
Phạm vi điện áp làm việc:	12v-24v
Không- tốc độ tải:	50-100r/min
Trọng lượng:	100g

2.4.6 Cảm biến khoảng cách vật cản E3F-DS10C4

Cảm biến E3F-DS10C4 được sử dụng phổ biến trong các ứng dụng tự động hóa, như 1 cảm biến tiệm cận để nhận biết vật đến gần, cảm biến trong các băng chuyền tự động, mạch đếm sản phẩm, ... E3F-DS10C4 Cảm Biến Khoảng Cách Vật Cản dùng ánh sáng hồng ngoại để phát hiện vật cản với ưu điểm ít nhiễu do cảm biến phát ra tia hồng ngoại với dải tần số riêng biệt.

Chức năng:

Cảm biến E3F-DS10C4 đóng vai trò quan trọng trong việc phát hiện vị trí của hộp sản phẩm trên băng tải. Khi sản phẩm di chuyển đến đúng vị trí cần dán tem (trên băng tải dài 90cm do động cơ giảm tốc điều khiển), cảm biến sẽ phát hiện vật cản trong khoảng

cách 3–10cm và gửi tín hiệu về bộ điều khiển để điều khiển van điện từ kích hoạt xilanh hạ xuống cố định tem gián lên sản phẩm.



Hình 2.16 Cảm Biến Khoảng Cách Vật Cản E3F-DS10C4 3-10cm, NPN-NO

Nguyên lí hoạt động:

- Phát tia IR: LED hồng ngoại phát ra tia sáng liên tục.
- Phản xạ: Nếu có vật nằm trong phạm vi (3–10cm), tia IR phản xạ ngược lại.
- Thu tín hiệu: Bộ thu (phototransistor) nhận tia phản xạ.
- Chuyển đổi: Tia sáng → tín hiệu điện.
- Xử lý: Mạch xử lý so sánh cường độ ánh sáng với mức ngưỡng.
- Kích hoạt ngõ ra:
 - Có vật: Tín hiệu mạnh → transistor NPN dẫn → ngõ ra nối GND (ON).
 - Không vật: Tín hiệu yếu → transistor ngắt → ngõ ra hở (OFF).

Bảng thông số kĩ thuật:

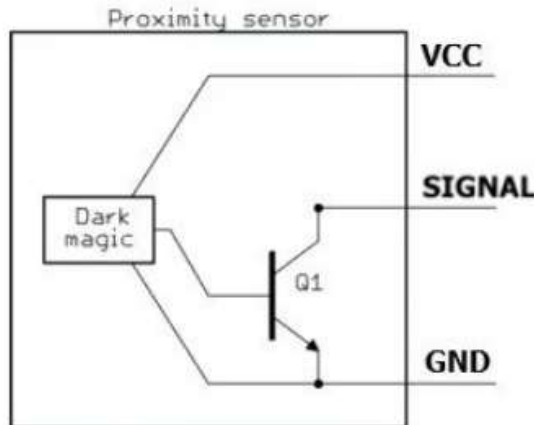
Bảng 2.10 Bảng thông số kĩ thuật Cảm Biến Khoảng Cách Vật Cản E3F-DS10C4 3-10cm, NPN-NO

Thông số	Giá trị
Model	OMDHON E3F-DS10C4
Số dây tín hiệu	3 dây (2 dây cấp nguồn và 1 dây tín hiệu)
Điện áp cấp	6–36V DC (khuyến nghị 6–24V DC)
Khoảng cách phát hiện	3–10 cm (có thể điều chỉnh qua biến trở)
Ngõ ra	NPN, thường mở (NO)
Dòng ngõ ra tối đa	<300 mA
Nhiệt độ hoạt động	-40°C đến 70°C
Góc khuếch tán	3~5 độ
Kích thước	Đường kính 18 mm, chiều dài 68 mm
Chất liệu vỏ	Nhựa ABS, đồ keo chống nước và chống va đập

Chiều dài dây	1 m
---------------	-----

Kết nối dây cảm biến:

- Nâu: VCC (6–36V DC)
- Đen: Ngõ ra tín hiệu (cần có điện trở kéo lên VCC, khoảng 1–10 k Ω)
- Xanh dương: GND



Hình 2.17 Sơ đồ nối dây cảm biến khoảng cách vật cản E3F-DS10C4 3-10cm, NPN-NO

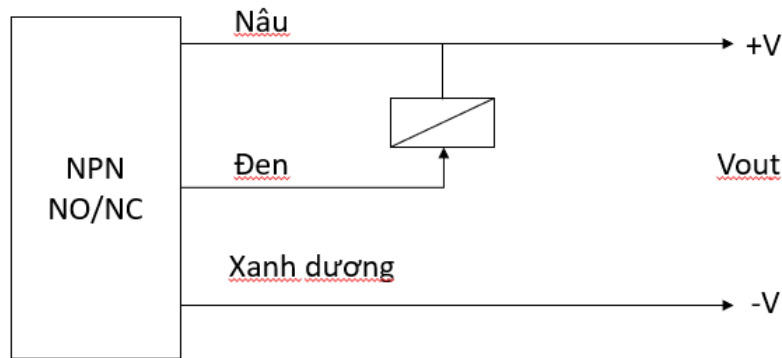
2.4.7 E3Z-D61 Cảm Biến Quang Khuếch Tán Khoảng Dò 5-100mm NPN 12~24VDC

E3Z-D61 là cảm biến quang khuếch tán (Diffuse Reflective Sensor) thuộc dòng E3Z nổi tiếng của hãng Omron – Nhật Bản. Với thiết kế nhỏ gọn, bền bỉ, khoảng dò ổn định trong phạm vi 5–100mm, E3Z-D61 là giải pháp lý tưởng cho các ứng dụng phát hiện vật thể ở khoảng cách ngắn trong môi trường công nghiệp tự động hóa.

Cảm biến sử dụng nguồn điện 12~24VDC và có ngõ ra NPN, phù hợp với hầu hết các hệ thống điều khiển PLC hoặc relay hiện nay.



Hình 2.18 E3Z-D61 Cảm Biến Quang Khuếch Tán Khoảng Dò 5-100mm NPN 12~24VDC



Hình 2.19 Ngõ ra NPN

Chức năng:

- Phát hiện tem có mặt tại vị trí dán, tức là kiểm tra tem đã sẵn sàng tại đầu cơ cấu dán tem.
- Nếu có tem, cảm biến sẽ gửi tín hiệu ngõ ra NPN (LOW), bộ điều khiển nhận tín hiệu → kích hoạt motor giảm tốc → kéo tem xuống và thực hiện thao tác dán tem.
- Nếu không có tem: Không dán, tránh lỗi không tem.
- Tốc độ phản hồi nhanh, thích hợp cho băng tải: Thời gian đáp ứng < 1ms giúp cảm biến không bỏ sót sản phẩm, kể cả khi băng tải chạy nhanh.

Thời gian đáp ứng < 1ms giúp cảm biến không bỏ sót sản phẩm, kể cả khi băng tải chạy nhanh.

Nguyên lí hoạt động:

- Phát tia sáng: LED phát tia sáng (thường là hồng ngoại hoặc đỏ).
- Phản xạ khuếch tán: Nếu có vật trong phạm vi 5–100mm, tia sáng phản xạ ngược lại.
- Thu tín hiệu: Phototransistor nhận tia phản xạ và tạo ra tín hiệu điện.
- Xử lý: Mạch bên trong so sánh cường độ ánh sáng phản xạ với ngưỡng đặt trước.
- Ngõ ra:
 - Có vật thể: Tín hiệu đủ mạnh → NPN dẫn → ngõ ra nối GND → ON.
 - Không có vật: Tín hiệu yếu → NPN ngắt → ngõ ra hở mạch → OFF.

Thông số kĩ thuật:

Bảng 2.11 Thông số kĩ thuật E3Z-D61 Cảm Biến Quang Khuếch Tán Khoảng Dò 5-100mm NPN 12~24VDC

Thông số	Giá trị
----------	---------

Kiểu dáng	Khối (hình chữ nhật)
Loại cảm biến	Quang khuếch tán
Khoảng cách dò	5 → 100 mm
Loại ngõ ra	NPN (thường mở - NO)
Điện áp DC tối đa	24V DC
Nguồn ánh sáng	Hồng ngoại (IR LED)
Dòng điện ngõ ra max	100 mA
Vật liệu vỏ	PBT (nhựa kỹ thuật)
Nhiệt độ hoạt động	-25°C đến +55°C

2.4.8 Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F

Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F của Sharp là một thiết bị đo khoảng cách ngắn, sử dụng công nghệ quang học để xác định khoảng cách giữa cảm biến và vật thể. Với phạm vi đo từ 2 đến 15 cm, nó phù hợp cho các ứng dụng cần phát hiện vật thể ở khoảng cách rất gần, chẳng hạn như robot dọn dẹp, thiết bị tự động hóa hoặc các ứng dụng tương tác người-máy.

Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F dùng để xử lý hộp sản phẩm lỗi hoặc biến dạng biến dạng bằng phương pháp xác định bề mặt gián tem có bị sai lệch vị trí so với khoảng cách cảm biến hồng ngoại.



Hình 2.20 Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F

Chức năng:

- ❖ Phát hiện sản phẩm lỗi biến dạng:
 - Cảm biến GP2Y0A51SK0F đo khoảng cách giữa cảm biến và bề mặt hộp. Nếu hộp bị bóp méo, cong vênh hoặc kích thước không chuẩn → khoảng cách sẽ lệch khỏi giá trị chuẩn.
 - Hệ thống phân tích dữ liệu khoảng cách để xác định hộp lỗi.

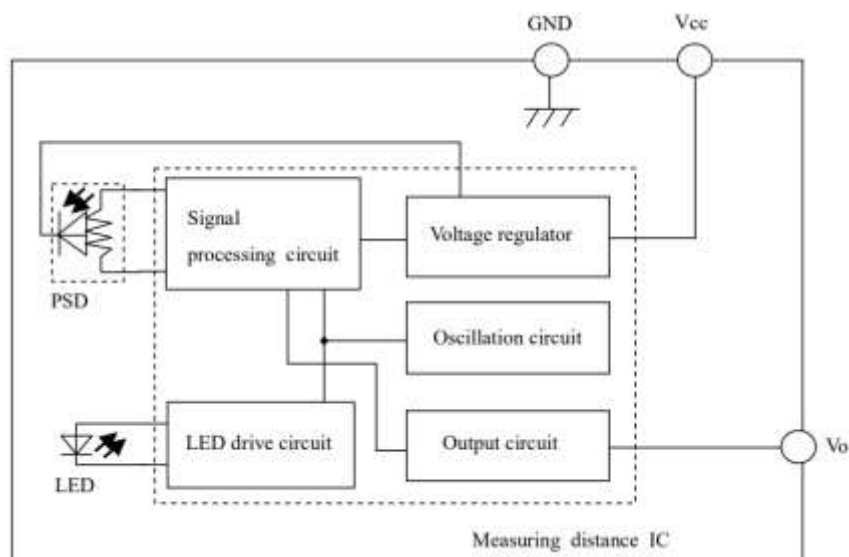
- ❖ Kích hoạt cơ cấu loại bỏ hộp lỗi, khi phát hiện hộp lỗi:
 - Gửi tín hiệu điều khiển van điện từ. van điện từ kích hoạt xi lanh khí nén để đẩy hộp ra khỏi dây chuyền băng tải.
- ❖ Đồng bộ với hệ thống điều khiển tự động, tích hợp với PLC để:
 - Ghi nhận trạng thái hoạt động.
 - Đếm số lượng sản phẩm lỗi/đạt.
 - Đồng bộ với băng tải và bộ dán tem.

Thông số kĩ thuật:

Bảng 2.12 Thông số kỹ thuật của cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F

Thông số	Giá trị
Phạm vi đo:	2 cm đến 15 cm
Điện áp hoạt động:	4.5 V đến 5.5 V
Dòng tiêu thụ trung bình:	12 mA /1s
Tín hiệu đầu ra:	Điện áp analog tỉ lệ với khoảng cách đo
Chu kỳ cập nhật:	16.5 ± 4 ms
Kích thước:	$27.0 \times 10.8 \times 12.0$ mm
Trọng lượng:	2.7 g
Nhiệt độ hoạt động:	-10°C đến $+60^{\circ}\text{C}$
Loại đầu nối:	JST ZH 3 chân

Sơ đồ mô tả:



Hình 2.21 Sơ đồ cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F

- ❖ Giải thích từng khối chức năng:

- Vcc (Voltage Supply): Chân cấp nguồn (4.5V – 5.5V DC) giúp toàn bộ cảm biến hoạt động.
 - GND (Ground): Chân nối đất, dùng làm tham chiếu chung cho mạch.
 - Vo (Output Voltage): Ngõ ra điện áp analog thay đổi theo khoảng cách đến vật thể. Vật càng gần → điện áp càng cao.
 - LED (Infrared Emitting Diode): Phát ra tia hồng ngoại chiếu vào vật thể.
 - LED Drive Circuit: Điều khiển LED phát ra ánh sáng đều và đúng tần số, giúp chống nhiễu từ môi trường.
 - PSD (Position Sensitive Detector): Cảm biến quang đặc biệt phát hiện vị trí tia phản xạ chiếu vào → xác định khoảng cách.
 - Signal Processing Circuit: Xử lý tín hiệu từ PSD → chuyển thành điện áp analog ổn định gửi ra Vo.
 - Voltage Regulator: Ổn định điện áp cho các khối bên trong cảm biến, dù nguồn vào có dao động nhẹ.
 - Oscillation Circuit: Tạo tín hiệu dao động điều khiển LED và đồng bộ xử lý tín hiệu, giảm nhiễu từ ng
- ❖ Nguyên lý hoạt động:
- Phát xạ: Mạch điều khiển LED được kích hoạt bởi mạch dao động, làm cho LED phát ra một chùm tia hồng ngoại theo dạng xung.
 - Phản xạ: Chùm tia hồng ngoại này chiếu vào vật thể và phản xạ trở lại phía cảm biến.
 - Thu nhận & Xác định vị trí: Tia hồng ngoại phản xạ được thu nhận bởi PSD. Do nguyên lý triangulation, nếu vật thể ở gần, tia phản xạ sẽ chiếu vào một vị trí khác trên PSD so với khi vật thể ở xa.
 - Xử lý tín hiệu: PSD chuyển đổi vị trí ánh sáng thành tín hiệu điện. Mạch xử lý tín hiệu sẽ đọc tín hiệu này và chuyển đổi nó thành một điện áp tương tự.
 - Ngõ ra: Điện áp tương tự đã xử lý được đưa ra chân Vo. Điện áp này sẽ thay đổi theo khoảng cách:
 - Vật thể càng gần: Điện áp Vo thường sẽ càng cao.
 - Vật thể càng xa: Điện áp Vo thường sẽ càng thấp.

2.4.9 Van điện từ (solenoid) khí nén LAIZE 4V210-08

Van điện từ (solenoid) khí nén LAIZE 4V210-08 là loại van 5/2 (5 cổng, 2 vị trí) phổ biến, được sử dụng rộng rãi trong các hệ thống khí nén để điều khiển hướng dòng khí,

thường dùng cho xilanh tác động kép. Sản phẩm này có nhiều lựa chọn điện áp, bao gồm 24V DC, phù hợp với nhiều ứng dụng công nghiệp và tự động hóa.

Van điện từ (solenoid) khí nén LAIZE 4V210-08 sử dụng để điều khiển xilanh đóng và mở để xử lý như hộp lỗi, cố định tem cho hộp sau khi gián và đẩy hộp lỗi sau khi gián tem lỗi.



Hình 2.22 Van điện từ (solenoid) khí nén LAIZE 4V210-08

Chức năng:

- Điều khiển hướng dòng khí: Van này cho phép chuyển hướng dòng khí từ cổng cấp khí (P) đến cổng ra (A hoặc B) tùy thuộc vào trạng thái của van, giúp điều khiển chuyển động của xi lanh khí nén.
- Điều khiển chuyển động của xi lanh: Khi van được kích hoạt (cấp điện vào cuộn coil), nó thay đổi vị trí của lõi van, từ đó thay đổi hướng dòng khí và điều khiển chuyển động của xi lanh.
- Đảo chiều chuyển động: Van có thể đảo chiều dòng khí, giúp xi lanh di chuyển theo hai hướng: đẩy ra và rút về.

Thông số kỹ thuật:

Bảng 2.13 Thông số kỹ thuật Van điện từ (solenoid) khí nén LAIZE 4V210-08

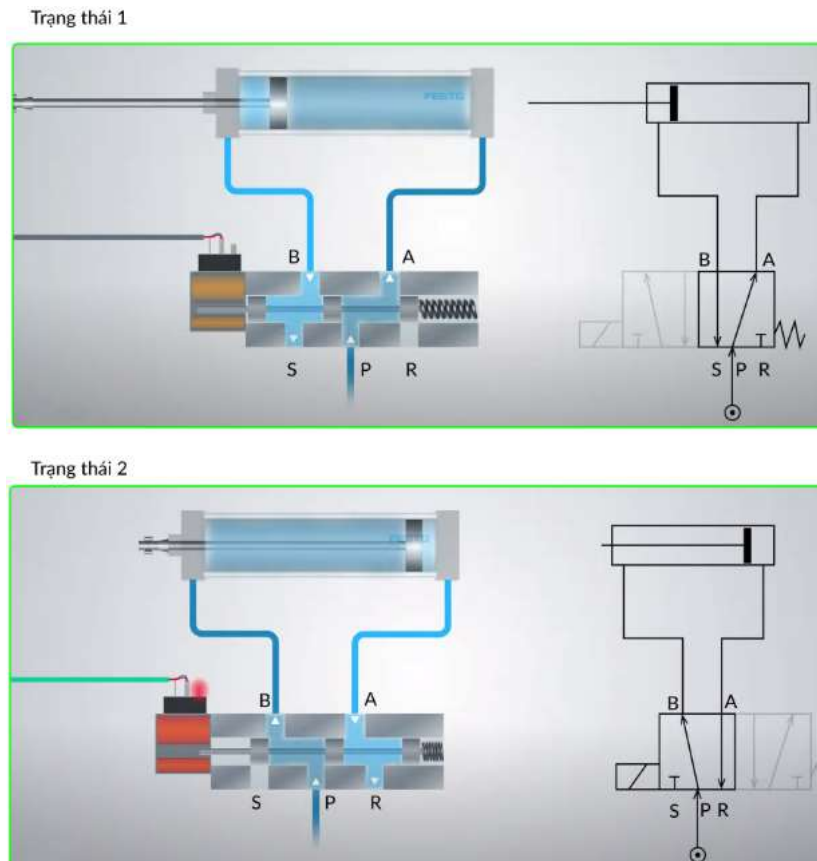
Thông số	Giá trị
Loại van:	5 cổng, 2 vị trí, tác động đơn
Điện áp cuộn coil:	24V DC
Áp suất hoạt động:	0.15 – 0.8 MPa
Áp suất chịu đựng tối đa:	1.2 MPa

Nhiệt độ hoạt động:	-20 ~ 70°C
Kích thước cổng:	G1/4"
Lưu lượng hiệu dụng (Cv):	0.89
Tần số hoạt động tối đa:	5 chu kỳ/giây
Thời gian đáp ứng:	Kích hoạt: ≤ 0.05 giây Hồi vị: ≤ 0.1 giây
Công suất cuộn coil:	DC: 4.8 W AC: 5.4 VA
Tuổi thọ cơ học:	≥ 15,000,000 lần

Cấp nguồn cho Van điện từ:



Hình 2.23 Cấp nguồn cho van điện từ.



