

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



**ĐỒ ÁN TỐT NGHIỆP  
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG**

**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG  
ĐẾM SỐ LƯỢNG TRONG KHÂU  
ĐÓNG GÓI BÁNH KẸO**

**GVHD: NGUYỄN DUY THẢO  
SVTH: NGUYỄN PHI HÙNG  
MSSV: 16141171  
SVTH: PHẠM TRỌNG HIỀN  
MSSV: 16141149**



**Tp. Hồ Chí Minh, tháng 08/2020**

**BỘ GIÁO DỤC & ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH**

---



# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG**

**ĐỀ TÀI:**

**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐẾM SỐ  
LƯỢNG TRONG KHÂU ĐÓNG GÓI BÁNH KẸO**

**GVHD: Th.S Nguyễn Duy Thảo**

**SVTH: 1. Nguyễn Phi Hùng 16141171**

**2. Phạm Trọng Hiền 16141149**

**Tp. Hồ Chí Minh - 8/2020**

Tp. HCM, ngày tháng 8 năm 2020

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên:	Nguyễn Phi Hùng	MSSV: 16141171
	Phạm Trọng Hiền	MSSV: 16141149
Chuyên ngành:	Điện tử công nghiệp	Mã ngành: 41
Hệ đào tạo:	Đại học chính quy	Mã hệ: 1
Khóa:	2016	Lớp: 16141DT1A

### I. TÊN ĐỀ TÀI: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐẾM SỐ LƯỢNG TRONG KHÂU ĐÓNG GÓI BÁNH KẸO

#### II. NHIỆM VỤ

##### 1. Các số liệu ban đầu:

(ghi những thông số, tập tài liệu tin hiệu, hình ảnh, ...).....  
.....  
.....  
.....  
.....

##### 2. Nội dung thực hiện:

(ghi những nội dung chính cần thực hiện như trong phần tổng quan).....  
.....  
.....  
.....  
.....

#### III. NGÀY GIAO NHIỆM VỤ:

#### IV. NGÀY HOÀN THÀNH NHIỆM VỤ:

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: ThS. Nguyễn Duy Thảo

CÁN BỘ HƯỚNG DẪN

BM. ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

## LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên 1: Nguyễn Phi Hùng

Lớp:16141DT1B .....MSSV:16141171

Họ tên sinh viên 2: Phạm Trọng Hiền

Lớp:16141DT1A.....MSSV:16141149

Tên đề tài:

THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐẾM SỐ LƯỢNG TRONG KHẤU  
ĐÓNG GÓI BÁNH KẸO

<i>Tuần/ngày</i>	<i>Nội dung</i>	<i>Xác nhận GVHD</i>
Tuần 1 (9-15/03/2020)	Nhận đồ, tìm hiểu đề tài	
Tuần 2 (16-22/03/2020)	Tìm hiểu hướng làm đề tài, chọn vi xử lý điều khiển, ngôn ngữ lập trình	
Tuần 3 (23-29/03/2020)	Tìm hiểu Raspberry Pi 3 mode B+ và ngôn ngữ lập trình Matlab	
Tuần 4 (30-5/04/2020)	Tiến hành lập trình đếm số lượng giữa máy tính và Raspberry	
Tuần 5 (6-12/04/2020)	Lập trình chạy độc lập chương trình đếm số lượng trên Raspberry	
Tuần 6, 7, 8 (13-3/05/2020)	Tiến hành thiết kế phần cứng cho toàn bộ hệ thống.	
Tuần 9, 10, 11, 12 (4-31/05/2020)	Chạy thử nghiệm hệ thống và hiệu chỉnh sao cho hệ thống hoạt động ổn định	
Tuần 13, 14, 15 (1-21/06/2020)	Hoàn chỉnh báo cáo	
	Hoàn thành nhiệm vụ đồ án	

GV HƯỚNG DẪN  
(Ký và ghi rõ họ và tên)

# LỜI CAM ĐOAN

Đề tài này là do tôi tự thực hiện dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Người thực hiện đề tài

Nguyễn Phi Hùng

Phạm Trọng Hiền

# LỜI CẢM ƠN

Nhóm thực hiện đồ án xin gửi lời cảm ơn đến thầy GVHD Th.S Nguyễn Duy Thảo đã trực tiếp hướng dẫn, tham gia đóng góp, gợi ý các ý kiến, chia sẻ nhiều những kinh nghiệm và tận tình giúp đỡ cũng như tạo điều kiện tốt và thoải mái nhất để chúng em có thể hoàn thành tốt đề tài này.

Nhóm xin gửi lời cảm ơn đến các quý thầy cô khoa Điện – Điện tử đã giúp đỡ và tạo điều kiện tốt cho chúng em thực hiện đề tài. Ngoài ra, chúng em cũng cảm ơn các bạn học ở lớp 16141DT1 cũng như đã chia sẻ và giúp đỡ chúng em rất nhiều trong đề tài này.

Xin chân thành cảm ơn tất cả!

Nhóm thực hiện đề tài

Nguyễn Phi Hùng

Phạm Trọng Hiền

# MỤC LỤC

Trang bìa .....	i
Nhiệm vụ đồ án .....	ii
Lịch trình .....	iii
Cam đoan .....	iv
Lời cảm ơn .....	v
Mục lục.....	vi
Liệt kê hình vẽ.....	ix
Liệt kê bảng vẽ .....	xi
Tóm tắt .....	xii
Chương 1. TỔNG QUAN .....	1
1.1 ĐẶT VẤN ĐỀ.....	1
1.2 MỤC TIÊU.....	2
1.3 NỘI DUNG NGHIÊN CỨU .....	2
1.4 GIỚI HẠN.....	2
1.5 BỐ CỤC .....	3
Chương 2. CƠ SỞ LÝ THUYẾT.....	4
2.1 TỔNG QUAN VỀ XỬ LÝ ẢNH.....	4
2.1.1 Thu nhận ảnh (Image Acquisition) .....	4
2.1.2 Tiền xử lý (Image processing) .....	5
2.1.3 Phân đoạn (Segmentation) hay phân vùng ảnh.....	5
2.1.4 Biểu diễn ảnh (Image Representation).....	5
2.1.5 Nhận dạng và nội suy ảnh (Image Recognition and Interpretation) .....	5
2.1.6 Cơ sở tri thức (Knowledge Base).....	6
2.1.7 Mô tả .....	6
2.1.8 Nhi phân hóa ngưỡng động.....	6
2.1.9 Phân vùng ảnh .....	7
2.1.10 Thuật toán đánh nhãn.....	8
2.1.11 Phương pháp phát hiện biên.....	8
2.2 TỔNG QUAN VỀ MATLAB.....	9

2.2.1 Tổng quan.....	9
2.2.2 Các hàm xử lý trong Matlab.....	10
2.2.3 Hàm xử lý phát hiện hình tròn .....	11
2.3 GIỚI THIỆU RASPBERRY PI 3 .....	13
2.3.1 Giới thiệu.....	13
2.3.2 Thông tin cấu hình Raspberry Pi 3 Module B+ .....	14
2.3.3 Ứng dụng.....	15
2.4 GIỚI THIỆU VỀ CAMERA PI .....	15
2.4.1 Giới thiệu.....	15
2.4.2 Thông tin cấu hình Camera Pi v2.1.....	16
2.4.3 Ứng dụng.....	17
2.5 GIỚI THIỆU VỀ CẢM BIẾN HỒNG NGOẠI E18-D80NK .....	17
2.5.1 Giới thiệu.....	17
2.5.2 Thông số kỹ thuật.....	18
2.5.3 Ứng dụng.....	18
2.6 GIỚI THIỆU VỀ ĐỘNG CƠ SERVO MG996R.....	19
2.6.1 Giới thiệu động cơ servo MG996R.....	19
2.6.2 Thông số kỹ thuật.....	19
2.6.3 Ứng dụng.....	19
2.7 GIỚI THIỆU MODULE RELAY 5V .....	20
2.7.1 Tổng quan về module relay 5V .....	20
2.8 GIỚI THIỆU HỆ THỐNG BĂNG TẢI.....	21
2.8.1 Giới thiệu.....	21
2.8.2 Cấu tạo .....	21
2.8.3 Ứng dụng.....	22
2.9. CÀI ĐẶT HỖ TRỢ PHẦN CỨNG MATLAB CHO RPI.....	22
Chương 3: TÍNH TOÁN VÀ THIẾT KẾ .....	35
3.1 TỔ CHỨC CÁC KHỐI TRONG HỆ THỐNG.....	35
3.2 THIẾT KẾ CÁC KHỐI TRONG HỆ THỐNG .....	36
3.2.1. Khối thu tín hiệu hình ảnh (camera) .....	36
3.2.2 Khối xử lý trung tâm (Raspberry Pi 3B+).....	39
3.2.3 Khối cảm biến .....	40
3.2.4 Khối hiển thị.....	41
3.2.5 Khối nguồn.....	41
3.2.6 Sơ đồ nguyên lý toàn mạch.....	43
Chương 4: THI CÔNG HỆ THỐNG .....	44



4.1 GIỚI THIỆU .....	44
4.2 THI CÔNG HỆ THỐNG .....	44
4.2.1 Chuẩn bị phần cứng .....	44
4.2.2 Lắp ráp và kiểm tra .....	45
4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH.....	48
4.3.1 Thi công mô hình .....	49
4.4 LẬP TRÌNH HỆ THỐNG.....	49
4.4.1 Nhúng chương trình xuống Raspberry Pi .....	49
4.4.2 Lưu đồ giải thuật .....	55
4.5 TÀI LIỆU HƯỚNG DẪN .....	61
Chương 5. KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ .....	70
5.1 KẾT QUẢ .....	70
5.1.1 Tổng quan kết quả đạt được .....	70
5.2 KẾT QUẢ THỰC NGHIỆM .....	70
5.2.1 Mô hình sản phẩm thực tế.....	70
5.2.2 Kết quả thực nghiệm từ việc đếm số lượng .....	72
5.3 NHẬN XÉT – ĐÁNH GIÁ.....	77
5.3.1 Nhận xét .....	77
5.3.2. Đánh giá .....	78
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	80
6.1 KẾT LUẬN.....	80
6.1.1 Kết quả thu được .....	80
6.1.2 Những mặt hạn chế .....	80
6.2 HƯỚNG PHÁT TRIỂN .....	80
TÀI LIỆU THAM KHẢO.....	82
PHỤ LỤC.....	82

## LIỆT KÊ HÌNH VẼ

<b>Hình</b>	<b>Trang</b>
Hình 2. 1: Phát hiện tâm đường tròn.....	12
Hình 2. 2: Phát hiện bán kính đường tròn.....	12
Hình 2. 3: Raspberry Pi 3B+.....	13
Hình 2. 4: Sơ đồ Raspberry Pi 3 Module B+.....	14
Hình 2. 5: Camera Pi v2.1.....	16
Hình 2. 6: Sơ đồ khối Camera Raspberry Pi.....	17
Hình 2. 7: Cảm biến hồng ngoại E18-D80NK.....	18
Hình 2. 8: Động cơ servo MG996R.....	19
Hình 2. 9: Module Relay 1 kênh.....	20
Hình 2. 10: Mô hình băng tải.....	21
Hình 2. 11: Get Hardware Support Package.....	22
Hình 2. 12: Cửa sổ “Support Package Installer”.....	23
Hình 2. 13: Giao diện cài Package cho Raspberry.....	23
Hình 2. 14: Đăng nhập cho cửa sổ “MathWorks Account Log In”.....	24
Hình 2. 15: Matlab Support Package for Raspberry Pi Hardware.....	24
Hình 2. 16: Cửa sổ Add -On.....	25
Hình 2. 17: Lựa chọn phiên bản Raspberry Pi.....	25
Hình 2. 18: Lựa chọn cách cài đặt.....	26
Hình 2. 19: Kiểm tra kết nối Raspberry Pi.....	27
Hình 2. 20: Kiểm tra kết nối Raspberry thành công.....	28
Hình 2. 21: Cài đặt gói hỗ trợ Raspberry Pi.....	28
Hình 2. 22: Lựa chọn cài đặt.....	29
Hình 2. 23: Download hệ điều hành Mathworks Raspbian.....	30
Hình 2. 24: Kiểm tra hệ điều hành Mathworks Raspbian.....	31
Hình 2. 25: Kết nối thông qua mạng LAN.....	31
Hình 2. 26: Kết nối thông qua WIFI.....	32
Hình 2. 27: Kết nối thông qua cáp ethernet.....	32
Hình 2. 28: Cấu hình WIFI.....	33
Hình 2. 29: Chọn ổ đĩa để lưu trữ.....	34
Hình 2. 30: Cài đặt gói hỗ trợ Raspberry.....	34
Hình 2. 31: Kết nối thông qua cáp ethernet.....	35
Hình 2. 32: Chọn ổ đĩa để lưu trữ.....	36
Hình 2. 33: Cài đặt gói hỗ trợ Raspberry.....	37
Hình 2. 34: Kết nối Raspberry và Matlab thành công.....	38
Hình 3. 1: Sơ đồ khối hệ thống.....	35
Hình 3. 2: Sơ đồ kết nối khối Camera.....	36

Hình 3. 3: Kết nối Camera thực tế .....	37
Hình 3. 4: Led 3 thanh .....	37
Hình 3. 5: Sơ đồ bố trí buồng chụp ảnh (nhìn từ trên xuống).....	38
Hình 3. 6: Sơ đồ bố trí buồng chụp ảnh (nhìn từ mặt bên) .....	38
Hình 3. 7: Sơ đồ các cổng ngoại vi sử dụng .....	39
Hình 3. 8: Sơ đồ chân Raspberry Pi 3+.....	40
Hình 3. 9: Nguồn 5V 2A cung cấp cho Raspberry Pi.....	42
Hình 3. 10: Nguồn 12V 2A cung cấp cho băng tải .....	42
Hình 3. 12: Sơ đồ nguyên lý toàn mạch.....	43
Hình 4. 1: Băng tải và động cơ DC .....	46
Hình 4. 2: Máng đưa sản phẩm sau khi phân loại.....	46
Hình 4. 3: Buồng chụp ảnh nhìn từ phía dưới lên trên.....	47
Hình 4. 4: Khuôn kẹo trên solid work.....	47
Hình 4. 5: Khuôn kẹo thực tế .....	48
Hình 4. 6: Mô hình thực tế nhìn từ phía trước .....	49
Hình 4. 7: Mô hình thực tế nhìn từ trên xuống .....	49
Hình 4. 8: Kết nối Raspberry và Matlab thành công. ....	50
Hình 4. 9: Kết nối tới board Raspberry Pi .....	50
Hình 4. 10: Biên dịch chương trình có tên counter_7 lên Raspberry Pi .....	50
Hình 4. 11: Biên dịch thành công.....	51
Hình 4. 12: Đường dẫn kết quả .....	51
Hình 4. 13: Tập được tạo với tên counter_7.elf trên đường dẫn thu được .....	52
Hình 4. 14: Chạy chương trình vừa được tạo.....	53
Hình 4. 15: Kết quả thu được .....	53
Hình 4. 16: Kết quả thu được .....	54
Hình 4. 17: Thông báo biên dịch không thành công.....	54
Hình 4. 18: Hàm được hỗ trợ cho việc biên dịch .....	55
Hình 4. 19: Lưu đồ hoạt động của hệ thống.....	56
Hình 4. 20: Lưu đồ chụp và lưu ảnh .....	58
Hình 4. 21: Lưu đồ chương trình xử lý ảnh .....	59
Hình 4. 22: Lưu đồ chương trình đếm số lượng .....	60
Hình 4. 23: Công tắc ở vị trí ON.....	61
Hình 4. 24: Chạy chương trình đếm 7 viên/hộp.....	62
Hình 4. 25: Khuôn kẹo trước khi đưa vào buồng chụp.....	63
Hình 4. 26: Khuôn kẹo được đưa vào buồng chụp .....	63
Hình 4. 27: Kết quả hiện thị trên màn hình.....	64
Hình 4. 28: Số lượng kẹo đúng thì servo không tác động.....	65
Hình 4. 29: Khuôn kẹo tiếp theo trước khi đưa vào buồng chụp.....	65
Hình 4. 30: Khuôn kẹo tiếp theo được đưa vào buồng chụp .....	66
Hình 4. 31: Kết quả hiện thị trên màn hình.....	66
Hình 4. 32: Số lượng kẹo sai thì servo tác động kẹo xuống máng .....	67
Hình 4. 33: Công tắc ở vị trí OFF .....	67
Hình 4. 34: Kết quả của quá trình chạy hệ thống được lưu lại trong file text. ....	68

Hình 5. 1: Mô hình thực tế nhìn từ phía trước .....	71
Hình 5. 2: Mô hình thực tế nhìn từ trên xuống .....	72
Hình 5. 3: Kết quả chụp từ camera (3 viên).....	72
Hình 5. 4: Kết quả sau khi xử lý (3 viên).....	72
Hình 5. 5: Kết quả chụp từ camera (4 viên).....	73
Hình 5. 6: Kết quả sau khi xử lý (4 viên).....	73
Hình 5. 7: Kết quả chụp từ camera (4 viên).....	73
Hình 5. 8: Kết quả sau khi xử lý (4 viên).....	73
Hình 5. 9: Kết quả chụp từ camera (7 viên).....	74
Hình 5. 10: Kết quả sau khi xử lý (7 viên).....	74
Hình 5. 11: Kết quả của quá trình chạy lưu trong file text. ....	74
Hình 5. 12: Kết quả chụp từ camera (4 viên).....	74
Hình 5. 13: Kết quả sau khi xử lý (4 viên).....	74
Hình 5. 14: Kết quả chụp từ camera (7 viên).....	75
Hình 5. 15: Kết quả sau khi xử lý (7 viên).....	75
Hình 5. 16: Kết quả chụp từ camera (7 viên).....	75
Hình 5. 17: Kết quả sau khi xử lý (7 viên).....	75
Hình 5. 18: Kết quả chụp từ camera (8 viên).....	75
Hình 5. 19: Kết quả sau khi xử lý (8 viên).....	75
Hình 5. 20: Kết quả của quá trình chạy lưu trong file text .....	76
Hình 5. 21: Kết quả chụp từ camera .....	76
Hình 5. 22: Kết quả sau khi xử lý .....	76
Hình 5. 23: Kết quả chụp từ camera .....	77
Hình 5. 24: Kết quả sau khi xử lý .....	77

# LIỆT KÊ BẢNG

## **Bảng**

	<b>Trang</b>
Bảng 2. 1: Các hàm xử lý xác định bán kính trong Matlab .....	13
Bảng 4. 1: Danh sách các linh kiện.....	44
Bảng 5. 1: Bảng đánh giá .....	78

# TÓM TẮT

Đề tài “Thiết kế và thi công hệ thống đếm số lượng trong khâu đóng gói bánh kẹo” là mô hình đếm số lượng sản phẩm. Được viết bằng ngôn ngữ Matlab, chương trình sau khi hoàn thiện được nhúng xuống kit Raspberry để chạy như một ứng dụng chạy độc lập. Kết quả thực hiện của đề tài đã nhận dạng được những sản phẩm có hình dạng (hình tròn) cùng với việc đếm được sản phẩm theo đúng yêu cầu số lượng đã đặt trước.

# Chương 1. TỔNG QUAN

## 1.1 ĐẶT VẤN ĐỀ

Những năm qua, ngành bánh kẹo Việt Nam có tốc độ tăng trưởng cao và ổn định, với sản lượng hàng năm trên 150 ngàn tấn, doanh thu năm 2014 đạt 27 ngàn tỉ đồng. Như vậy, ta thấy Việt Nam là thị trường tiềm năng của ngành công nghiệp sản xuất bánh kẹo. Sự khẳng định của các thương hiệu Việt không chỉ ở số lượng, chất lượng và sự đa dạng hóa sản phẩm.[5]

Trong quy trình sản xuất bánh kẹo, sau khi tạo hình kẹo theo hình dáng có sẵn trên khuôn, kẹo sẽ được tách ra khuôn đi qua hệ thống băng chuyền hẹp được làm nguội nhờ quạt gió thổi khí nén vào. Tiếp theo đến công đoạn chọn kẹo nhờ công nhân đếm số lượng và kiểm tra ngoại quang bằng mắt trước trên băng tải đang chạy, trước khi đưa đến công đoạn đóng gói thành phẩm tiếp theo.

Giai đoạn chọn kẹo sẽ mất rất nhiều thời gian, và tốn nhiều chi phí nhân công nhằm tránh việc máy gói kẹo bị tắc do có quá nhiều kẹo trong bao bì và do kích thước kẹo không đồng nhất nên nhóm em tiến hành dùng xử lý ảnh để kiểm tra đếm đúng số lượng kẹo và phù hợp với kích thước của bao bì.[2]

Hiện nay, đã có rất nhiều nghiên cứu, cụ thể trong đó là những Đồ Án Tốt Nghiệp (ĐATN) được thực hiện tạo ra những mô hình thực tế như năm 2019, Phạm Thị Thanh Thảo – Phan Trần Hoài Vũ với đề tài “*Nghiên cứu, thiết kế và thi công mô hình đếm và phân loại sản phẩm theo cân nặng, màu sắc*”, đề tài này đã sử dụng xử lý ảnh thực hiện các tác vụ trên Kit Raspberry Pi và hệ điều hành Raspbian để nhận biết và phân loại cà chua. [3]. Rõ ràng, mô hình trên cho thấy rất thiết thực và mang lại nhiều giá trị khi áp dụng vào thực tế. Đề tài ĐATN tiếp theo là của bạn Võ Sĩ Nguyên với đề tài “*Thiết kế và thi công hệ thống đếm số lượng cá giống*” cũng sử dụng kit Raspberry Pi 3+ và camera Pi Rev 1.3 đếm kích thước cá 1 đến 2cm và số lượng dưới 50 con.[4]

Sau khi xem qua các đề tài tham khảo, với mục đích áp dụng xử lý ảnh trong việc đếm số lượng nhằm giảm thiểu thời gian khi đếm thủ công và ít tốn chi phí nhân công, nhóm chọn đề tài: “*Thiết kế và thi công hệ thống đếm số lượng trong khâu đóng gói bánh kẹo*”.

### 1.2 MỤC TIÊU

Mục tiêu nghiên cứu của đề tài: “*Thiết kế và thi công hệ thống đếm số lượng trong khâu đóng gói bánh kẹo*” là hiểu rõ về ứng dụng xử lý ảnh trong môi trường công nghiệp. Ngoài ra, nhóm đề tài mong muốn thi công được một hệ thống có khả năng phân loại số lượng sản phẩm, được viết bằng ngôn ngữ Matlab, chương trình sau khi hoàn thiện được nhúng xuống kit Raspberry để chạy như một ứng dụng chạy độc lập.

### 1.3 NỘI DUNG NGHIÊN CỨU

Trong quá trình thực hiện Đồ án tốt nghiệp với đề tài: “*Thiết kế và thi công hệ thống đếm số lượng trong khâu đóng gói bánh kẹo*”, nhóm sẽ tập trung giải quyết và hoàn thành được những nội dung sau:

- **NỘI DUNG 1:** Tìm hiểu phương pháp phân loại theo số lượng.
- **NỘI DUNG 2:** Thiết kế thi công mô hình hệ thống.
- **NỘI DUNG 3:** Viết chương trình xử lý ảnh bằng ngôn ngữ Matlab và nhúng xuống kit Raspberry Pi.
- **NỘI DUNG 4:** Hoàn thiện hệ thống điều khiển, mô hình, tiến hành chạy mẫu.
- **NỘI DUNG 5:** Viết báo cáo thực hiện.
- **NỘI DUNG 6:** Bảo vệ đồ án.

### 1.4 GIỚI HẠN

Hệ thống nhận dạng ảnh chụp được từ camera với kích thước 640x480 với các phương pháp tiền xử lý ảnh đầu vào và thuật toán nhận dạng, đếm đối tượng ảnh và đánh nhãn đối tượng.

Các thông số giới hạn của đề tài bao gồm:

- Thiết kế mô hình băng tải vận chuyển bánh kẹo.
- Thiết kế cơ cấu gạt sản phẩm khi không đúng số lượng yêu cầu mỗi túi.
- Thiết kế hệ thống buồng chụp ảnh có đủ độ sáng.
- Dùng Raspberry Pi để xử lý ảnh phân loại số lượng gửi thông số lên màn hình máy tính.



### 1.5 BỐ CỤC

- **Chương 1:** Tổng Quan

Chương này trình bày đặt vấn đề dẫn nhập lí do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bố cục đề án.

- **Chương 2:** Cơ Sở Lý Thuyết

Nội dung chương bao gồm lí thuyết tổng quan về xử lý ảnh.

Giới thiệu về phân cứng của mô hình.

Giới thiệu về các phương pháp xử lý ảnh đầu vào và thuật toán xác định tâm, đếm số lượng và đánh nhãn đối tượng.

Cài đặt gói hỗ trợ phân cứng Matlab cho Raspberry Pi.

- **Chương 3:** Thiết Kế và Tính Toán

Từ những cơ sở lí thuyết có được, nhóm đề tài sẽ tiến hành trình bày sơ đồ khối, thiết kế các khối có trong hệ thống, thiết kế chương trình cho hệ thống đếm đối tượng sau khi được chụp viết trên phần mềm Matlab, sau đó nhúng trên Raspberry Pi để chạy độc lập.

- **Chương 4:** Thi công hệ thống

Sau khi thực hiện tính toán và thiết kế, nhóm đề tài sẽ tiến hành thi công hệ thống và trình bày lại quá trình thi công trong chương này.

Tiến hành lập trình cho hệ thống.

- **Chương 5:** Kết Quả, Nhận Xét và Đánh Giá

Nêu các kết quả đạt được khi thực hiện chương trình, phân tích, nhận xét, đánh giá kết quả đạt được.

- **Chương 6:** Kết Luận và Hướng Phát Triển

Tóm tắt kết quả đạt được, những hạn chế và nêu lên hướng phát triển trong tương lai.

# Chương 2. CƠ SỞ LÝ THUYẾT

## 2.1 TỔNG QUAN VỀ XỬ LÝ ẢNH

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh đặc biệt là máy tính chuyên dụng riêng cho nó.

Xử lý ảnh là kỹ thuật áp dụng trong việc tăng cường và xử lý các ảnh thu nhận từ các thiết bị như camera, webcam... Do đó, xử lý ảnh đã được ứng dụng và phát triển trong rất nhiều lĩnh vực quan trọng như:

- Trong lĩnh vực quân sự: xử lý và nhận dạng ảnh quân sự.
- Trong lĩnh vực giao tiếp: nhận dạng ảnh, xử lý âm thanh, đồ họa.
- Trong lĩnh vực an ninh, bảo mật: nhận diện khuôn mặt người, nhận diện vân tay, mẫu mắt, ...
- Trong lĩnh vực giải trí: trò chơi điện tử.
- Trong lĩnh vực y tế: xử lý ảnh y sinh, chụp X quang, MRI,...

Sau đây, ta sẽ xét các bước cần thiết trong quá trình xử lý ảnh. Đầu tiên, ảnh tự nhiên từ thế giới bên ngoài được thu nhận qua các thiết bị thu (như camera, máy chụp ảnh). Trước đây, ảnh thu qua camera là các ảnh tương tự (loại camera ống kính CCIR). Gần đây với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ camera, sau đó nó được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo. Mặt khác ảnh có thể được quét từ vệ tinh chụp trực tiếp bằng máy quét ảnh.

### 2.1.1 Thu nhận ảnh (Image Acquisition)

Ảnh có thể nhận qua camera màu hoặc trắng đen. Thường ảnh nhận qua camera là ảnh tương tự (loại camera ống kính chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã số hóa (như loại CCD – Charge Coupled Device) là loại photodiode tạo cường độ sáng tại mỗi điểm ảnh.

Camera thường dùng là loại quét dòng, ảnh tạo ra có dạng hai chiều. Chất lượng ảnh thu nhận được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh).

### **2.1.2 Tiền xử lý (Image processing)**

Sau bộ thu nhận, ảnh có thể nhiễu độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để làm ảnh rõ hơn, nét hơn.

### **2.1.3 Phân đoạn (Segmentation) hay phân vùng ảnh**

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu chữ về địa chỉ hoặc tên người thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

### **2.1.4 Biểu diễn ảnh (Image Representation)**

Đây là phần sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết ở các vùng lân cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (Feature Extration) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp phân biệt ký tự này với ký tự khác.

### **2.1.5 Nhận dạng và nội suy ảnh (Image Recognition and Interpretation)**

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt chữ số và nét gạch ngang trên phong bì thư có thể nội suy thành mã điện thoại. Có nhiều cách phân loại ảnh khác nhau về ảnh. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản:

- Nhận dạng theo tham số.
- Nhận dạng theo cấu trúc.

## **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

---

Một số đối tượng nhận dạng khá phổ biến hiện nay được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản, nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người,...

### **2.1.6 Cơ sở tri thức (Knowledge Base)**

Như đã nói ở trên, ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiều. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý ảnh theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây các cơ sở tri thức được phát huy.

### **2.1.7 Mô tả**

Ảnh sau khi số hóa sẽ lưu vào bộ nhớ, hoặc truyền sang các khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ các ảnh thô, đòi hỏi dung lượng bộ nhớ cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Thông thường, các ảnh được gọi là các đặc trưng ảnh như: biên ảnh, vùng ảnh.