Hình 2.24 Sơ đồ nguyên lý hoạt động

❖ Cấu tạo chính:

- P – Cổng cấp khí nén (từ máy nén).
- A, B – Cổng làm việc (nối với 2 đầu xi lanh).
- R, S – Cổng xả khí.
- Solenoid – Tạo lực hút kéo con trượt khi có điện.
- Lò xo hồi – Đẩy con trượt về vị trí ban đầu khi mất điện.
- Con trượt – Điều hướng khí trong van.

❖ Nguyên lý hoạt động:

- Trạng thái 1 – Không cấp điện ($VDT = 0$):
 - Lò xo đẩy con trượt về bên trái.
 - Khí nén từ $P \rightarrow A \rightarrow$ vào xi lanh làm pít-tông thu về.
 - Khí từ đầu kia của xi lanh $\rightarrow B \rightarrow$ xả ra ngoài qua S.
- Trạng thái 2 – Cấp điện ($VDT = 1$):
 - Solenoid hút con trượt sang phải.
 - Khí nén từ $P \rightarrow B \rightarrow$ vào xi lanh làm pít-tông duỗi ra.
 - Khí từ đầu còn lại $\rightarrow A \rightarrow$ xả qua R.

2.4.10 Xi lanh TN10-100

Xi lanh TN10-100 là dòng xi lanh khí nén trực kép (2 ty) với đường kính 10mm và hành trình 100mm, được sử dụng rộng rãi trong các ứng dụng tự động hóa, lắp ráp và các hệ thống khí nén công nghiệp.



Hình 2.25 Xi lanh TN10-100

Chức năng:

❖ Thiết kế và ứng dụng trong mô hình:

- Lực đẩy của xilank: TN10 có piston nhỏ ($\varnothing 10\text{mm}$), vì vậy cần áp suất khí đủ cao ($\sim 0.5\text{--}0.7\text{ MPa}$) để tạo lực đẩy phù hợp.
- Tốc độ đẩy: Có thể điều chỉnh bằng van tiết lưu khí gắn trên co nối của xi lanh
- Vị trí lắp đặt: Xi lanh nên được gắn song song với băng tải, đầu đẩy hướng vuông góc để đảm bảo hộp được đẩy ra đúng hướng.

❖ cách hoạt động:

- Hộp chạy đến vị trí có gắn cảm biến khoảng cách vật cản E3F-DS10C4 3-10cm, NPN-NO .
- Cảm biến phát hiện có hộp lỗi đến sau đó gửi tín hiệu đến bộ điều khiển PLC.
- Bộ điều khiển PLC điều khiển van điện từ mở khí vào xi lanh kích hoạt xi lanh đẩy hộp ra ngoài.
- Sau thời gian định sẵn, PLC đóng van điện từ $\rightarrow$ xi lanh rút lại, sẵn sàng cho lần tiếp theo.

Thông số kỹ thuật:

Bảng 2.14 Thông số kỹ thuật Van điện từ (solenoid) khí nén LAIZE 4V210-08

Thông số	Giá trị
Đường kính piston:	10mm
Hành trình:	10–100mm
Cách thức tác động:	Tác động kép (hai ty)
Áp suất làm việc:	0.1–1.0 MPa
Áp suất tối đa:	1.5 MPa
Nhiệt độ môi trường:	-10°C đến 60°C
Kích thước cổng khí:	Ren M5
Lưu chất:	Khí nén sạch

2.5 Thiết kế mô hình dán tem

Mục tiêu thiết kế

Thay thế công đoạn dán tem thủ công hiện tại bằng một mô hình bán tự động/tự động hoàn toàn giúp:

- Tăng năng suất và độ chính xác khi dán tem.
- Giảm lỗi như lệch tem, dán sai vị trí, hoặc dán lên hộp móp méo.
- Góp phần hướng đến dây chuyền sản xuất thông minh.

2.5.1 Thiết kế tủ điện điều khiển

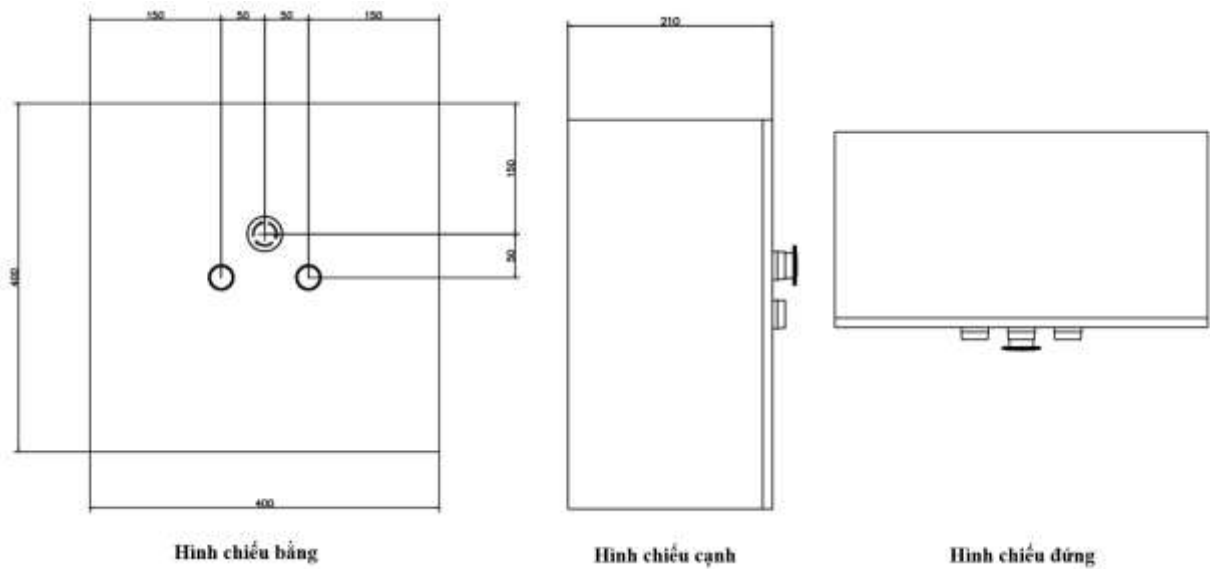
Tủ điện điều khiển là bộ phận trung tâm của hệ thống, nơi tập trung các thiết bị đóng cắt, bảo vệ và điều khiển quá trình vận hành máy dán tem tự động. Nó giúp đảm bảo hệ thống vận hành an toàn, ổn định và có thể mở rộng dễ dàng.

Kích thước tổng thể tủ điện: 40cm (dài) × 21cm (rộng) × 40cm(cao)

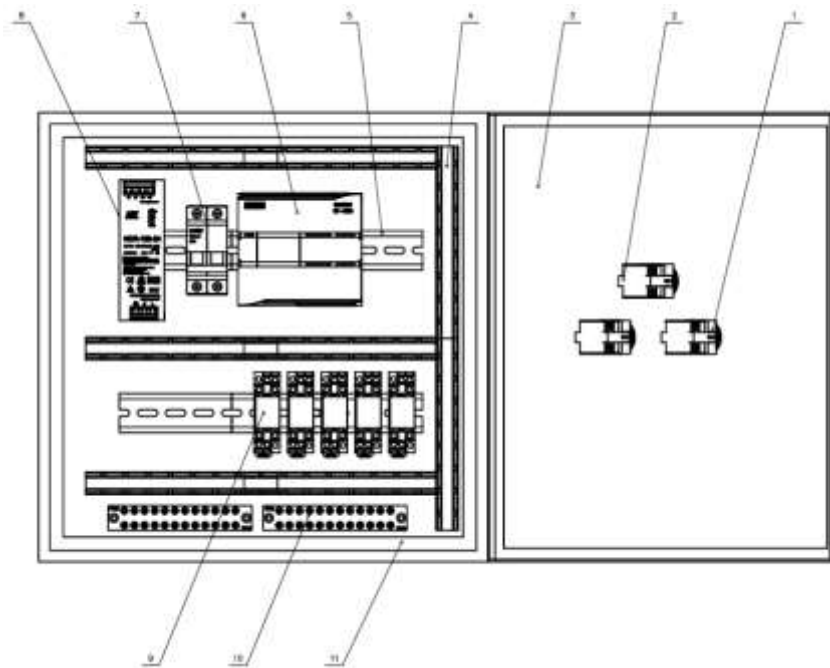
Chất liệu vỏ tủ: tôn mạ sơn tĩnh điện

Chức năng chính

- Cung cấp và phân phối điện cho các thiết bị trong hệ thống.
- Điều khiển hoạt động của xy lanh khí nén, cảm biến, động cơ dán tem.
- Giao tiếp với phần mềm xử lý ảnh và camera.
- Bảo vệ hệ thống khi có sự cố quá dòng, ngắn mạch,...
- Hiển thị trạng thái hoạt động và cho phép vận hành thủ công khi cần.



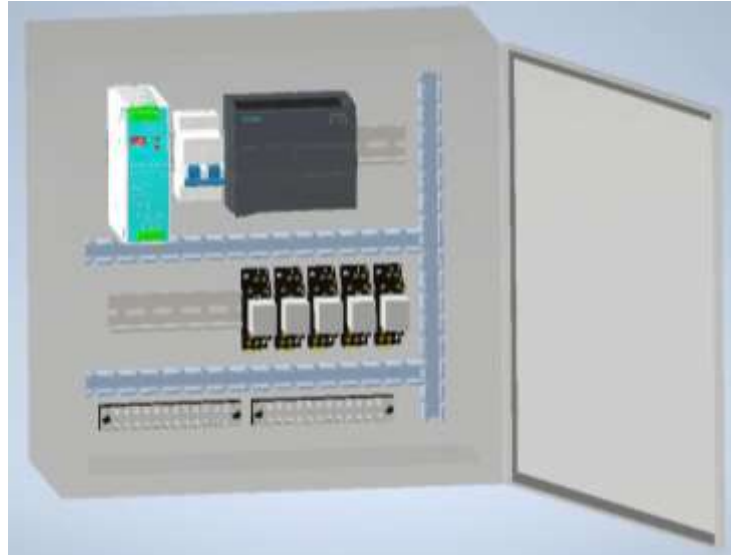
Hình 2.26 Kích thước bản vẽ mặt cắt tủ điều khiển.



Hình 2.27 Bản vẽ bố trí thiết bị trong tủ điều khiển

Chú thích:

- | | |
|-------------------------|--------------------------|
| 1. Nút nhấn Start, Stop | 7. CCB |
| 2. Nút nhấn EMG | 8. Nguồn tổ ong |
| 3. Cánh tủ | 9. Role |
| 4. Máng điện | 10. Cầu đấu |
| 5. Thanh đỡ thiết bị | 11. Vỏ trong tủ thiết bị |
| 6. PLC Siemens | |



Hình 2.28 Bản vẽ thiết kế tủ điều khiển 3D

2.5.2 Phần cơ khí

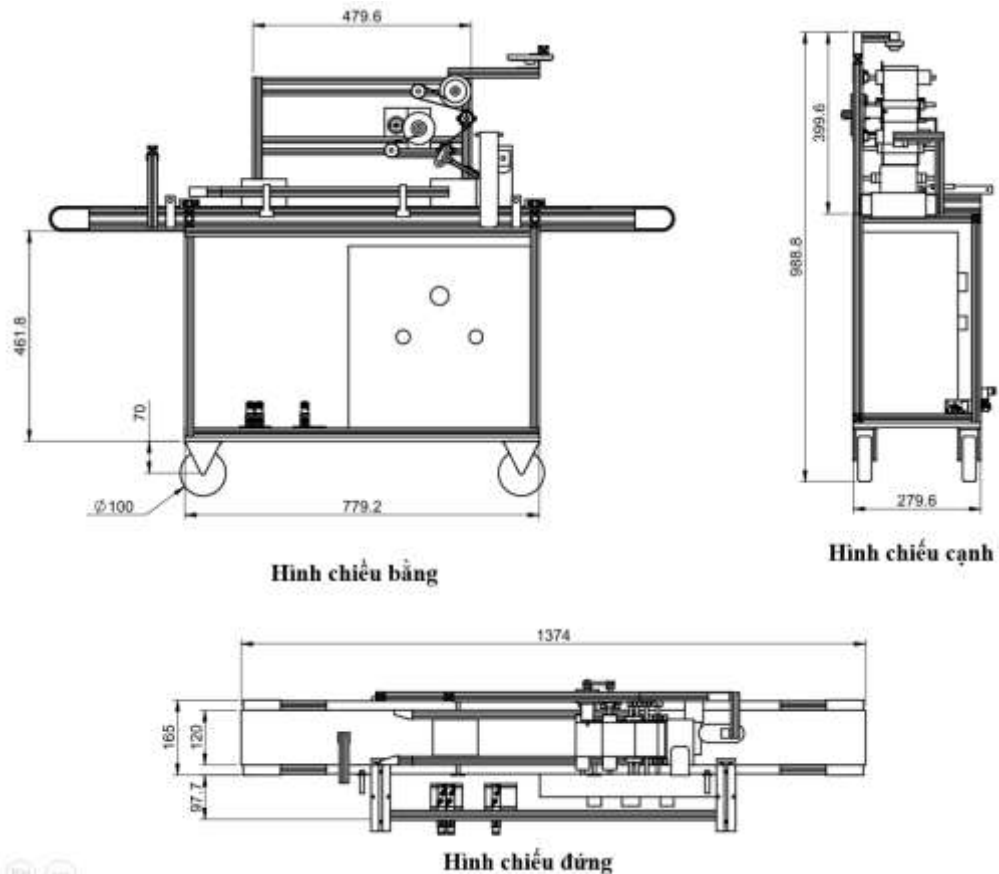
Phần cơ khí đóng vai trò là khung sườn và cơ cấu vận hành chính của hệ thống, bao gồm:

Kích thước tổng thể khung: dài 138cm (dài) $\times$ 28cm rộng $\times$ 99cm (cao)

Chất liệu khung: Nhôm định hình 3030, chắc chắn và dễ lắp ráp.

Các thành phần chính:

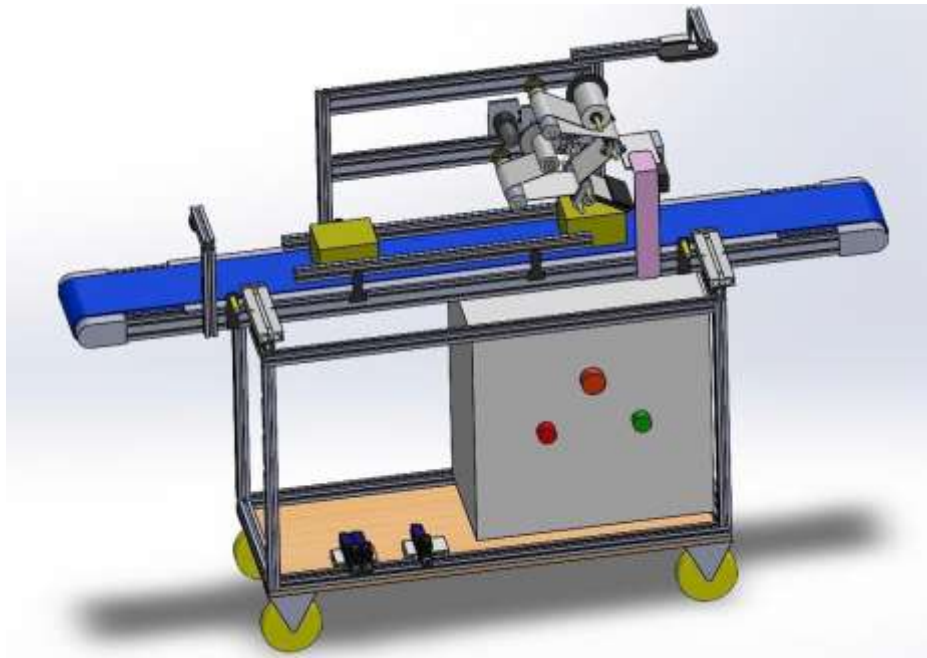
- Khung chính (Bạc): Là bộ khung chịu lực, tạo nền móng để lắp ráp các cụm cơ cấu khác như băng tải, cụm dán tem, cảm biến,...
- Tủ điện (xám nhạt): Lắp đặt ở bên dưới khung, chứa bộ điều khiển PLC, rơ le, nguồn, công tắc,...
- Sàn đỡ (cam nhạt): Sử dụng vật liệu gỗ, giúp tăng độ vững chắc và để đặt tủ điều khiển.
- Bánh xe (vàng): Gắn ở 4 chân khung giúp dễ dàng di chuyển mô hình.



Hình 2.29 Kích thước thiết kế khung cơ khí

2.5.3 Phân điều khiển – khí nén

- Xi lanh khí nén (Xi lanh ép): Điều khiển động tác dán tem (đưa tem tiếp xúc hộp), loại bỏ những sản phẩm có bề mặt dán tem lỗi và sản phẩm sau khi dán tem lỗi.
- Van điện từ 5/2: Dùng để điều khiển hướng khí cấp vào xi lanh. Tín hiệu điều khiển đến van được cấp từ PLC.
- Van điện từ được đặt ở vị trí trên mặt bằng (sàn đỡ) của mô hình. Bố trí hợp lý, dễ thay thế sửa chữa khi có vấn đề
- Nguồn khí: Đảm bảo cung cấp khí nén ổn định từ máy nén khí.



Hình 2.30 Bản vẽ thiết kế mô hình dán tem

2.6 Kết luận

Trong chương này, nhóm đã hoàn thiện việc xây dựng quy trình công nghệ tổng thể cho hệ thống dán tem tự động, từ giai đoạn kiểm tra đầu vào đến kiểm tra tem sau dán, nhằm đảm bảo tính chính xác và ổn định của toàn bộ dây chuyền. Quy trình được phân tích chặt chẽ theo trình tự công nghệ thực tế và được mô hình hóa thành các bước điều khiển rõ ràng, phục vụ cho quá trình lập trình và vận hành hệ thống.

Việc lựa chọn thiết bị được thực hiện dựa trên các tiêu chí kỹ thuật như độ chính xác, tốc độ xử lý, khả năng tương thích và chi phí đầu tư. Các thành phần như cảm biến quang, xi lanh khí nén, camera, bộ cấp tem, PLC điều khiển, và máy tính xử lý hình ảnh đều được lựa chọn phù hợp với yêu cầu kỹ thuật của mô hình. Thông số của từng thiết bị đã được nghiên cứu kỹ lưỡng để đảm bảo sự phù hợp và đồng bộ trong toàn hệ thống.

Về thiết kế tủ điện điều khiển, nhóm đã xác định đầy đủ các thiết bị điện cần thiết như nguồn tổ ong, role trung gian, nút nhấn, đèn báo, PLC, bộ chia khí và các thiết bị bảo vệ. Sơ đồ điện được thiết kế khoa học, dễ bảo trì và mở rộng. Toàn bộ hệ thống điều khiển được phân chia thành các nhóm chức năng rõ ràng nhằm thuận tiện cho việc lập trình và khắc phục sự cố.

Đối với phần thiết kế mô hình, hệ thống dán tem được xây dựng với kích thước mô phỏng thu gọn nhưng vẫn đảm bảo đầy đủ chức năng như hệ thống thực tế: từ kiểm tra trước dán, dán tem, đến kiểm tra sau dán. Cơ khí mô hình được bố trí hợp lý, giúp trực

quan hóa rõ quy trình vận hành và có thể được sử dụng làm nền tảng thử nghiệm trước khi triển khai thực tế trên dây chuyền.

Nhìn chung, toàn bộ hệ thống đã được lên kế hoạch và thiết kế một cách đồng bộ, khả thi và sẵn sàng đưa vào vận hành thử nghiệm. Đây là cơ sở quan trọng để chuyển sang các bước tiếp theo như lập trình điều khiển, xử lý ảnh và kiểm thử hệ thống tổng thể.

CHƯƠNG 3. THIẾT LẬP CẤU HÌNH CÁC THIẾT BỊ TRÊN PHẦN MỀM VÀ XÂY DỰNG THUẬT TOÁN ĐIỀU KHIỂN

3.1 Xây dựng thuật toán điều khiển hệ thống trên phần mềm TIA Portal

3.1.1 Giới thiệu phần mềm TIA Portal.

TIA Portal hay Totally Integrated Automation Portal là một nền tảng phần mềm toàn diện do Siemens phát triển để lập trình, cấu hình và quản lý các hệ thống tự động hóa công nghiệp. Nền tảng này cung cấp một môi trường làm việc thống nhất cho nhiều thiết bị khác nhau, bao gồm bộ điều khiển logic lập trình (PLC), giao diện giám sát (HMI), hệ thống truyền thông công nghiệp và hệ thống truyền động. TIA Portal được sử dụng rộng rãi trong các ngành sản xuất, dây chuyền sản xuất, hệ thống SCADA và các ứng dụng Internet vạn vật (IoT), cho phép các kỹ sư tối ưu hóa các quy trình tự động hóa từ thiết kế đến vận hành. Nền tảng này tích hợp nhiều công cụ thiết yếu vào một giao diện duy nhất, bao gồm lập trình PLC, thiết kế HMI, cấu hình truyền động, hệ thống tích hợp an toàn và tích hợp SCADA. Một trong những tính năng chính của TIA Portal là khả năng hỗ trợ nhiều ngôn ngữ lập trình, chẳng hạn như LAD, FBD, SCL và STL. Ngoài ra, nền tảng này còn cung cấp một công cụ mô phỏng có tên là PLCSIM, cho phép người dùng kiểm tra các chương trình PLC mà không cần phần cứng. Trong các ứng dụng thực tế, TIA Portal được sử dụng trong tự động hóa nhà máy, kiểm soát dây chuyền sản xuất và rô bốt công nghiệp. Nó cũng được sử dụng trong các hệ thống SCADA để thu thập dữ liệu và giám sát quy trình. Hơn nữa, nền tảng này đóng vai trò quan trọng trong sản xuất thông minh, cho phép kết nối các thiết bị IoT và phân tích dữ liệu dựa trên đám mây. TIA Portal cũng được sử dụng trong các thiết lập giáo dục và nghiên cứu, chẳng hạn như phòng thí nghiệm và các dự án luận án kỹ thuật, cung cấp một nền tảng toàn diện cho việc học tập và thử nghiệm. Nhìn chung, TIA Portal là một công cụ mạnh mẽ giúp hợp lý hóa các quy trình tự động hóa công nghiệp, nâng cao hiệu quả và năng suất.



Hình 3.1 Phần mềm TIA portal

3.1.2 Một số chức năng chính.

a. Lập trình PLC (SIMATIC S7-1200, S7-1500, v.v.):

Viết chương trình điều khiển bằng các ngôn ngữ như LAD, FBD, STL, SCL.

- LAD (Ladder Logic - Ladder Diagram)
 - Dựa trên sơ đồ rơ-le (relay logic), giống như bản vẽ mạch điện.
 - Phổ biến trong điều khiển tuần tự và logic đơn giản.
 - Cú pháp cơ bản: sử dụng tiếp điểm (contacts) và cuộn dây (coils)
- FBD (Function Block Diagram)
 - Lập trình bằng cách kết nối các khối hàm (function blocks)
 - Giống như thiết kế mạch điện tử (logic gate AND, OR, Timer, Counter).
 - Phù hợp với các thuật toán phức tạp hơn LAD
- SCL (Structured Control Language)
 - Là ngôn ngữ dạng văn bản, tương tự Pascal/C.
 - Mạnh mẽ trong xử lý dữ liệu số, toán học, mảng, vòng lặp.
 - Dễ bảo trì và mở rộng với code dài.
 - Cú pháp cơ bản

```
IF "Start" THEN
```

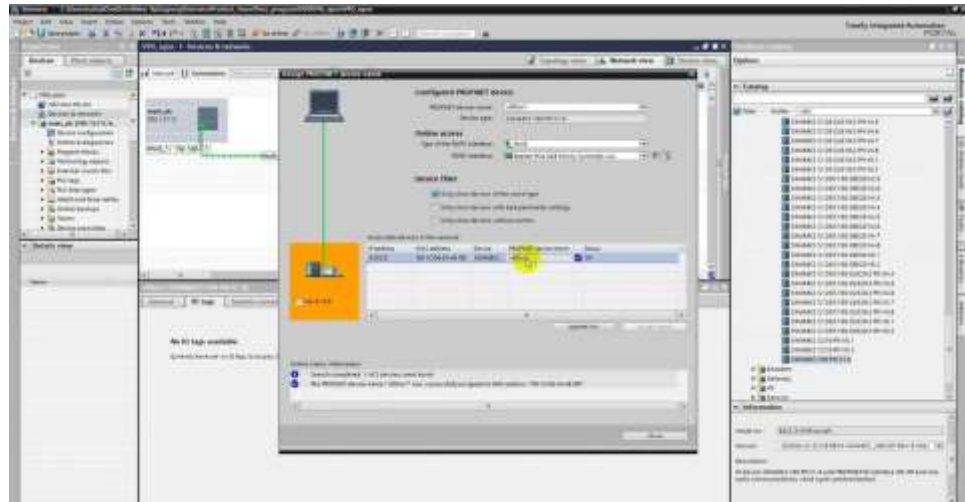
```
    "Motor" := TRUE;
```

```
ELSE
```

```
    "Motor" := FALSE;
```

```
END_IF;
```

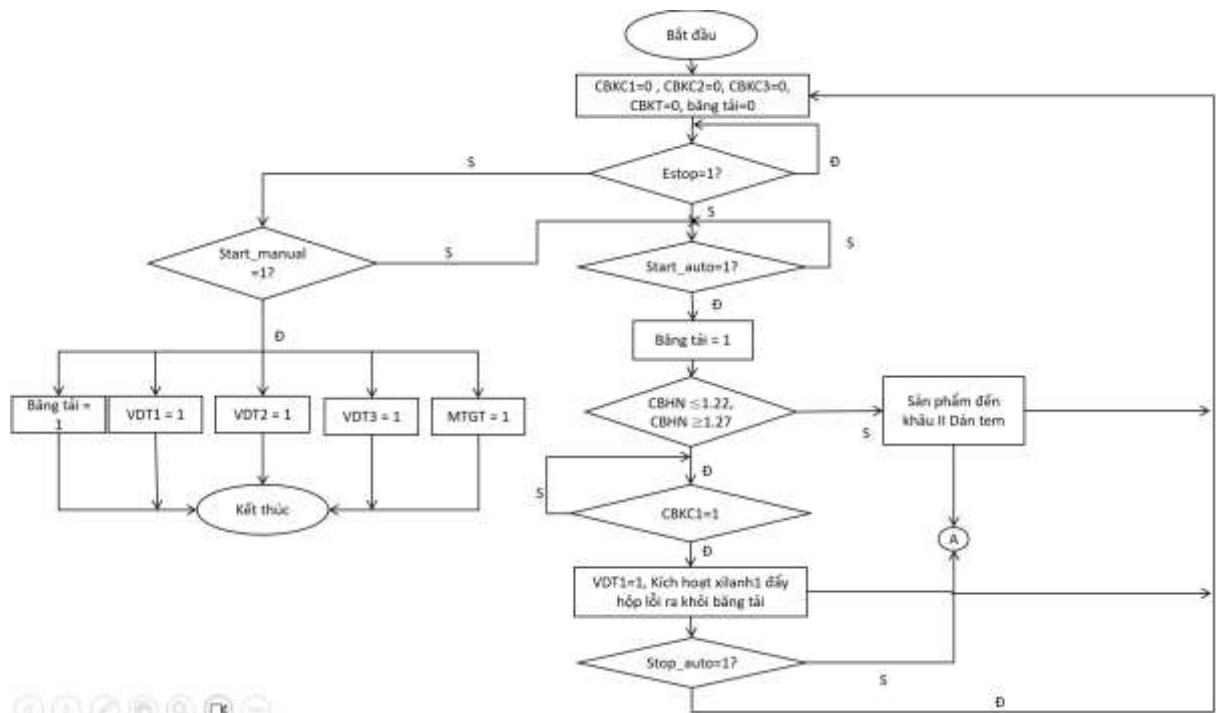
- STL (Statement List)
 - Ngôn ngữ Assembly-like, dùng các lệnh ngắn gọn.
 - Tốc độ xử lý nhanh, tiết kiệm bộ nhớ.
 - Khó đọc hơn LAD/FBD, phù hợp với người có kinh nghiệm
- Tạo khối chức năng (Function Block), khối dữ liệu (Data Block).
- Kiểm tra lỗi và gỡ lỗi chương trình.



Hình 3.2 Giao diện phần mềm TIA portal

3.1.3 Lưu đồ thuật toán

❖ Khâu I: xử lý lỗi biến dạng bề mặt dán tem.



Hình 3.3 Lưu đồ thuật toán khâu I

Giải thích sơ đồ khối khâu I:

- Tất cả các tín hiệu từ cảm biến và trạng thái băng tải được đặt về 0 (OFF hoặc không hoạt động), xét điều kiện nút dừng khẩn cấp Estop:

- Đ (Đúng - Estop được nhấn): Nếu Estop=1 (được nhấn), hệ thống sẽ quay ngược lại chính nó, tức là hệ thống sẽ dừng hoàn toàn và chờ đợi cho đến khi nút Estop được nhả ra (tín hiệu Estop trở lại 0).
 - S (Sai - Estop không nhấn): Nếu Estop không được nhấn (tín hiệu Estop là 0), hệ thống sẽ tiếp tục luồng hoạt động chính sang kiểm tra Start_auto=1?.
- Bắt đầu: Hộp sản phẩm đặt lên băng tải đưa đến khâu I xử lý bề mặt dán tem.

Lựa chọn chế độ chạy Auto:

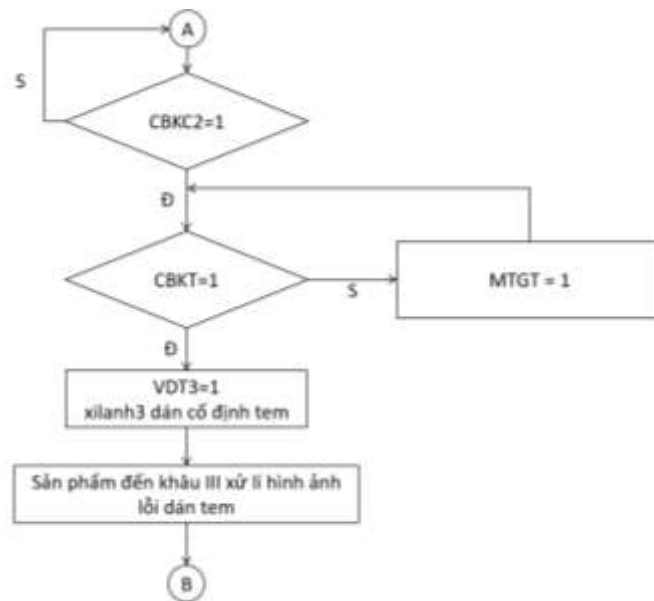
- Xét điều kiện Start_auto=1?:
 - Nếu điều kiện Start_auto=1 là đúng, thì sẽ Set bang_tai =1 (động cơ giảm tốc GB37-550) cho băng chuyền tải hoạt động.
 - Nếu sai thì sẽ quay lại kiểm tra điều kiện Start_auto=1 cho đến khi điều kiện đúng và hoàn thành 1 vòng lặp..
- Tiếp theo xét điều kiện CBHN(Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F) sẽ kiểm tra bề mặt hộp:
 - Nếu bề mặt đạt chuẩn (khoảng cách cảm biến đo được nằm trong khoảng 1.22 đến 1.27), hộp được chuyển thẳng đến khâu dán tem (khâu II dán tem).
 - Nếu bề mặt không đạt chuẩn (khoảng cách cảm biến đo được nằm ngoài 1.22 đến 1.27), hộp được xác định là lỗi. Đối với hộp lỗi, hệ thống sẽ tiếp tục di chuyển đến CBKC(Cảm biến khoảng cách vật cản E3F-DS10C4 3-10cm, NPN-NO).
- Xét điều kiện CBKC1=1 (Cảm biến khoảng cách vật cản E3F-DS10C4 3-10cm, NPN-NO):
 - Nếu CBKC khác 1 thì chưa phát hiện hộp lỗi, hệ thống tiếp tục đợi.
 - Nếu CBKC=1 phát hiện hộp lỗi, kích hoạt VDT1(Van điện từ (solenoid) khí nén LAIZE 4V210-08) để điều khiển xilanh1 sẽ đẩy hộp lỗi ra khỏi băng tải.
- Nhấn nút dừng STOP sẽ dừng hoạt động băng tải, và chờ cho đến khi nhấn nút start để chạy băng tải hoạt động

Lựa chọn chế độ chạy Manual:

- Đ (Đúng - Start_manual được nhấn): Nếu người vận hành muốn chạy ở chế độ thủ công, hệ thống sẽ thực hiện các hành động thủ công.

- BĂNG tải = 1: Băng tải được kích hoạt.
 - VDT1 = 1: Van điện từ 1 được kích hoạt
 - VDT2 = 1: Van điện từ 2 được kích hoạt
 - VDT3 = 1: Van điện từ 3 được kích hoạt
 - MTGT = 1: Motor giảm tốc được kích hoạt.
- Sai thì sẽ kiểm tra lại điều kiện Start_manual=1 có đúng chưa và sẽ hoàn thành 1 vòng lặp.
- Kết thúc: Sau khi các hành động thủ công được thực hiện, quy trình chế độ thủ công kết thúc.