### **2.1.8 Nhị phân hóa ngưỡng động**

Nhị phân hóa theo ngưỡng tự động với ý tưởng sau:

- Chia tấm ảnh thành nhiều khu vực, cửa sổ khác nhau
- Dùng một thuật toán để tìm một giá trị T phù hợp với từng khu vực, cửa sổ (Region).
- Áp dụng phương pháp nhị phân hóa cho từng khu vực, cửa sổ (Region) với T phù hợp.

Điều quan trọng trong kỹ thuật này là phải tìm một giá trị T phù hợp với từng khu vực, cửa sổ (Region) hoặc cả tấm ảnh. Có rất nhiều phương pháp để tìm T, ở nội dung tiếp theo sẽ giới thiệu thuật toán giúp tìm kiếm giá trị T này.

- Thuật toán Otsu

Bước 1: Xác định T1. Giá trị cho T1 ban đầu nên chọn là  $(0+255) / 2 = 128$ .

Bước 2: Phân loại thành 2 nhóm điểm ảnh.

## **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

---

- Loại 1 (Type1): chứa tất cả các điểm ảnh có giá trị cường độ sáng  $\leq T$ .
- Loại 2 (Type2): chứa tất cả các điểm ảnh có giá trị cường độ sáng  $> T$ .

Bước3: Tính giá trị cường độ sáng trung bình (iAverage) cho Type1 (iAverage1) và Type2 (iAverage2).

Bước 4: Tính giá trị T2 theo công thức  $(iAverage1 + iAverage2) / 2$ .

Bước 5: So sánh T1 và T2.

- Nếu giá trị chênh lệch của T1 và T2  $\leq \Delta$  (một giá trị cho trước) thì T2 chính là T cần tìm.
- Nếu giá trị chênh lệch của T1 và T2  $> \Delta$  thì quay lại Bước 1.

### **2.1.9 Phân vùng ảnh**

Là bước xác định vật thể trong ảnh, bằng cách trích lọc ra vùng có cùng tính chất dựa vào sự đồng nhất về mức xám giữa những pixel. Sau khi phân vùng ảnh, ta sẽ có được vật thể xác định trên ảnh tách biệt với các chi tiết thừa trong hình. Hạn chế là ta chỉ nhận được các vật tách biệt nhau còn với các ảnh nhiều chi tiết, kết quả phân vùng sẽ không chính xác.

Đối với cấu trúc ảnh nhị phân, các pixel chỉ mang giá trị 0 hoặc 1 nên việc phân vùng ảnh rất quan trọng nhằm loại bỏ nhiễu, phân tách các vật thể gần nhau hay định rõ một vật thể với các đường liên kết nhỏ. Phương pháp khả thi nhất cho việc phân vùng ảnh nhị phân là làm mảnh và làm đầy:

- Làm mảnh: loại bỏ hoàn toàn các nhiễu trong vật thể hay trên biên, tách biệt các vật với nhau.
- Làm đầy: đưa các vùng ảnh khôi phục như ban đầu.

Tùy theo mục đích và yêu cầu hệ thống, ta có thể sử dụng một vài bước trên để lấy thông tin cần thiết cho việc xử lý ảnh. Đó có thể đơn giản là nhận biết sự khác nhau giữa 2 hình nhằm xác định chuyển động hoặc phức tạp cần đến độ chính xác cao như nhận biết dấu vân tay đối tượng định trước ...

### 2.1.10 Thuật toán đánh nhãn

Khi nhận data hình, ta nhận từng pixel một theo chiều từ trái sang phải, từ trên xuống dưới. Việc xử lý có thể được thực hiện giữa các lines hoặc các frames hình. Do hạn chế về tài nguyên của các board xử lý hình ảnh, mà cụ thể là Ram, ta không thể lưu cả 1 frames ảnh để dành đến cuối khung hình mới xử lý, vậy ta chọn cách thực thi thuật toán giữa các lines.

Cấu trúc của đối tượng trong hình có thể đơn giản như hình vuông, chữ nhật ... nhưng cũng có thể rất phức tạp như hình sao, hình lục giác... độ lớn cũng khác nhau, trải dài trên nhiều lines và ở mọi vị trí trên hình ảnh. Ta chỉ có thể nhận và xử lý lines từ trên xuống dưới, các lines đã qua xử lý không thể giữ lại nên nếu gặp một phần của đối tượng nằm ở lines bất kì bên dưới, không thể dùng so sánh để biết nó có thuộc đối tượng nào ở trên hay không.

Cách giải quyết vấn đề này là đánh nhãn đối tượng. Với mỗi pixel nhận được, ta đánh số cho pixel đó, các pixel ở cạnh nhau mà cùng thuộc vật thể( có giá trị "1" khi nhận từ bước tiền xử lý) sẽ được gán cùng một nhãn( nhãn là một giá trị nào đó theo thứ tự số). Vậy cần lưu lại ít nhất 1 line ở trên để so sánh, điều này có thể thực hiện được bởi 1 line gồm 1024 Pixels, không quá nhiều. Do hạn chế của cách hoạt động trên phần cứng, ta chỉ có thể so sánh mỗi pixels ở trên và phía trước nó chứ không thể so sánh toàn bộ các pixels xung quanh.

### 2.1.11 Phương pháp phát hiện biên

#### ❖ Phương pháp phát hiện biên trực tiếp

Phương pháp phát hiện biên này nhằm làm nổi biên dựa vào sự biến thiên về giá trị độ sáng của điểm ảnh. Kỹ thuật chủ yếu dùng phát hiện biên ở đây là kỹ thuật đạo hàm.

- Nếu lấy đạo hàm bậc nhất của ảnh ta có phương pháp Gradient. Phương pháp này bao gồm kỹ thuật Gradient và kỹ thuật la bàn.
- Nếu lấy đạo hàm bậc hai của ảnh ta có phương pháp Laplace.

### ❖ Phương pháp phát hiện biên gián tiếp

Là quá trình phân vùng dựa vào phép xử lý kết cấu đối tượng, cụ thể là dựa vào sự biến thiên nhỏ và đồng đều độ sáng của các điểm ảnh thuộc một đối tượng. Nếu các vùng của ảnh được xác định thì đường phân ranh giữa các vùng đó chính là biên ảnh cần tìm. Việc phát hiện biên và phân vùng đối tượng là hai bài toán đối ngẫu. Từ phát hiện biên ta có thể tiến hành phân lớp đối tượng, như vậy là đã phân vùng được ảnh. Và ngược lại, khi đã phân vùng được ảnh nghĩa là đã phân lập được thành các đối tượng, từ đó có thể phát hiện được biên cần tìm.

### ❖ Quy trình phát hiện biên

Bước 1: Do ảnh ghi được thường có nhiễu, bước đầu tiên của quá trình này là giảm nhiễu của ảnh cần xử lý.

Bước 2: Làm nổi biên sử dụng các toán tử phát hiện biên.

Bước 3: Định vị biên. Chú ý rằng kỹ thuật nổi biên gây tác dụng phụ là gây nhiễu làm một số biên giả xuất hiện do vậy cần loại bỏ biên giả.

Bước 4: Liên kết và trích chọn biên.

## 2.2 TỔNG QUAN VỀ MATLAB.

### 2.2.1 Tổng quan

Matlab (Matrix Laboratory) là một môi trường tính toán số và lập trình, được thiết kế bởi công ty MathWorks, là ngôn ngữ lập trình thực hành bậc cao được sử dụng nhiều để giải các bài toán kỹ thuật. Matlab tích hợp việc tính toán thể hiện kết quả cho phép lập trình, giao diện làm việc rất dễ dàng cho người sử dụng. Dữ liệu cùng với thư viện được lập trình sẵn cho phép người dùng có được những ứng dụng như:

- Tính toán các phép toán học thông thường, tính toán ma trận.
- Lập trình tạo ra những ứng dụng mới.
- Cho phép mô phỏng các mô hình thực tế.
- Phân tích, khảo sát, hiển thị dữ liệu.
- Với phần mềm đồ họa cực mạnh.

- Matlab giúp đơn giản hóa việc giải quyết các bài toán tính toán kỹ thuật so với các ngôn ngữ lập trình truyền thống như C, C++, và Fortran.

Matlab là một hệ thống tương giao chúng cho phép giải quyết các vấn đề liên quan đến lập trình bằng máy tính, đặc biệt sử dụng các phép tính về ma trận hay vector và có thể sử dụng ngôn ngữ C hoặc Fortran lập trình rồi thực hiện ứng dụng lập trình đó bằng các câu lệnh gọi từ Matlab.

Matlab được sử dụng trong nhiều lĩnh vực, bao gồm xử lý tín hiệu và ảnh, truyền thông, thiết kế điều khiển tự động, đo lường kiểm tra, phân tích mô hình tài chính, hay tính toán sinh học. Matlab cung cấp giải pháp chuyên dụng gọi là Toolbox. Toolbox là một tập hợp toàn diện các hàm của Matlab (M-file).

Hệ thống Matlab gồm 5 phần chính:

- Ngôn ngữ Matlab: Cho phép lập trình từ các ứng dụng nhỏ đến phức tạp. Đó là các ngôn ngữ cao về ma trận và mảng, các dòng lệnh, các hàm, cấu trúc dữ liệu vào.
- Môi trường làm việc: Bao gồm các phương tiện cho việc quản lý các biến trong không gian làm việc Workspace cũng như xuất nhập dữ liệu. Nó cũng bao gồm các công cụ phát triển, quản lý, gỡ rối và định hình M-file.
- Xử lý đồ họa: Bao gồm các lệnh cao cấp cho trực quan hóa dữ liệu hai chiều và ba chiều, xử lý ảnh, ảnh động. Cung cấp các giao diện tương tác giữa người sử dụng và máy tính.
- Thư viện toán học: Các hàm cơ bản như cộng, trừ, nhân, chia, sin, cos ... và các hàm phức tạp như tính ma trận nghịch đảo, trị riêng, chuyển đổi fourier, laplace, symbolic library.
- Giao diện người dùng (Application Program Interface): Cho phép viết chương trình tương tác với các ngôn ngữ khác C, Fortran ...
- Simulink là một chương trình đi kèm với Matlab, là một hệ thống tương tác với việc mô phỏng các hệ thống động phi tuyến, mô phỏng mạch.

### 2.2.2 Các hàm xử lý trong Matlab

Các hàm chính hiển thị ảnh trong matlab:

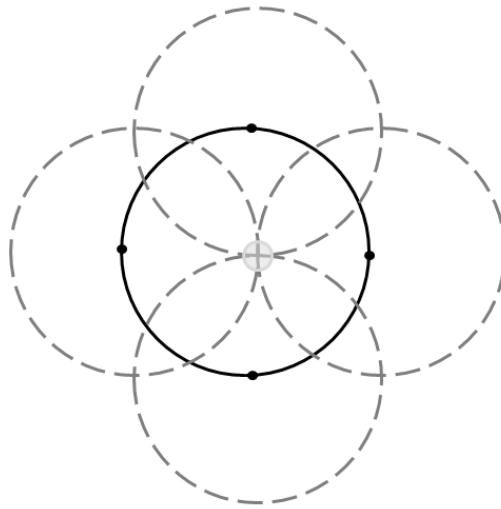


- Hàm **image(x,y,c)**: Hiển thị hình ảnh biểu diễn bởi ma trận  $c$  kích thước  $m \times n$  lên hệ trục tọa độ,  $x,y$  là các vector xác định vị trí của các pixel  $c(m,n)$ .
- Hàm **imagesc**: Tương tự hàm image, dữ liệu ảnh sẽ được co giãn để sử dụng toàn bộ bản đồ màu hiện hành.
- Hàm **imshow**: Cho phép hiển thị ảnh trên một figure và tự động thiết lập giá trị các đối tượng image, axes, figure để hiển thị hình ảnh.
- Các hàm khác được sử dụng trong đề tài:
- **int2str(x)**: Chuyển đổi số kiểu integer thành chuỗi ký tự.
- **size(a)**: Trả về giá trị là ma trận có dạng  $[x,y]$  là kích thước của ma trận  $a$ .
- **rgb2gray(RGB)** : Chuyển đổi ảnh màu sang ảnh xám
- **im2bw(I, level)**: Chuyển đổi hình ảnh xám sang hình ảnh nhị phân, dựa trên ngưỡng.
- **bwareaopen(BW, P)**: Loại bỏ các đối tượng nhỏ khỏi hình ảnh nhị phân
- **medfilt2(A)**: Lọc trung bình ma trận  $A$  theo hai chiều.
- **bwboundaries(BW)**: Đánh dấu biên giới các vật thể, cũng như ranh giới của lỗ bên trong các đối tượng này(BW).
- **bwlabel(BW)**: Kết nối các thành phần có trong hình ảnh nhị phân.
- **text(x,y,z,'string','PropertyName',PropertyValue)**: Tạo đối tượng văn bản trong các trục hiện tại.

### 2.2.3 Hàm xử lý phát hiện hình tròn

Hàm **imfindcircles** sử dụng thuật toán dựa trên Hough Transform với mục đích tìm hình tròn trong ảnh. Hàm này được sử dụng rộng rãi vì nó có thể xác định được đối tượng trong các chất lượng ảnh khác nhau như, ánh sáng hay đối tượng xếp chồng lên nhau.

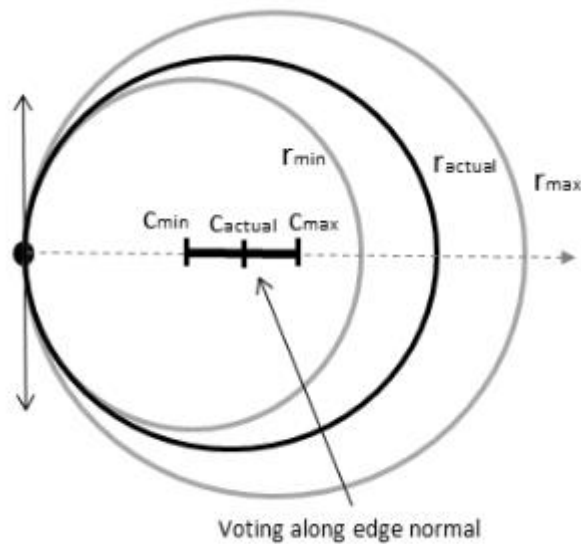
- **Phát hiện tâm**: Các điểm ảnh của một vòng tròn thì có xu hướng tích lũy thành mảng xung quanh tâm vòng tròn. Do đó, tâm vòng tròn được xác định từ đỉnh của mảng tích lũy.



*Hình 2. 1: Phát hiện tâm đường tròn*

Hình trên cho thấy một ví dụ về các điểm ảnh (chấm đen) nằm trên đường tròn (nét liền) và từ các điểm ảnh đó sẽ là tâm của các đường tròn (nét đứt), giao điểm các đường tròn này thì trùng khớp với tâm đường tròn thực tế.

- **Phát hiện bán kính:** Hàm imfindcircles sẽ cung cấp hai thuật toán để tìm bán kính đường tròn trong ảnh : **Phase – Coding** và **Two – Stage**



*Hình 2. 2: Phát hiện bán kính đường tròn*

### Trong đó:

Bảng 2. 1: Các hàm xử lý xác định bán kính trong Matlab

$r_{\min}$	Bán kính nhỏ nhất tìm được
$r_{\max}$	Bán kính lớn nhất tìm được
$r_{\text{actual}}$	Bán kính thực tế
$C_{\min}$	Tâm của đường tròn có $r_{\min}$
$C_{\max}$	Tâm của đường tròn có $r_{\max}$
$C_{\text{actual}}$	Tâm của đường tròn có $r_{\text{actual}}$

- **Vẽ đường viền:** đường viền có thể được giải thích đơn giản là một đường cong nối tất cả các điểm liên tục (dọc theo đường biên), có cùng màu sắc hoặc cường độ. Các đường viền là một công cụ hữu ích để phân tích hình dạng, phát hiện và nhận dạng đối tượng.

## 2.3 GIỚI THIỆU RASPBERRY PI 3

### 2.3.1 Giới thiệu



Hình 2. 3: Raspberry Pi 3B+

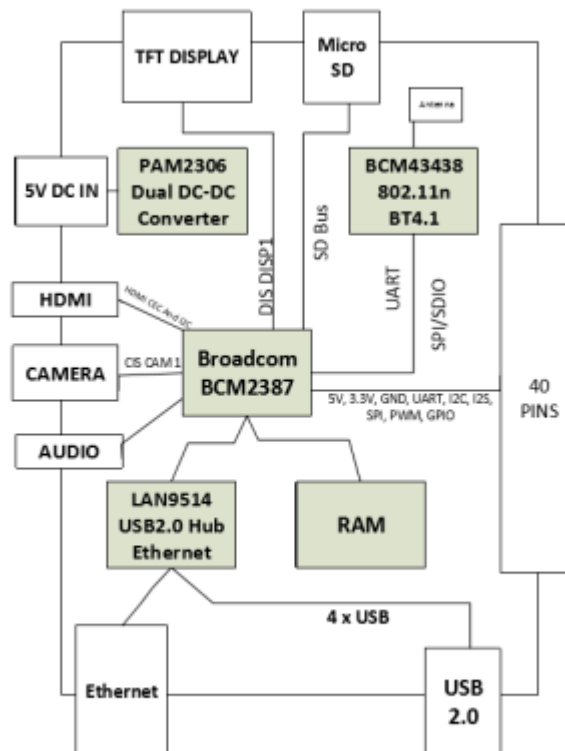
Raspberry Pi là một máy tính chỉ có một board mạch kích thước chỉ bằng một cái thẻ ATM. Người ta đã tích hợp mọi thứ cần thiết trong đó để bạn sử dụng như một cái máy vi tính, chạy hệ điều hành Linux, được phát triển tại Anh bởi Raspberry

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Pi Foundation với mục đích thúc đẩy việc giảng dạy về khoa học máy tính cơ bản trong các trường học và các nước đang phát triển.

Raspberry Pi xây dựng xoay quanh bộ xử lý SoC Broadcom BCM2835 bao gồm: CPU, GPU, bộ xử lý âm thanh /video, và các tính năng khác... Raspberry Pi có hai phiên bản, Model A và Model B. Ở đề tài này sử dụng Kit Raspberry Pi 3 Model B+ vì thông dụng hơn.

### 2.3.2 Thông tin cấu hình Raspberry Pi 3 Module B+



Hình 2. 4: Sơ đồ Raspberry Pi 3 Module B+

#### Thông tin cấu hình Raspberry Pi 3B+:

- Bộ xử lý Broadcom BCM2835 tốc độ xử lý 1.2ghz 64-bit quad-core ARM Cortex- A53.
- Mạng Wireless LAN chuẩn 802.11 b/g/n.
- Bộ xử lý đa phương tiện Videocore IV® Dual Core.
- Bộ nhớ Ram 1GB.
- Hỗ trợ tất cả các bản phân phối ARM GNU / Linux mới nhất và Windows 10 iot.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

- Đầu nối microusb cho nguồn điện 2,5A 5VDC.
- Cổng mạng 1 x 10/100.
- Đầu nối video / âm thanh 1 x HDMI.
- Đầu nối video / âm thanh 1 x RCA.
- 4 x USB 2.0 ports.
- 40 GPIO pins.
- Chip antenna.
- Kết nối hiển thị DSI.
- Khe cắm thẻ nhớ microsd.
- Kích thước: 85 x 56 x 17 mm.

**Ưu điểm:** Giá rẻ, nhỏ gọn, GPU mạnh, phục vụ cho nhiều mục đích. Khả năng hoạt động liên tục, giá thành rẻ hơn so với máy tính thông thường...

**Nhược điểm:** Máy tính nhúng xử lý vẫn còn chậm so với máy tính để bàn hay laptop. Không chạy được một số chương trình nặng.

### 2.3.3 Ứng dụng

Hệ thống máy tính cho ô tô: chúng ta có thể dùng Raspberry Pi làm hệ thống máy cho xe hơi, Sử dụng Raspberry Pi để khắc laser, chế tạo robot,...

## 2.4 GIỚI THIỆU VỀ CAMERA PI

### 2.4.1 Giới thiệu

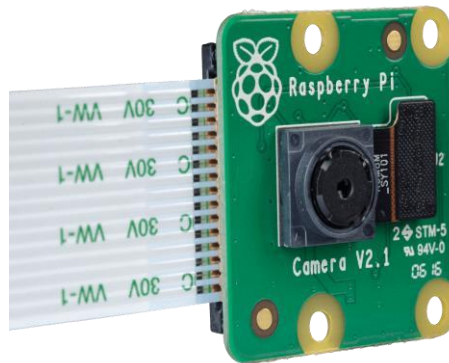
Raspberry Pi Camera là module camera được chính Raspberry Pi Foundation thiết kế và đưa vào sản xuất đại trà từ tháng 5/2013. Camera module ra đời đã làm thoả lòng rất nhiều tín đồ yêu thích Raspberry. Trước khi xuất hiện camera, điều duy nhất bạn có thể làm để thêm khả năng nhận biết hình ảnh, quay phim, chụp hình cho RPi là sử dụng 1 webcam cắm vào cổng USB. Với các webcam Logitech tích hợp sẵn định dạng xuất mjpeg sẽ giúp Raspberry xử lý nhanh hơn. Nhưng các webcam Logitech lại có giá thành khá cao, nhất là các webcam có độ phân giải lớn. Bạn không tốn thêm cổng USB nào cho Camera vì Camera được gắn chắc chắn vào socket CSI. Điều này giúp hạn chế tình trạng nghẽn băng thông cho chip xử lý USB trên mạch Raspberry. Chiều dài cáp nối camera đã được tính toán cẩn thận khi vừa đạt được độ

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

dài cần thiết trong khi vẫn đảm bảo tốc độ truyền hình ảnh từ module về Raspberry Pi. Tháng 4/2016, Raspberry Pi Foundation ra mắt thế hệ thứ 2 của sản phẩm. Camera Module với nâng cấp đáng kể nhất là sử dụng sensor Sony IMX219 8 Megapixel.

Raspberry Pi Camera Module V2 có một cảm biến 8-megapixel của Sony IMX219 (so với cảm biến 5-megapixel OmniVision OV5647 trên Camera Module phiên bản cũ).



Hình 2. 5: Camera Pi v2.1

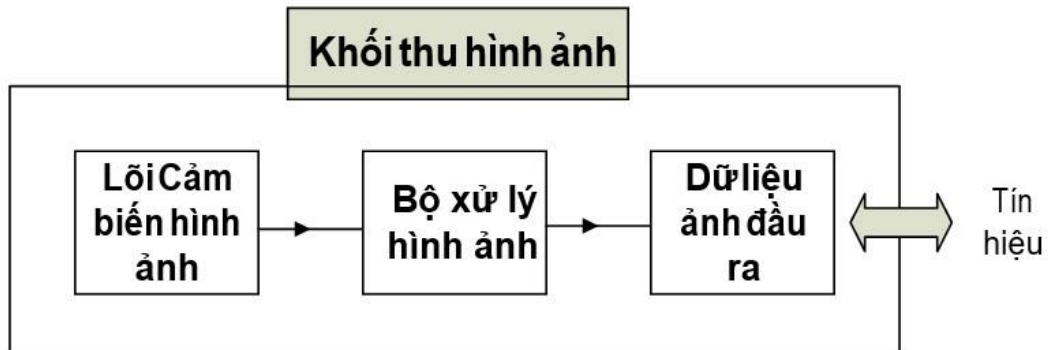
Camera Module có thể được sử dụng để quay video độ nét cao, cũng như chụp hình ảnh tĩnh. Nó khá dễ dàng để sử dụng cho người mới bắt đầu, nhưng cũng có rất nhiều giải pháp mở rộng để cung cấp cho người dùng yêu cầu cao. Có rất nhiều demo của người dùng về công dụng của Camera Module như chụp Time-Lapse, Slow - Motion và rất nhiều ứng dụng khác.

Raspberry Pi Camera Module V2 là một bước nhảy vọt về chất lượng hình ảnh, màu sắc trung thực và hiệu suất ánh sáng thấp. Đặc biệt nó hỗ trợ video lên tới 1080P30, 720P60 và video mode VGA90, cũng như chế độ chụp hình. Dĩ nhiên, nó vẫn sử dụng đoạn cáp 15cm qua cổng CSI trên Raspberry Pi. Chiếc camera này tương thích với tất cả các phiên bản của Raspberry Pi 1, 2 và 3.

### 2.4.2 Thông tin cấu hình Camera Pi v2.1

- Ống kính tiêu cự cố định.
- Cảm biến độ phân giải 8 megapixel cho khả năng chụp ảnh kích thước 3280 x 2464.
- Hỗ trợ video 1080p30, 720p60 và 640x480p90.
- Kích thước 25mm x 23mm x 9mm.

- Trọng lượng chỉ hơn 3g.
- Kết nối với Raspberry Pi thông qua cáp ribbon đi kèm dài 15 cm.
- Camera Module được hỗ trợ với phiên bản mới nhất của Raspbian.



Hình 2. 6: Sơ đồ khối Camera Raspberry Pi

### 2.4.3 Ứng dụng

Bạn có thể dùng Camera và Raspberry để thiết lập hệ thống phát hiện chuyển động. Hệ thống này hoạt động bằng cách sử dụng hình ảnh từ Camera kết hợp với chương trình xử lý hình ảnh và đưa ra lệnh điều khiển nếu phát hiện thấy có chuyển động. Camera cũng đặc biệt hữu ích với các bạn yêu thích làm phim, bạn có thể sử dụng để quay những góc quay khó hoặc những cảnh quay độc mà chỉ với máy quay gọn nhẹ nhất mới làm được. Ngoài ra, bạn có thể sử dụng camera để quay các đoạn phim time-lapse (ghép nhiều hình lại với nhau) đang được rất nhiều người dùng trên thế giới thực hiện.

## 2.5 GIỚI THIỆU VỀ CẢM BIẾN HỒNG NGOẠI E18-D80NK

### 2.5.1 Giới thiệu

Cảm biến vật cản hồng ngoại E18-D80NK dùng ánh sáng hồng ngoại để xác định khoảng cách tới vật cản cho độ phản hồi nhanh và rất ít nhiễu do sử dụng mắt nhận và phát tia hồng ngoại theo tần số riêng biệt. Cảm biến có thể chỉnh khoảng cách mong muốn thông qua biến trở.



Hình 2. 7: Cảm biến hồng ngoại E18-D80NK

### 2.5.2 Thông số kỹ thuật

Thông số kỹ thuật cảm biến E18-D80NK:

- Nguồn điện cung cấp: 5VDC.
- Khoảng cách phát hiện: 3 ~ 80cm.
- Có thể điều chỉnh khoảng cách qua biến trở.
- Dòng kích ngõ ra: 300mA.
- Ngõ ra dạng NPN cực thu hở giúp tùy biến được điện áp ngõ ra.
- Chất liệu sản phẩm: nhựa.
- Có led hiển thị ngõ ra màu đỏ.
- Kích thước: 1.8cm (D) x 7.0cm (L).

Sơ đồ chân:

- Màu nâu: VCC, nguồn dương 5VDC.
- Màu xanh dương: GND, nguồn âm 0VDC
- Màu đen: Chân tín hiệu ngõ ra cực thu hở NPN.

### 2.5.3 Ứng dụng

Các ứng dụng của cảm biến E18-D80NK: Xe dò tự động tránh vật cản, trong các dây chuyền sản xuất tự động.



### 2.6 GIỚI THIỆU VỀ ĐỘNG CƠ SERVO MG996R

#### 2.6.1 Giới thiệu động cơ servo MG996R

Động cơ RC Servo MG996 là loại thường được sử dụng nhiều nhất trong các thiết kế robot hoặc dẫn hướng xe. Động cơ RC Servo MG996 có lực kéo mạnh, các khớp và bánh răng được làm hoàn toàn bằng kim loại nên có độ bền cao, động cơ được tích hợp sẵn driver điều khiển động cơ bên trong theo cơ chế phát xung - quay góc nên rất dễ sử dụng.



*Hình 2. 8: Động cơ servo MG996R*

#### 2.6.2 Thông số kỹ thuật

- **Chủng loại:** Analog RC Servo.
- **Điện áp hoạt động:** 4.8-6.6VDC
- **Kích thước:** 40mm x 20mm x 43mm
- **Trọng lượng:** 55g.

#### 2.6.3 Ứng dụng

Trong ngành công nghiệp động cơ servo được sử dụng trong các máy công cụ, đóng gói, tự động hóa nhà máy, xử lý vật liệu, in chuyển đổi. Ngoài ra chúng còn được sử dụng để chế tạo cánh tay robot vì sử dụng chuyển đổi trơn tru và chính xác.

### 2.7 GIỚI THIỆU MODULE RELAY 5V

#### 2.7.1 Tổng quan về module relay 5V



Hình 2. 9: Module Relay 1 kênh

Module 1 relay 5V với opto cách ly kích H/L với opto cách ly nhỏ gọn, có opto và transistor cách ly giúp cho việc sử dụng trở nên an toàn với board mạch chính, mạch được sử dụng để đóng ngắt nguồn điện công suất cao AC hoặc DC, có thể chọn đóng khi kích mức cao hoặc mức thấp bằng Jumper. Tiếp điểm đóng ngắt gồm 3 tiếp điểm NC (thường đóng), NO (thường mở) và COM (chân chung) được cách ly hoàn toàn với board mạch chính, ở trạng thái bình thường chưa kích NC sẽ nối với COM, khi có trạng thái kích COM sẽ chuyển sang nối với NO và mất kết nối với NC.