❖ **Khâu II: Dán tem cho hộp sản phẩm.**

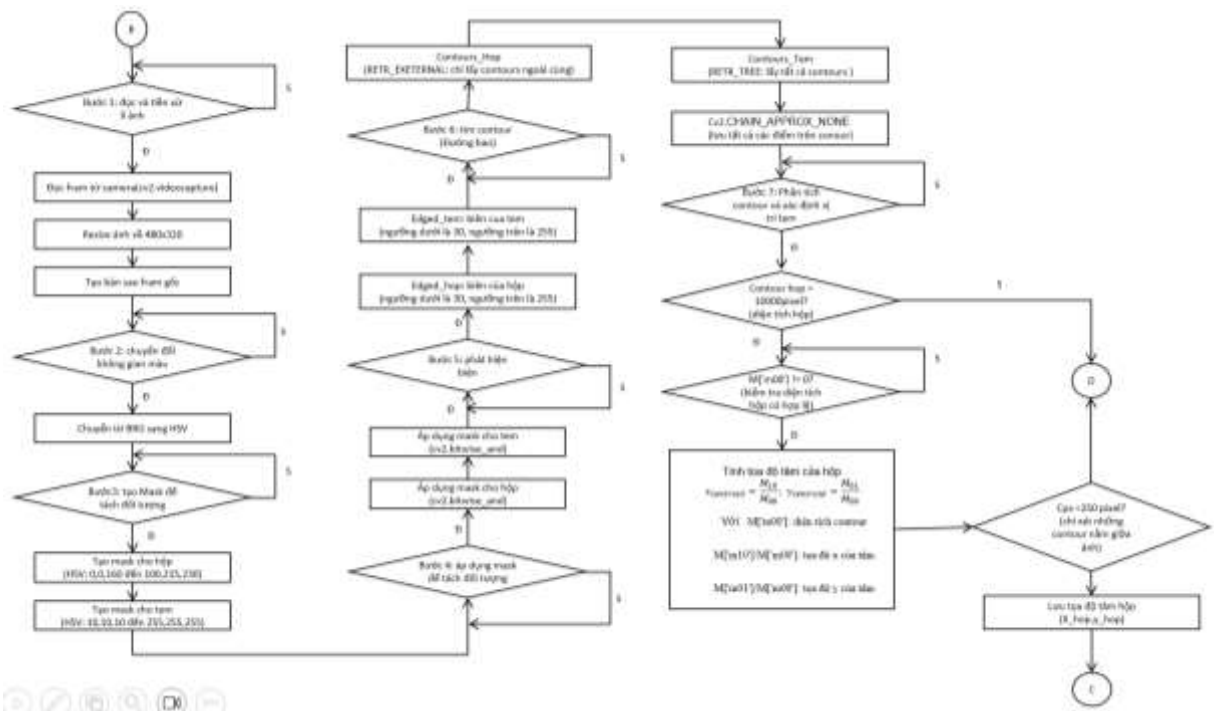


Hình 3.4 Lưu đồ thuật toán khâu II

Giải thích sơ đồ khối khâu II:

- Hộp sản phẩm được băng tải đưa đến khâu dán tem.
- Tại đây hệ thống xét điều kiện $CBKC2 = 1$ (Cảm biến khoảng cách vật cản E3F-DS10C4 3-10cm, NPN-NO):
 - Nếu $CBKC2=1$ (đúng), khi hộp đã đúng vị trí, hệ thống xét thêm điều kiện $CBKT=1$ (E3Z-D61 cảm biến quang khuếch tán khoảng dò 5-100mm NPN 12~24VDC) cho đến khi tem có mặt và sẵn sàng để dán tại cơ cấu dán tem. Nếu không có tem, $MTGT=1$ (motor giảm tốc 24V) sẽ được kích hoạt và quay lại kiểm tra điều kiện $CBKT=1$ có đúng hay chưa. Sau đó khi cả hộp và tem đều đúng vị trí, kích hoạt VDT3 (Van điện từ (solenoid) khí nén LAIZE 4V210-08) để điều khiển xilanh3 để cố định tem lên hộp.

- Nếu CBKC2 = 1(sai), (tức là không phát hiện thấy hộp), quy trình quay lại điểm A và tiếp tục kiểm tra. Điều này tạo thành một vòng lặp đợi cho đến khi hộp sản phẩm đến đúng vị trí và được CBKC2 phát hiện.
 - Cuối cùng, hộp sản phẩm đã dán tem được chuyển đến khâu kiểm tra bằng xử lý hình ảnh để đảm bảo tem được dán chính xác và không có lỗi.
- ❖ **Khâu III: camera xử lý ảnh dán tem lỗi.**



Hình 3.5 Lưu đồ thuật toán khâu III

Giải thích sơ đồ khối khâu III:

- Hộp sản phẩm đã dán tem đi vào khâu xử lý ảnh.
- Bước 1: đọc và tiền xử lý ảnh.
 - Đọc frame từ camera bằng cv2.VideoCapture
 - Resize hình ảnh về kích thước cố định (480x320) để xử lý đồng nhất
 - Tạo bản sao của frame gốc để sử dụng sau này
- Bước 2: chuyển đổi không gian màu.

Chuyển từ không gian màu BGR sang HSV để dễ dàng tách màu hơn

- Hue (Màu sắc): Giá trị từ 0-179 (OpenCV scale).
- Saturation (Độ bão hòa): 0-255.
- Value (Độ sáng): 0-255.
- Bước 3: tạo mask để tách đối tượng.
- Bước 4: áp dụng mask để tách đối tượng.
- Bước 5: phát hiện biên.
- Bước 6: tìm đường bao(contour).
- Bước 7: phân tích contour và xác định vị trí tem.
- Xét điều kiện camera xử lý hình ảnh bằng công cụ camera và thuật toán xử lý ảnh sử dụng thư viện OpenCV và lập trình ngôn ngữ Python bằng phương pháp xác định tâm của tem dán lên sản phẩm tiêu chuẩn đạt chuẩn so với tâm tem sản phẩm sau khi dán (tính toán độ lệch giữa tâm hộp và tâm tem theo chiều dài (ΔX) và chiều rộng (ΔY) của hộp):
 - Nếu tem được dán đúng vị trí (độ lệch nằm trong ngưỡng cho phép: $0 \leq \Delta X \leq 20$ và $0 \leq \Delta Y \leq 10$ với đơn vị là pixel), hộp sản phẩm được chuyển thẳng đến khâu tiếp theo.
 - Nếu tem bị dán sai vị trí (độ lệch vượt quá ngưỡng cho phép), hộp được xác định là dán tem lỗi. Đối với hộp dán tem lỗi, hệ thống sẽ gửi tín hiệu đến cảm biến và xét điều kiện CBKC3=1 (Cảm Biến Khoảng Cách Vật Cản E3F-DS10C4 3-10cm, NPN-NO) cho đến khi hộp đến vị trí đẩy hộp dán tem bị lỗi ra khỏi băng tải (điều kiện này tạo thành một vòng lặp đợi cho đến khi hộp sản phẩm đến đúng vị trí và được CBKC3 phát hiện). Khi hộp lỗi đã đúng vị trí, kích hoạt VDT2 (Van điện từ (solenoid) khí nén LAIZE 4V210-08) để điều khiển xilanh 2 sẽ đẩy hộp lỗi ra khỏi băng tải.

3.1.4 Bảng phân kênh

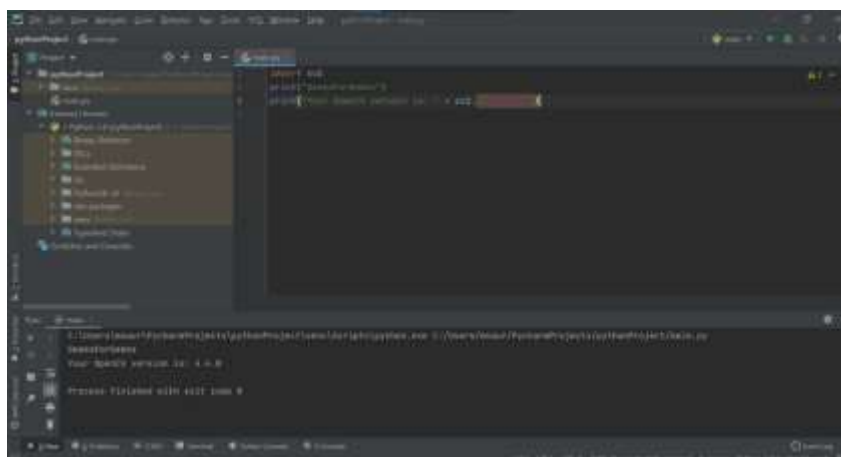
Bảng 3.1 Bảng phân kênh Input/Output

Địa chỉ PLC	Tên	Loại (input/output)	Thiết bị thực tế
%IO.0	Start	Input	Nút nhấn start
%IO.1	Stop	Input	Nút nhấn stop
%IO.2	"Cam_bien_1"	Input	Cảm biến khoảng cách hồng ngoại GP2Y0A51SK0F
%IO.4	"Cam_bien_nhan"	Input	Cảm biến quang khuếch tán khoảng cách 5-100mm NPN 12~24VDC
%IO.5	Cam_bien_3	Input	Cảm biến khoảng cách vật cản E3F-DS10C4
%IW64"	Tag_1" VALUE	Input	Cảm biến khoảng cách vật cản E3F-DS10C4
%Q0.0	"Bang_tai"	Output	Động cơ giảm tốc E3F-DS10C4
%Q0.1	"Xi_lanh_3"	Output	Van điện từ 3
%Q0.2	"Xi_lanh_1"	Output	Van điện từ 1
%Q0.3	Xi_lanh_2	Output	Van điện từ 2
%Q0.4	"Dong_co"	Output	Motor giảm tốc 24VDC

3.2 Xây dựng thuật toán xử lý ảnh trên phần mềm Pycharm

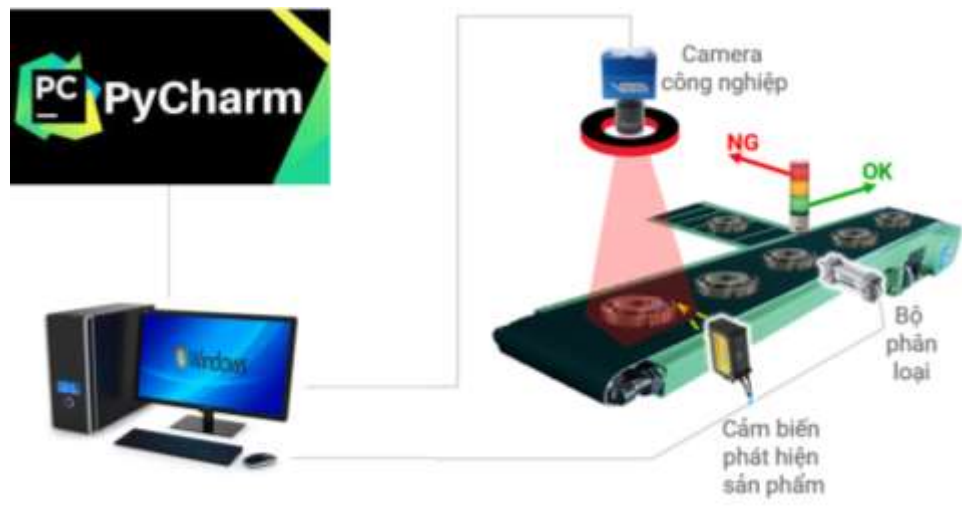
3.2.1 Giới thiệu về phần mềm

PyCharm là một trong những Integrated Development Environment (IDE - Môi trường phát triển tích hợp) phổ biến nhất dành cho lập trình Python, được phát triển bởi JetBrains. Nó cung cấp nhiều tính năng mạnh mẽ hỗ trợ phát triển ứng dụng Python một cách hiệu quả, từ viết code, debug, kiểm thử đến triển khai.



Hình 3.6 Giao diện phần mềm Pycharm

3.2.2 Giới thiệu về lý thuyết xử lý ảnh



Hình 3.7 Mô tả quá trình xử lý ảnh

Xử lý ảnh (Image Processing) là quá trình phân tích và biến đổi ảnh nhằm cải thiện chất lượng ảnh hoặc trích xuất các thông tin hữu ích từ ảnh số. Trong công nghiệp, xử lý ảnh được ứng dụng phổ biến trong các hệ thống thị giác máy (machine vision) để giám sát, kiểm tra, phân loại và điều khiển theo thời gian thực.

- ❖ Xử lý ảnh bao gồm các phương pháp toán học và thuật toán để thao tác trên dữ liệu ảnh nhằm:
 - Cải thiện chất lượng ảnh (khử nhiễu, tăng độ tương phản, chỉnh màu).
 - Trích xuất thông tin (phát hiện biên, phân đoạn ảnh, nhận dạng đối tượng).
 - Nén ảnh (giảm kích thước tệp mà vẫn giữ chất lượng).
 - Phân tích ảnh (đo lường, đếm đối tượng, phân loại).
- ❖ Các loại ảnh trong xử lý ảnh

Trong xử lý ảnh, các loại ảnh được phân loại dựa trên cách biểu diễn màu sắc và thông tin pixel. Ảnh nhị phân (Binary Image) là dạng đơn giản nhất, chỉ chứa hai mức màu: 0 (đen) và 1 (trắng), thường được ứng dụng trong nhận dạng ký tự (OCR) hoặc phát hiện đối tượng do khả năng tách biệt rõ ràng giữa đối tượng và nền. Ảnh xám (Grayscale Image) phức tạp hơn, mỗi pixel có giá trị từ 0 (đen tuyền) đến 255 (trắng tinh), biểu diễn bằng ma trận 2D, thích hợp cho các bài toán phân tích cường độ sáng như xử lý ảnh y tế. Ảnh màu (Color Image) sử dụng các mô hình màu như RGB (Red, Green, Blue), trong đó mỗi kênh màu được mã hóa 8-bit (0-255), giúp biểu diễn hình ảnh chân thực; ngoài ra còn có các mô hình khác như HSV (phân tích màu sắc, độ bão hòa, giá trị), CMYK (dùng trong in ấn), hoặc YCbCr (mã hóa video). Cuối cùng, ảnh đa

phổ và siêu phổ (Multispectral & Hyperspectral) chứa nhiều kênh phổ, ghi nhận thông tin ở các bước sóng khác nhau, được ứng dụng rộng rãi trong viễn thám, nông nghiệp thông minh và chẩn đoán y tế tiên tiến. Mỗi loại ảnh có đặc điểm và công dụng riêng, phù hợp với từng bài toán cụ thể trong xử lý ảnh.

❖ Xử lý ảnh bao gồm ba cấp độ chính:

- Tiền xử lý (Pre-processing): Cải thiện chất lượng ảnh, làm nổi bật đặc trưng quan trọng, loại bỏ nhiễu.
- Xử lý trung gian (Intermediate processing): Phát hiện biên, phân vùng ảnh, trích xuất đặc trưng.
- Xử lý cao cấp (High-level processing): Nhận dạng đối tượng, hiểu nội dung ảnh, phân tích hành vi, ...

Trong phạm vi đề tài, xử lý ảnh ở cấp độ trung gian và tiền xử lý là trọng tâm chính, đặc biệt liên quan đến việc phát hiện tem, hộp và xác định vị trí dán tem.

- Ưu điểm của xử lý ảnh trong công nghiệp:
- Tự động hóa cao: Không cần can thiệp thủ công.
- Độ chính xác cao: Xử lý dựa trên pixel → kiểm tra chính xác đến từng milimet.
- Phản hồi nhanh: Cho phép loại bỏ sản phẩm lỗi ngay lập tức.
- Khả năng tích hợp: Có thể kết hợp với PLC, SCADA, MES, ...

❖ Các kỹ thuật xử lý ảnh phổ biến:

– Biến Đổi Fourier (Fourier Transform)

- Chức năng: Chuyển ảnh từ miền không gian (spatial domain) sang miền tần số (frequency domain).
- Ứng dụng: Lọc nhiễu (noise filtering) và Nén ảnh (image compression).

– Biến đổi Wavelet (Wavelet Transform)

- Chức năng: Phân tích ảnh ở nhiều mức độ chi tiết (multi-resolution analysis).
- Ứng dụng: Nén ảnh, đặc biệt trong chuẩn JPEG2000.

– Toán tử hình thái học (Morphological Operations)

- Chức năng: Thao tác trên hình dạng của đối tượng trong ảnh nhị phân.
- Các phép toán chính: Dilation (giãn nở), Erosion (bào mòn) và Opening & Closing khử nhiễu và làm mượt biên.

– Phát hiện biên (Edge Detection)

- Chức năng: Tìm và làm nổi bật ranh giới giữa các vùng có cường độ khác nhau trong ảnh.
- Các toán tử phổ biến:
 - Sobel: Dựa trên gradient, đơn giản và hiệu quả cho phát hiện biên ngang/dọc.
 - Prewitt: Tương tự Sobel, độ chính xác thấp hơn một chút.
 - Canny: Tối ưu hóa biên mảnh, liên tục và giảm nhiễu – rất phổ biến trong thực tế.

3.2.3 Các bước để xử lý ảnh

Kiểm tra vị trí dán tem bằng cách xác định tọa độ tâm của hộp và tọa độ tâm của tem, sau đó so sánh. Nếu lệch quá ngưỡng cho phép tem bị dán sai vị trí.

❖ Các bước xử lý ảnh:

Bước 1: Đọc và tiền xử lý ảnh

```
ret, frame = cap.read()

if ret:
    frame = cv2.resize(frame, dsize=(480, 320), interpolation=cv2.INTER_LINEAR)
    frame_org = frame.copy()
```

Đọc frame từ camera bằng cv2.VideoCapture

Resize hình ảnh về kích thước cố định (480x320) để xử lý đồng nhất

Tạo bản sao của frame gốc để sử dụng sau này

Bước 2: Chuyển đổi không gian màu

```
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

Chuyển từ không gian màu BGR sang HSV để dễ dàng tách màu hơn

- Hue (Màu sắc): Giá trị từ 0-179 (OpenCV scale).
- Saturation (Độ bão hòa): 0-255.
- Value (Độ sáng): 0-255.

HSV (Hue-Saturation-Value) giúp nhận diện màu sắc tốt hơn trong 5 sáng khác nhau

Bước 3: Tạo Mask để tách đối tượng

```
mask_hop = cv2.inRange(hsv, lowerb: (0, 0, 160), upperb: (100, 215, 230))
mask_tem = cv2.inRange(hsv, lowerb: (10, 10, 10), upperb: (255, 255, 255))
```

Tạo mask cho hộp với ngưỡng màu từ (0,0,160) đến (100,215,230) trong HSV

- Hue: từ 0 đến 100 → gồm đỏ, vàng nhạt, cam nhạt, nâu nhạt
- Saturation: từ 0 → 215 → chấp nhận cả vùng ít màu (xám, trắng nhạt...)
- Value (sáng): từ 160 đến 230 → lọc các vùng sáng rõ

Tạo mask cho tem với ngưỡng màu rộng hơn từ (10,10,10) đến (255,255,255)

- Hue: 10–255 → bao gần như toàn bộ dải màu sắc
- Saturation: 10–255 → tránh lọc vùng trắng/đen
- Value: 10–255 → chấp nhận mọi độ sáng trừ hoàn toàn đen

cv2.inRange giữ lại các pixel nằm trong khoảng màu xác định

Bước 4: Áp dụng Mask để tách đối tượng

```
result_hop = cv2.bitwise_and(frame, frame, mask=mask_hop)
result_tem = cv2.bitwise_and(frame, frame, mask=mask_tem)
```

Áp dụng mask lên frame gốc để tách riêng hình ảnh hộp và tem

cv2.bitwise_and thực hiện phép AND bitwise giữa frame và mask

Bước 5: Phát hiện biên (Edge detection)

Áp dụng thuật toán Canny Edge Detection để xác định các biên có độ tương phản rõ rệt trong ảnh. Đây là bước quan trọng nhằm tách biệt vật thể (tem, hộp) ra khỏi nền.

- Mục đích: Xác định các đường viền sắc nét – là biên của vật thể (hộp, tem).
- Tham số: threshold1 và threshold2 để điều chỉnh độ nhạy biên.
- Ngưỡng dưới là 30, ngưỡng trên là 255

```
edged_hop = cv2.Canny(result_hop, 30, 255)
edged_tem = cv2.Canny(result_tem, 30, 255)
```

Bước 6: Tìm contour (Đường bao)

Sử dụng hàm cv2.findcontour() để tính diện tích mỗi contour, loại bỏ các vùng nhiễu nhỏ hoặc không đúng kích thước chuẩn. Đồng thời, xác định số cạnh của contour (bằng polygonal approximation) để xác minh xem đó có phải là hình chữ nhật hay không.

- Mục đích: Lọc ra những contour có diện tích phù hợp (ví dụ: loại bỏ vật thể nhỏ, không phải tem hay hộp).
- Ý nghĩa: Hộp và tem thường có diện tích lớn nhất hoặc nằm trong khoảng diện tích mong đợi.