Thông số kỹ thuật:

- Sử dụng điện áp DC 5V.
- Mỗi relay tiêu thụ dòng khoảng 80mA.
- Điện thế đóng ngắt tối đa: AC250V ~ 10A hoặc DC30V ~ 10A.
- Có đèn báo đóng ngắt trên mỗi relay.
- Có thể chọn mức tín hiệu kích 0 hoặc 1 qua jumper.
- Kích thước: 1.97 in x 1.02 in x 0.75 in (5.0 cm x 2.6 cm x 1.9 cm).
- Khối lượng: 17 g.

### 2.8 GIỚI THIỆU HỆ THỐNG BĂNG TẢI

#### 2.8.1 Giới thiệu

Băng tải (băng chuyền) hiểu đơn giản là một máy cơ khí dùng để vận chuyển các đồ vật từ điểm này sang điểm khác, từ vị trí A sang vị trí B. Thay vì vận chuyển sản phẩm bằng công nhân vừa tốn thời gian, chi phí nhân công lại tạo ra môi trường làm việc lộn xộn thì băng chuyền tải có thể giải quyết điều đó. Nó giúp tiết kiệm sức lao động, số lượng nhân công, giảm thời gian và tăng năng suất lao động.

Vì vậy băng chuyền, băng tải là một trong những bộ phận quan trọng trong dây chuyền sản xuất, lắp ráp của các nhà máy, xí nghiệp. Góp phần tạo nên một môi trường sản xuất hiện đại, khoa học và giải phóng sức lao động mang lại hiệu quả kinh tế cao cho công ty.



*Hình 2. 10: Mô hình băng tải*

#### 2.8.2 Cấu tạo

- Khung băng tải: thường được làm bằng nhôm định hình, thép sơn tĩnh điện hoặc inox.
- Dây băng tải: thường là dây băng PVC dày 2mm và 3mm hoặc dây băng PU độ dày 1.5mm.
- Động cơ chuyển động: là động cơ giảm tốc công suất 0.2KW, 0.4KW, 0.75KW, 1.5KW, 2.2KW.

- Bộ điều khiển bằng chuyên: thường gồm có biến tần, sensor, timer, PLC...
- Cơ cấu truyền động gồm có: rulo kéo, con lăn đỡ, không xích...

### 2.8.3 Ứng dụng

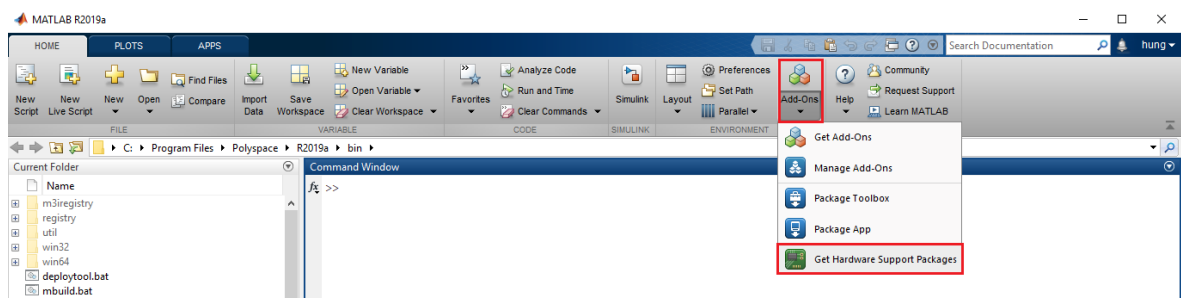
Băng tải, băng chuyền được ứng dụng trong các ngành sản xuất công nghiệp, chế biến thực phẩm, khai thác, nhằm giúp hỗ trợ tối đa cho doanh nghiệp trong quá trình vận chuyển hàng hóa nhanh chóng, an toàn, tiện lợi. Ngoài ra với hệ thống băng tải còn giúp doanh nghiệp tối ưu hóa được chi phí, tiết kiệm thời gian, hạn chế nguồn nhân lực mang lại hiệu quả kinh tế cao đồng thời còn giúp cho hệ thống sản xuất ngày càng được tự động hóa theo hướng hiện đại.

## 2.9. CÀI ĐẶT HỖ TRỢ PHẦN CỨNG MATLAB CHO RPI

### ❖ Kết nối Raspberry với Matlab.

Gói hỗ trợ MATLAB® cho phần cứng Raspberry Pi cho phép người sử dụng giao tiếp với Raspberry Pi từ xa thông qua máy tính có cài đặt MATLAB. Chúng ta có thể thu thập dữ liệu từ các cảm biến và thiết bị hình ảnh được kết nối với Raspberry Pi và xử lý chúng trong MATLAB. Bạn cũng có thể giao tiếp với các phần cứng khác thông qua các chân GPIO, nối tiếp, I2C và SPI. Ở đề tài này nhóm sử dụng Module Raspberry 3B+ và Matlab 2019a. Để cài đặt gói Package làm như sau:

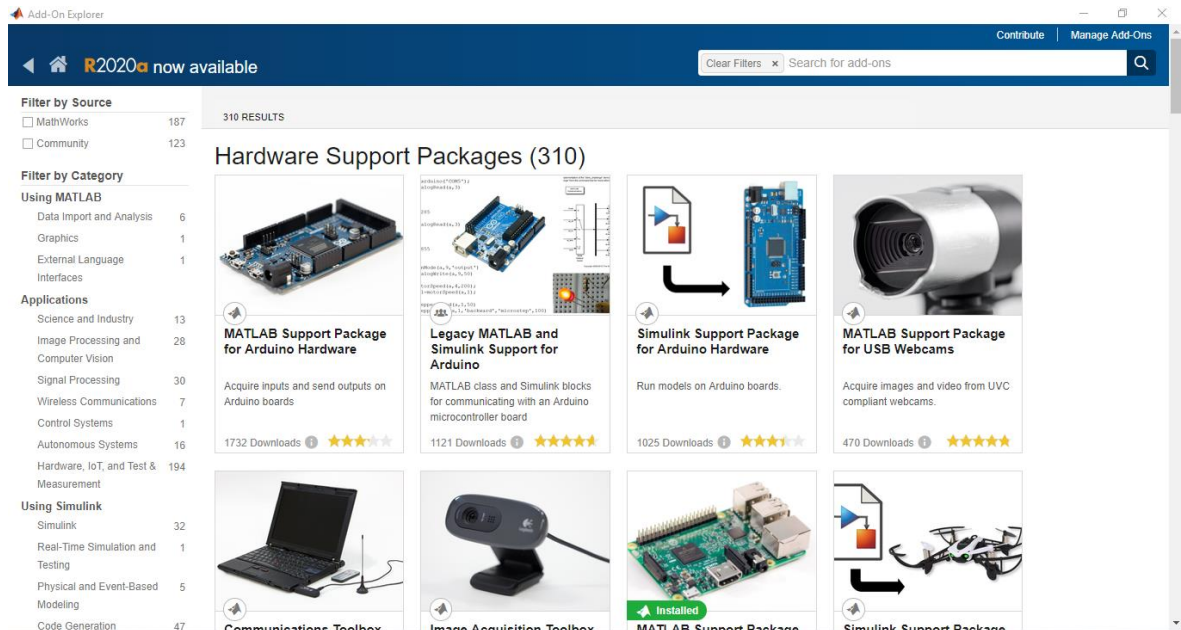
Từ giao diện Matlab, click vào **Add-Ons** chọn **Get Hardware Support Package**.



Hình 2. 11: Get Hardware Support Package.

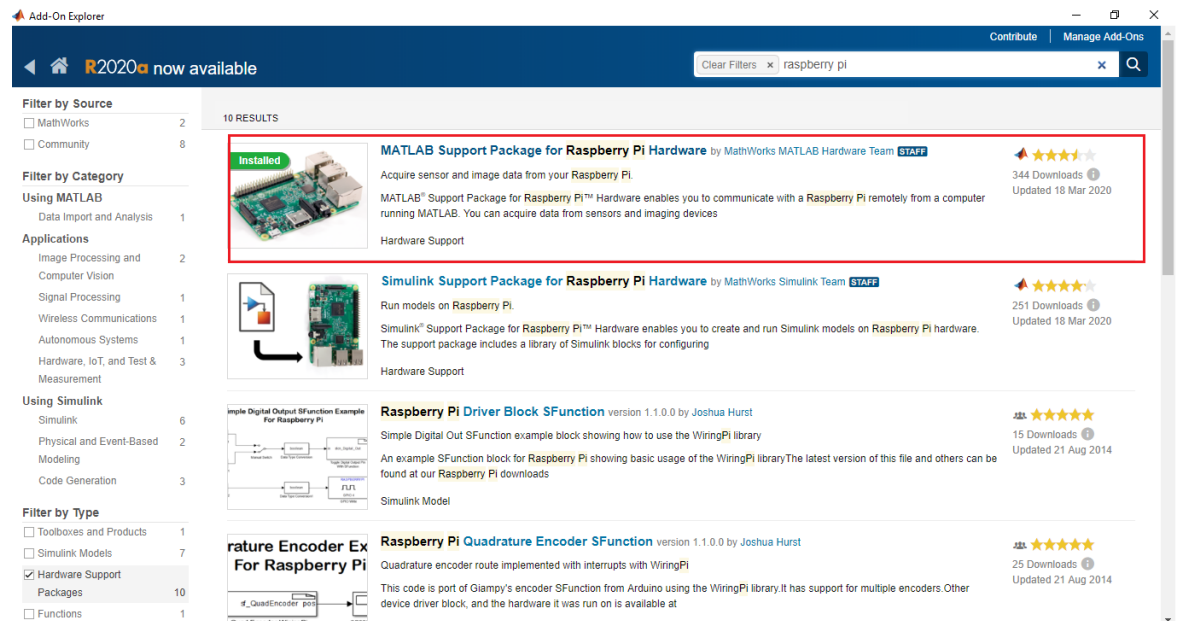
Cửa sổ Support Package Installer hiện lên

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2. 12: Cửa sổ “Support Package Installer”.

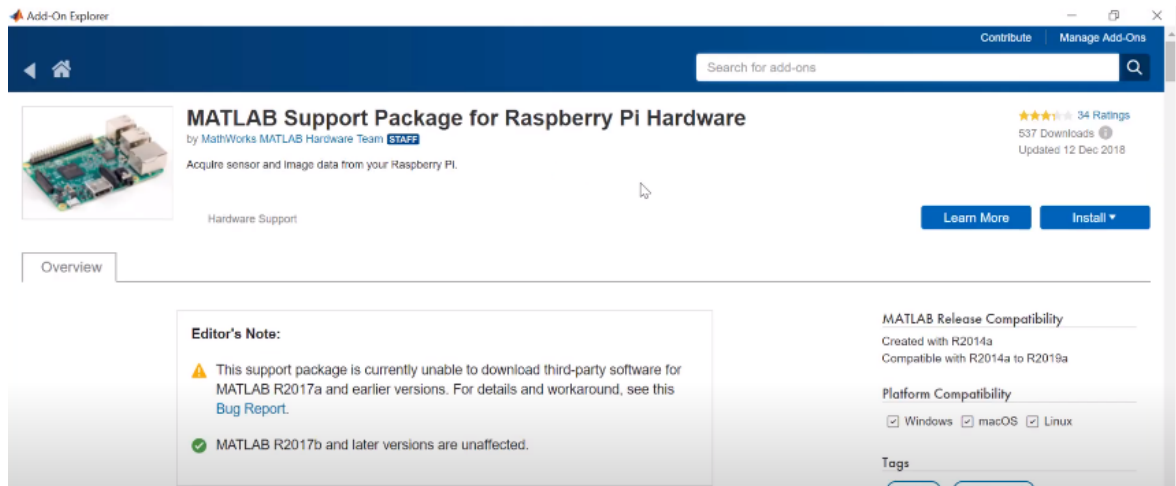
Ở thanh tìm kiếm gõ “raspberry pi”. Kết quả tìm kiếm hiện ra, tiến hành chọn gói MATLAB Support Package for Raspberry Pi Hardware.



Hình 2. 13: Giao diện cài Package cho Raspberry

Chọn **Install** và tiến hành cài đặt Package cho Raspberry.

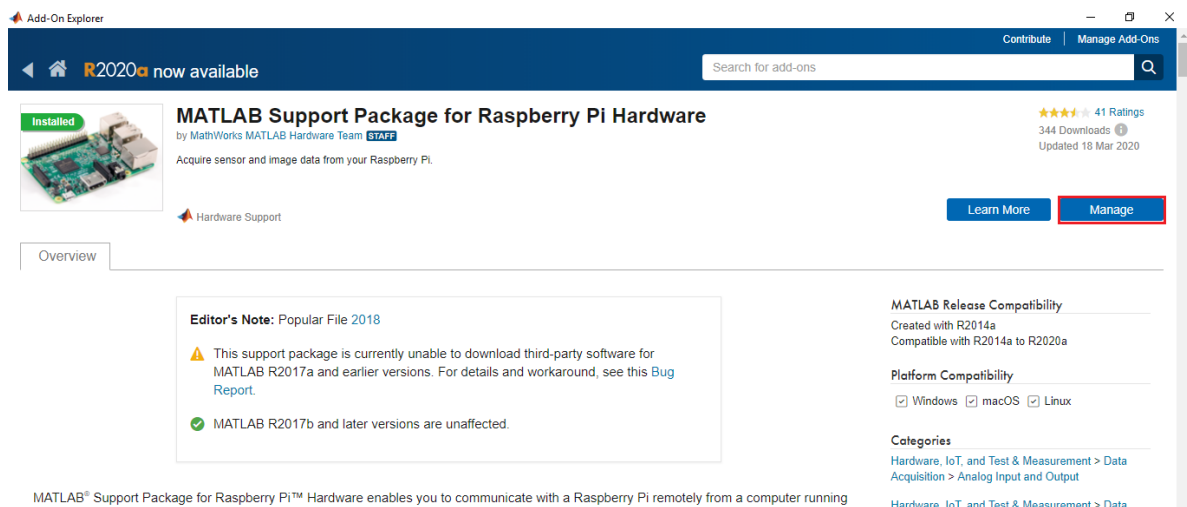
## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2. 14: Đăng nhập cho cửa sổ “MathWorks Account Log In”

Tiến hành đăng nhập với tài khoản Matlab, nhấn **Accept** để cài đặt. Nhấn **Setup later** để kết thúc phần cài đặt.

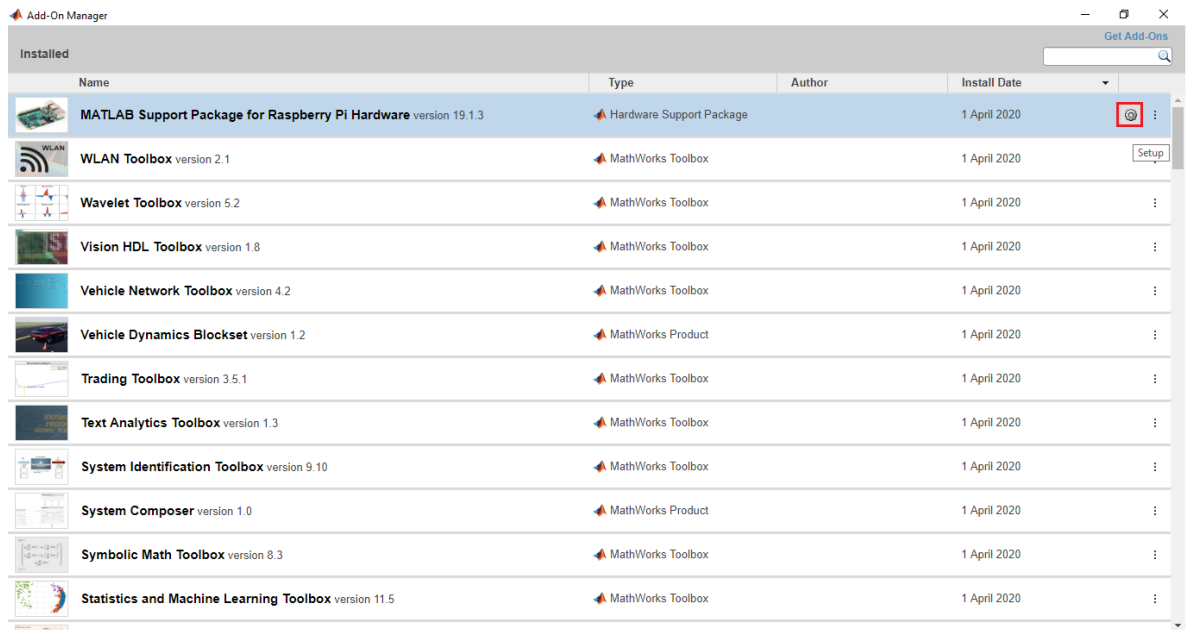
Sau khi cài đặt sau trở lại cửa sổ MATLAB Support Package for Raspberry Pi Hardware. Nút **Install** chuyển thành **Manage**, chọn **Manage**.



Hình 2. 15: Matlab Support Package for Raspberry Pi Hardware

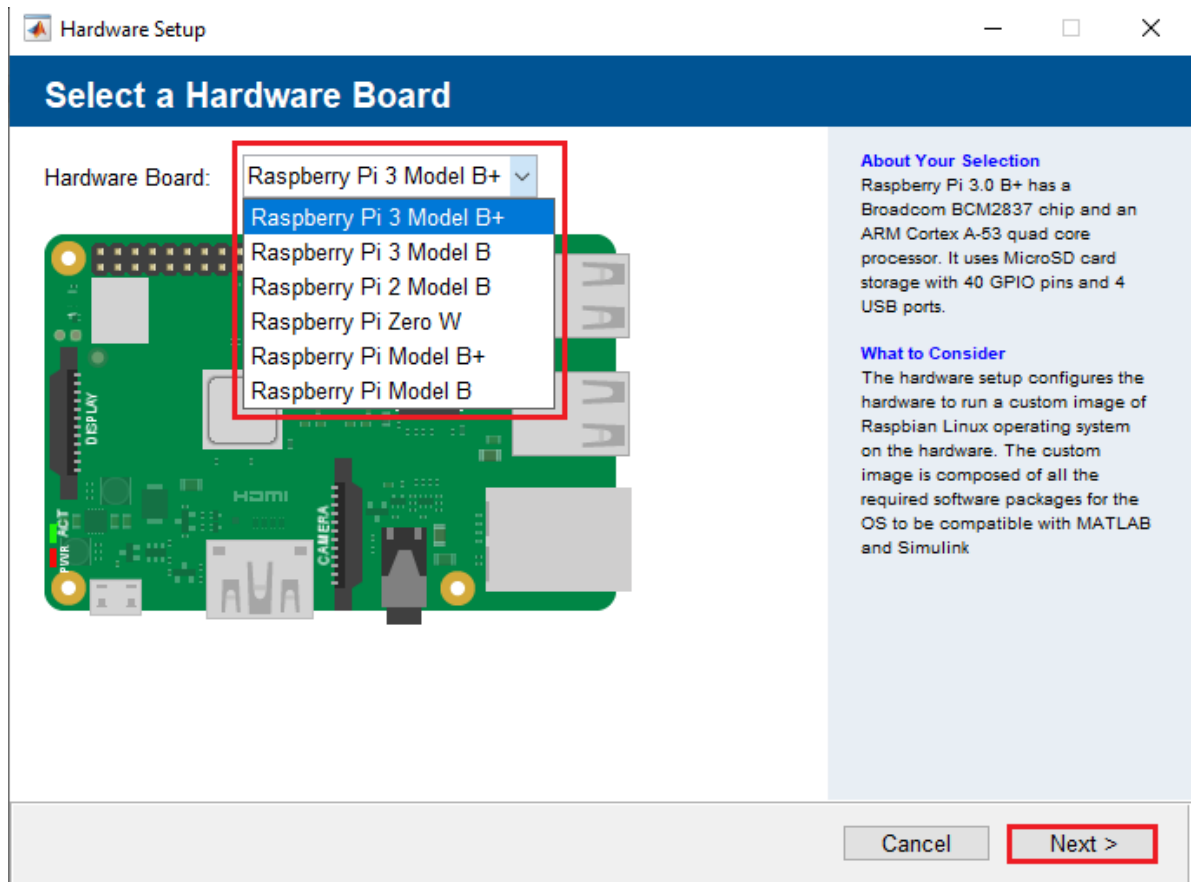
Cửa sổ **Add-On Manager** hiện lên, chọn **Setup** (có hình bánh răng của như hình dưới)

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



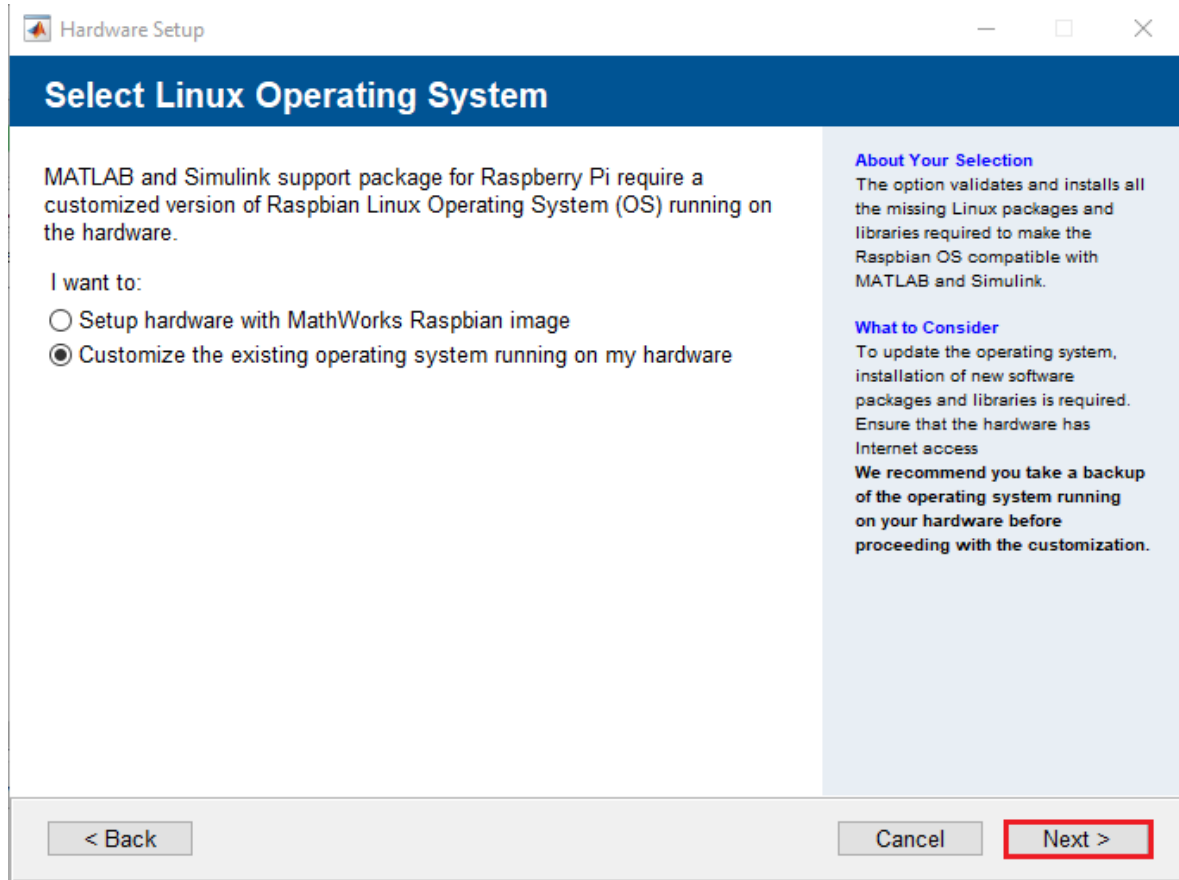
Hình 2. 16: Cửa sổ Add -On

Cửa sổ cài đặt Hardware Setup hiện lên, lựa chọn module Raspberry Pi phù hợp và chọn Next.



Hình 2. 17: Lựa chọn phiên bản Raspberry Pi

Chọn **Customize the existing operating system running on my hardware** và nhấn **Next**.



Hình 2. 18: Lựa chọn cách cài đặt

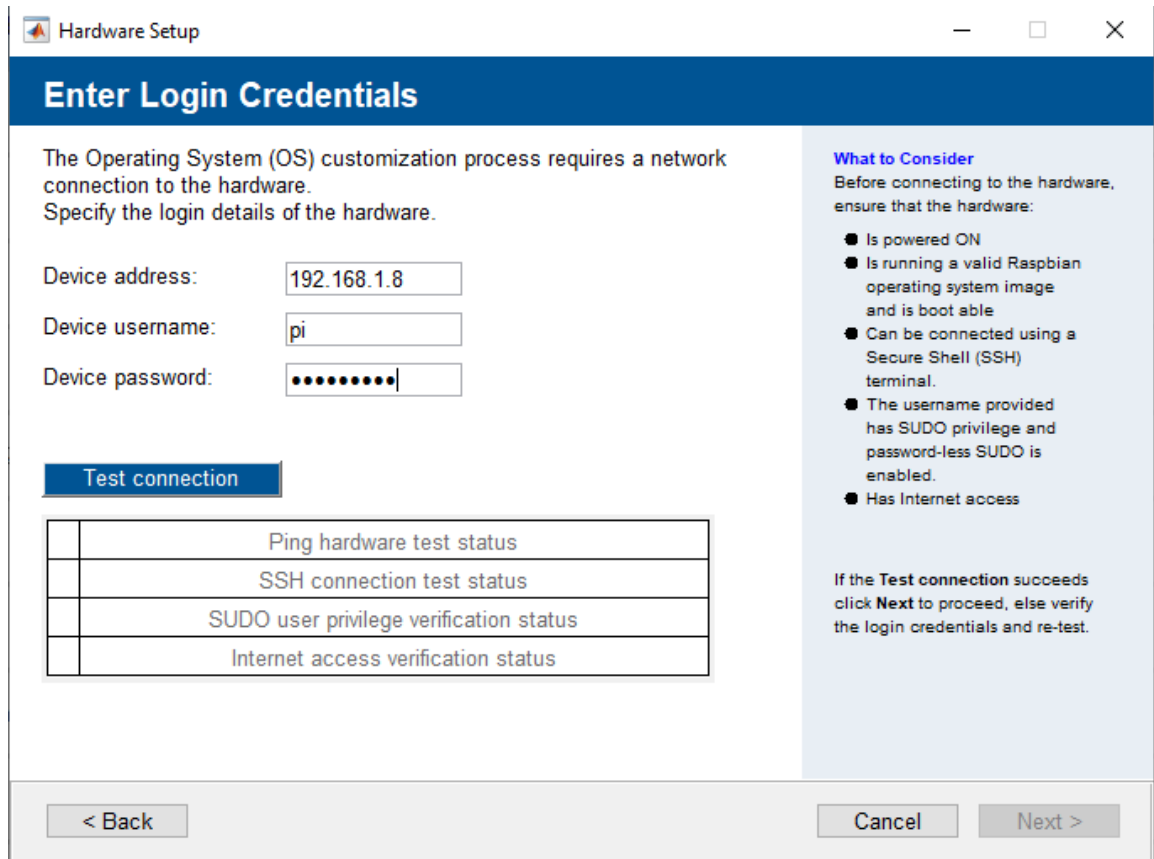
Ở bước này có 2 lựa chọn:

- Cài đặt phần cứng với hệ điều hành Matlab Raspbian.
- Cài đặt hệ điều hành có sẵn chạy trên phần cứng.

Ở phần này, nhóm lựa chọn cách cài đặt hệ điều hành có sẵn chạy trên phần cứng, cách cài đặt còn lại sẽ được giới thiệu ở phần tiếp theo.

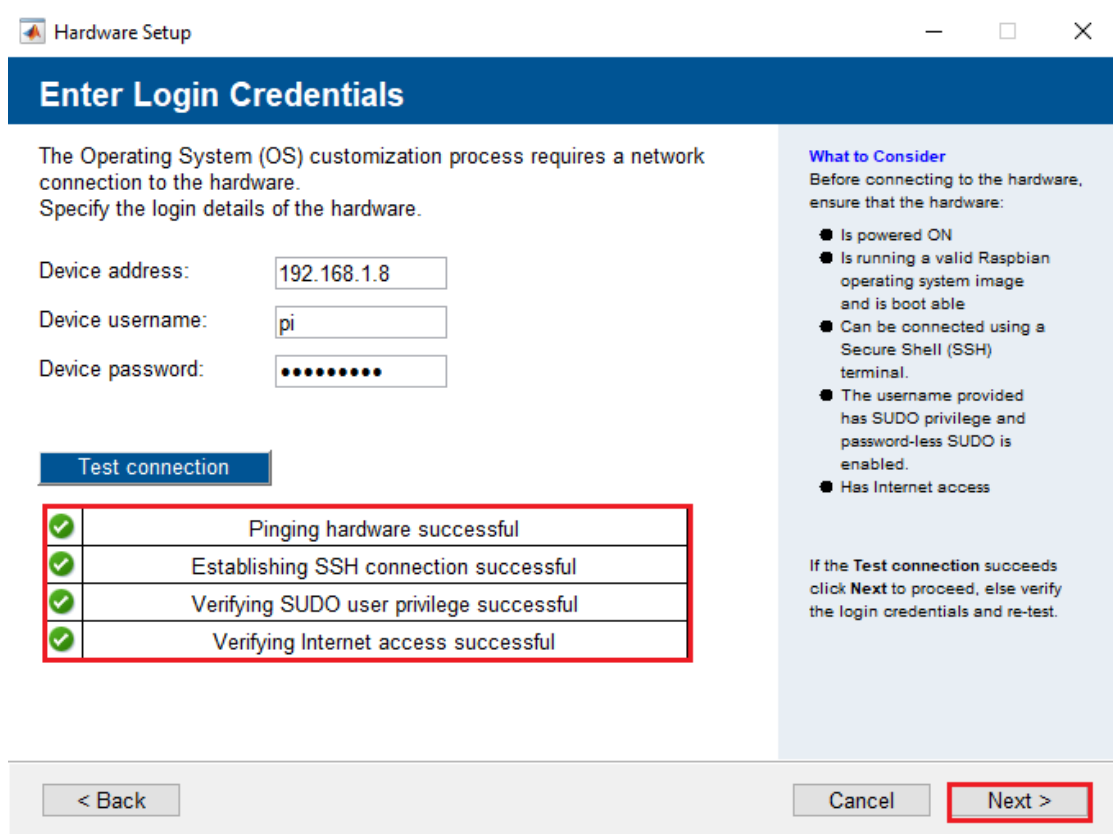
**Cách cài đặt thứ nhất:**





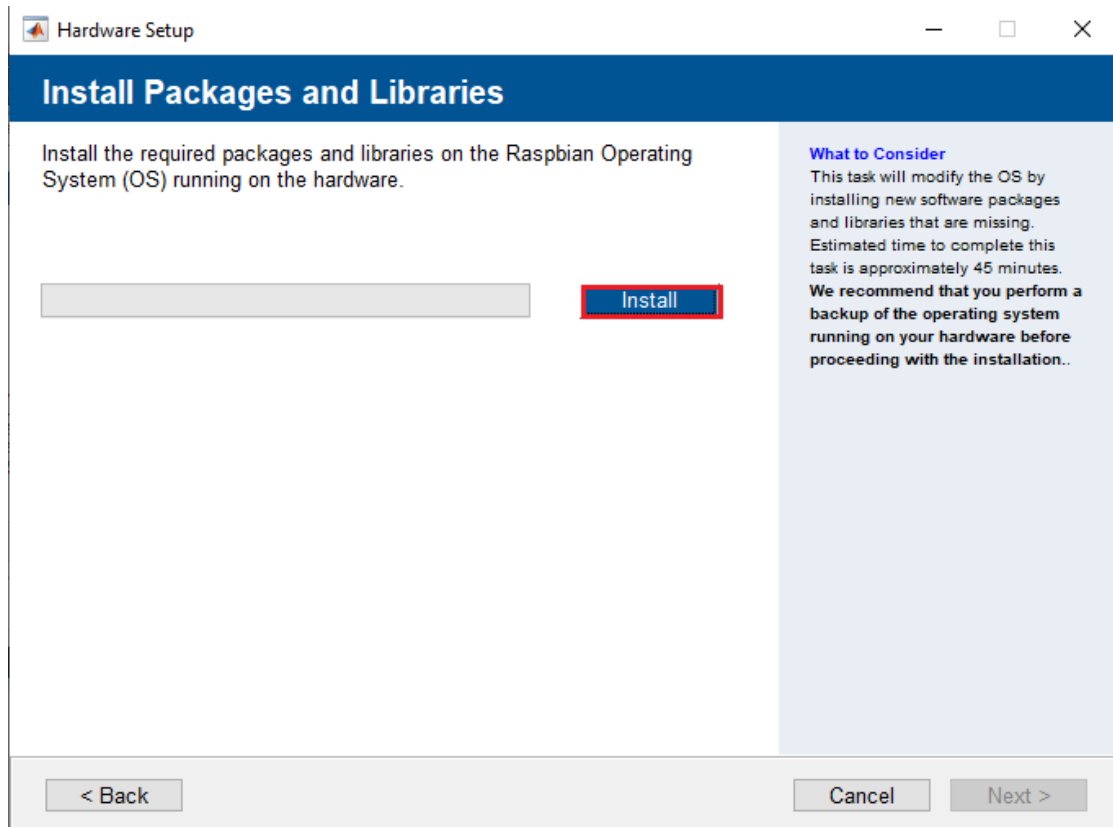
Hình 2. 19: Kiểm tra kết nối Raspberry Pi

Tiến hành nhập địa chỉ của Raspberry, tên thiết bị và mật khẩu. Sau đó chọn **Test connection** để kiểm tra việc kết nối. Quá trình sẽ tốn một khoảng thời gian ngắn để kiểm tra, nếu việc kiểm tra kết nối thành công sẽ có thông báo như hình sau:



Hình 2. 20: Kiểm tra kết nối Raspberry thành công

Kết nối thành công, nhấn **Next** cho đến khi xuất hiện **Install**.



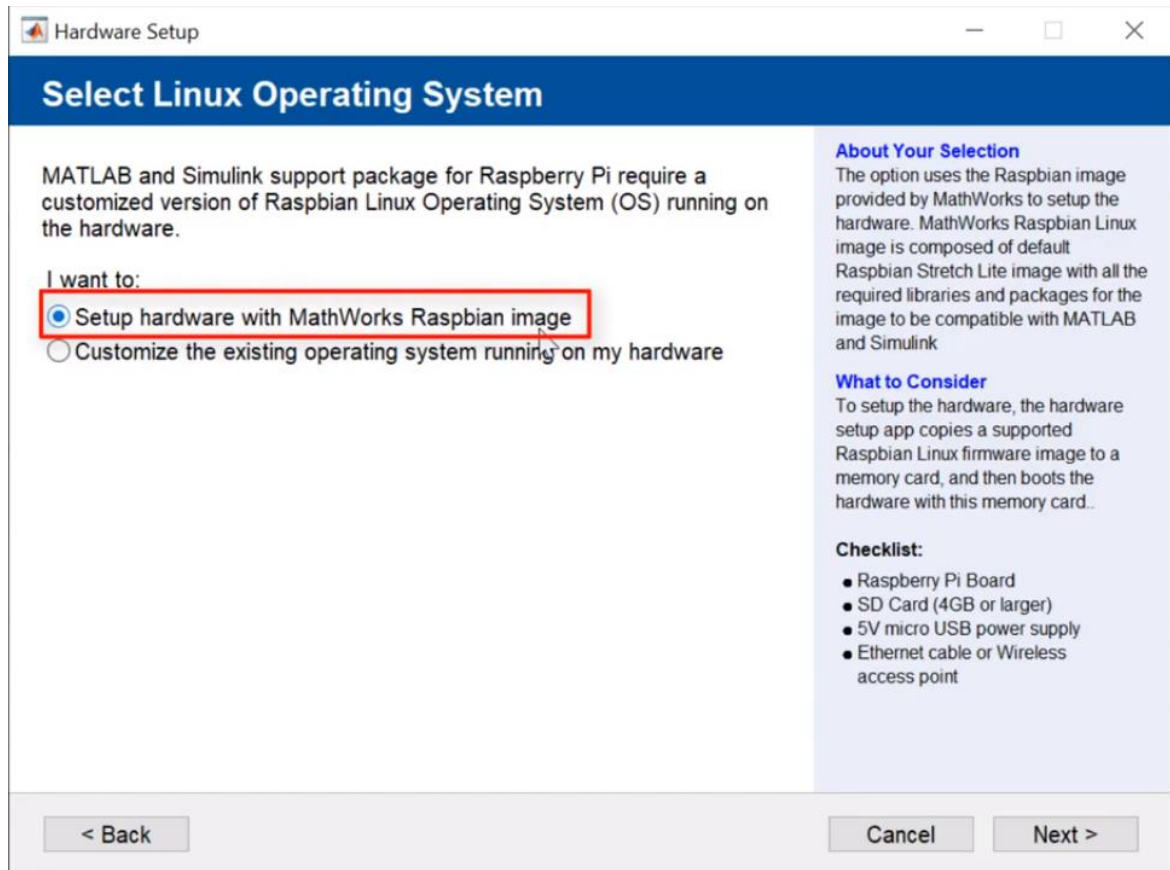
Hình 2. 21: Cài đặt gói hỗ trợ Raspberry Pi

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Nhấn **Install** để tiến hành cài đặt cho đến khi hoàn thành, chọn **Reboot later** để hoàn tất việc cài đặt theo cách thứ nhất.

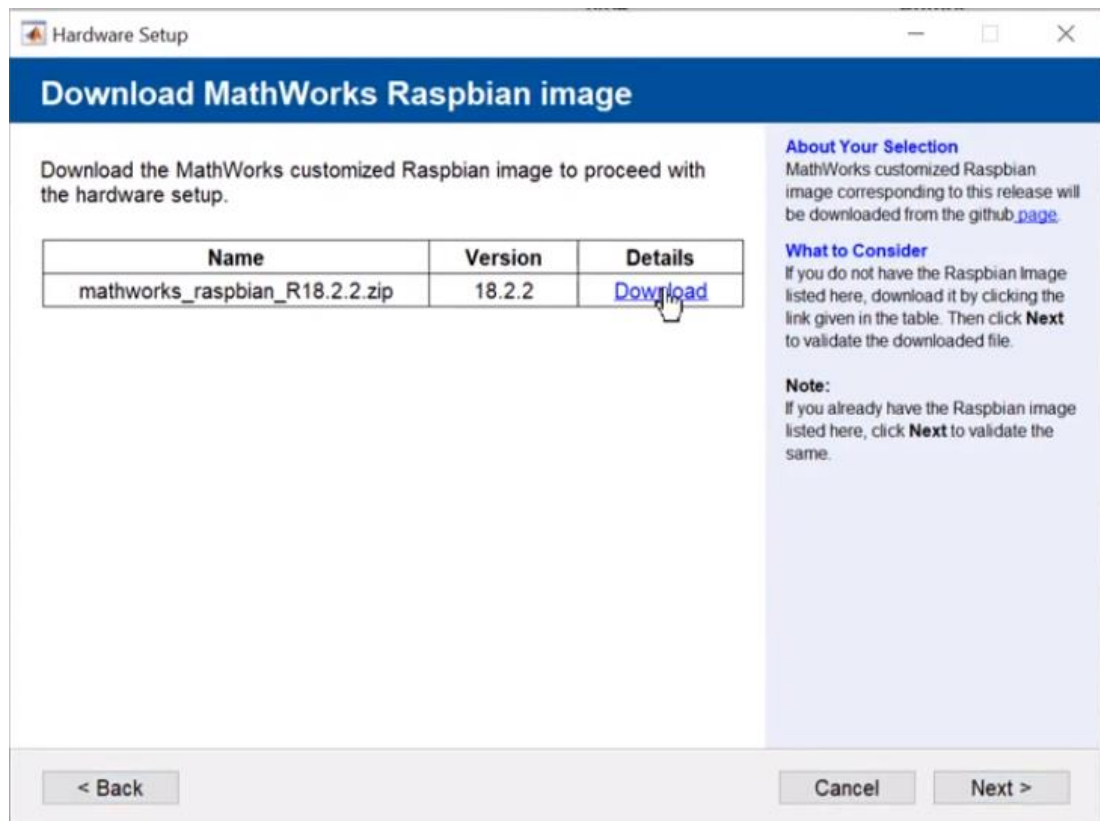
### Cách cài đặt thứ hai:

Ở bước lựa chọn cài đặt, chọn “**Setup hardware with Mathworks Raspbian image**”. Sau đó nhấn **Next**.



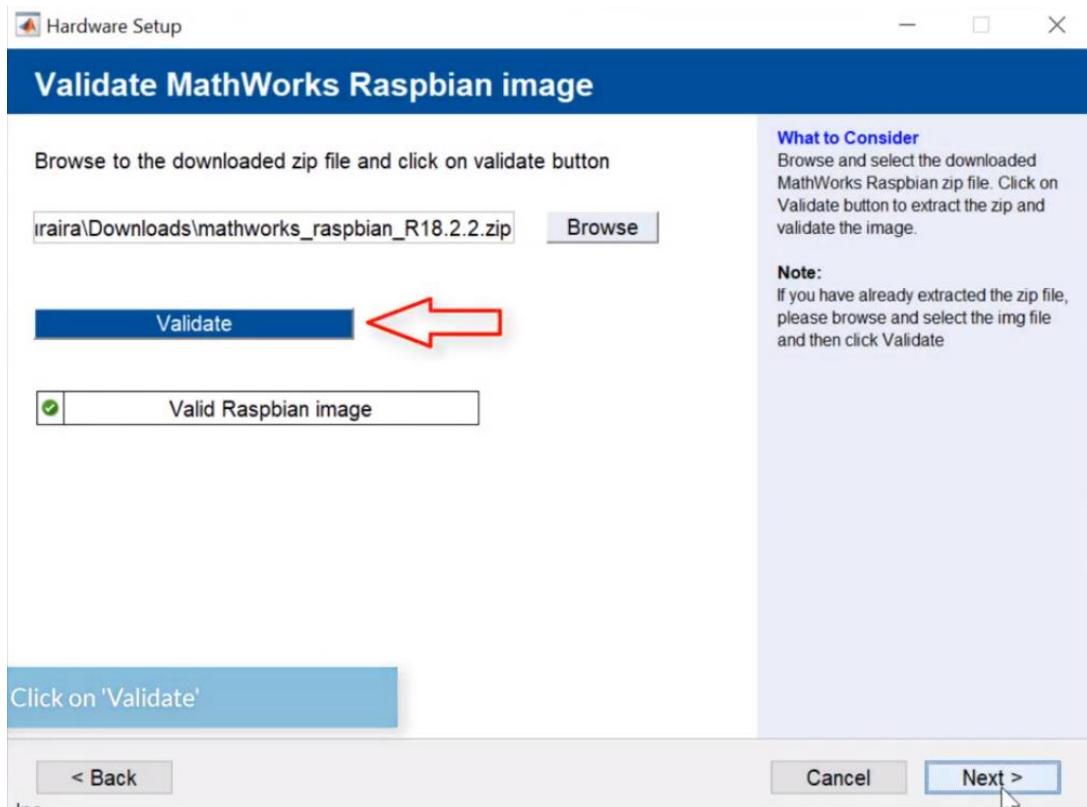
Hình 2. 22: Lựa chọn cài đặt

Click vào **Download** để tải, sau khi tải hoàn tất chọn **Next** để tiếp tục.



Hình 2. 23: Download hệ điều hành Mathworks Raspbian

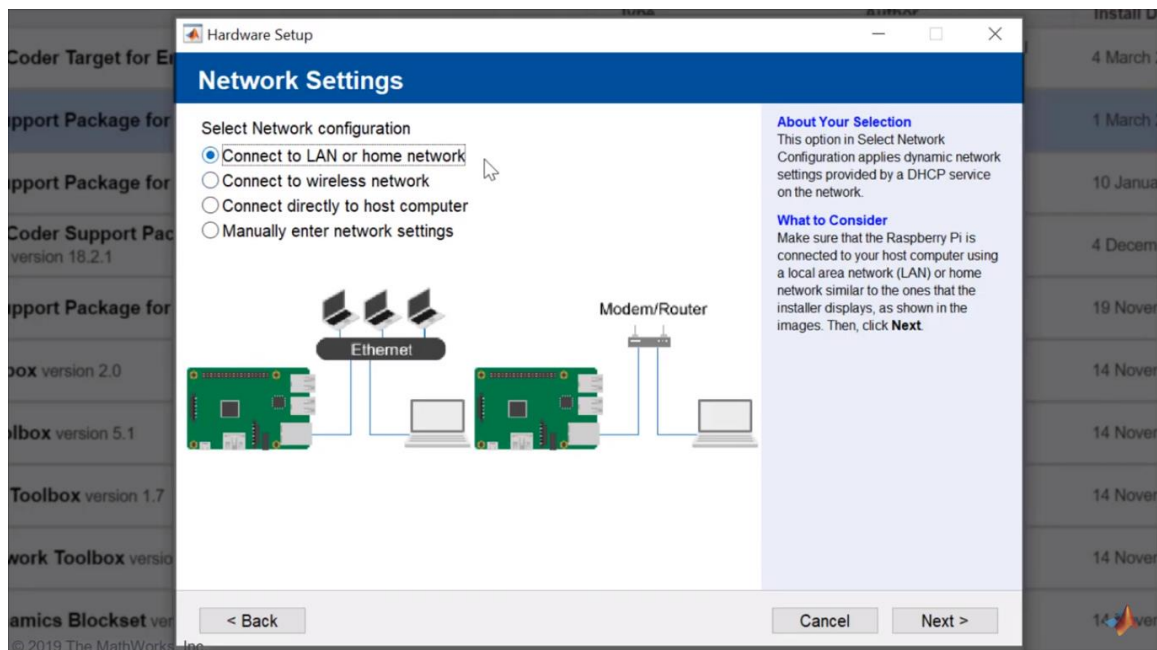
Click vào **Browse** để chọn lại đường dẫn tệp vừa tải xuống ở bước trước, sau đó Click vào **Validate** để kiểm tra. Chọn **Next** để tiếp tục.



Hình 2. 24: Kiểm tra hệ điều hành Mathworks Raspbian

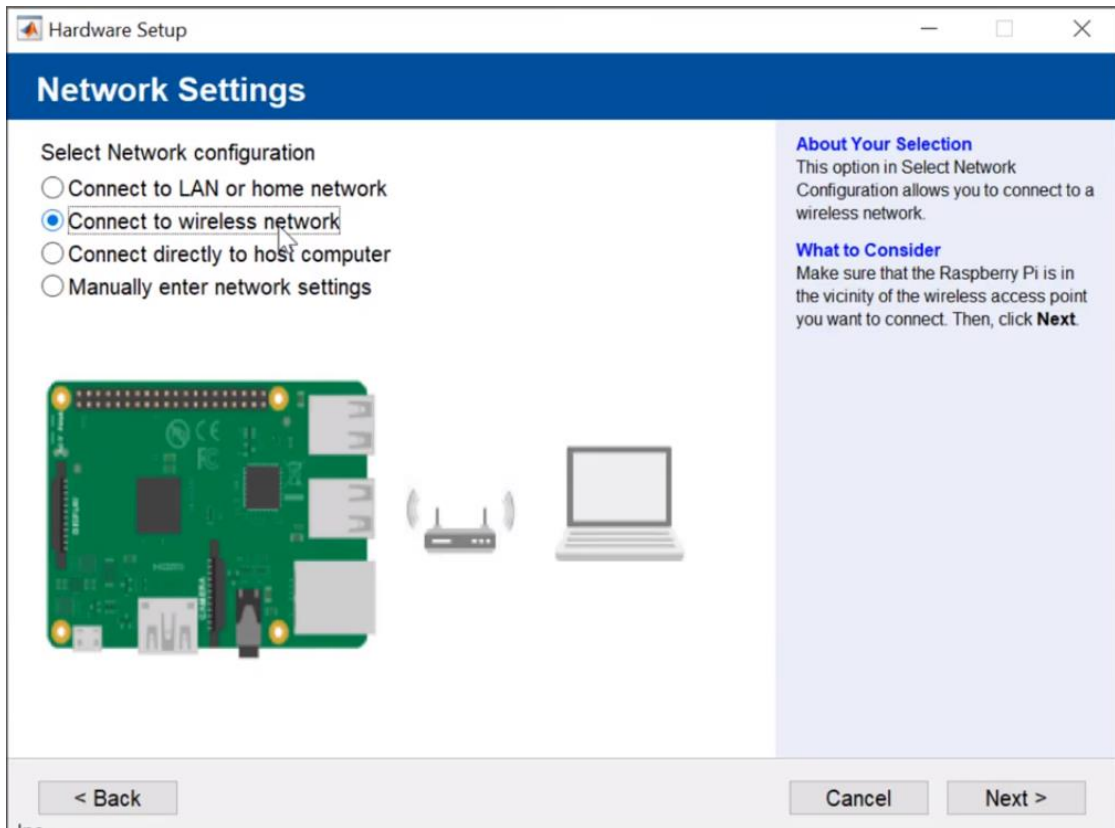
Ở bước này có 4 lựa chọn:

- Kết nối thông qua mạng LAN.



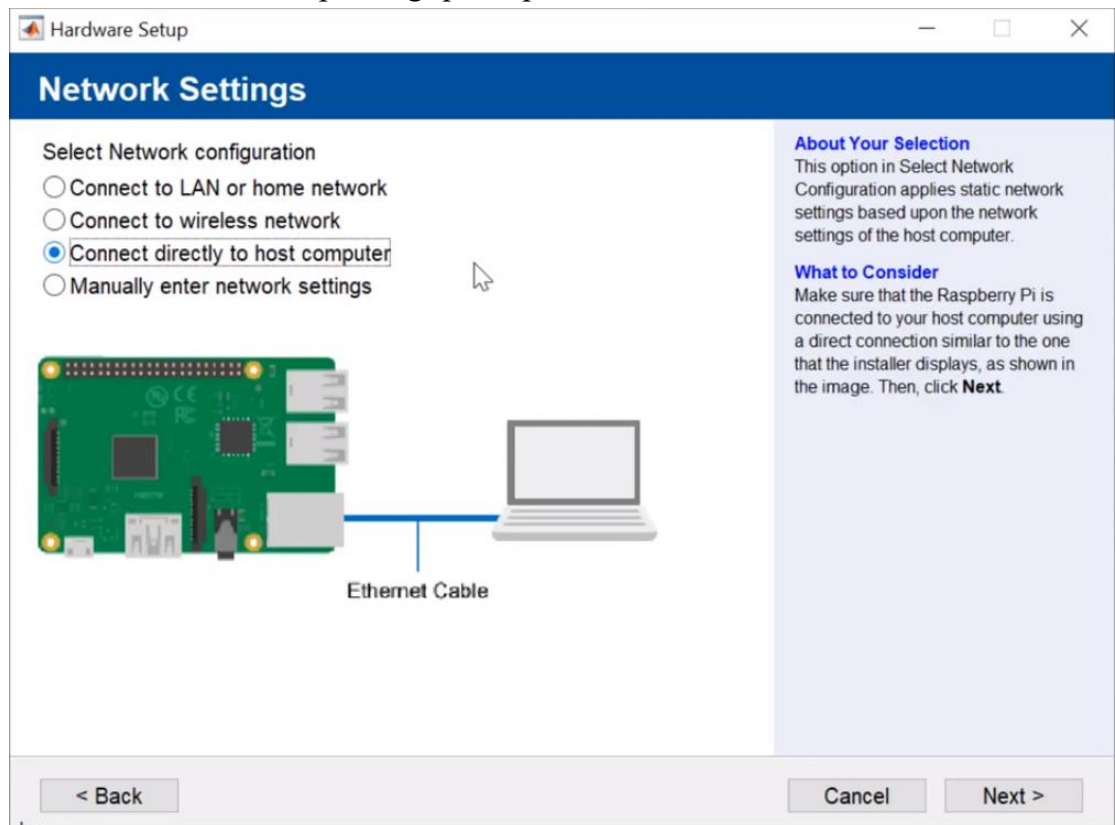
Hình 2. 25: Kết nối thông qua mạng LAN

- Kết nối thông qua WIFI.



Hình 2. 26: Kết nối thông qua WIFI

- Kết nối trực tiếp thông qua cáp ethernet.



Hình 2. 27: Kết nối thông qua cáp ethernet

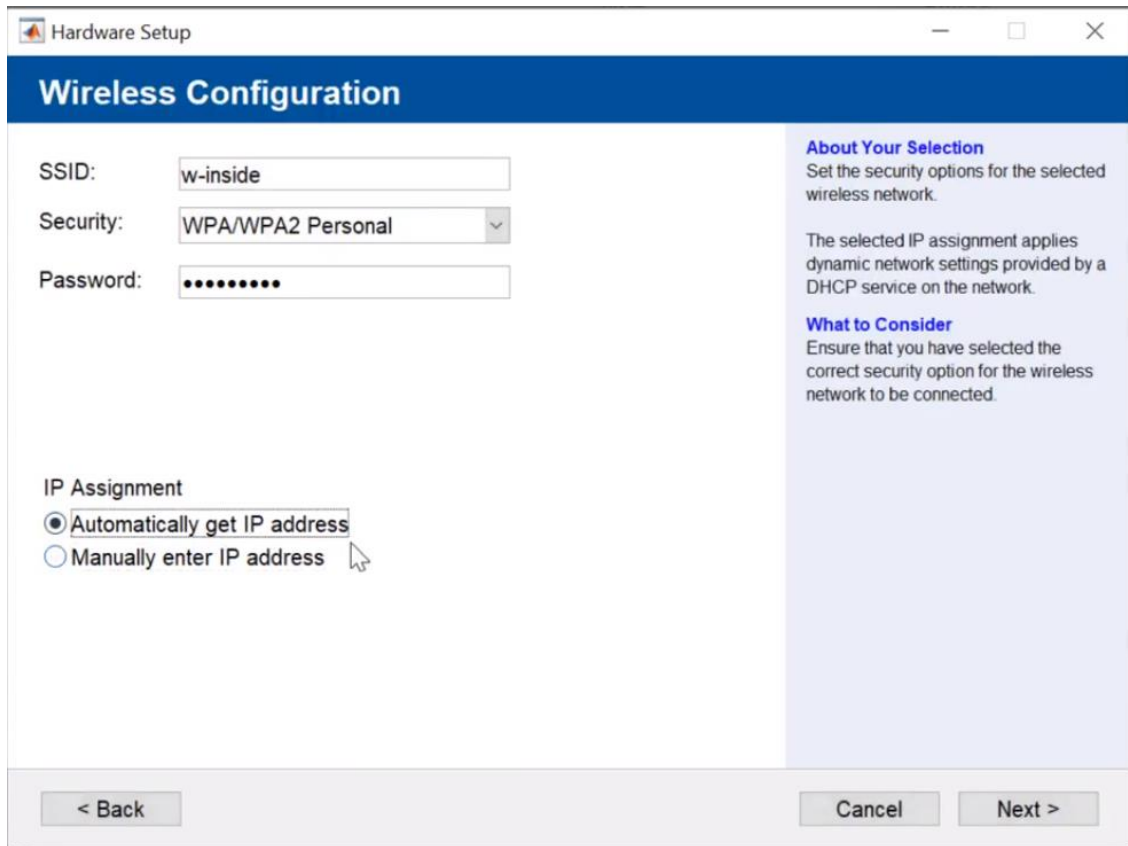
## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- Kết nối kiểu mặc định

Tùy thuộc vào việc Raspberry đã được kết nối tới Wifi hay mạng LAN.

Ở phần này các cách cài đặt là gần như nhau chỉ khác việc thiết lập cấu hình kết nối để cài đặt nên nhóm sẽ trình bày hai cách cài đặt phức tạp hơn.

### Cách cài đặt thông qua WIFI.

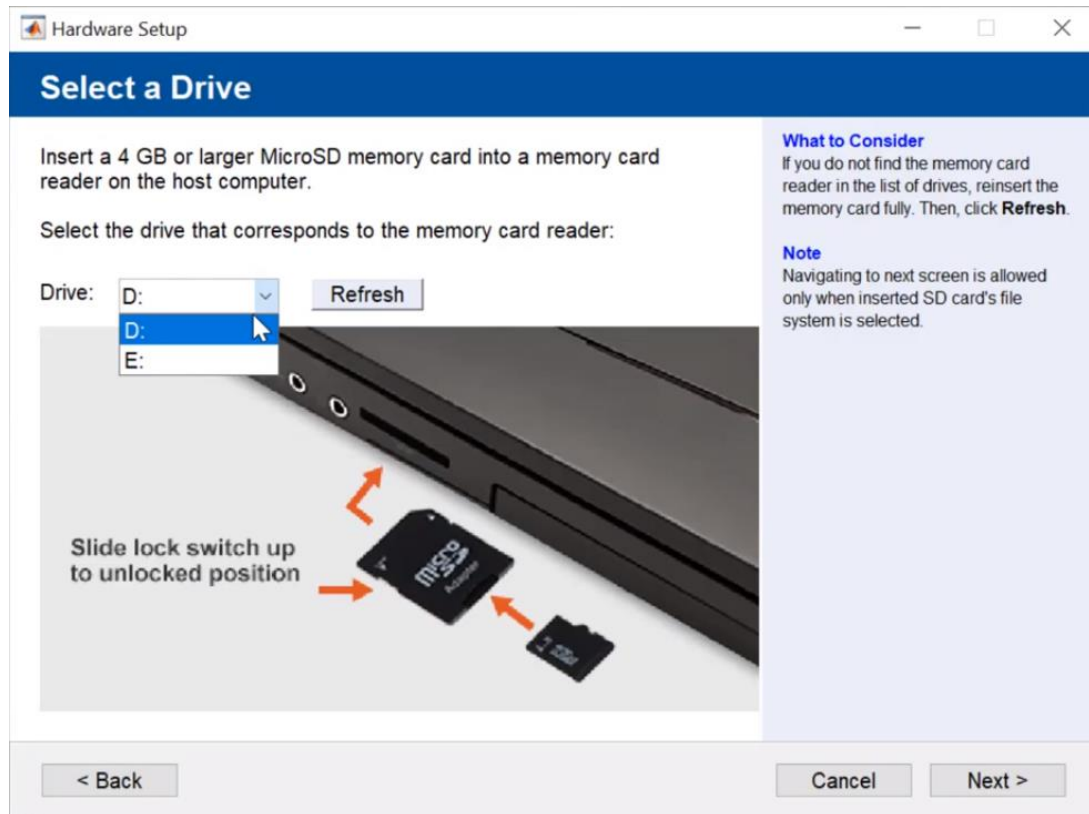


Hình 2. 28: Cấu hình WIFI

Nhập **SSID** là tên wifi muốn truy cập, **password** là mật khẩu của wifi đó.