```
contours_hop, hierarchy_hop = cv2.findContours(edged_hop, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
contours_tem, hierarchy_tem = cv2.findContours(edged_tem, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
```

Tìm contours (đường bao) của hộp và tem từ ảnh biên

Với hộp: chỉ lấy contours ngoài cùng (RETR_EXTERNAL)

Với tem: lấy tất cả contours (RETR_TREE)

CHAIN_APPROX_NONE lưu tất cả các điểm trên contour

Bước 7: Phân tích contour và xác định vị trí

Xử lý contour của hộp

```
for cnt in contours_hop:
    a = cv2.contourArea(cnt)
    if a > 10000:
        M = cv2.moments(cnt)
        if M['m00'] != 0:
            cpx = int(M['m10'] / M['m00'])
            cpy = int(M['m01'] / M['m00'])
            if cpx < 250:
                cv2.drawContours(frame_org, contours=[cnt], -1, color=(0, 255, 0), thickness=2)
                cv2.circle(frame_org, center=(cpx, cpy), radius=7, color=(0, 0, 255), -1)
                self.x_hop = cpx
                self.y_hop = cpy
        break
```

Xử lý contour của tem

```
for cnt in contours_tem:
    a = cv2.contourArea(cnt)
    if 2000 < a < 15000:
        M = cv2.moments(cnt)
        if M['m00'] != 0:
            cpx = int(M['m10'] / M['m00'])
            cpy = int(M['m01'] / M['m00'])
            if cpx < 250:
                cv2.drawContours(frame_org, contours=[cnt], -1, color=(255, 255, 0), thickness=2)
                cv2.circle(frame_org, center=(cpx, cpy), radius=7, color=(255, 0, 255), -1)
                self.x_tem = cpx
                self.y_tem = cpy
        break
```

Lọc contours theo diện tích:

- Hộp: diện tích > 10000 pixel
- Tem: diện tích từ 2000-15000 pixel

Dùng moment hình ảnh để tính toán tâm hình học của tem và hộp:

$$x_{centroid} = \frac{M_{10}}{M_{00}}; y_{centroid} = \frac{M_{01}}{M_{00}} \quad (3.1)$$

Việc tính tâm cho phép kiểm tra mức độ lệch vị trí dán tem thông qua khoảng cách giữa tâm tem và tâm hộp.

M['m00']: diện tích contour

M['m10']/M['m00']: tọa độ x của tâm

M['m01']/M['m00']: tọa độ y của tâm

- Mục đích: Tính toán tọa độ tâm của contour (điểm trung bình trọng tâm).
- Ứng dụng: Dùng để so sánh tâm hộp và tâm tem.

Vẽ contour và đánh dấu tâm lên hình ảnh gốc

Lưu tọa độ tâm vào các biến x_hop, y_hop, x_tem, y_tem

Cuối cùng, chương trình kiểm tra độ lệch giữa tâm hộp và tâm tem để xác định tem có bị dán sai vị trí hay không:

```
if (abs(self.x_hop - self.x_tem) >= self.lech_x_max or abs(self.y_hop - self.y_tem) >= self.lech_y_max) \
    and self.x_hop > 0 and self.x_tem > 0 and self.y_hop > 0 and self.y_tem > 0:
    self.sai_nhan = 1
    print(self.sai_nhan)
print(self.x_hop, self.x_tem, self.y_hop, self.y_tem)
```

Khoảng cách giữa hai tâm được so sánh với một ngưỡng đã cài đặt sẵn. Nếu vượt ngưỡng, hệ thống sẽ cảnh báo “tem dán sai vị trí” và loại sản phẩm khỏi dây chuyền.

- Mục tiêu: So sánh tọa độ tâm hộp và tem.
- Ngưỡng lệch cho phép: Đặt tùy theo yêu cầu kỹ thuật (3 mm tương đương vài pixel trên ảnh).
- Note: Nếu tâm tem khác tâm hộp lớn hơn một ngưỡng mình cài đặt thì sai vị trí.

3.2.4 Điều khiển PLC và hệ thống phản ứng

❖ Điều khiển PLC (Siemens S7 qua thư viện snap7)

Chương trình giao tiếp với PLC để điều khiển các thiết bị phần cứng. Các lệnh điều khiển đều thông qua đọc/ghi dữ liệu vào Data Block (DB) của PLC.

❖ **Các chức năng điều khiển:**

- Khởi động/Dừng hệ thống:

```
def Start(self):
    reading = plc.db_read(1, 0, 1)
    snap7.util.set_bool(reading, byte_index: 0, bool_index: 0, value: True)
    plc.db_write(1, 0, reading)

def Stop(self):
    reading = plc.db_read(1, 0, 1)
    snap7.util.set_bool(reading, byte_index: 0, bool_index: 0, value: False)
    plc.db_write(1, 0, reading)
```

- Chế độ Auto/Manual:

```
def Auto(self):
    reading = plc.db_read(1, 0, 1)
    snap7.util.set_bool(reading, byte_index: 0, bool_index: 1, value: False)
    plc.db_write(1, 0, reading)

def Manual(self):
    reading = plc.db_read(1, 0, 1)
    snap7.util.set_bool(reading, byte_index: 0, bool_index: 1, value: True)
    plc.db_write(1, 0, reading)
```

- Điều khiển thiết bị công nghiệp:

```
def BangTai(self):
    reading = plc.db_read(1, 0, 1)
    temp = snap7.util.get_bool(reading, byte_index: 0, bool_index: 2)
```

❖ **Giải thích PLC Communication:**

- plc.db_read(DB_number, offset, size): Đọc dữ liệu từ PLC.
- snap7.util.set_bool(): Ghi giá trị boolean vào vị trí bit chỉ định.
- DB1 được sử dụng làm vùng nhớ trao đổi dữ liệu:

- Bit 0: Start/Stop
- Bit 1: Auto/Manual
- Bit 2-6: Điều khiển thiết bị.

3.2.5 Thiết kế giao diện GUI trên Python

Giao diện người dùng đồ họa (GUI – Graphical User Interface) là một phần quan trọng trong các hệ thống điều khiển tự động. GUI giúp người vận hành có thể theo dõi, điều khiển và tương tác với hệ thống một cách trực quan và dễ dàng hơn so với giao diện dòng lệnh.

Python là ngôn ngữ lập trình bậc cao, dễ học và có hệ sinh thái phong phú. Trong thiết kế GUI, Python cung cấp nhiều thư viện hỗ trợ như:

- Tkinter: thư viện GUI tích hợp sẵn trong Python, dễ dùng cho các ứng dụng đơn giản.
- PyQt hoặc PySide: hỗ trợ giao diện chuyên nghiệp và phức tạp hơn.
- Kivy: hỗ trợ giao diện trên đa nền tảng, cả máy tính và thiết bị cảm ứng.
- CustomTkinter: phiên bản mở rộng của Tkinter với giao diện hiện đại, thân thiện hơn.

Trong mô hình dán tem, thư viện Tkinter thường được lựa chọn do tính đơn giản, đủ dùng cho ứng dụng điều khiển giám sát trong xưởng.

Các thành phần trong giao diện GUI

Bảng 3.2 Các thành phần trong giao diện BUI

Thành phần	Vai trò chính
Cửa sổ chính (Main window)	Nơi chứa tất cả các thành phần giao diện
Nút nhấn (Button)	Kích hoạt các hành động: bắt đầu chạy và dừng hệ thống, ...
Nhãn (Label)	Hiển thị thông tin: số lượng sản phẩm đạt lỗi, biến dạng.
Khung ảnh (Canvas/Label + ảnh)	Hiển thị ảnh từ camera sau dán tem

Luồng hoạt động giao diện

Giao diện hoạt động theo quy trình:

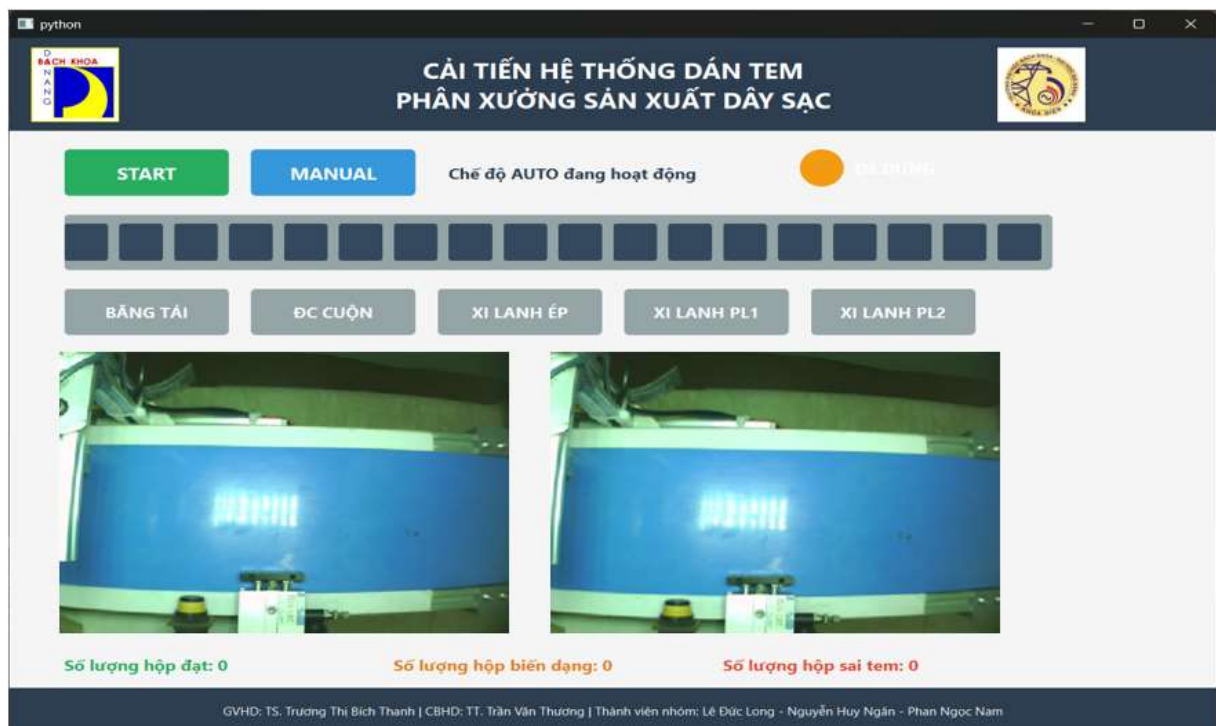
1. Người vận hành mở ứng dụng giao diện.
2. Giao diện kết nối với camera và hệ thống điều khiển (PLC, cảm biến,...).
3. Ảnh sản phẩm được chụp, xử lý và hiển thị.

4. Kết quả kiểm tra (đạt, sai tem, biến dạng) được hiển thị kèm số lượng.
5. Người dùng có thể bắt đầu/dừng hệ thống bằng các nút điều khiển.

Tích hợp xử lý ảnh trong GUI

- Ảnh từ camera được xử lý bằng thư viện OpenCV.
- Ảnh đầu ra sau xử lý được chuyển đổi thành định dạng hiển thị (PIL.Image → PhotoImage) để gắn vào GUI.
- Trạng thái (ví dụ: tem lệch tâm) được biểu diễn bằng màu sắc hoặc nhãn hiển thị.

3.2.6 Màn hình chính của hệ thống



Hình 3.8 Màn hình điều khiển hệ thống

Các chức năng có trên màn hình hệ thống

Bảng 3.3 Các chức năng có trên màn hình hệ thống

STT	Nút chức năng	Mô tả
1	Start	Khởi động hệ thống dán tem tự động.
2	Stop	Dừng hệ thống khẩn cấp hoặc kết thúc quy trình.
3	Auto	Chế độ tự động: Hệ thống tự xử lý ảnh và dán tem không cần can thiệp thủ công.
4	Manual	Chế độ thủ công: Người vận hành kiểm tra ảnh, căn chỉnh tem từng bước.
5	Băng tải	Nạp nguyên liệu vào hệ thống.

6	Đc cuộn	Điều khiển động cơ cuộn dây di chuyển sản phẩm qua băng tải.
7	Xi lanh ép	Điều khiển xi lanh ép tem lên bề mặt sản phẩm.
8	Xi lanh pl1/pl2	Các xi lanh phân loại các sản phẩm lỗi
9	Màn hình camera	Màn hình hiển thị chi tiết hình ảnh của sản phẩm khi đi qua camera
10	Số lượng hộp	Hiển thị số lượng các loại sản phẩm gồm: Sản phẩm đạt yêu cầu, sản phẩm lỗi biến dạng, sản phẩm lỗi tem.

3.3 Kết luận

Trong chương này, nhóm đã tập trung triển khai hai nội dung cốt lõi của hệ thống dán tem tự động: xây dựng thuật toán điều khiển toàn bộ quy trình hoạt động và thiết lập cấu hình cho các thiết bị phần cứng trên nền tảng phần mềm điều khiển. Kết quả đạt được có thể tóm tắt như sau:

- Đã xây dựng được chương trình điều khiển tổng thể của hệ thống, bao gồm các trạng thái: kiểm tra đầu vào, cấp tem, dán tem, kiểm tra sau dán và phân loại sản phẩm. Quy trình này đảm bảo tính tự động hóa và tính logic, giúp thiết bị hoạt động liên tục và chính xác.
- Hoàn thiện thuật toán xử lý ảnh tích hợp với hệ thống camera, giúp phát hiện các lỗi như móp hộp, hộp thiếu, lệch vị trí dán tem, tem bị bong tróc... Việc xử lý được triển khai trên phần mềm Python sử dụng thư viện OpenCV, giúp hệ thống đưa ra tín hiệu điều khiển chính xác cho PLC.
- Đã cấu hình thành công các thiết bị điều khiển chính: PLC, camera, cảm biến, motor dán tem và các cơ cấu chấp hành, thông qua phần mềm lập trình tương ứng như TIA Portal, Python GUI. Các thiết bị được giao tiếp với nhau qua tín hiệu I/O và truyền thông phù hợp.
- Giao diện điều khiển được thiết kế đơn giản, trực quan, dễ thao tác, hiển thị trạng thái hoạt động của hệ thống và kết quả kiểm tra ảnh theo thời gian thực, giúp vận hành và xử lý sự cố dễ dàng.

CHƯƠNG 4. KIỂM NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ ĐẠT ĐƯỢC

4.1 Mô hình hệ thống



Hình 4.1 Ảnh tổng thể của mô hình

4.1.1 Chi tiết các bộ phận của hệ thống

Cơ cấu xử lý hộp bị biến dạng



Hình 4.2 Cơ cấu xử lý hộp biến dạng

Cơ cấu dán tem



Hình 4.3 Cơ cấu dán tem

Cơ cấu xử lý ảnh



Hình 4.4 Cơ cấu xử lý ảnh

Tủ điện PLC



Hình 4.5 Tủ điện PLC

4.1.2 Video hoạt động chi tiết

[Video hoạt động chi tiết của hệ thống](#)

SVTH1: Lê Đức Long
SVTH2: Nguyễn Huy Ngân
SVTH3: Phan Ngọc Nam

CBHD1: TS. Trương Thị Bích Thanh
CBHD2: TT. Trần Văn Thương

4.2 Đánh giá hiệu quả vận hành thực tế của sản phẩm

4.2.1 Phân loại sản phẩm

❖ Sản phẩm đạt yêu cầu là:

- Vị trí dán tem chính xác

Tem được dán gần chính giữa bề mặt hộp, không bị lệch quá giới hạn cho phép.

Khoảng cách đều từ tem đến mép hộp cho thấy hệ thống dán tem căn chỉnh tốt.

- Tem không bị nhăn, gấp hay bong

Bề mặt tem phẳng, không có dấu hiệu bong tróc hay gấp nếp.

Điều này cho thấy lực ép và tốc độ dán của xi lanh dán tem ổn định.

- Kích thước tem đúng tiêu chuẩn.

Tem có kích thước đồng đều, không bị rách hoặc sai kích thước.

- Không có lỗi trên hộp.

Hộp không bị móp, méo hay biến dạng tại vùng dán tem, giúp bề mặt dán ổn định.



Hình 4.6 Sản phẩm đạt yêu cầu

❖ Sản phẩm biến dạng bề mặt dán tem

- Hộp bị móp và nhăn nheo

Bề mặt dán không còn phẳng, điều này có thể gây lỗi dán tem như: tem bị bong, dán lệch, không dính hoàn toàn hoặc không dán được.

- Khoảng cách đo được ngoài giới hạn cho phép

Theo tiêu chuẩn hệ thống: Nếu khoảng cách từ cảm biến đến bề mặt lớn hơn 1.27V ($\approx 7.6cm$) (hộp bị lõm vào) hoặc bé hơn 1.27V ($\approx 7.6cm$) (hộp bị phồng lên) thì hộp sẽ bị xem là hộp biến dạng.



Hình 4.7 Sản phẩm biến dạng bề mặt dán tem

❖ Sản phẩm lỗi dán tem

- Lỗi lệch theo trục Y (lệch ngang)

Là hiện tượng tem bị dán sai vị trí theo phương ngang so với vị trí chuẩn (trái hoặc phải).

- Nguyên nhân phổ biến:

Sản phẩm trên băng chuyền bị lệch hướng di chuyển.

Đầu dán tem không căn giữa so với sản phẩm.

- Lỗi lệch theo trục X (lệch dọc):

Là hiện tượng tem bị dán lệch theo phương dọc so với vị trí chuẩn (lên hoặc xuống).

- Nguyên nhân phổ biến:

Tốc độ băng tải không phù hợp.

Cảm biến phát hiện vị trí sản phẩm sai hoặc trễ.



Hình 4.8 Sản phẩm lỗi dán tem

4.2.2 Kết quả vận hành thực tế

Hệ thống sử dụng cảm biến hồng ngoại để quét bề mặt dán tem của sản phẩm. Nếu phát hiện biến dạng, PLC sẽ điều khiển xi lanh đẩy sản phẩm lỗi ra khỏi băng chuyền và tự động tăng giá trị đếm "Số lượng hộp biến dạng" lên 1.



Hình 4.9 Lỗi bề mặt dán tem

Sản phẩm đạt tiêu chuẩn về hình dạng sẽ được băng tải vận chuyển tới vị trí dán tem. Khi cảm biến quang phát hiện sản phẩm đi qua, PLC sẽ nhận tín hiệu và điều khiển xi lanh khí nén thực hiện quy trình dán tem tự động.



Hình 4.10 Xi lanh ép thực hiện dán tem

Quy trình xử lý lỗi vị trí tem: Khi hệ thống phát hiện tem bị lệch vị trí, PLC sẽ kích hoạt xi lanh đẩy sản phẩm lỗi ra khỏi băng chuyền và tự động tăng biến đếm "Số lượng hộp sai tem" lên 1. Các sản phẩm đạt chuẩn tiếp tục quy trình với biến đếm "Số lượng hộp đạt" được cập nhật tương ứng.



Hình 4.11 Xi lanh đẩy hộp dán tem bị lỗi

❖ Giao diện giám sát.

Hệ thống giám sát đã thể hiện hiệu quả cao trong quá trình vận hành thực tế, liên tục cập nhật chính xác số lượng sản phẩm lên màn hình điều khiển sau mỗi lần kiểm tra. Mỗi khi hộp sản phẩm đi qua trạm kiểm tra, hệ thống cảm biến và camera sẽ tự động phân tích chất lượng tem, sau đó truyền tín hiệu về PLC để xử lý. Ngay lập tức, màn hình hiển thị đầy đủ các thông số quan trọng bao gồm tổng số sản phẩm đã xử lý, số sản phẩm đạt chuẩn và không đạt chuẩn.



Hình 4.12 Màn hình giám sát cập nhật số lượng sản phẩm

4.3 Đánh giá và cải tiến

- Môi trường thử nghiệm:

Nhà xưởng 50m², nhiệt độ 25±2°C.

Nguồn điện 220V±5%.

- Số lượng kiểm tra 300 hộp chia làm 3 đợt kiểm tra (Trong đó có 15 hộp bị biến dạng bề mặt dán tem)

4.3.1 Đánh giá độ chính xác

Bảng 4.1 Bảng đánh giá độ chính xác

Thông số	Đợt 1	Đợt 2	Đợt 3
Số lượng sản phẩm	100	100	100
Tốc độ dán tem (tem/phút)	18 tem/phút	20 tem/phút	17 tem/phút
Tỷ lệ phát hiện lỗi biến dạng bề mặt	100%	100%	100%
Tỷ lệ dán tem đạt yêu cầu	94%	96%	97%
Tỷ lệ phát hiện tem lỗi	98%	97%	97%

4.3.2 Đánh giá tính ổn định.

Hệ thống được kiểm tra qua 3 đợt vận hành liên tục trong điều kiện nhà xưởng tiêu chuẩn. Kết quả cho thấy tỷ lệ hoạt động ổn định là 95 – 97%.

Nguyên nhân chính gây gián đoạn:

- Kẹt cơ khí do tem/hộp lệch vị trí
- Nhiều tín hiệu: Nhiều sáng phần xử lý ảnh, nhiều cảm biến tại cơ cấu dán tem
- Mất tín hiệu với PLC

4.3.3 Đánh giá kỹ thuật

Bảng 4.2 Đánh giá kỹ thuật

Vấn đề	Nguyên nhân	Giải pháp khắc phục
Kẹt hộp khi dán tem	Xi lanh ép qua sâu	Chỉnh sửa đầu vào khí nén ở van xi lanh
Mất kết nối PLC	Dây cáp kết nối không tốt	Kiểm tra lại dây cáp và cổng kết nối
Nhiều độ sáng	Do ánh sáng đèn và môi trường xung quanh	Chỉnh sửa góc độ chiếu sáng và che chắn ánh sáng xung quanh

4.3.4 Đánh giá tổng quát và cải tiến

Hệ thống dán tem tự động sau quá trình kiểm nghiệm đã đáp ứng cơ bản các yêu cầu kỹ thuật đề ra, thể hiện qua khả năng vận hành ổn định với tỷ lệ chính xác đạt 95-97% và tốc độ xử lý trung bình 15 - 20 tem/phút. Tuy nhiên, vẫn tồn tại một số hạn chế cần được khắc phục để nâng cao hiệu quả tổng thể.

❖ **Ưu điểm nổi bật:**

- Độ chính xác trong phát hiện và xử lý sản phẩm lỗi đạt 97%
- Cơ chế vận hành tự động giảm sự phụ thuộc vào nhân công thủ công
- Thời gian phục hồi sự cố nhanh đảm bảo tính liên tục của dây chuyền

❖ **Hạn chế cần cải tiến:**

- Độ nhạy với điều kiện môi trường: Hệ thống vẫn bị ảnh hưởng bởi nhiễu ánh sáng và dao động điện áp, dẫn đến một số cảnh báo sai. Giải pháp đề xuất bao gồm lắp đặt thêm bộ lọc nhiễu và sử dụng nguồn điện ổn định hơn.
- Khả năng xử lý sản phẩm đa dạng: Hiện tại, hệ thống chưa tối ưu cho các loại hộp có bề mặt cong hoặc chất liệu đặc biệt. Cần nghiên cứu thêm về cơ cấu ép tem linh hoạt để mở rộng phạm vi ứng dụng.
- Chi phí bảo trì: Một số linh kiện như cảm biến và xi lanh cần được theo dõi thường xuyên để tránh hao mòn nhanh. Việc chuyển sang sử dụng các thiết bị có tuổi thọ cao hơn sẽ giúp giảm chi phí vận hành dài hạn.

❖ **Định hướng và phát triển:**

Trong tương lai, hệ thống cần được tích hợp thêm các công nghệ như trí tuệ nhân tạo (AI) để nâng cao khả năng tự học và thích nghi với các dạng lỗi phức tạp. Đồng thời, việc số hóa quy trình giám sát thông qua nền tảng IoT sẽ giúp theo dõi và tối ưu hiệu suất theo thời gian thực.

KẾT LUẬN

Sau một quá trình nghiên cứu, thiết kế và triển khai thực hiện, nhóm sinh viên đã hoàn thành đề tài: “Cải tiến hệ thống dán tem của phân xưởng sản xuất dây sạc”.

Qua quá trình thực hiện, nhóm rút ra những kết luận tổng quát như sau:

- Đề tài đã xây dựng thành công một mô hình thu nhỏ mô phỏng quy trình dán tem tự động có tích hợp hệ thống xử lý ảnh nhằm kiểm tra sản phẩm trước và sau khi dán.
- Hệ thống được thiết kế đồng bộ bao gồm phần cơ khí, tủ điện điều khiển, hệ thống camera kiểm tra và phần mềm xử lý ảnh, đảm bảo hoạt động theo đúng quy trình công nghệ đã đề ra.
- Mô hình không chỉ dừng lại ở tính trình diễn mà còn có khả năng áp dụng thực tế sau khi nâng cấp công suất và khả năng chịu tải.

Kết quả đạt được:

- Thiết kế được một hệ thống cơ điện tử hoàn chỉnh bao gồm: bộ cấp tem, cơ cấu dán, hệ thống camera kiểm tra, thiết bị điều khiển (PLC), và giao diện điều khiển trên máy tính.
- Xây dựng thuật toán xử lý ảnh cơ bản (dùng thư viện OpenCV) để xác định được lỗi trước và sau dán như: bề mặt dán tem bị móp méo, vị trí lệch, tem thiếu,...

Đóng góp của đề tài:

- Góp phần giải quyết bài toán thực tế về năng suất và chất lượng trong khâu đóng gói – dán tem sản phẩm dây sạc.
- Là tài liệu tham khảo và nền tảng thử nghiệm hữu ích cho các đề tài nghiên cứu ứng dụng công nghệ xử lý ảnh trong dây chuyền sản xuất.

Giới hạn và đề xuất:

- Do giới hạn về thời gian và điều kiện thực tế, mô hình mới dừng lại ở mức mô hình thu nhỏ, chưa đạt hiệu suất công nghiệp.
- Độ chính xác xử lý ảnh còn phụ thuộc vào điều kiện ánh sáng và độ phân giải camera.

- Cần nghiên cứu mở rộng với các công nghệ cao hơn như AI/Deep Learning để tăng khả năng nhận diện lỗi phức tạp hơn trong tương lai.

Kiến nghị

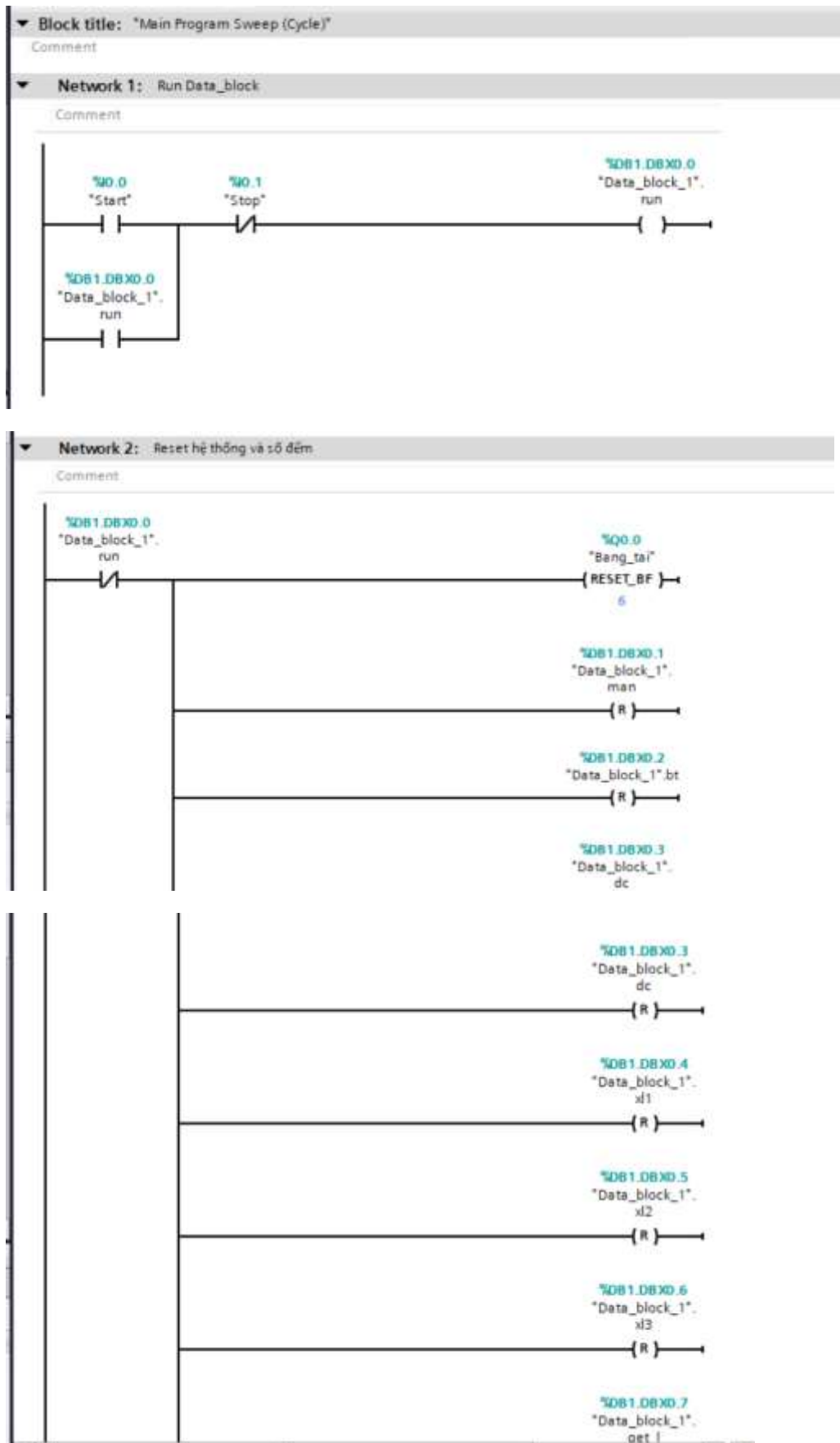
- Đề nghị nhà trường, khoa và giảng viên tiếp tục tạo điều kiện cho sinh viên tiếp cận thiết bị thực tế, đặc biệt là các module xử lý ảnh công nghiệp chuyên dụng.
- Mong muốn có cơ hội hợp tác với doanh nghiệp để triển khai thử nghiệm thực tế và nâng cấp mô hình lên cấp độ sản xuất.

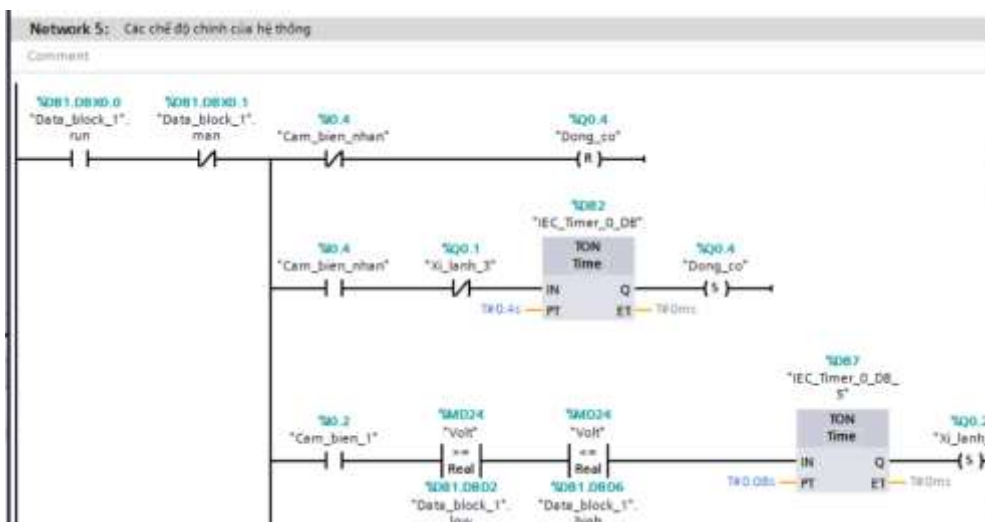
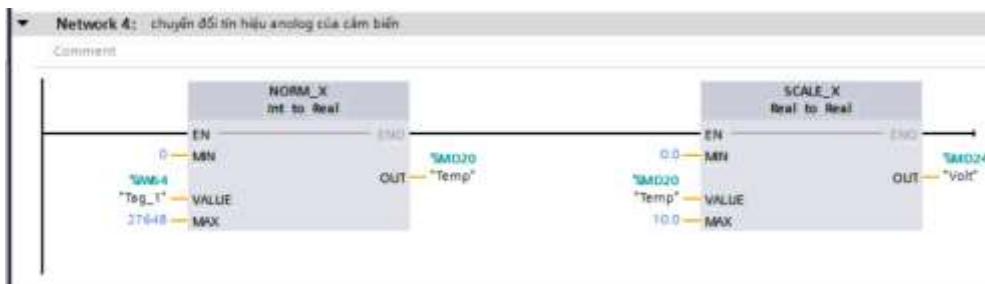
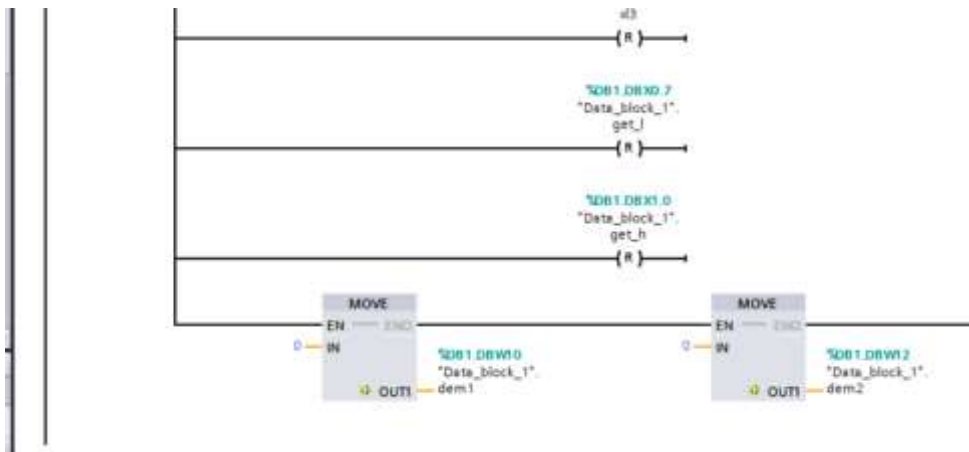
TÀI LIỆU THAM KHẢO

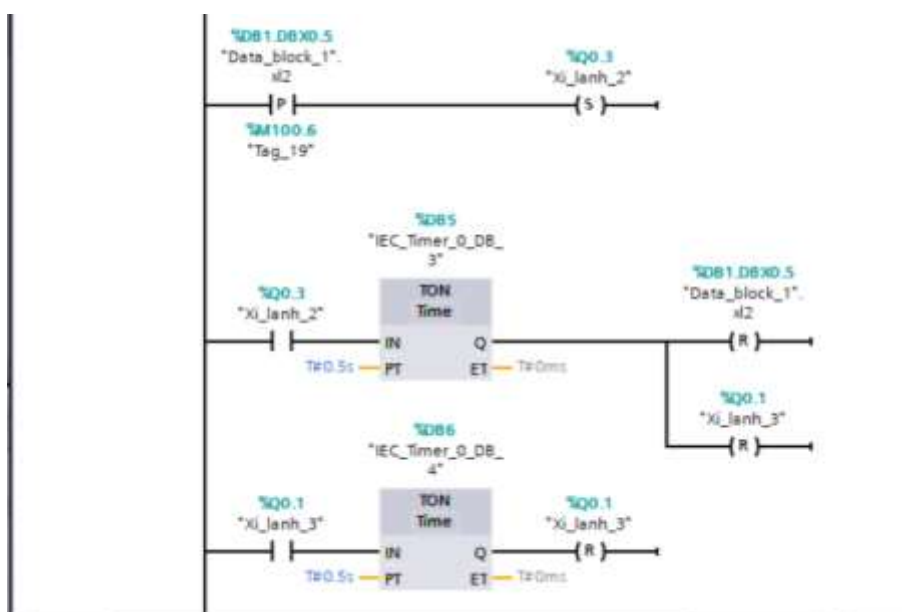
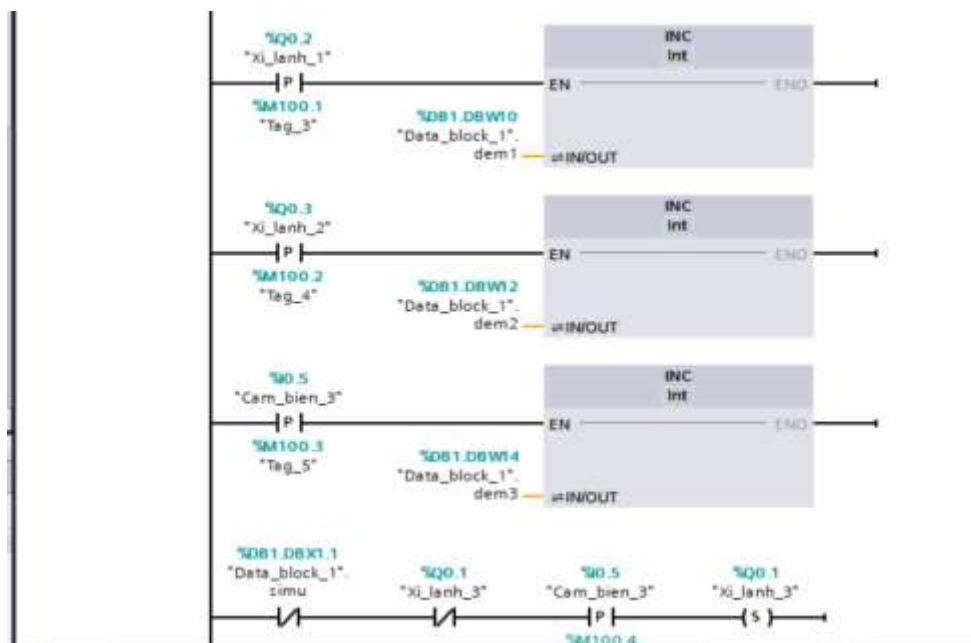
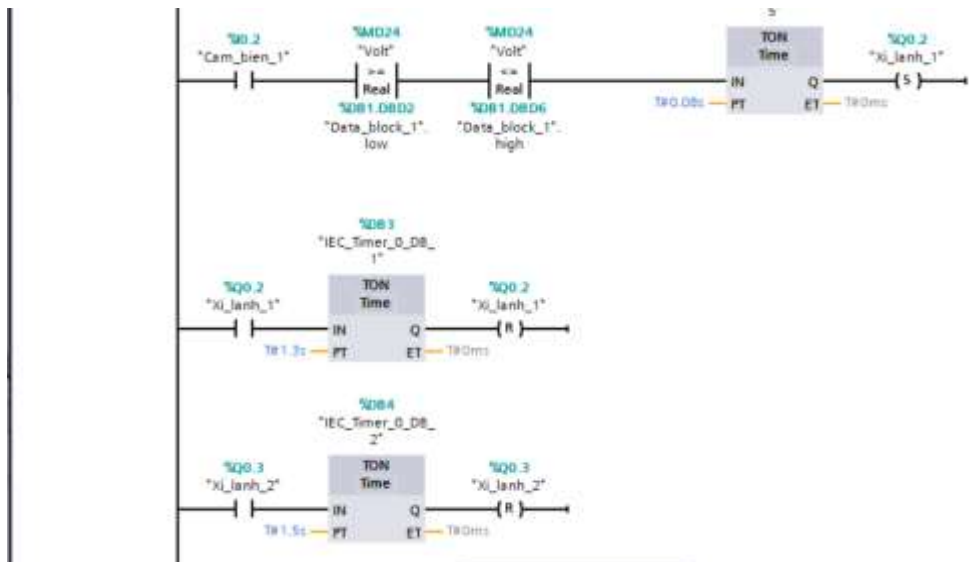
- [1]. [OpenCV Team. \(2023\). OpenCV Documentation.](#)
- [2]. Rosebrock, A. (2022). Deep Learning for Computer Vision with Python. PyImageSearch.
- [3]. Bradski, G. & Kaehler, A. (2020). Learning OpenCV 4 Computer Vision with Python 3 (3rd ed.). Packt Publishing.
- [4]. <http://siemens-vietnam.vn/product/simatic-s7-1200-cpu-1212c-dc-dc-dc-6es7212-1ae40-0xb0/>
- [5]. <https://www.alldatasheet.com/datasheetpdf/pdf/1134370/XLSEMI/XL4016E1.html>
- [6]. <https://hshop.vn/cam-bien-vat-can-hong-ngoai-e3f-ds10c4-adjustable-ir-infrared-proximity-sensor>
- [7]. <https://media.digikey.com/pdf/Data%20Sheets/Omron%20PDFs/E3Z.pdf>
- [8]. <https://dientutuyetnga.com/products/gp2y0a51sk0f-cam-bien-khoang-cach-hong-ngoai>
- [9]. <https://auvietco.com.vn/van-dien-tu-khi-nen-airtac-4v210-08>

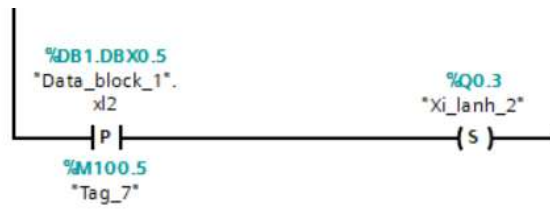
PHỤ LỤC

Code PLC



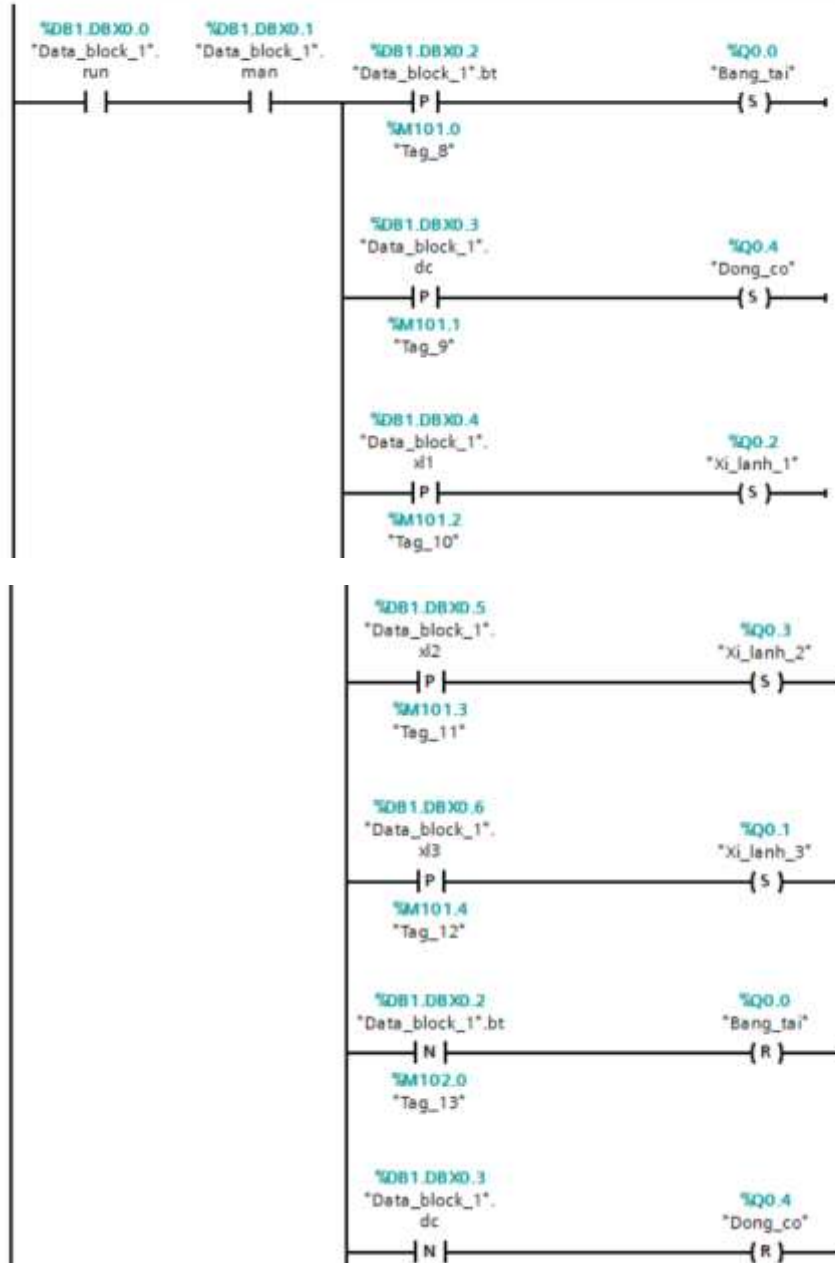


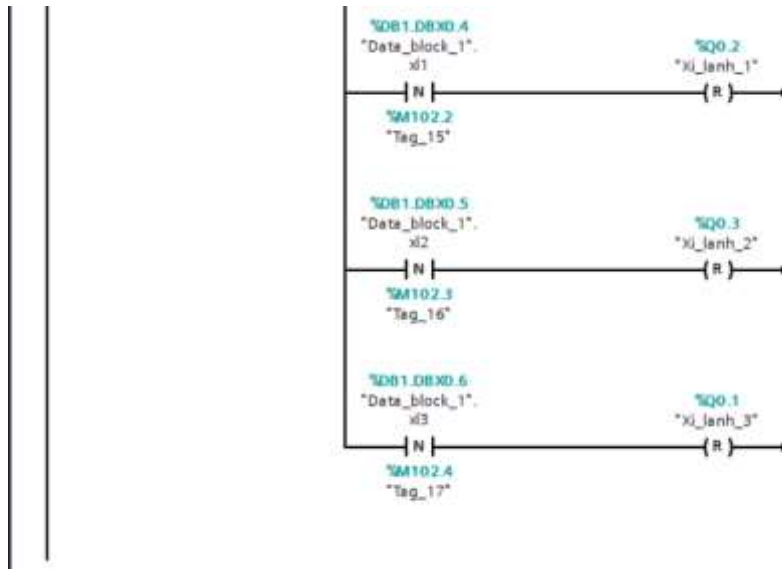




Network 6: Chế độ man

Comment





Code Python