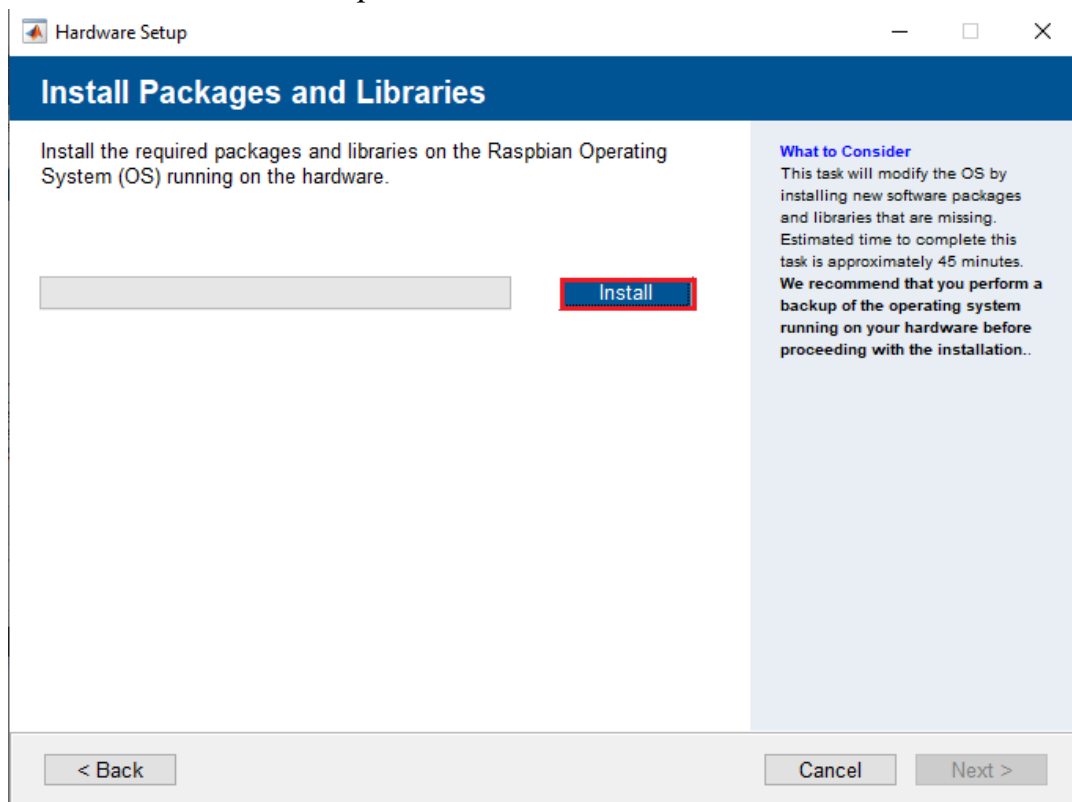
Chọn vào **Automatically get IP address**.

Chọn **Next** để tiếp tục.



Hình 2. 29: Chọn ổ đĩa để lưu trữ

Cắm thẻ nhớ có dung lượng lớn hơn 4GB vào máy tính, chọn **Refresh** để lựa chọn ổ đĩa. Chọn **Next** để tiếp tục.

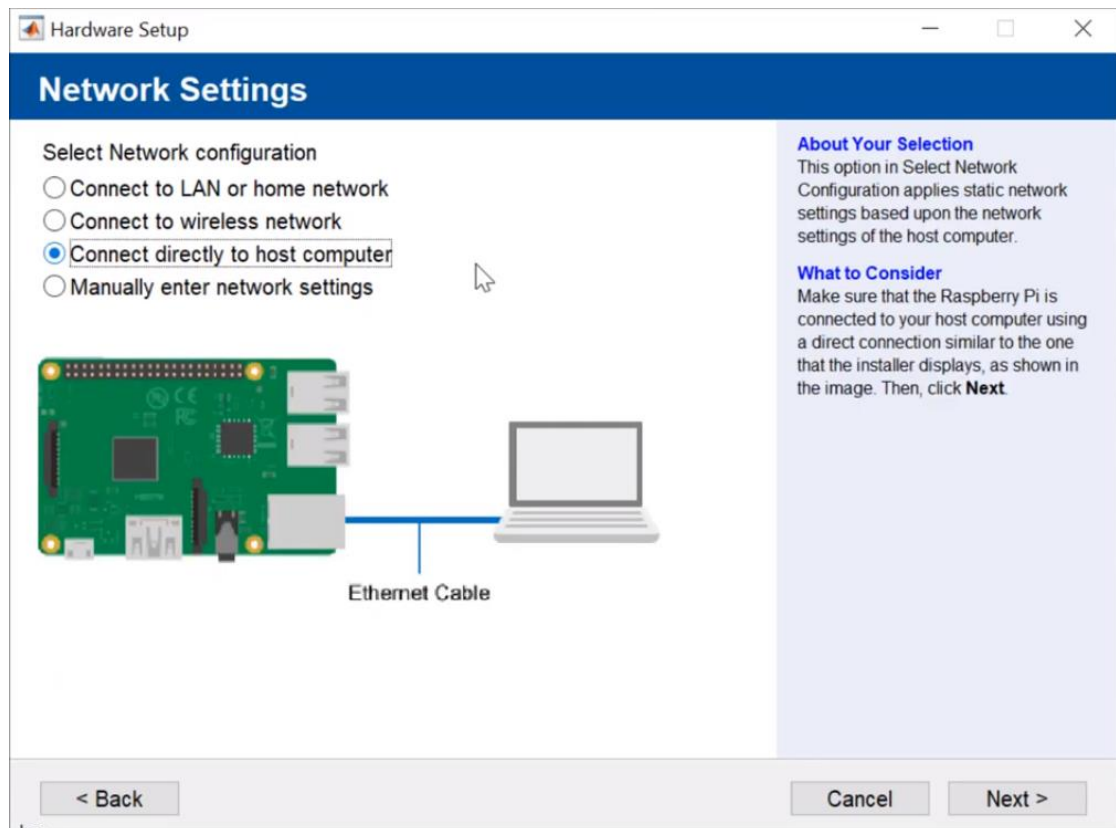


Hình 2. 30: Cài đặt gói hỗ trợ Raspberry



Tiến hành cài đặt cho đến khi hoàn thành, chọn **Reboot later** để hoàn tất.

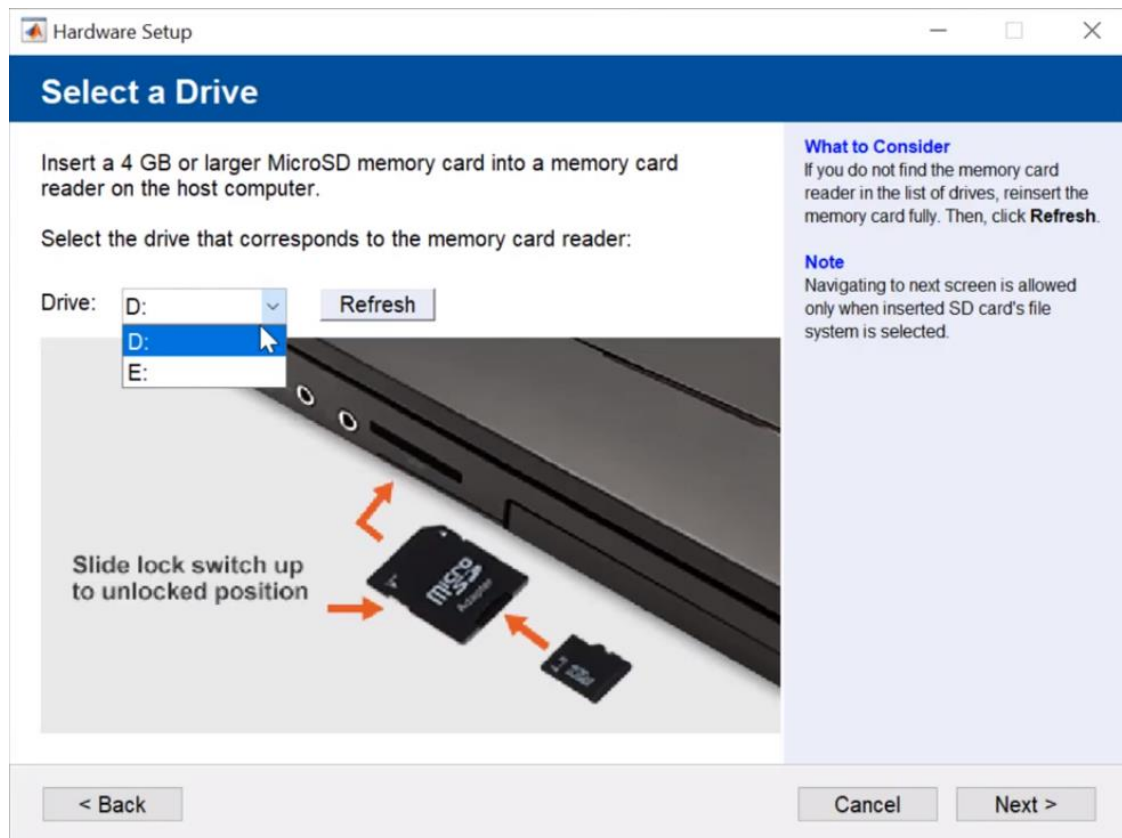
### Cách cài đặt thông qua cáp Ethernet.



Hình 2. 31: Kết nối thông qua cáp ethernet

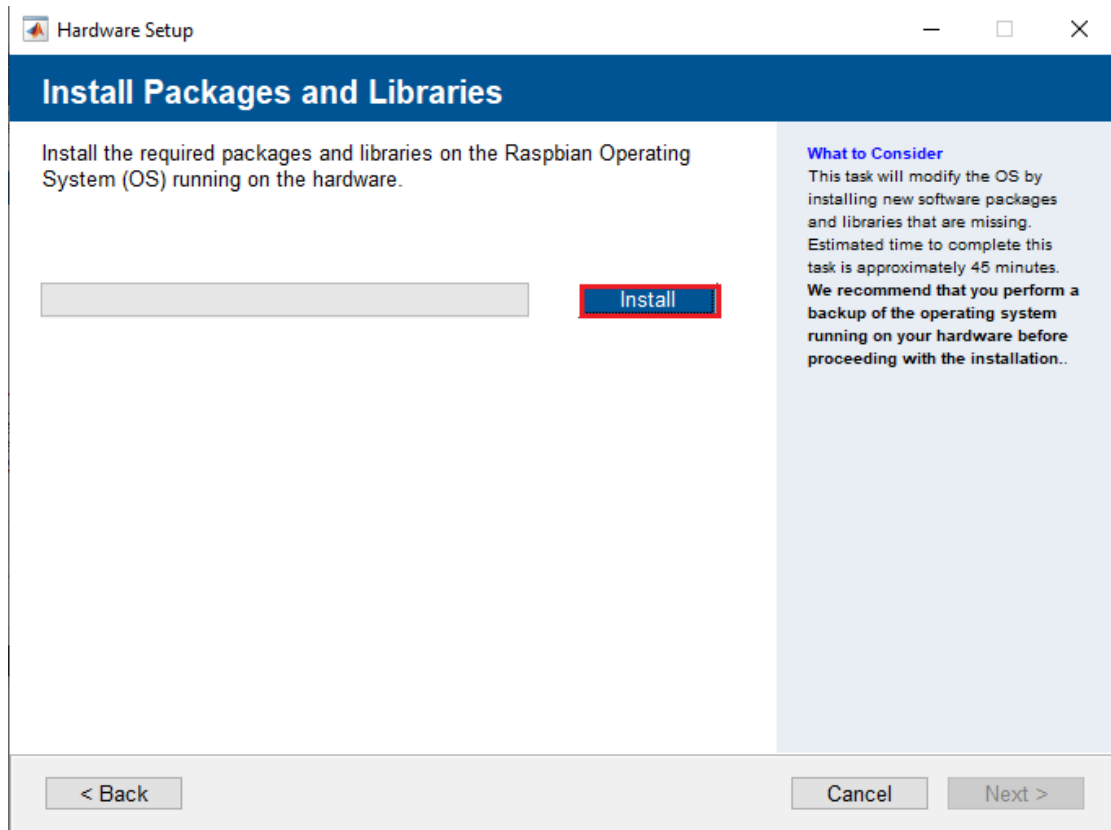
Sử dụng cáp ethernet kết nối giữa laptop và Raspberry Pi. Đảm bảo rằng laptop đã được phép chia sẻ mạng.

Nếu kết nối thành công thì nút **Next** hiện lên, chọn **Next** để tiếp tục.



Hình 2. 32: Chọn ổ đĩa để lưu trữ

Cắm thẻ nhớ có dung lượng lớn hơn 4GB vào máy tính, chọn **Refresh** để lựa chọn ổ đĩa. Chọn **Next** để tiếp tục.



Hình 2. 33: Cài đặt gói hỗ trợ Raspberry

Tiến hành cài đặt cho đến khi hoàn thành, chọn **Reboot later** để hoàn tất việc cài đặt theo cách kết nối thông qua mạng cáp Ethernet.

### ❖ Kiểm tra kết nối Raspberry Pi với Matlab.

Sau khi đã hoàn tất việc cài đặt, tiến hành khởi động phần Matlab để kiểm tra việc kết nối giữa Raspberry Pi với Matlab.

Trên Command Window gõ lệnh:

***“mypi = raspi(ipaddress,username,password)”***.

Với ipaddress là địa chỉ của Raspberry Pi, username và password lần lượt là tên và mật khẩu của Raspberry Pi.

Đây là câu lệnh để kiểm tra kết nối giữa Raspberry với Matlab và các Packages đã được cài đặt để hỗ trợ cho Raspberry hay chưa.

```
Command Window
>> mypi=raspi('192.168.1.8','pi','raspberrypi')

mypi =

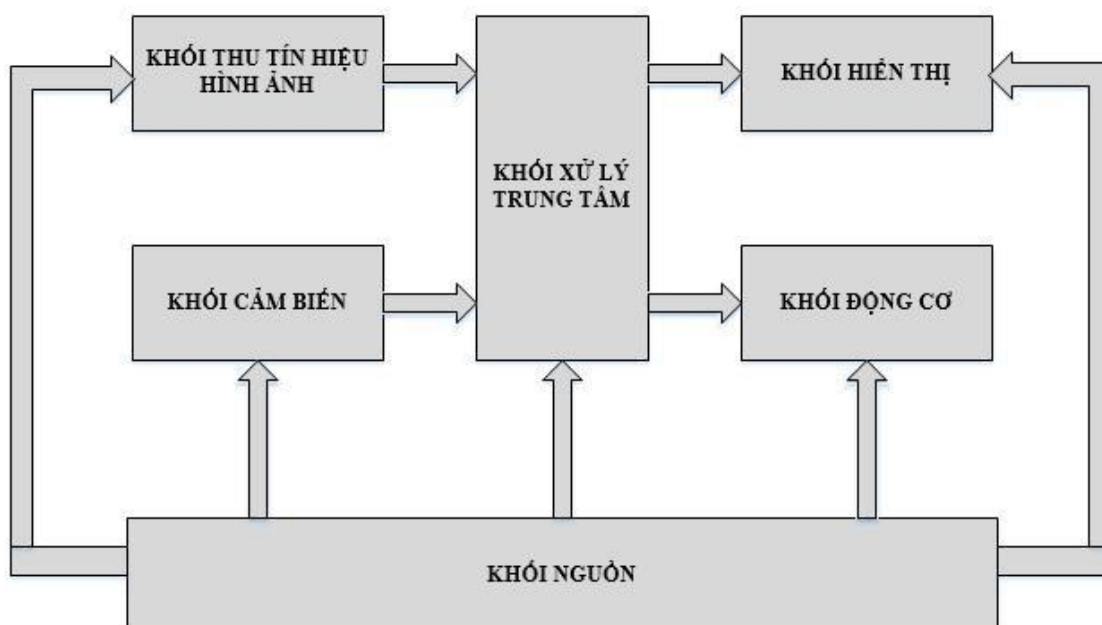
  raspi with properties:
    DeviceAddress: '192.168.1.8'
    Port: 18734
    BoardName: 'Raspberry Pi 3 Model B+'
    AvailableLEDs: {'led0'}
    AvailableDigitalPins: [4,5,6,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27]
    AvailableSPIChannels: {'CE0','CE1'}
    AvailableI2CBuses: {'i2c-1'}
    AvailableWebcams: {'/dev/video10','/dev/video11','/dev/video12'}
    I2CbusSpeed: 100000

  Supported peripherals
```

Hình 2. 34: Kết nối Raspberry và Matlab thành công.

## Chương 3: TÍNH TOÁN VÀ THIẾT KẾ

### 3.1 TỔ CHỨC CÁC KHỐI TRONG HỆ THỐNG



Hình 3. 1: Sơ đồ khối hệ thống

#### ❖ Chức năng từng khối

**Khối xử lý trung tâm :** đảm nhận nhiệm vụ xử lý chính, nhận tín hiệu hình ảnh từ khối thu tín hiệu hình ảnh (camera), nhận tín hiệu từ cảm biến, sau đó xử lý hình ảnh, so sánh đưa ra kết luận về số lượng sản phẩm, hiển thị và lưu trữ kết quả lên màn hình máy tính và điều khiển servo cho phù hợp.

**Khối thu tín hiệu hình ảnh:** có chức năng thu nhận tín hiệu hình ảnh từ thực tế chuyển về tín hiệu điện và gửi dữ liệu cho khối xử lý trung tâm. Ở đây dùng Camera Raspberry Pi làm khối thu tín hiệu hình ảnh.

**Khối hiển thị:** nhận tín hiệu từ khối xử lý trung tâm để hiển thị kết quả lên màn hình máy tính rời.

**Khối điều khiển động cơ:** bao gồm relay, băng chuyền, servo. Nếu Raspberry nhận dữ liệu số viên kẹo nhỏ khác số lượng đặt trước thì Raspberry sẽ tiến hành điều khiển chân tín hiệu của servo quay một góc 45 độ để gạt viên kẹo ra khỏi băng tải. Nếu dữ liệu nhận được bằng đúng số lượng đặt trước thì sẽ không làm gì cả, viên kẹo tiếp tục đi tiếp trên băng tải.

**Khối nguồn:** dùng nguồn 5V để nuôi khối xử lý trung tâm, dùng nguồn 12V để nuôi khối điều khiển ngoại vi.

**Khối cảm biến:** khi cảm biến hồng ngoại phát hiện có sản phẩm thì kích relay dùng băng tải và camera bắt đầu chụp. Sau khi Raspberry Pi thu được hình ảnh và đã xử lý hình ảnh xong thì thả kích relay để băng tải chuyển kẹo đi tiếp.

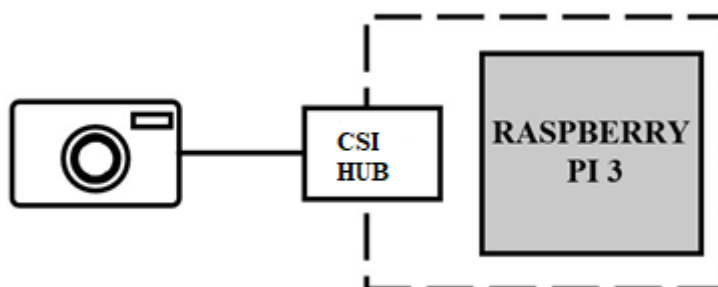
### 3.2 THIẾT KẾ CÁC KHỐI TRONG HỆ THỐNG

#### 3.2.1. Khối thu tín hiệu hình ảnh (camera)

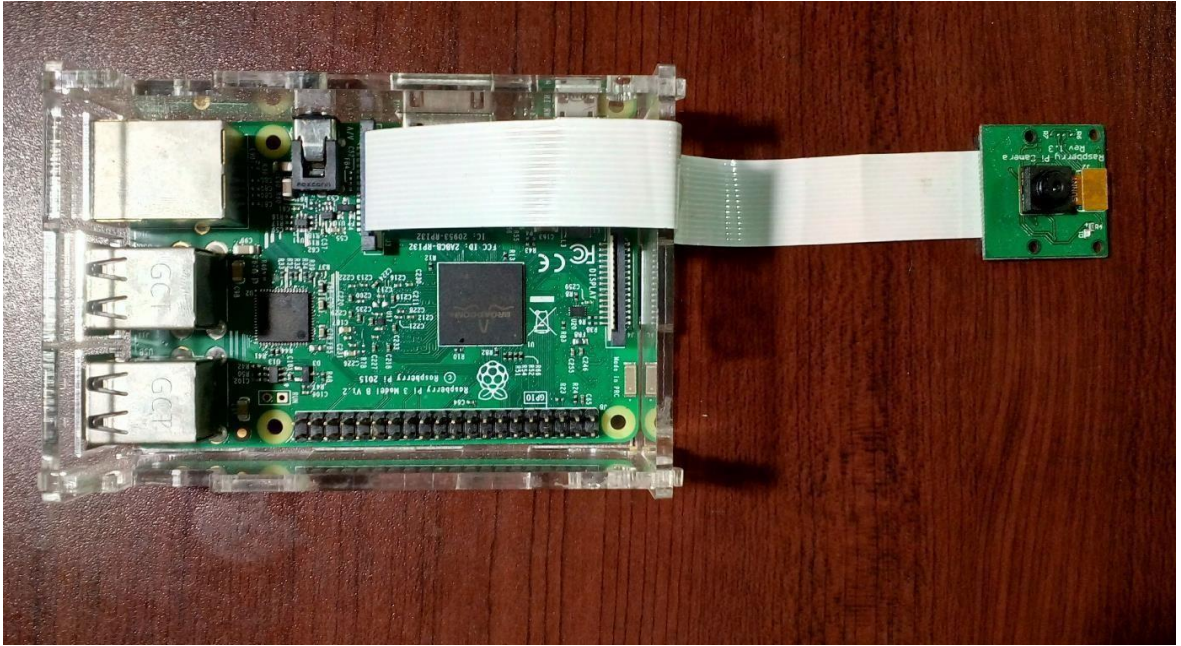
Được xem như là con mắt của mô hình. Có chức năng thu thập tín hiệu hình ảnh từ thực tế rồi gửi dữ liệu cho khối Raspberry Pi.

Ở đây để có chất lượng hình ảnh tốt để xử lý ảnh dễ dàng ta dùng Camera có độ phân giải từ 8 Megapixels, nên chọn Camera Raspberry Pi V2.1 làm khối thu tín hiệu hình ảnh. Vì khối có ảnh hưởng lớn đến chất lượng ảnh cùng với việc nhận dạng và phân loại sản phẩm, chính vì vậy nhóm chọn Camera Pi V2.1 có độ phân giải tốt 8 Megapixels. Megapixel (MP) là đơn vị đo độ phân giải của thiết bị quang. Giá trị MP được tính bằng tích độ rộng với chiều cao số lượng điểm ảnh. Có thể hiểu ngắn gọn megapixel sẽ quy định kích thước bức ảnh chụp. Một MP tương ứng với một triệu điểm ảnh (1.000 x 1.000 pixel). Như vậy 8 MP tương ứng với 8 triệu điểm ảnh. Cùng với kích thước nhỏ gọn phù hợp cho việc thí nghiệm trong mô hình nhỏ, kích thước thực tế: 25 x 23 x 9mm. Camera Pi được kết nối trực tiếp với máy tính nhưng Raspberry thông qua cổng giao tiếp ngoại vi CSI Camera có 15 chân.

Để tăng cường độ sáng cũng như việc tránh ánh sáng nhiễu từ bên ngoài, nhóm đã thêm một buồng chụp có 4 bóng đèn led với công suất 1W và điện áp sử dụng là 3V để tăng cường cho chất lượng ảnh.



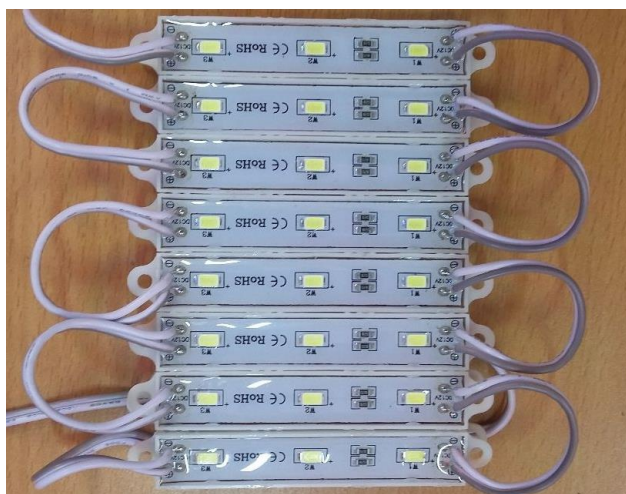
Hình 3. 2: Sơ đồ kết nối khối Camera



*Hình 3. 3: Kết nối Camera thực tế*

Ngoài ra, để camera có thể thu được hình ảnh của kẹo với chất lượng ảnh chân thực về hình dạng phục vụ cho công việc đếm số lượng, nhóm đề tài sử dụng một buồng chụp ảnh với chức năng chính là tạo khung đặt camera, cung cấp nguồn sáng trắng ổn định và phù hợp cho môi trường chụp ảnh, cách ly kẹo với những nguồn sáng gây nhiễu từ môi trường bên ngoài.

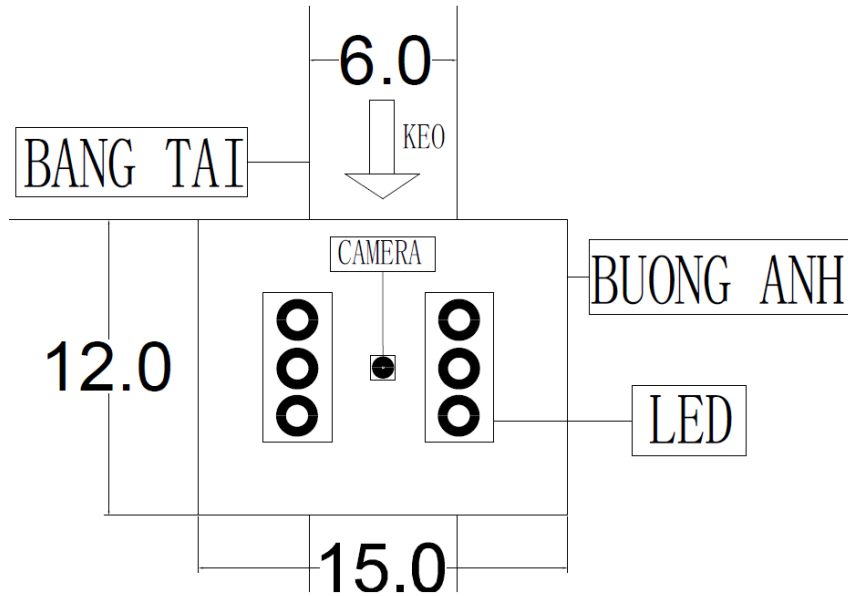
Để ổn định nguồn sáng trắng, nhóm sử dụng thanh led hắt 3 bóng 5050 kích thước 75x12mm sử dụng bóng led loại SMD 5x5mm cho chất lượng ánh sáng tốt, độ hắt độ rọi cao, tuổi thọ bóng hàng vạn giờ. Thanh led làm bằng nhựa cho khả năng chống chịu, độ bền khá, led sử dụng nguồn 12VDC, công suất 0.72W/1 thanh.



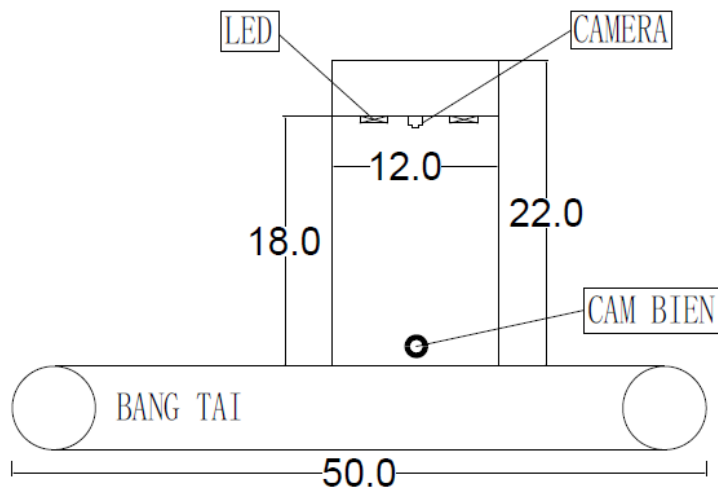
*Hình 3. 4: Led 3 thanh*

### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Về kích thước, buồng chụp ảnh được thiết kế phù hợp với kích thước của băng tải, với các thông số được thể hiện ở hình sau:



Hình 3. 5: Sơ đồ bố trí buồng chụp ảnh (nhìn từ trên xuống)



## MO HÌNH NHIN NGANG

Hình 3. 6: Sơ đồ bố trí buồng chụp ảnh (nhìn từ mặt bên)

Kích thước của buồng chụp ảnh:

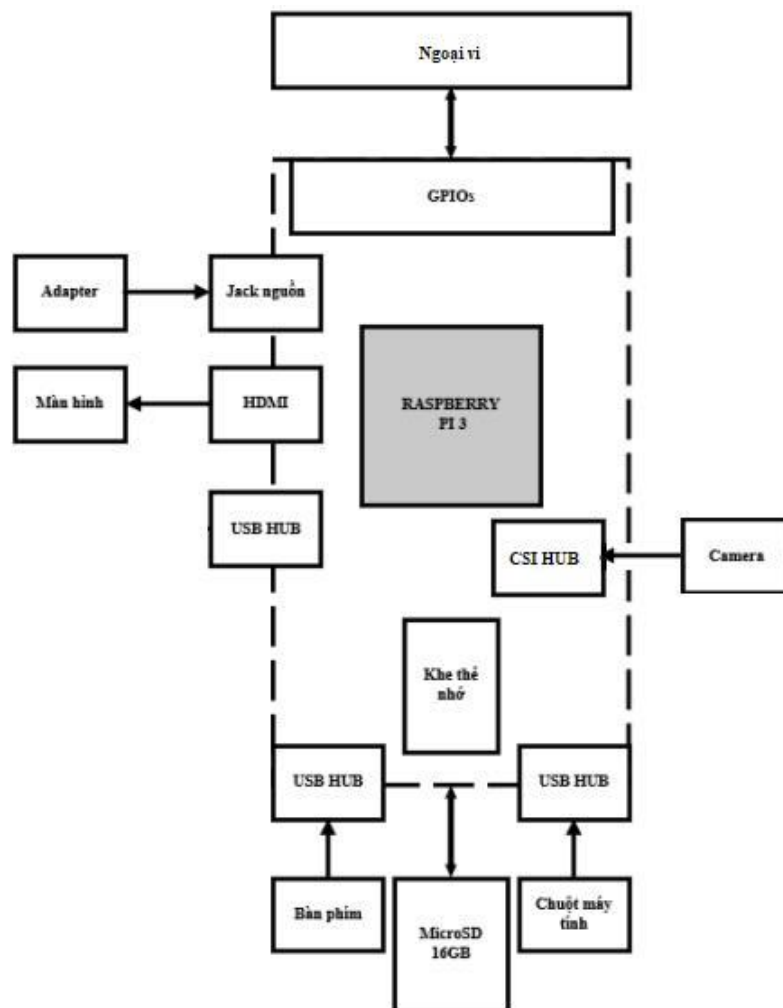
- Chiều dài:  $CD = 15$  cm.
- Chiều rộng:  $CR = 12$  cm.
- Chiều cao:  $CC = 22$  cm.



- Khoảng cách từ camera đến băng tải:  $D \approx 18 \text{ cm}$ .

### 3.2.2 Khối xử lý trung tâm (Raspberry Pi 3B+)

Chức năng của khối xử lý trung tâm là nhận tín hiệu hình ảnh, sau đó xử lý ảnh, để đưa kết quả ra khối hiển thị và ngoại vi xử lý. Vì vấn đề xử lý ảnh cần nhiều tài nguyên và tốc độ xử lý cao, nên ta không sử dụng vi điều khiển thông thường để xử lý. Chính vì vậy, nhóm đề tài đã sử dụng máy tính nhúng Raspberry Pi 3 Model B+ làm khối xử lý trung tâm. Máy tính nhúng Raspberry Pi có CPU với tốc độ xử lý lên tới 1.2GHz. Chính vì vậy việc sử dụng vào mô hình là hợp lý. Máy tính hỗ trợ rất tốt việc giao tiếp các thiết bị ngoại vi và module từ bên ngoài. Raspberry Pi có nhiều cổng giao tiếp nhưng trong phạm vi đề tài chỉ sử dụng một số cổng giao tiếp: CSI để kết nối camera, 6 chân GPIO để giao tiếp với servo, cảm biến hồng ngoại, mạch kích relay, công tắc, 1 cổng HDMI kết nối màn hình giao diện, 1 chuột máy tính, 1 bàn phím.



Hình 3. 7: Sơ đồ các cổng ngoại vi sử dụng

## CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

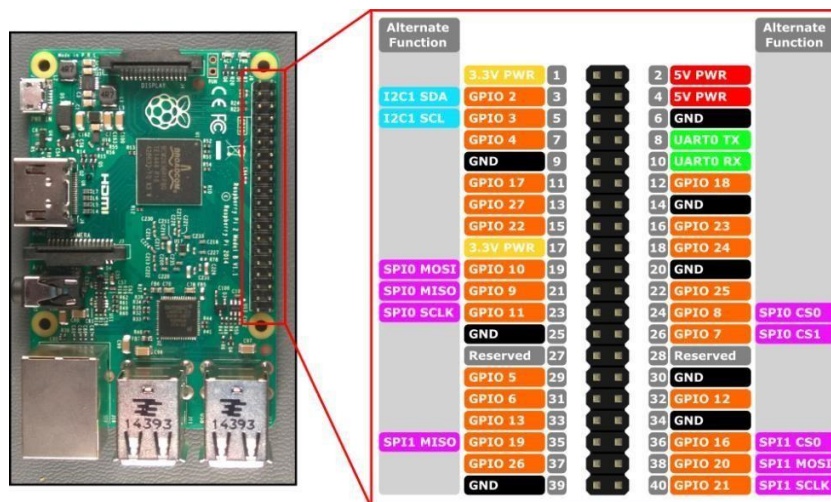
**Cổng USB:** kit Raspberry có 4 cổng USB (Universal Serial Bus) nhưng chỉ sử dụng 2 cổng để kết nối với bàn phím và chuột. Với chuẩn USB 2.0 tốc độ cao, đường truyền đạt tốc độ tối đa đến 480 Mbps. Cáp USB gồm hai sợi nguồn (+5V và dây GND) cùng một cặp gồm hai sợi dây xoắn để mang dữ liệu.

**Cổng HDMI:** dùng một cổng HDMI (High-Definition Multimedia Interface) để kết nối với màn hình LCD để hỗ trợ lập trình, truyền dữ liệu theo chuẩn giao tiếp hình ảnh kỹ thuật số, hỗ trợ âm thanh kỹ thuật số 8 kênh.

**Cổng CSI Camera:** mắt của mô hình này chính là Camera được kết nối thông qua cổng CSI Camera có 15 chân.

**Nguồn:** dòng hoạt động của Raspberry 500-1000mA.

**Chọn thẻ nhớ lưu dữ liệu:** muốn chạy được chương trình trên máy tính nhưng ta cần phải có hệ điều hành được cài sẵn trên thẻ nhớ. Vì hệ điều hành Raspbian chiếm 4GB dung lượng chưa kể các dữ liệu, chương trình, phần mềm liên quan đến, chính vì vậy chọn loại thẻ nhớ tối thiểu 8GB, ở đây nhóm chọn loại thẻ nhớ MicroSD 16GB có tốc độ đọc lên tới 48MB/s vì tốc độ đọc ảnh hưởng tới tốc độ xử lý dữ liệu của các chương trình chính nên 48MB/s là tốc độ hợp lý.



Hình 3. 8: Sơ đồ chân Raspberry Pi 3+

### 3.2.3 Khối cảm biến

Cảm biến: có chức năng phát hiện khi có chức năng phát hiện kẹo đi qua trên băng tải, và kích xung cho module relay để băng tải dừng và camera bắt đầu chụp.

### 3.2.4 Khôi hiển thị

Có chức năng hiển thị kết quả được nhận từ khối xử lý trung tâm. Sử dụng màn hình máy tính để hiển thị kết quả.

### 3.2.5 Khối nguồn

Do hệ thống sử dụng nhiều loại tải với mức điện áp và dòng điện hoạt động khác nhau nên việc sử dụng một bộ nguồn chung cho toàn bộ hệ thống không đảm bảo về độ an toàn và tính ổn định cho hệ thống. Nhóm đề tài thay thế bằng việc sử dụng những bộ nguồn độc lập và được thi công sẵn, với ưu điểm đáp ứng được mức điện áp và dòng điện một cách ổn định, giá thành hợp lý và được bán rộng rãi trên thị trường, tiết kiệm thời gian thi công mạch nguồn, an toàn cho người sử dụng...

Hệ thống bao gồm 3 bộ nguồn: một adapter (5V, 2A) sử dụng cho Raspberry Pi; một bộ nguồn(12V, 2A) dùng cho băng tải và đèn (buồng chụp ảnh); nguồn 220V cung cấp cho màn hình LCD.

Đối với Raspberry Pi 3B+, các thông số sử dụng mức dòng điện sẽ được trình bày ở bảng sau:

Bảng 3.1: Dòng tiêu thụ của hệ thống

Thiết bị	Dòng tiêu thụ	Ghi chú
Raspberry Pi	500-1000mA	
Module Camera	250mA	
Cổng HDMI	50mA	
Bàn phím và chuột	100mA - 1000mA	
Cảm biến hồng ngoại	300mA.	
Module relay 5V	80mA	
Servo MG996G	30mA	
Tổng	2010mA	Chọn nguồn 2A

Từ bảng thống kê ta thấy thiết bị ngoại vi tốn rất nhiều năng lượng( khoảng 2010mA) . Nhóm quyết định mua nguồn 5V 2A để mạch chạy ổn định.

## CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

---

*Thông tin các linh kiện sử dụng cho khối nguồn*

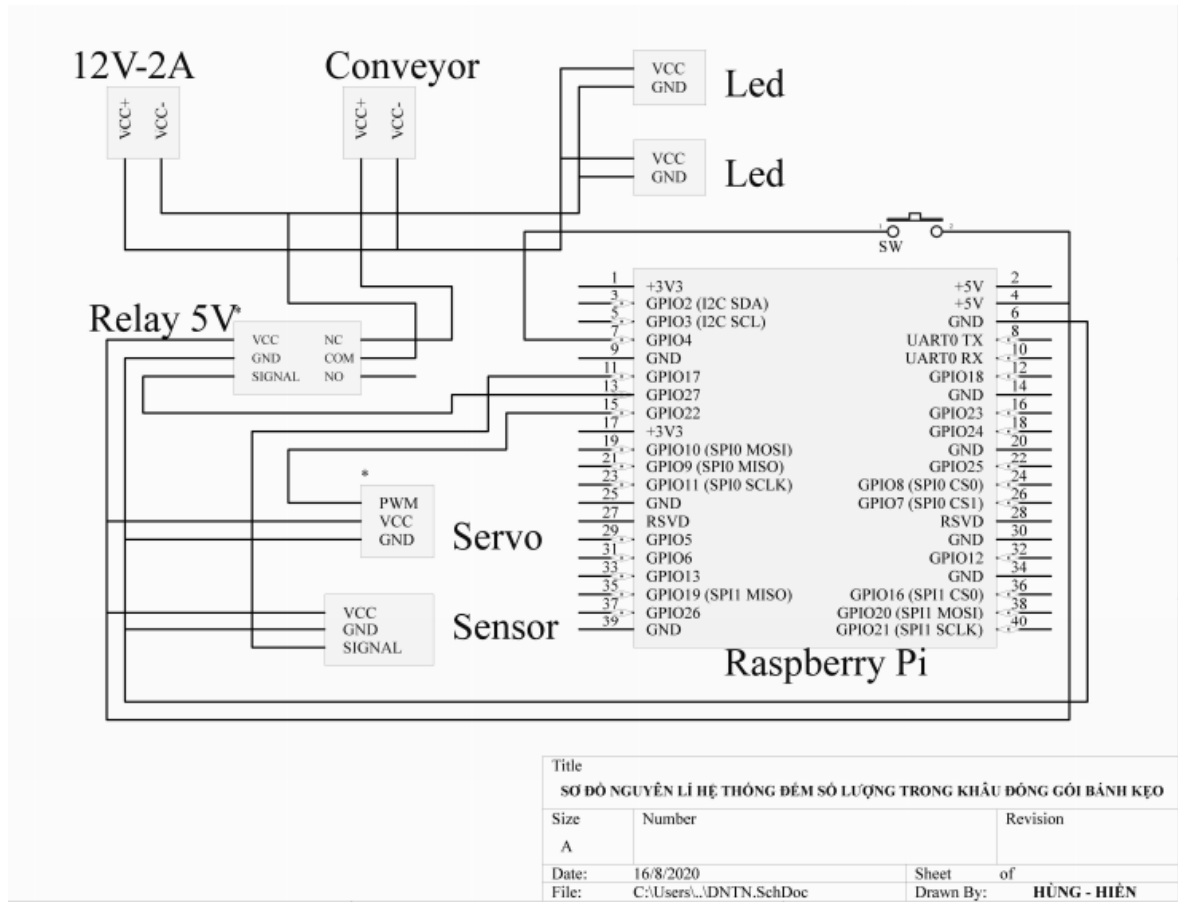


*Hình 3. 9: Nguồn 5V 2A cung cấp cho Raspberry Pi*



*Hình 3. 10: Nguồn 12V 2A cung cấp cho băng tải*

**3.2.6 Sơ đồ nguyên lý toàn mạch**



Hình 3. 11: Sơ đồ nguyên lý toàn mạch

## Chương 4: THI CÔNG HỆ THỐNG

### 4.1 GIỚI THIỆU

Sau khi tính toán và thiết kế, nhóm đề tài tiến hành thi công mô hình đếm số lượng trong khâu đóng gói bánh kẹo. Mô hình bao gồm:

- 1 mô hình băng tải (gồm 1 băng tải, 1 relay, 1 servo MG996): tải kẹo đến các vị trí servo đã định sẵn, nhằm đẩy kẹo xuống rãnh theo đúng số lượng theo yêu cầu.
- 1 buồng chụp ảnh (bao gồm 1 camera, 1 cảm biến hồng ngoại, 2 bóng đèn led siêu sáng) có chức năng dừng băng tải khi cảm biến phát hiện cho kẹo, chụp ảnh và xử lý số lượng kẹo trên băng tải.
- 1 Kit Raspberry Pi 3 Model B+.
- 1 màn hình máy tính.

### 4.2 THI CÔNG HỆ THỐNG

#### 4.2.1 Chuẩn bị phần cứng

Sau khi tính toán và thiết kế, chúng ta chuẩn bị linh kiện tiến hành lắp ráp.

*Bảng 4. 1: Danh sách các linh kiện.*

STT	Tên linh kiện	Giá trị	Dạng vỏ	Chú thích
1	Băng tải		Nhôm và nhựa	Bao gồm: đế, ròng rọc, dây băng tải
2	Kit Raspberry	Điện áp hoạt động: 5V Dòng: 1A	Kit Raspberry	Có quạt tản nhiệt
3	Camera Pi	Điện áp hoạt động: 5V Dòng: 250mA	Camera Pi	
4	Servo MG996R	Điện áp hoạt động: 5V Dòng: 900mA	Nhựa cứng	

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

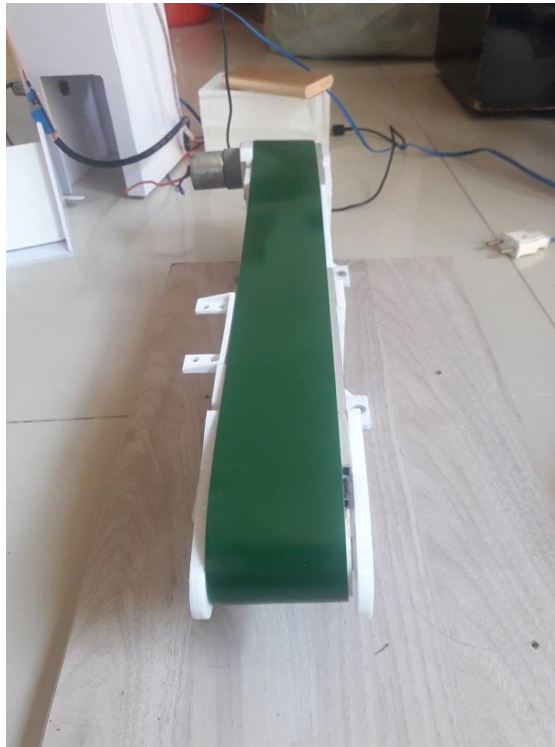
5	Cảm biến	Điện áp hoạt động: 5V Dòng: 300mA	Nhựa	
6	Relay	Điện áp hoạt động: 5V Dòng cho phép: 10A Dòng kích: 5mA	Nhựa	
7	Led siêu sáng	Điện áp hoạt động: 5V Dòng: 15mA	Led	
8	Bo mạch		Nhựa cứng	
9	Nguồn tổ ong	Điện áp vào: 220 VAC Điện áp ra: 12VDC Dòng ra: 2A	Thép	
10	Adapter	Điện áp vào: 220 VAC Điện áp ra: 5VDC Dòng ra: 2A	Nhựa cứng	

### 4.2.2 Lắp ráp và kiểm tra

Là phân kết nối giữa các linh kiện và các module, tạo thành một khối thống nhất với nhau.

#### ❖ Phần băng tải

Lắp băng tải và động cơ.



*Hình 4. 1: Băng tải và động cơ DC*

### ❖ Phần máng trượt

Máng trượt chia làm hai phần:

- Phần thứ nhất là máng khi đếm số lượng sản phẩm không theo yêu cầu được servo gạt ra ngoài.
- Phần thứ hai là máng đưa sản phẩm sau khi được đếm số lượng đúng.

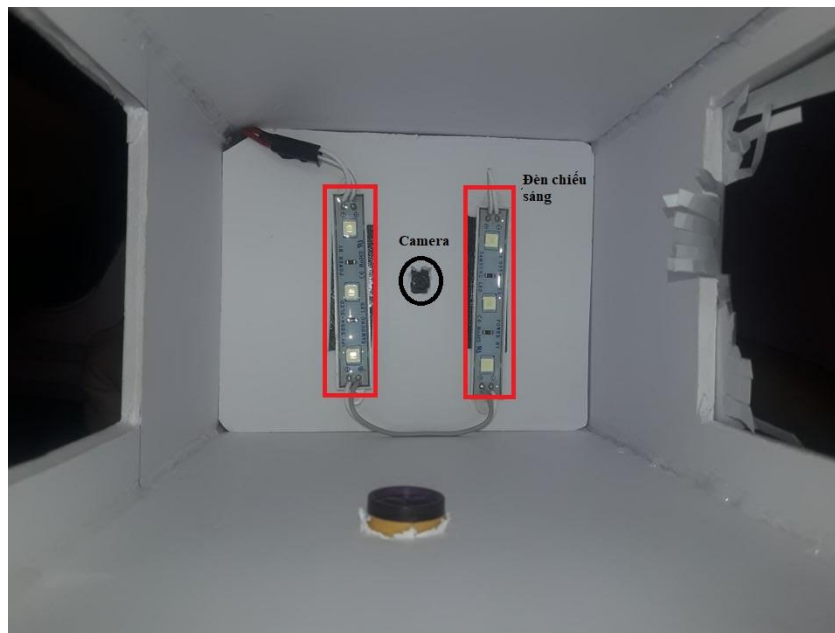


*Hình 4. 2: Máng đưa sản phẩm sau khi phân loại*

Chúng được sắp xếp và gắn kết với băng tải, tiến hành cố định máng trượt.



### ❖ Phần hộp camera

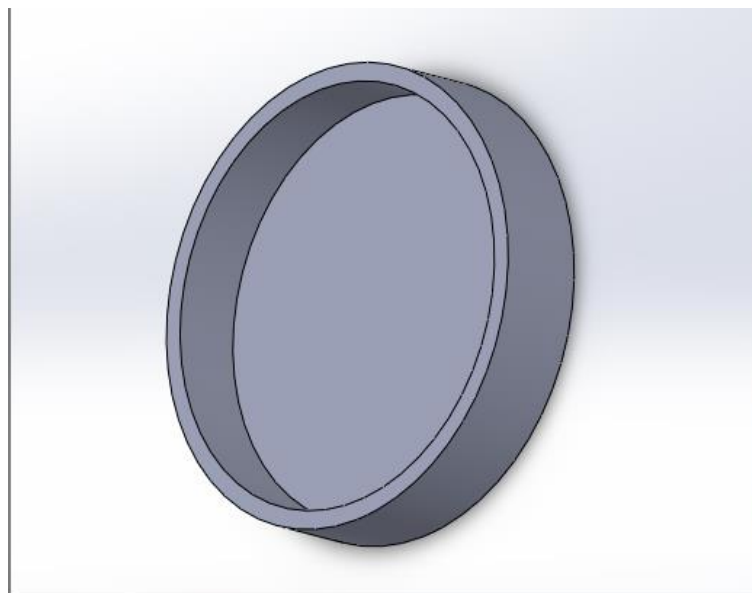


Hình 4. 3: Buồng chụp ảnh nhìn từ phía dưới lên trên

Led trên cùng của buồng chụp ảnh để đảm bảo cung cấp ổn định nguồn sáng trắng cho việc chụp ảnh.

### ❖ Phần máng đựng kẹo

Tiến hành vẽ máng đựng kẹo trên Solid work rồi đi in 3D sản phẩm.



Hình 4. 4: Khuôn kẹo trên solid work



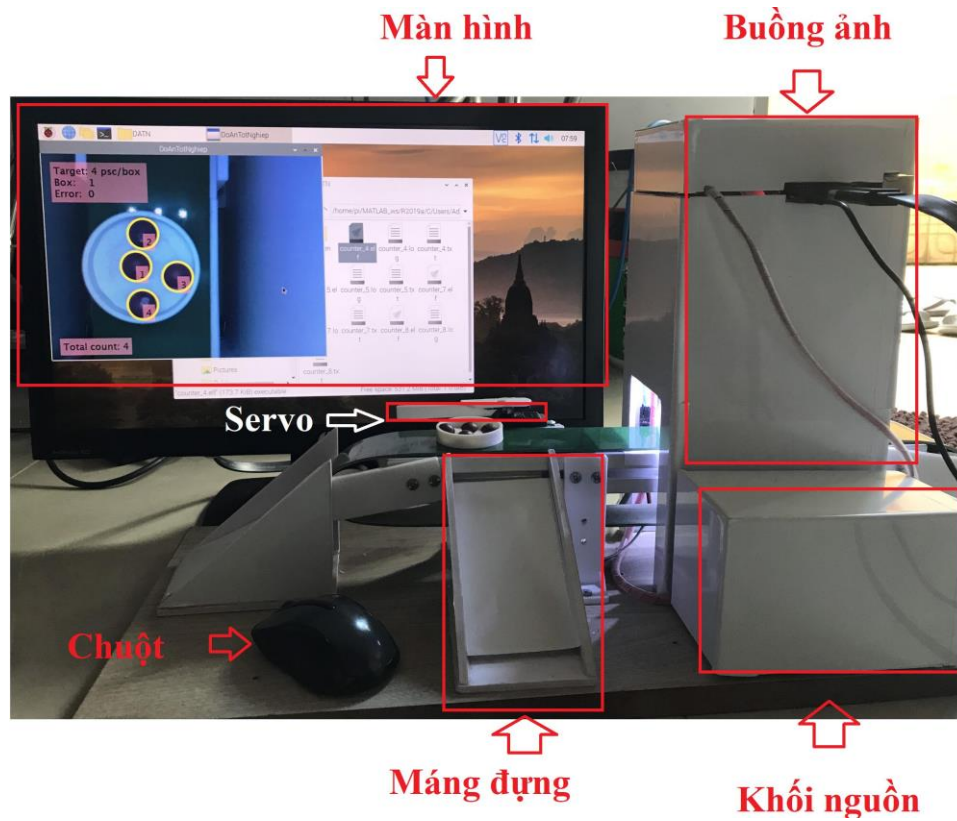
*Hình 4. 5: Khuôn kẹo thực tế*

### 4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH

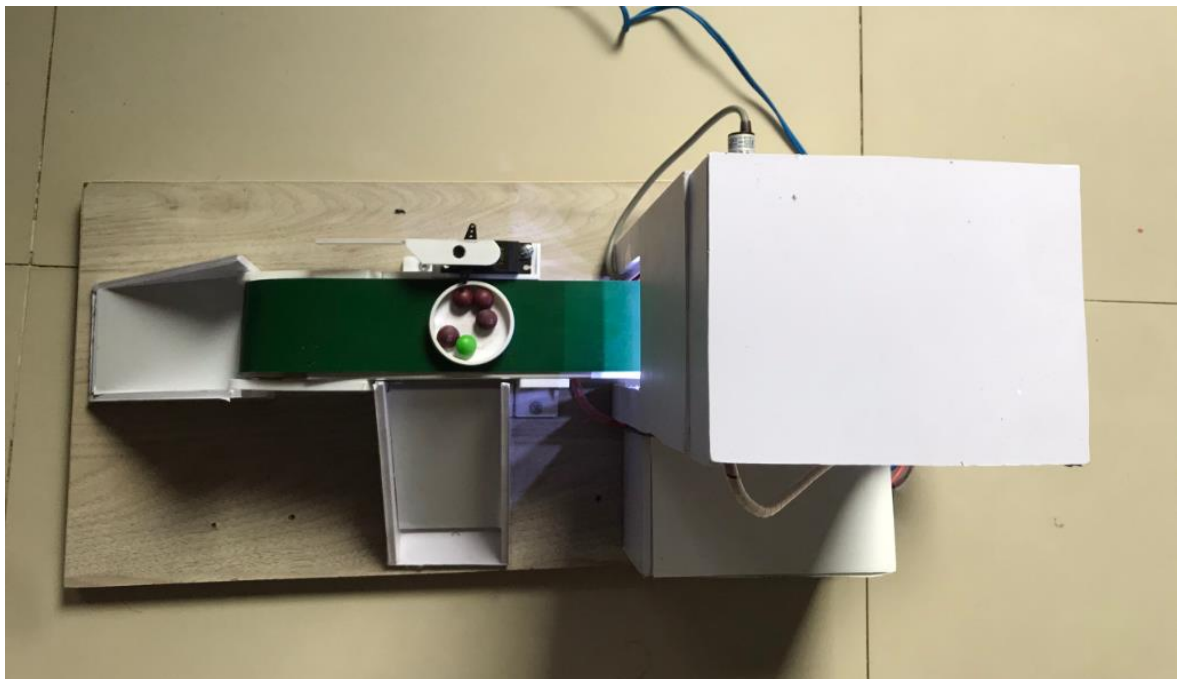
Sau khi thực hiện công đoạn lắp ráp và kiểm tra nhóm tiến hành bố trí các thành phần và hoàn thiện hệ thống để dễ dàng vận chuyển, vận hành, đảm bảo an toàn khi sử dụng.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### 4.3.1 Thi công mô hình



Hình 4. 6: Mô hình thực tế nhìn từ phía trước



Hình 4. 7: Mô hình thực tế nhìn từ trên xuống

## 4.4 LẬP TRÌNH HỆ THỐNG

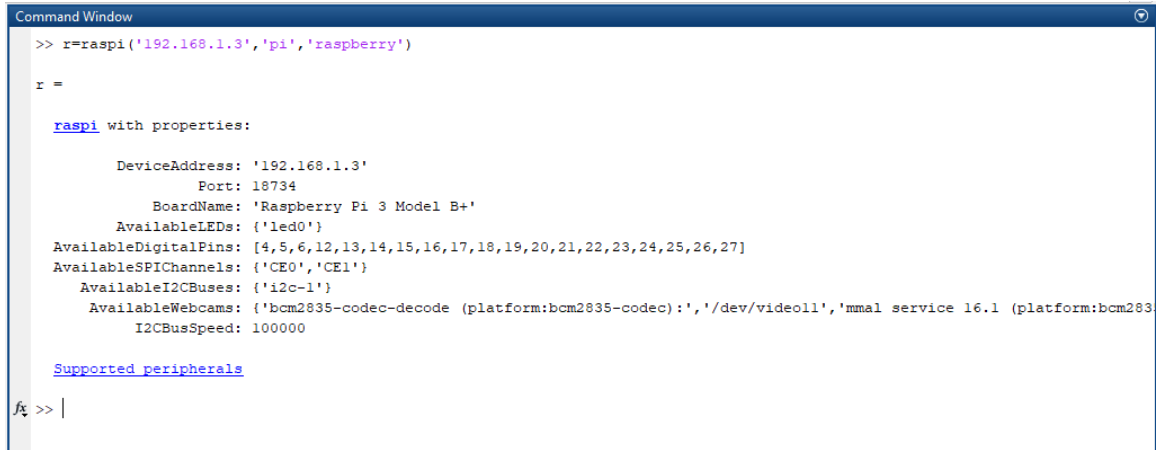
### 4.4.1 Nhúng chương trình xuống Raspberry Pi

- ❖ Biên dịch code Matlab lên Raspberry Pi để chạy độc lập

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Sau khi kết nối thành công Raspberry Pi với Matlab cũng như hoàn thành việc lập trình để tiến hành biên dịch và tổng hợp code lên Raspberry Pi cần làm theo những bước sau.