```

9   from PyQt6 import QtCore, QtGui, QtWidgets
10  import anap7
11  import cv2
12  import numpy as np
13  from threading import Timer
14  import math
15  from PyQt6.QtGui import QImage
16
17  plc = snap7.client.Client()
18  plc.connect('192.168.0.1', 0, 1)
19
20  cap = cv2.VideoCapture(0)
21  # cap = cv2.VideoCapture("v1.mp4")
22
23
24  class UI_MainWindow(object):
25      def __init__(self):
26          self.connected = 1
27          self.sai_nhan = 0
28          self.lech_x_max = 20
29          self.lech_y_max = 20
30          self.x_hop = 0
31          self.y_hop = 0
32          self.x_tem = 0
33          self.y_tem = 0
  
```

```

34  def setupUI(self, MainWindow):
35      MainWindow.setObjectName("MainWindow")
36      MainWindow.resize(1261, 836)
37      MainWindow.setStyleSheet("background-color: rgb(255, 178, 127);")
38      self.centralwidget = QtWidgets.QWidget(parent=MainWindow)
39      self.centralwidget.setObjectName("centralwidget")
40      self.labelCam1 = QtWidgets.QLabel(parent=self.centralwidget)
41      self.labelCam1.setGeometry(QtCore.QRect(260, 110, 488, 120))
42      self.labelCam1.setStyleSheet("background-color: rgb(170, 178, 127);")
43      self.labelCam1.setText("")
44      self.labelCam1.setObjectName("labelCam1")
45      self.labelCam2 = QtWidgets.QLabel(parent=self.centralwidget)
46      self.labelCam2.setGeometry(QtCore.QRect(770, 110, 488, 120))
47      self.labelCam2.setStyleSheet("background-color: rgb(170, 178, 127);")
48      self.labelCam2.setText("")
49      self.labelCam2.setObjectName("labelCam2")
50      self.labelLoai1 = QtWidgets.QLabel(parent=self.centralwidget)
51      self.labelLoai1.setGeometry(QtCore.QRect(288, 460, 381, 41))
52      font = QtGui.QFont()
53      font.setFamily("Times New Roman")
54      font.setPointSize(24)
55      font.setBold(True)
56      font.setItalic(False)
57      font.setWeight(75)
58      self.labelLoai1.setFont(font)
59      self.labelLoai1.setStyleSheet("color: rgb(170, 0, 127);")
  
```

```

60 self.labelLoai1.setStyleSheet("color: rgb(170, 0, 127);")
61 self.labelLoai1.setObjectName("labelLoai1")
62 self.labelLoai2 = QtWidgets.QLabel(parent=self.centralwidget)
63 self.labelLoai2.setGeometry(QtCore.QRect(288, 328, 381, 41))
64 font = QtGui.QFont()
65 font.setFamily("Times New Roman")
66 font.setPointSize(24)
67 font.setBold(True)
68 font.setItalic(False)
69 font.setWeight(75)
70 self.labelLoai2.setFont(font)
71 self.labelLoai2.setStyleSheet("color: rgb(170, 0, 127);")
72 self.labelLoai2.setObjectName("labelLoai2")
73 self.labelLoai3 = QtWidgets.QLabel(parent=self.centralwidget)
74 self.labelLoai3.setGeometry(QtCore.QRect(288, 388, 381, 41))
75 font = QtGui.QFont()
76 font.setFamily("Times New Roman")
77 font.setPointSize(24)
78 font.setBold(True)
79 font.setItalic(False)
80 font.setWeight(75)
81 self.labelLoai3.setFont(font)
82 self.labelLoai3.setStyleSheet("color: rgb(170, 0, 127);")
83 self.labelLoai3.setObjectName("labelLoai3")
84 self.label_svth = QtWidgets.QLabel(parent=self.centralwidget)

```

```

85 self.label_svth.setGeometry(QtCore.QRect(0, 688, 761, 111))
86 font = QtGui.QFont()
87 font.setFamily("Times New Roman")
88 font.setPointSize(14)
89 font.setItalic(True)
90 self.label_svth.setFont(font)
91 self.label_svth.setStyleSheet("background-color: rgb(255, 255, 255);")
92 self.label_svth.setObjectName("label_svth")
93 self.pushButtonStart = QtWidgets.QPushButton(parent=self.centralwidget)
94 self.pushButtonStart.setGeometry(QtCore.QRect(18, 118, 151, 51))
95 font = QtGui.QFont()
96 font.setFamily("Times New Roman")
97 font.setPointSize(18)
98 font.setBold(True)
99 font.setWeight(75)
100 self.pushButtonStart.setFont(font)
101 self.pushButtonStart.setStyleSheet("background-color: rgb(170, 255, 255);")
102 self.pushButtonStart.setObjectName("pushButtonStart")
103 self.pushButtonStop = QtWidgets.QPushButton(parent=self.centralwidget)
104 self.pushButtonStop.setGeometry(QtCore.QRect(18, 178, 151, 51))
105 font = QtGui.QFont()
106 font.setFamily("Times New Roman")
107 font.setPointSize(18)
108 font.setBold(True)
109 font.setWeight(75)
110 self.pushButtonStop.setFont(font)

```

```

111 self.pushButtonStop.setFont(font)
112 self.pushButtonStop.setStyleSheet("background-color: rgb(170, 255, 255);")
113 self.pushButtonStop.setObjectName("pushButtonStop")
114 self.pushButtonAuto = QtWidgets.QPushButton(parent=self.centralwidget)
115 self.pushButtonAuto.setGeometry(QtCore.QRect(18, 238, 151, 51))
116 font = QtGui.QFont()
117 font.setFamily("Times New Roman")
118 font.setPointSize(18)
119 font.setBold(True)
120 font.setWeight(75)
121 self.pushButtonAuto.setFont(font)
122 self.pushButtonAuto.setStyleSheet("background-color: rgb(170, 255, 255);")
123 self.pushButtonAuto.setObjectName("pushButtonAuto")
124 self.pushButtonManual = QtWidgets.QPushButton(parent=self.centralwidget)
125 self.pushButtonManual.setGeometry(QtCore.QRect(18, 298, 151, 51))
126 font = QtGui.QFont()
127 font.setFamily("Times New Roman")
128 font.setPointSize(18)
129 font.setBold(True)
130 font.setWeight(75)
131 self.pushButtonManual.setFont(font)
132 self.pushButtonManual.setStyleSheet("background-color: rgb(170, 255, 255);")
133 self.pushButtonManual.setObjectName("pushButtonManual")
134 self.label = QtWidgets.QLabel(parent=self.centralwidget)
135 self.label.setGeometry(QtCore.QRect(510, -20, 511, 81))
136 font = QtGui.QFont()
137 font.setFamily("Times New Roman")

```

```

127         font.setPointSize(24)
128         font.setBold(True)
129         font.setWeight(75)
130         self.Label.setFont(font)
131         self.Label.setObjectName("Label")
132         self.Label_2 = QtWidgets.QLabel(parent=self.centralwidget)
133         self.Label_2.setGeometry(QtCore.QRect(480, 40, 571, 71))
134         font = QtGui.QFont()
135         font.setFamily("Times New Roman")
136         font.setPointSize(24)
137         font.setBold(True)
138         font.setWeight(75)
139         self.Label_2.setFont(font)
140         self.Label_2.setObjectName("Label_2")
141         self.LabelLogo = QtWidgets.QLabel(parent=self.centralwidget)
142         self.LabelLogo.setGeometry(QtCore.QRect(0, -10, 161, 111))
143         font = QtGui.QFont()
144         font.setFamily("Times New Roman")
145         font.setPointSize(24)
146         font.setBold(True)
147         font.setWeight(75)
148         self.LabelLogo.setFont(font)
149         self.LabelLogo.setText("")
150         self.LabelLogo.setObjectName("LabelLogo")
151         self.Label_syth_2 = QtWidgets.QLabel(parent=self.centralwidget)
152         self.Label_syth_2.setGeometry(QtCore.QRect(740, 680, 501, 111))
153         font = QtGui.QFont()
154         font.setFamily("Times New Roman")

```

```

155         font.setPointSize(14)
156         font.setItalic(True)
157         self.Label_syth_2.setFont(font)
158         self.Label_syth_2.setStyleSheet("background-color: rgb(255, 255, 255);")
159         self.Label_syth_2.setObjectName("Label_syth_2")
160         self.pushButtonBangTai = QtWidgets.QPushButton(parent=self.centralwidget)
161         self.pushButtonBangTai.setGeometry(QtCore.QRect(10, 368, 152, 51))
162         font = QtGui.QFont()
163         font.setFamily("Times New Roman")
164         font.setPointSize(14)
165         font.setBold(True)
166         font.setWeight(75)
167         self.pushButtonBangTai.setFont(font)
168         self.pushButtonBangTai.setStyleSheet("background-color: rgb(170, 255, 255);\\n"
169 "background-color: rgb(170, 255, 127);")
170         self.pushButtonBangTai.setObjectName("pushButtonBangTai")
171         self.pushButtonDangCo = QtWidgets.QPushButton(parent=self.centralwidget)
172         self.pushButtonDangCo.setGeometry(QtCore.QRect(10, 430, 151, 51))
173         font = QtGui.QFont()
174         font.setFamily("Times New Roman")
175         font.setPointSize(14)
176         font.setBold(True)
177         font.setWeight(75)
178         self.pushButtonDangCo.setFont(font)
179         self.pushButtonDangCo.setStyleSheet("background-color: rgb(170, 255, 255);\\n"
180 "background-color: rgb(170, 255, 127);")
181         self.pushButtonDangCo.setObjectName("pushButtonDangCo")
182         self.pushButtonXiep = QtWidgets.QPushButton(parent=self.centralwidget)

```

```

193     self.pushButtonXlep.setGeometry(QRect(10, 470, 151, 51))
194     font = QtGui.QFont()
195     font.setFamily("Times New Roman")
196     font.setPointSize(16)
197     font.setBold(True)
198     font.setWeight(75)
199     self.pushButtonXlep.setFont(font)
200     self.pushButtonXlep.setStyleSheet("background-color: rgb(170, 255, 255);\n"
201 "background-color: rgb(170, 255, 127);")
202     self.pushButtonXlep.setObjectName("pushButtonXlep")
203     self.pushButtonXL1 = QtWidgets.QPushButton(parent=self.centralwidget)
204     self.pushButtonXL1.setGeometry(QRect(10, 530, 151, 51))
205     font = QtGui.QFont()
206     font.setFamily("Times New Roman")
207     font.setPointSize(16)
208     font.setBold(True)
209     font.setWeight(75)
210     self.pushButtonXL1.setFont(font)
211     self.pushButtonXL1.setStyleSheet("background-color: rgb(170, 255, 255);\n"
212 "background-color: rgb(170, 255, 127);")
213     self.pushButtonXL1.setObjectName("pushButtonXL1")
214     self.pushButtonXL2 = QtWidgets.QPushButton(parent=self.centralwidget)
215     self.pushButtonXL2.setGeometry(QRect(10, 590, 151, 51))
216     font = QtGui.QFont()
217     font.setFamily("Times New Roman")
218     font.setPointSize(16)
219     font.setBold(True)
220     font.setWeight(75)

```

```

221     self.pushButtonXL2.setFont(font)
222     self.pushButtonXL2.setStyleSheet("background-color: rgb(170, 255, 255);\n"
223 "background-color: rgb(170, 255, 127);")
224     self.pushButtonXL2.setObjectName("pushButtonXL2")
225     MainWindow.setCentralWidget(self.centralwidget)
226     self.menubar = QtWidgets.QMenuBar(parent=MainWindow)
227     self.menubar.setGeometry(QRect(0, 0, 1261, 21))
228     self.menubar.setObjectName("menubar")
229     MainWindow.setMenuBar(self.menubar)
230     self.statusbar = QtWidgets.QStatusBar(parent=MainWindow)
231     self.statusbar.setObjectName("statusbar")
232     MainWindow.setStatusBar(self.statusbar)
233
234     self.retranslateUi(MainWindow)
235     QtCore.QMetaObject.connectSlotsByName(MainWindow)
236
237     self.pushButtonStart.clicked.connect(self.Start)
238     self.pushButtonStop.clicked.connect(self.Stop)
239     self.pushButtonAuto.clicked.connect(self.Auto)
240     self.pushButtonManual.clicked.connect(self.Manual)
241     self.pushButtonBangTai.clicked.connect(self.BangTai)
242     self.pushButtonDangCo.clicked.connect(self.DangCo)
243     self.pushButtonXL1.clicked.connect(self.XL1)
244     self.pushButtonXL2.clicked.connect(self.XL2)
245     self.pushButtonXlep.clicked.connect(self.Xlep)
246
247     self.Timer_2()
248

```



```

249     def retranslateUi(self, MainWindow): 1usage
250         _translate = QtCore.QCoreApplication.translate
251         MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
252         self.labelLoai1.setText(_translate("MainWindow", "Số lượng hộp dẹt : 0"))
253         self.labelLoai2.setText(_translate("MainWindow", "Số lượng hộp biến dạng : 0"))
254         self.labelLoai3.setText(_translate("MainWindow", "Số lượng hộp sai tem : 0"))
255         self.label_svth.setText(_translate("MainWindow", "GVHD: TS. Trương Thị Bích Thanh\n"
256 "CBHD: TT. Trần Văn Thương"))
257         self.pushButtonStart.setText(_translate("MainWindow", "START"))
258         self.pushButtonStop.setText(_translate("MainWindow", "STOP"))
259         self.pushButtonAuto.setText(_translate("MainWindow", "AUTO"))
260         self.pushButtonManual.setText(_translate("MainWindow", "MANUAL"))
261         self.label.setText(_translate("MainWindow", "MÔ HÌNH DÀN TEM TỰ ĐỘNG"))
262         self.label_2.setText(_translate("MainWindow", "CỦA XƯỞNG SẢN XUẤT DÂY SẠC"))
263         self.label_svth_2.setText(_translate("MainWindow", "Thành viên nhóm: \n"
264 "Lê Đức Long - MSSV: 105200450\n"
265 "Nguyễn Huy Ngân - MSSV: 105200503\n"
266 "Phan Ngọc Nam - MSSV: 105200417"))
267         self.pushButtonBangTai.setText(_translate("MainWindow", "BĂNG TẢI"))
268         self.pushButtonDongCo.setText(_translate("MainWindow", "ĐC CUỘN"))
269         self.pushButtonXlep.setText(_translate("MainWindow", "XI LẠNH ÉP"))
270         self.pushButtonXL1.setText(_translate("MainWindow", "XI LẠNH PL1"))
271         self.pushButtonXL2.setText(_translate("MainWindow", "XI LẠNH PL2"))
272
273         self.labelLogo.setStyleSheet("image: url(logo.jpg);")
274         self.labelCam1.setStyleSheet("image: url(image1.jpg);")
275         self.labelCam2.setStyleSheet("image: url(image2.jpg);")

```

```

277     def Start(self): 1usage
278         reading = plc.db_read(1, 0, 1)
279         snap7.util.set_bool(reading, 0, 0, True)
280         plc.db_write(1, 0, reading)
281
282     def Stop(self): 1usage
283         reading = plc.db_read(1, 0, 1)
284         snap7.util.set_bool(reading, 0, 0, False)
285         plc.db_write(1, 0, reading)
286
287     def Auto(self): 1usage
288         reading = plc.db_read(1, 0, 1)
289         snap7.util.set_bool(reading, 0, 1, False)
290         plc.db_write(1, 0, reading)
291
292     def Manual(self): 1usage
293         reading = plc.db_read(1, 0, 1)
294         snap7.util.set_bool(reading, 0, 1, True)
295         plc.db_write(1, 0, reading)
296
297     def BangTai(self): 1usage
298         reading = plc.db_read(1, 0, 1)
299         temp = snap7.util.get_bool(reading, 0, 2)
300
301         if not temp:
302             snap7.util.set_bool(reading, 0, 2, True)
303         else:
304             snap7.util.set_bool(reading, 0, 2, False)

```

```

306         plc.db_write(1, 0, reading)
307
308     def DongCo(self): 1 usage
309         reading = plc.db_read(1, 0, 1)
310         temp = snap7.util.get_bool(reading, 0, 3)
311
312         if not temp:
313             snap7.util.set_bool(reading, 0, 3, True)
314         else:
315             snap7.util.set_bool(reading, 0, 3, False)
316
317         plc.db_write(1, 0, reading)
318
319     def XL1(self): 1 usage
320         reading = plc.db_read(1, 0, 1)
321         temp = snap7.util.get_bool(reading, 0, 4)
322
323         if not temp:
324             snap7.util.set_bool(reading, 0, 4, True)
325         else:
326             snap7.util.set_bool(reading, 0, 4, False)
327
328         plc.db_write(1, 0, reading)
329
330     def XL2(self): 1 usage
331         reading = plc.db_read(1, 0, 1)
332         temp = snap7.util.get_bool(reading, 0, 5)

```

```

334         if not temp:
335             snap7.util.set_bool(reading, 0, 5, True)
336         else:
337             snap7.util.set_bool(reading, 0, 5, False)
338
339         plc.db_write(1, 0, reading)
340
341     def Xlep(self): 1 usage
342         reading = plc.db_read(1, 0, 1)
343         temp = snap7.util.get_bool(reading, 0, 6)
344
345         if not temp:
346             snap7.util.set_bool(reading, 0, 6, True)
347         else:
348             snap7.util.set_bool(reading, 0, 6, False)
349
350         plc.db_write(1, 0, reading)
351
352
353     def Timer_2(self): 2 usages
354         reading = plc.db_read(1, 10, 6)
355         temp1 = snap7.util.get_int(reading, 0)
356         temp2 = snap7.util.get_int(reading, 2)
357         temp3 = snap7.util.get_int(reading, 4)
358
359         self.labelloa11.setText("Số lượng hộp đặt : " + str(temp3))
360         self.labelloa12.setText("Số lượng hộp biến dạng : " + str(temp1))

```

```

361         self.labelLoai3.setText("Số lượng hộp sai tem : " + str(temp2))
362
363     ret, frame = cap.read()
364
365     if ret:
366         frame = cv2.resize(frame, (480, 320), interpolation=cv2.INTER_LINEAR)
367
368         frame_org = frame.copy()
369         # frame_hop = frame.copy()
370         # frame_tem = frame.copy()
371
372         hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
373         mask_hop = cv2.inRange(hsv, (0, 0, 160), (100, 215, 230))
374         mask_tem = cv2.inRange(hsv, (10, 10, 10), (255, 255, 255))
375
376         # mask = cv2.inRange(hsv, (0, 50, 100), (100, 180, 255))
377
378         result_hop = cv2.bitwise_and(frame, frame, mask=mask_hop)
379         result_tem = cv2.bitwise_and(frame, frame, mask=mask_tem)
380
381         # result = cv2.bitwise_and(frame, frame, mask=mask)
382
383         edged_hop = cv2.Canny(result_hop, 30, 255)
384         edged_tem = cv2.Canny(result_tem, 30, 255)
385
386         # edged = cv2.Canny(result, 20, 255)
387
388         contours_hop, hierarchy_hop = cv2.findContours(edged_hop, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
389         contours_tem, hierarchy_tem = cv2.findContours(edged_tem, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
390
391         # contours, hierarchy = cv2.findContours(edged, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
392         self.x_tem = 0
393         self.y_tem = 0
394         self.x_hop = 0
395         self.y_hop = 0
396         self.sai_nhan = 0
397
398         for cnt in contours_hop:
399             a = cv2.contourArea(cnt)
400
401             if a > 10000:
402                 # approx = cv2.approxPolyDP(cnt, 0.1 * cv2.arcLength(cnt, True), True)
403                 #
404                 # if len(approx) == 4:
405                 M = cv2.moments(cnt)
406                 if M['m00'] != 0:
407                     cpx = int(M['m10'] / M['m00'])
408                     cpy = int(M['m01'] / M['m00'])
409                     if cpx < 250:
410                         cv2.drawContours(frame_org, [cnt], -1, (0, 255, 0), 2)
411                         cv2.circle(frame_org, (cpx, cpy), 7, (0, 0, 255), -1)
412                         self.x_hop = cpx
413                         self.y_hop = cpy
414                     break

```



```

416         for cnt in contours_tem:
417             a = cv2.contourArea(cnt)
418
419             if 2000 < a < 15000:
420                 # approx = cv2.approxPolyDP(cnt, 0.01 * cv2.arcLength(cnt, True), True)
421                 #
422                 # if len(approx) == 4:
423
424                 M = cv2.moments(cnt)
425                 if M['m00'] != 0:
426                     cpx = int(M['m10'] / M['m00'])
427                     cpy = int(M['m01'] / M['m00'])
428
429                     if cpx < 250:
430                         cv2.drawContours(frame_org, [cnt], -1, (255, 255, 0), 2)
431                         cv2.circle(frame_org, (cpx, cpy), 7, (255, 0, 255), -1)
432                         self.x_tem = cpx
433                         self.y_tem = cpy
434                     break
435
436
437         # for cnt in contours:
438         #     a = cv2.contourArea(cnt)
439         #
440         #     if 4000 < a < 20000:
441         #         approx = cv2.approxPolyDP(cnt, 0.01 * cv2.arcLength(cnt, True), True)
442         #

```

```

443         #     if len(approx) == 4:
444         #         M = cv2.moments(cnt)
445         #         if M['m00'] != 0:
446         #             cpx = int(M['m10'] / M['m00'])
447         #             cpy = int(M['m01'] / M['m00'])
448         #             cv2.drawContours(frame, [cnt], -1, (0, 255, 0), 2)
449         #             cv2.circle(frame, (cpx, cpy), 7, (0, 0, 255), -1)
450         #             self.x_tem = cpx
451         #             self.y_tem = cpy
452         #
453         #     if 10000 < a < 30000:
454         #         approx = cv2.approxPolyDP(cnt, 0.01 * cv2.arcLength(cnt, True), True)
455         #
456         #     if len(approx) == 4:
457         #         M = cv2.moments(cnt)
458         #         if M['m00'] != 0:
459         #             cpx = int(M['m10'] / M['m00'])
460         #             cpy = int(M['m01'] / M['m00'])
461         #             cv2.drawContours(frame, [cnt], -1, (0, 255, 255), 2)
462         #             cv2.circle(frame, (cpx, cpy), 7, (255, 0, 0), -1)
463         #             self.x_hop = cpx
464         #             self.y_hop = cpy
465         #
466         # if self.x_hop > 0 and self.x_tem > 0:
467
468         if (abs(self.x_hop - self.x_tem) >= self.lech_x_max or abs(self.y_hop - self.y_tem) >= self.lech_y_max)
469             and self.x_hop > 0 and self.x_tem > 0 and self.y_hop > 0 and self.y_tem > 0:
470                 self.sei_nhan = 1
471

```

```

471         print(self.sai_nhan)
472         print(self.x_hop, self.x_tem, self.y_hop, self.y_tem)
473
474         if self.sai_nhan == 1:
475             reading = plc.db_read(1, 0, 1)
476             snap7.util.set_bool(reading, 0, 5, True)
477             plc.db_write(1, 0, reading)
478
479             height, width, channel = frame.shape
480             bytesPerLine = 3 * width
481
482             qImg_hop = QImage(frame_org.data, width, height, bytesPerLine, QImage.Format.Format_BGR888)
483             qImg_tem = QImage(frame.data, width, height, bytesPerLine, QImage.Format.Format_BGR888)
484
485             pm_hop = QtGui.QPixmap.fromImage(qImg_hop)
486             pm_tem = QtGui.QPixmap.fromImage(qImg_tem)
487
488             self.labelCam1.setPixmap(pm_hop)
489             self.labelCam2.setPixmap(pm_tem)
490
491             Timer(0.1, self.Timer_2).start()
492
493
494 > if __name__ == "__main__":
495     import sys
496     app = QtWidgets.QApplication(sys.argv)
497     MainWindow = QtWidgets.QMainWindow()
498
499     ui = Ui_MainWindow()
500     ui.setupUI(MainWindow)
501     MainWindow.show()
502     sys.exit(app.exec())

```