### Bước 1: Kết nối Raspberry với Matlab



```
Command Window
>> r=raspi('192.168.1.3','pi','raspberrypi')

r =

  raspi with properties:

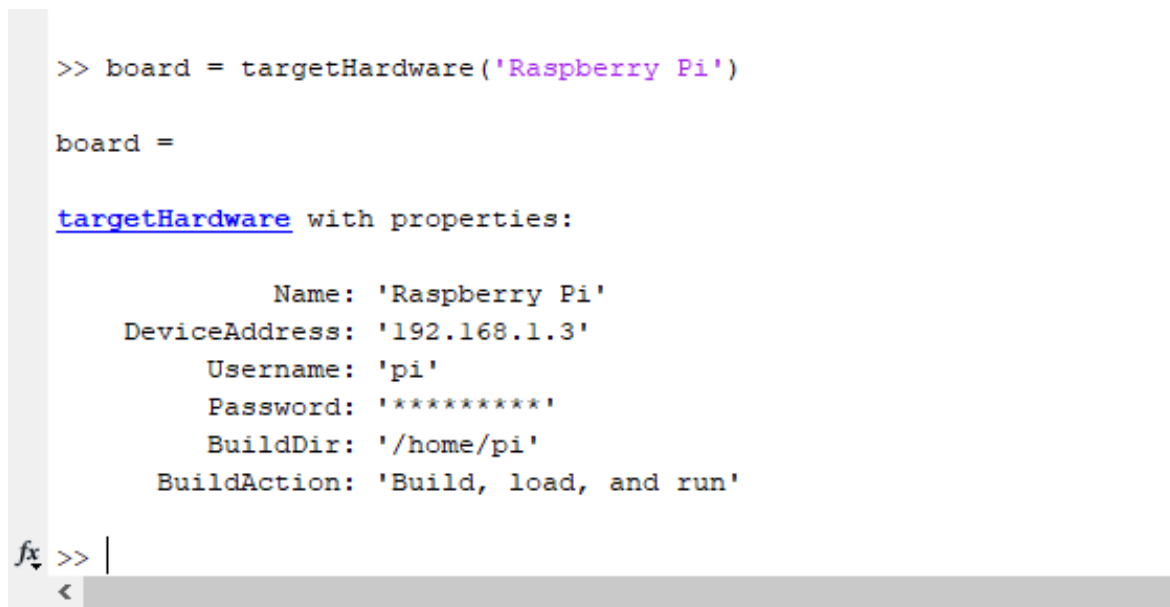
    DeviceAddress: '192.168.1.3'
      Port: 18734
    BoardName: 'Raspberry Pi 3 Model B+'
    AvailableLEDs: {'led0'}
AvailableDigitalPins: [4,5,6,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27]
AvailableSPIChannels: {'CE0','CE1'}
AvailableI2CBuses: {'i2c-1'}
AvailableWebcams: {'bcm2835-codec-decode (platform:bcm2835-codec):','/dev/video11','mmal service 16.1 (platform:bcm2835-codec)'}
      I2CBusSpeed: 100000

  Supported peripherals

fx >> |
```

Hình 4. 8: Kết nối Raspberry và Matlab thành công.

### Bước 2: Trên Command Window gõ lệnh: “**board=targetHardware('RaspberryPi')**”



```
>> board = targetHardware('Raspberry Pi')

board =

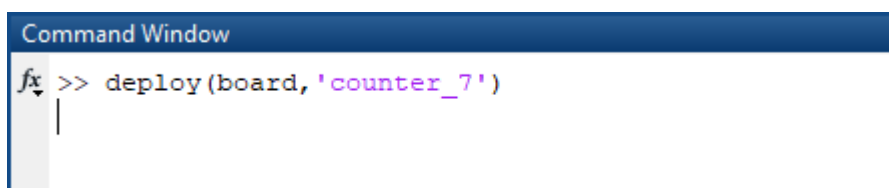
  targetHardware with properties:

    Name: 'Raspberry Pi'
DeviceAddress: '192.168.1.3'
  Username: 'pi'
  Password: '*****'
  BuildDir: '/home/pi'
BuildAction: 'Build, load, and run'

fx >> |
```

Hình 4. 9: Kết nối tới board Raspberry Pi

Bước 3: Trên Command Window gõ lệnh : “ **deploy(board,'FunctionName')**” với FunctionName là tên chương trình Matlab.



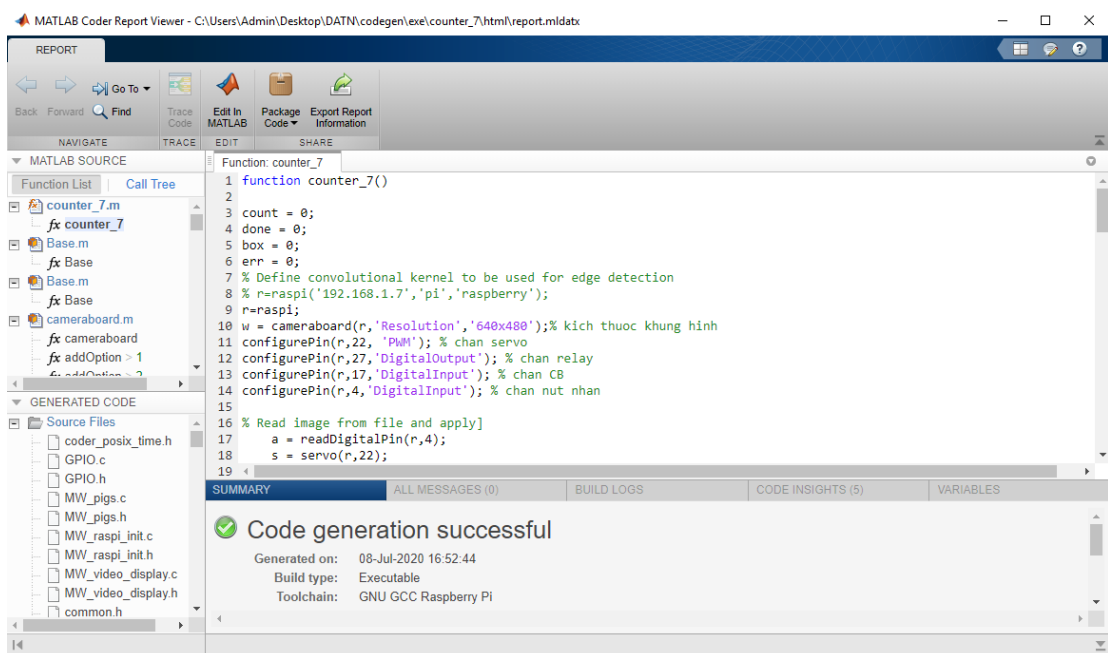
```
Command Window

fx >> deploy(board, 'counter_7')
|
```

Hình 4. 10: Biên dịch chương trình có tên counter\_7 lên Raspberry Pi

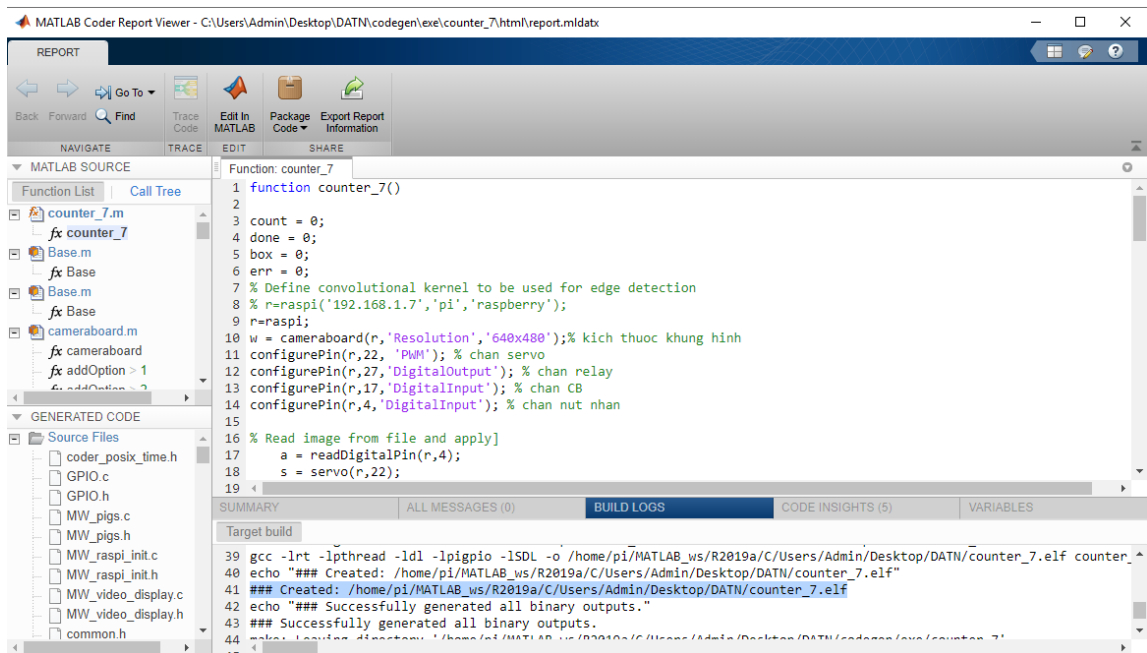
## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Nếu biên dịch thành công thì sẽ có thông báo như hình sau:



Hình 4. 11: Biên dịch thành công

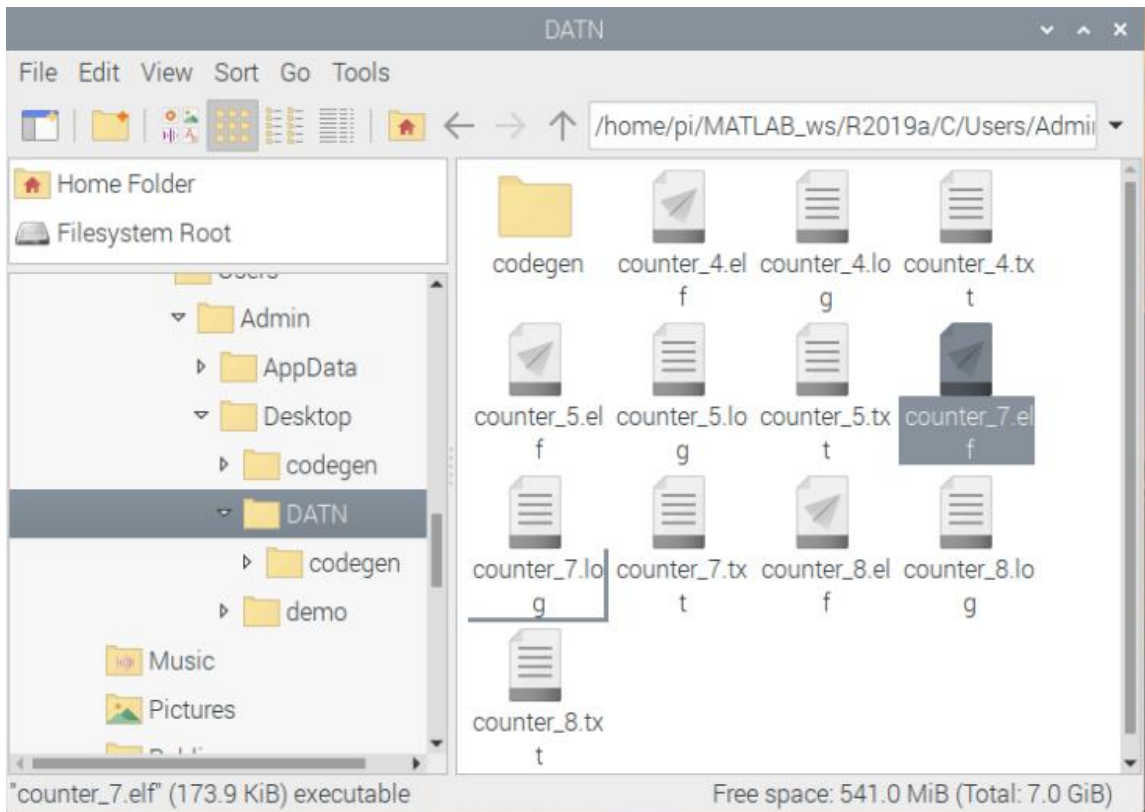
Tệp được tạo ra lưu trong đường dẫn ở thẻ BUILD LOGS.



Hình 4. 12: Đường dẫn kết quả

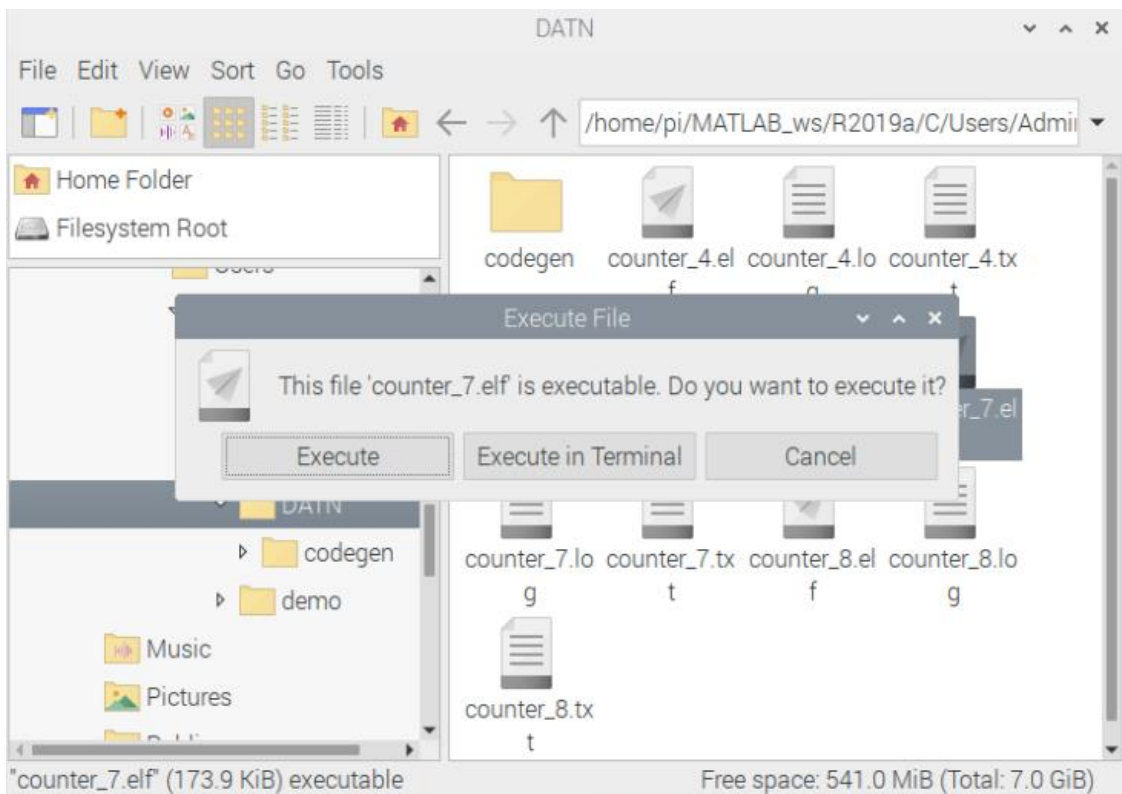
## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Truy cập vào đường dẫn trên vào Raspberry Pi để kiểm tra kết quả:

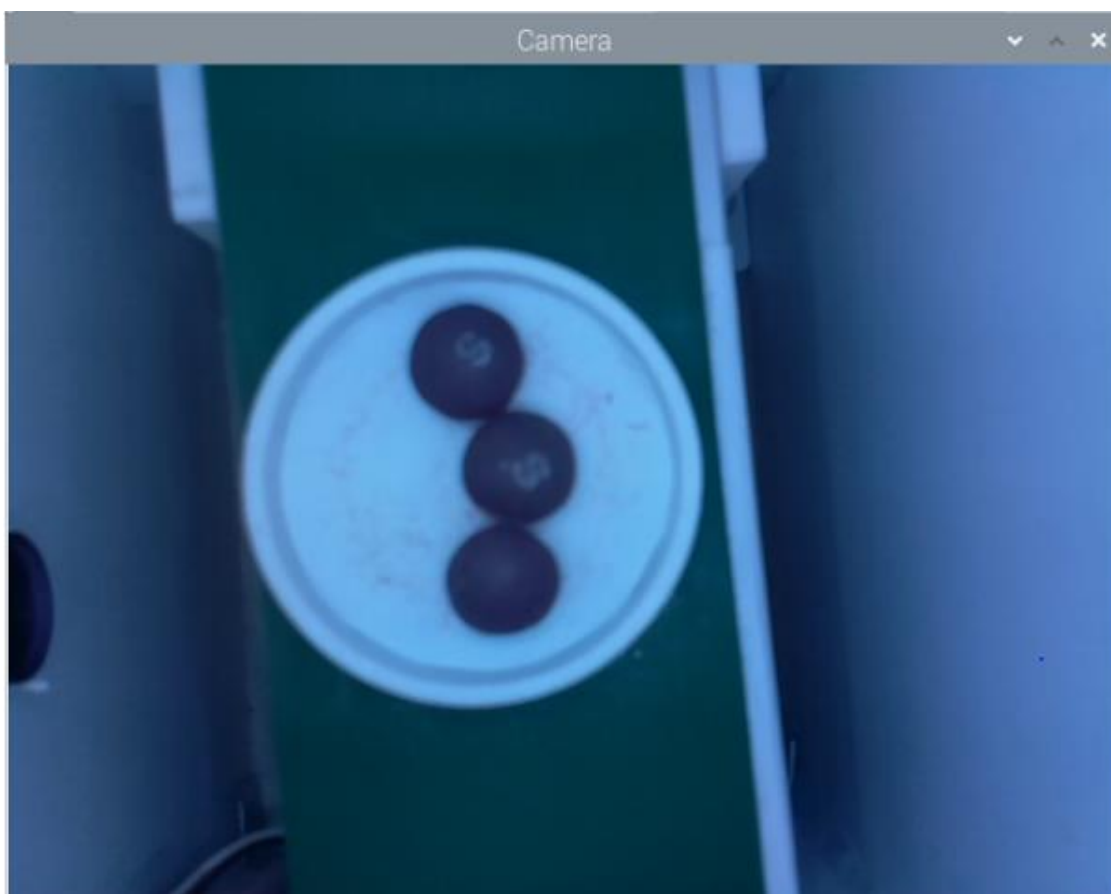


Hình 4. 13: Tập được tạo với tên counter\_7.elf trên đường dẫn thu được

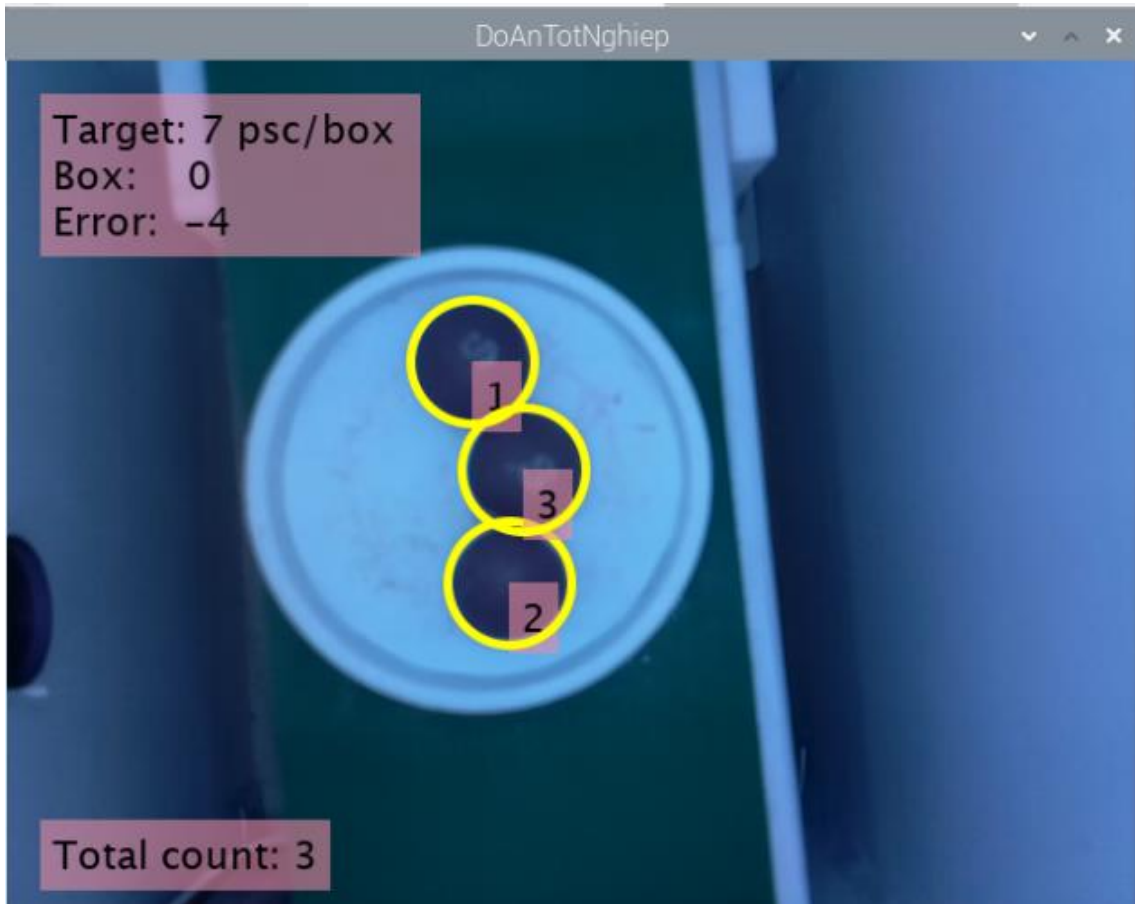
Chạy thử chương trình trên tập vừa tạo để kiểm tra kết quả. Nhấp chuột chọn **Execute**.



*Hình 4. 14: Chạy chương trình vừa được tạo*

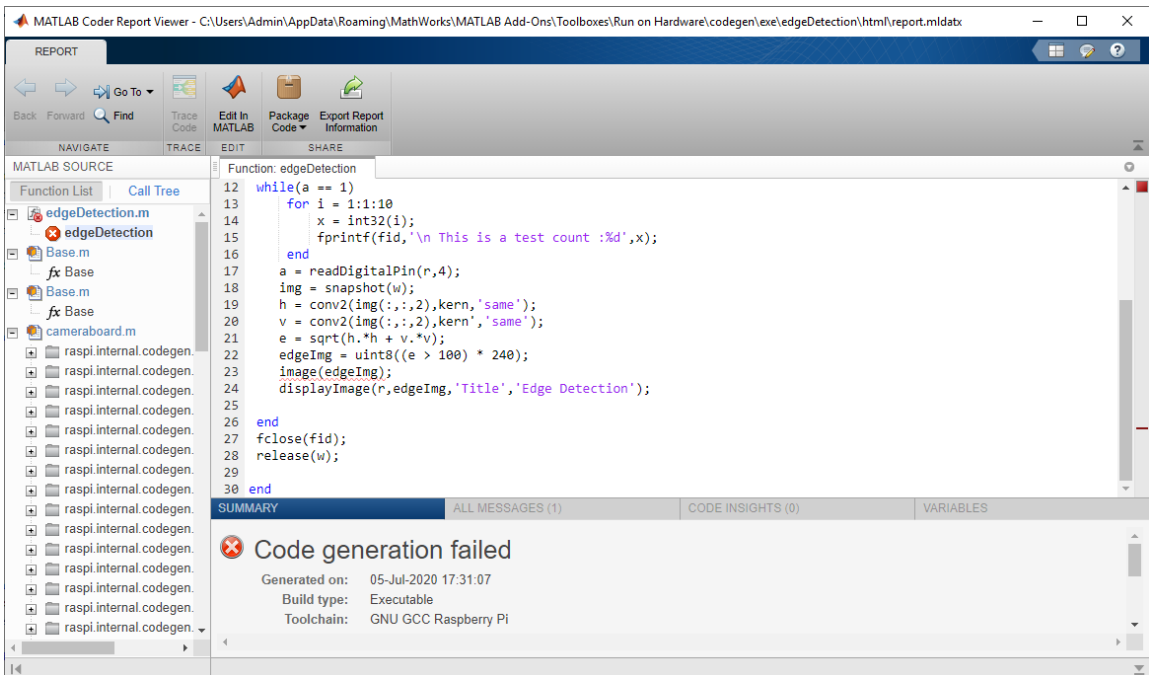


*Hình 4. 15: Kết quả thu được*



Hình 4. 16: Kết quả thu được

Trường hợp biên dịch lỗi sẽ có thông báo như sau:

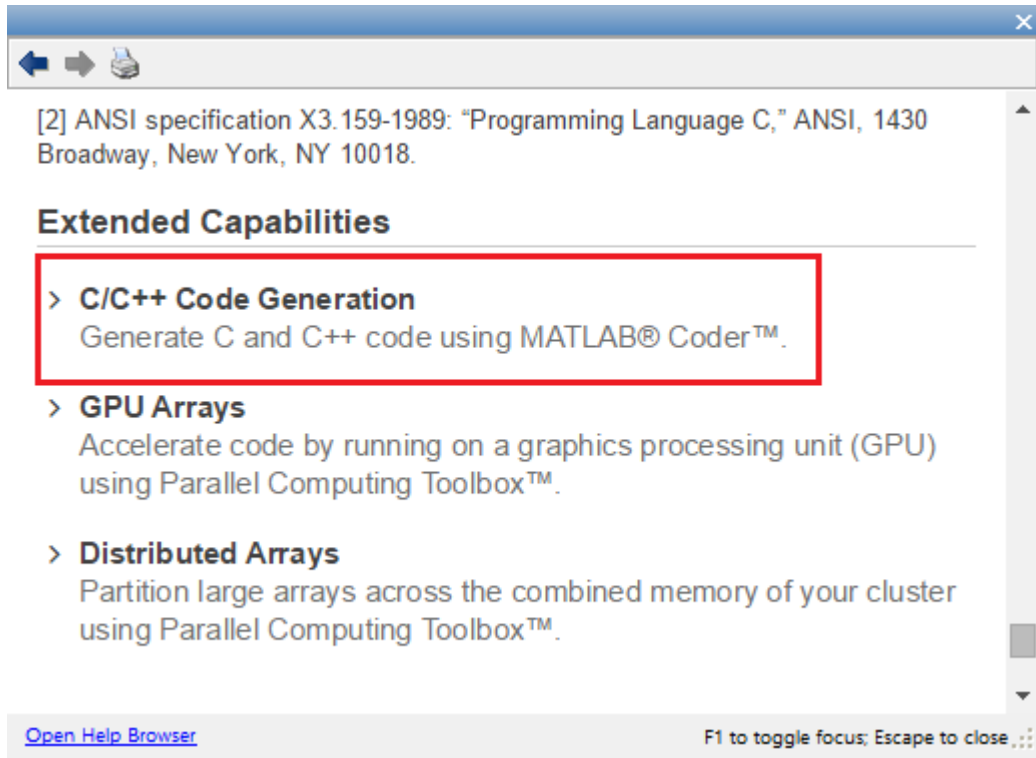


Hình 4. 17: Thông báo biên dịch không thành công



## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Đa phần lỗi biên dịch không thành công do các hàm sử dụng không phù hợp với việc hỗ trợ cho việc biên dịch lên Raspberry Pi. Cách kiểm tra các hàm sử dụng có phù hợp hay không: Bôi đen vào hàm cần kiểm tra và nhấn F1 trên bàn phím, sau đó kiểm tra trong cửa sổ hỗ trợ nếu có thông báo sau thì hàm đó được hỗ trợ cho việc biên dịch.



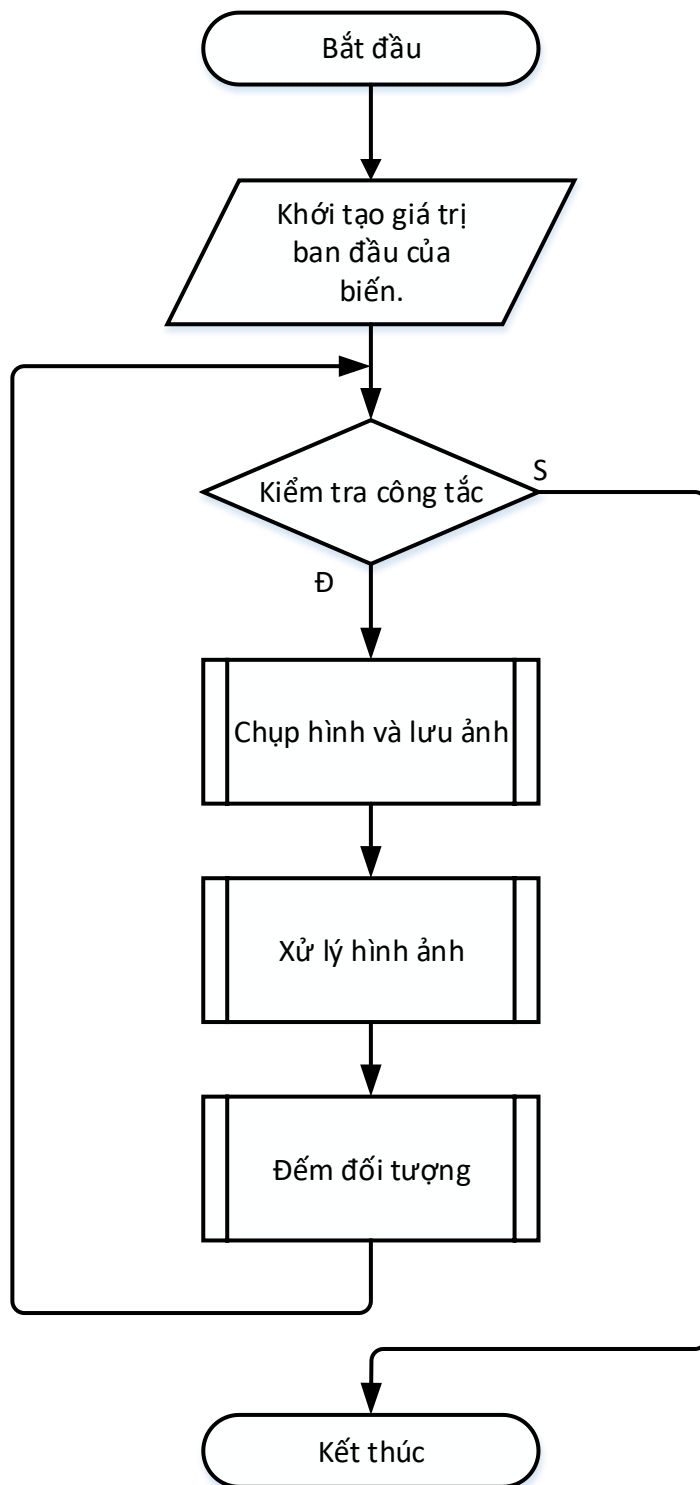
Hình 4. 18: Hàm được hỗ trợ cho việc biên dịch

Nếu không có thông báo đó thì chúng ta cần phải thay đổi các hàm khác để có thể biên dịch thành công.

### 4.4.2 Lưu đồ giải thuật

Từ yêu cầu đã xác định của đề tài nhóm xây dựng lưu đồ giải thuật chương trình chính như sau:

- Lưu đồ chương trình chính



Hình 4. 19: Lưu đồ hoạt động của hệ thống

### ➤ Giải thích lưu đồ

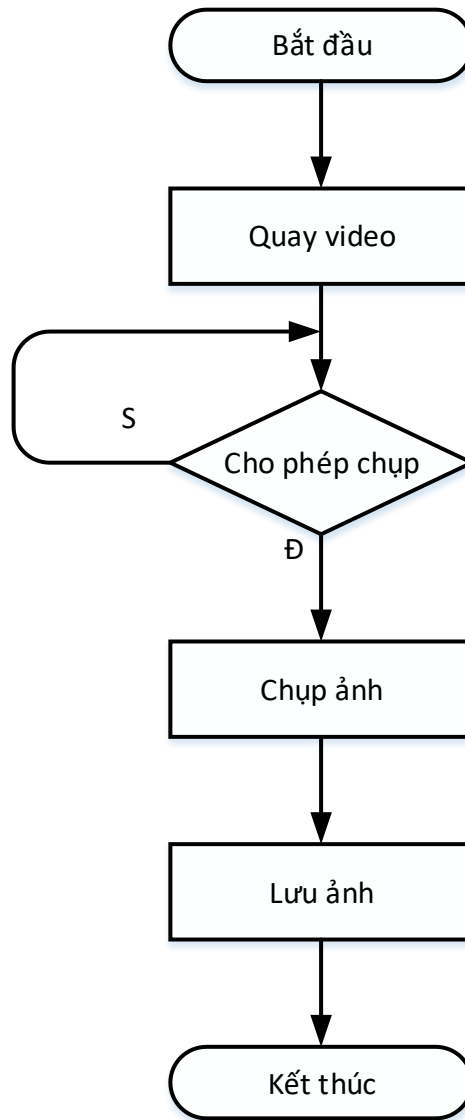
Đầu tiên, thực hiện khai báo thư viện, khởi tạo các biến, khai báo thông số, cấu hình các thiết bị ngoại vi.... Kiểm tra công tắc đã được chuyển sang chế độ ON hay chưa.

Nếu đã bật công tắc thì tiến hành mở ứng dụng trên màn hình máy tính, lúc này servo sẽ quay trả về vị trí 0 độ.

Tiếp tục quá trình vận hành, băng chuyền tải khuôn kẹo vào buồng chụp, sau khi cảm biến tiệm cận phát hiện có khuôn kẹo trên băng tải thì Raspberry Pi sẽ điều khiển relay để dừng băng tải, đồng thời ghi nhận tín hiệu hình ảnh thu được từ camera. Sau đó, Raspberry Pi sẽ xử lý, phân tích và cho ra kết luận cuối cùng về số lượng kẹo có trong khuôn. Khi đã có kết luận về số lượng kẹo thì Raspberry Pi điều khiển relay cho phép băng tải chạy trở lại, nếu số lượng kẹo đếm được khác với thông số đã cài đặt trước thì Raspberry Pi điều khiển servo gạt, ngược lại thì servo sẽ không bị tác động. Đồng thời, hệ thống hiển thị số lượng, số hộp, chênh lệch số lượng so với thông số đặt trước lên trên màn hình.

Quá trình tiếp tục lặp lại đối với khuôn kẹo tiếp theo cho đến khi ta dừng chương trình (công tắc chuyển sang chế độ OFF). Kết quả quá trình sẽ được lưu lại vào file.txt.

- **Chụp ảnh:** Thao tác chụp và lưu lại ảnh của camera.

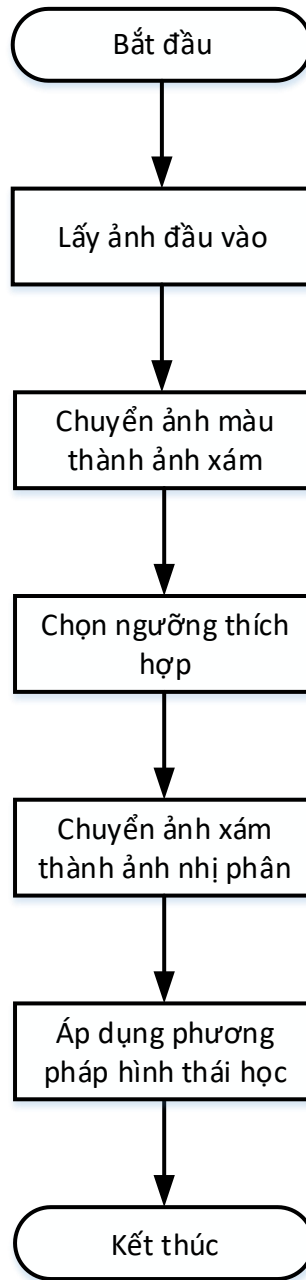


Hình 4. 20: Lưu đồ chụp và lưu ảnh

➤ **Giải thích lưu đồ:**

Khi hệ thống bắt đầu hoạt động, nếu có tín hiệu cho phép chụp từ Raspberry Pi, camera sẽ quay (ghi hình) quá trình khuôn kẹ ở trong buồng chụp. Với tín hiệu cho phép chụp từ Raspberry Pi (cảm biến tiệm cận hồng ngoại tích cực mức cao khi khuôn kẹ di chuyển đến trước cảm biến) camera sẽ chụp lại khung ảnh tại thời điểm và vị trí đó. Quá trình này lặp lại với khuôn kẹ khác khi tiến vào buồng chụp. Ảnh của khuôn kẹ trước sẽ được tự động thay thế bởi ảnh của khuôn kẹ sau.

- Xử lý ảnh

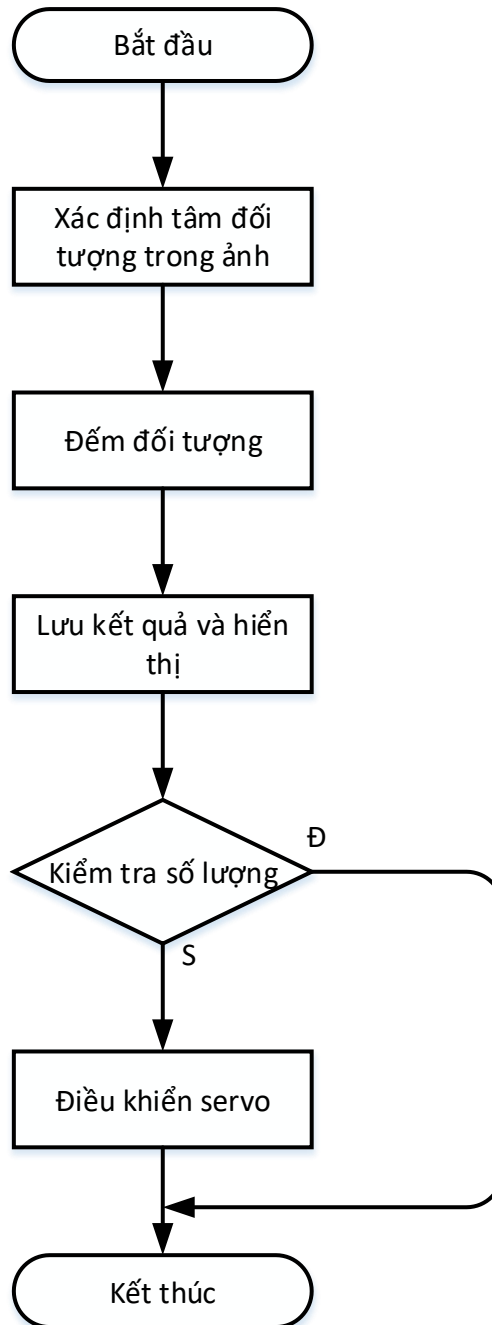


Hình 4. 21: Lưu đồ chương trình xử lý ảnh

➤ **Giải thích lưu đồ:**

Chương trình xử lý ảnh đầu vào được thực hiện qua các quá trình cụ thể. Đầu tiên, ảnh sẽ được chụp từ camera và chuyển ảnh đầu vào sang ảnh xám và tiếp theo tìm giá trị ngưỡng thích hợp. Sau đó, chuyển sang ảnh nhị phân. Kế tiếp sử dụng các phương pháp hình thái học để lọc nhiễu, thu các đối tượng trong ảnh cũng như loại bỏ các đối tượng không mong muốn.

- **Đếm đối tượng**



Hình 4. 22: Lưu đồ chương trình đếm số lượng

➤ **Giải thích lưu đồ:**

Sau khi ảnh đã qua chương trình xử lý ảnh, tiến hành áp dụng phương pháp tìm tâm của đối tượng trong ảnh. Dựa vào số lượng tâm tìm thấy để xác định số lượng kẹo có trong ảnh. Tiến hành so sánh số lượng đếm được với thông số đặt trước để điều khiển servo.

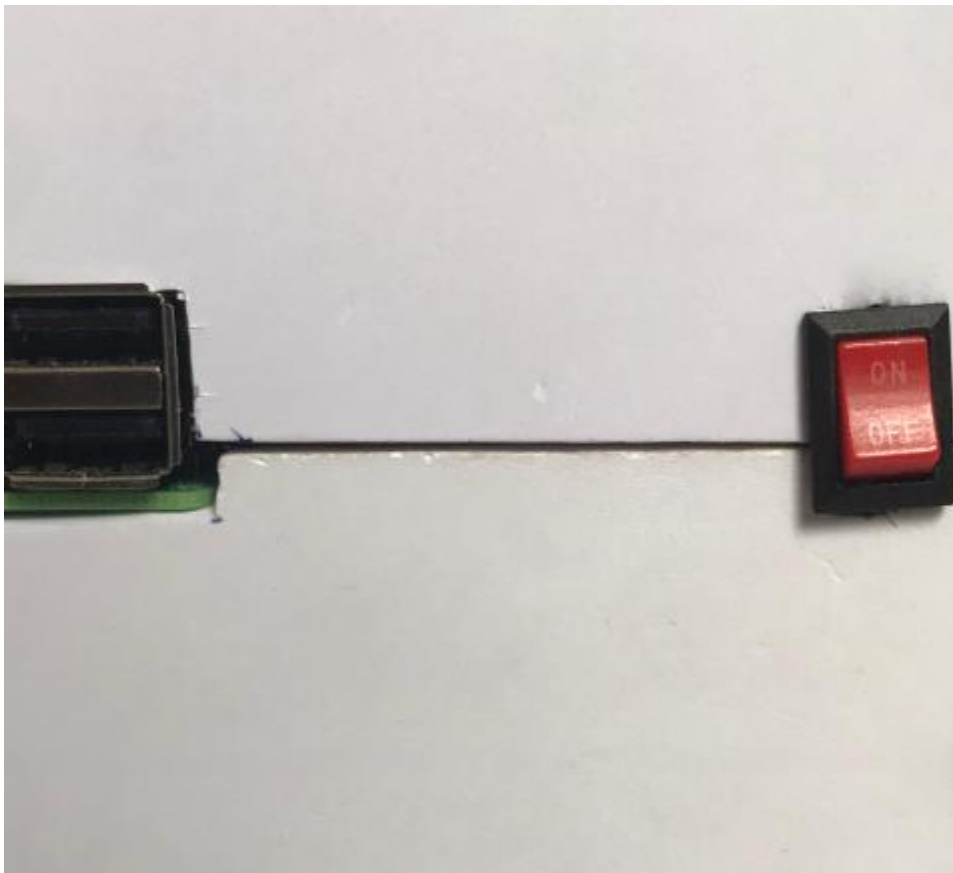
### 4.5 TÀI LIỆU HƯỚNG DẪN

#### ❖ Hướng dẫn sử dụng:

**Bước 1:** Cấp nguồn cho hệ thống: hệ thống sử dụng 3 bộ nguồn là :1 adapter (5V, 2A) sử dụng cho Raspberry Pi; 1 bộ nguồn (12V, 2A) dùng cho băng tải, đèn led siêu sáng (buồng chụp ảnh); 1 nguồn 220V cung cấp cho màn hình LCD. Khi cấp nguồn thì đèn báo hiệu trên Raspberry Pi, đèn led trong buồng chụp ảnh có điện sáng lên, băng chuyền hoạt động, màn hình LCD hiển thị hoạt động của hệ thống.

**Bước 2:** Chạy chương trình điều khiển hệ thống

Kiểm tra công tắc đã chuyển sang trạng thái ON.



Hình 4. 23: Công tắc ở vị trí ON

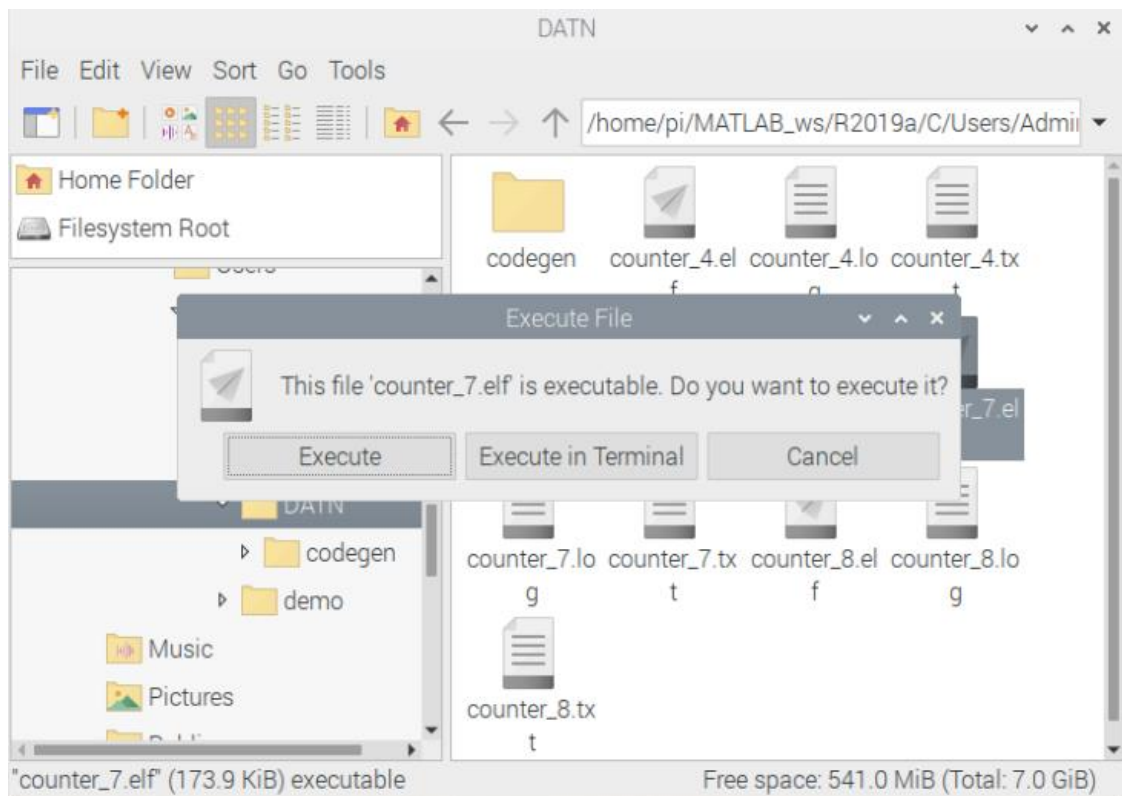
Trên giao diện desktop ta tiến hành dùng chuột mở code chương trình chính điều khiển hệ thống.

Ở đây nhóm đề tài đặt tên chương trình là :

- “counter\_4.elf”: là chương trình chạy đếm 4 viên/hộp.
- “counter\_5.elf”: là chương trình chạy đếm 5 viên/hộp.
- “counter\_7.elf”: là chương trình chạy đếm 7 viên/hộp.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

- “counter\_8.elf”: là chương trình chạy đếm 8 viên/hộp.

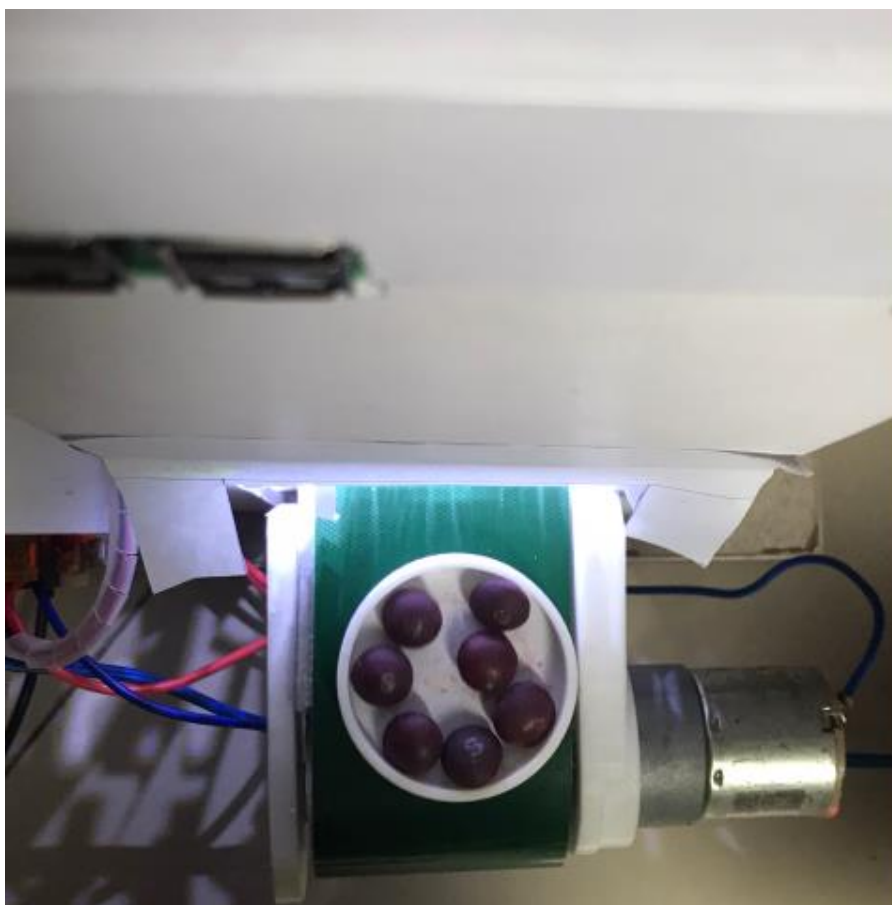


Hình 4. 24: Chạy chương trình đếm 7 viên/hộp

### Bước 3: Phân loại theo số lượng sản phẩm

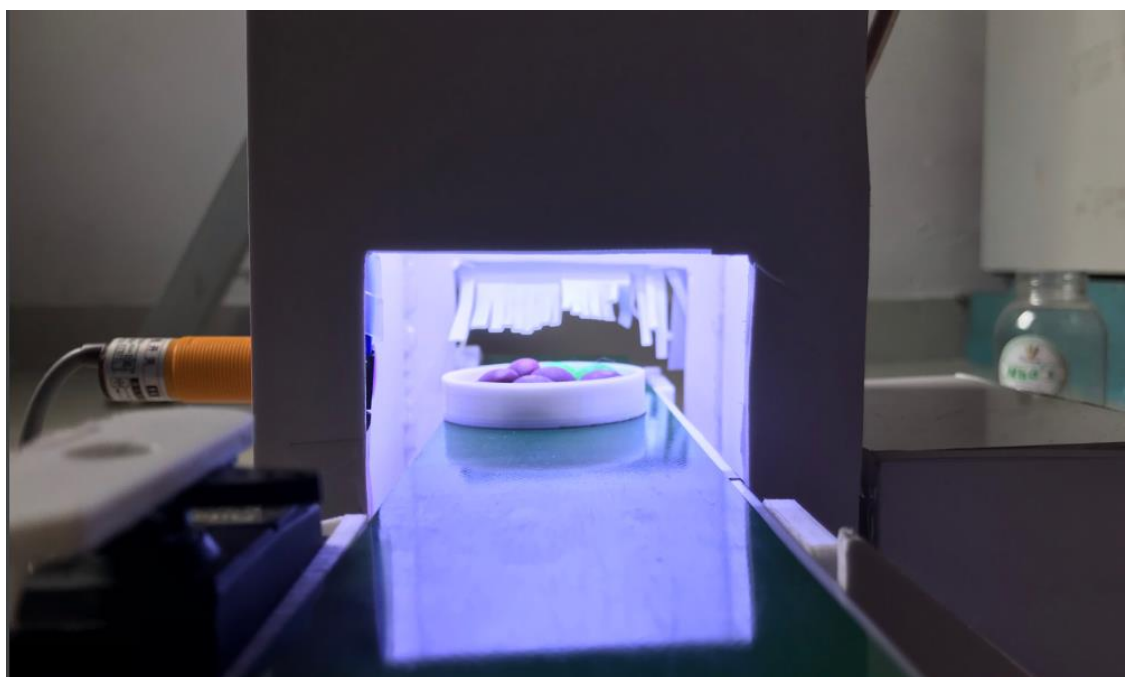
Kẹp sẽ được để trong khuôn ở đầu băng tải và sẽ di chuyển đến cuối băng tải hoặc được servo tác động đẩy xuống hộp chứa.





*Hình 4. 25: Khuôn kẹo trước khi đưa vào buồng chụp*

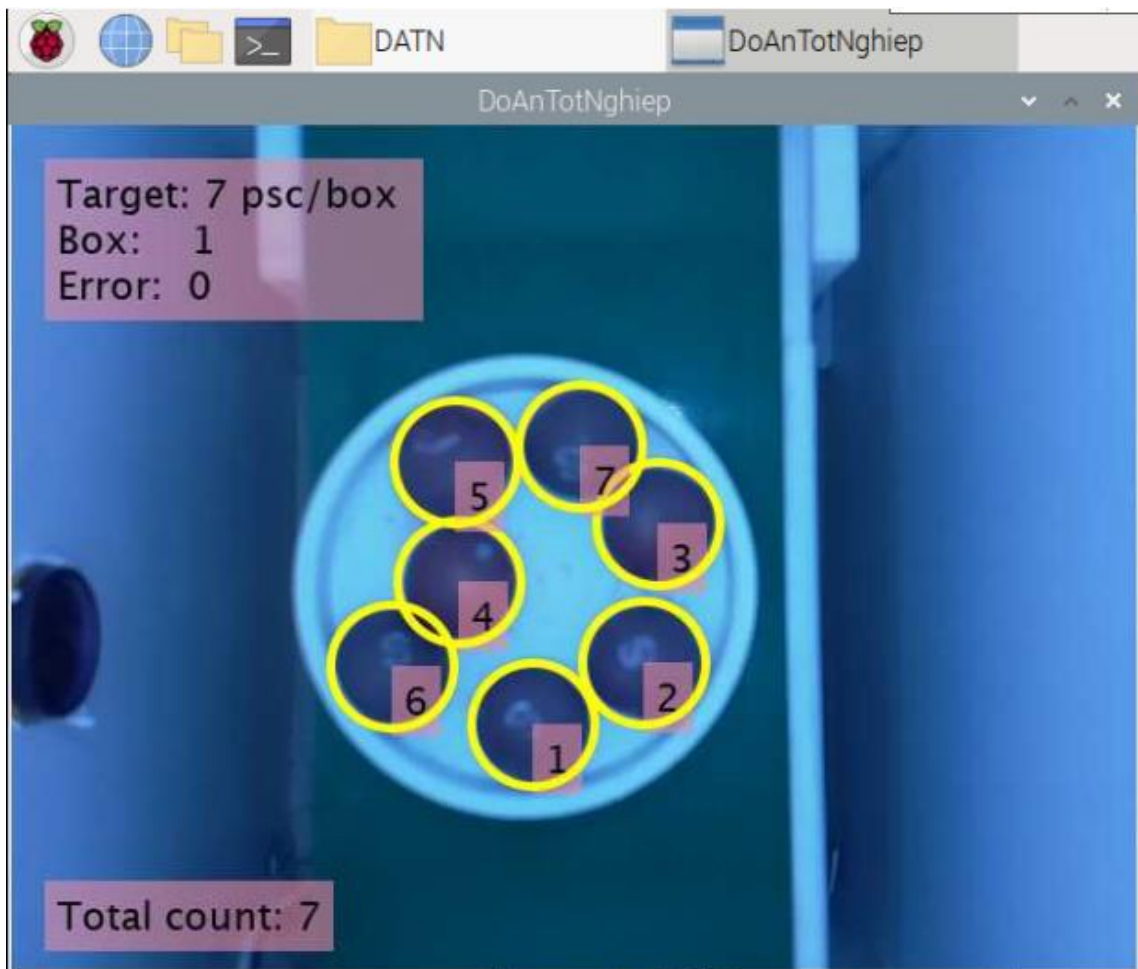
Sau khi cảm biến tác động thì băng tải sẽ dừng để camera tiến hành chụp hình ảnh gửi về khối xử lý trung tâm và tiến hành đếm sản phẩm.



*Hình 4. 26: Khuôn kẹo được đưa vào buồng chụp*

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Sau khi xử lý xong kết quả sẽ được xuất hiện trên màn hình LCD, nếu số lượng kẹo không đúng với số lượng đã đặt trước thì servo sẽ quay đẩy sản phẩm xuống hộp chứa lỗi. Kết quả sẽ được cập nhật trên màn hình và file lưu trữ kết quả.



Hình 4. 27: Kết quả hiện thị trên màn hình

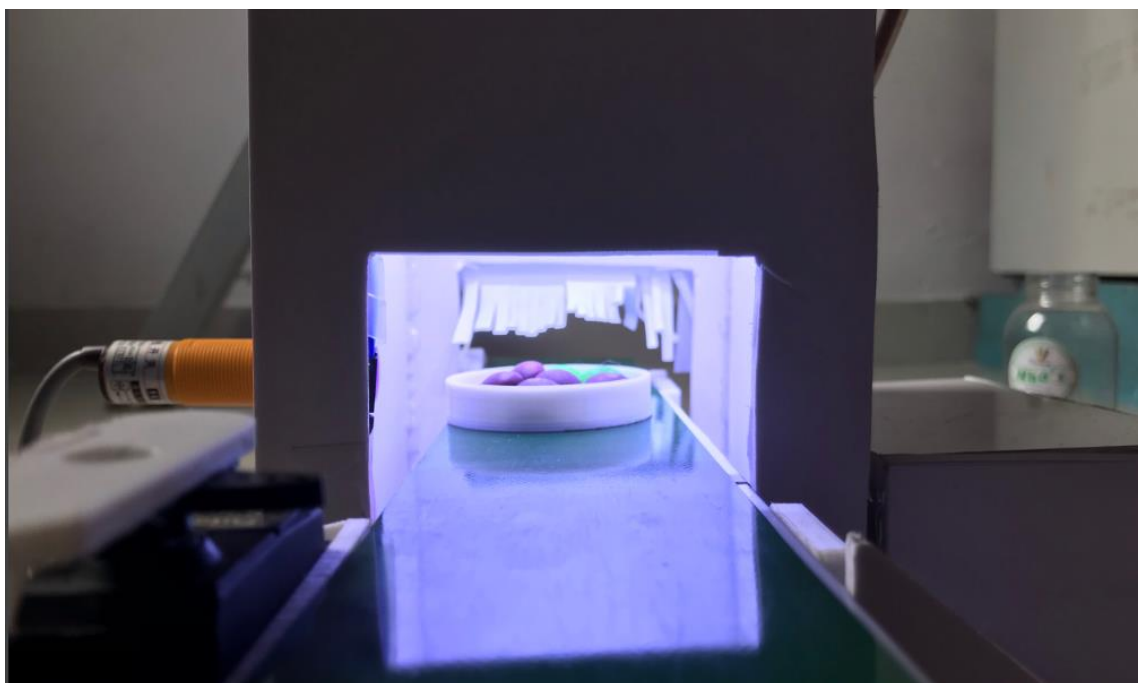


*Hình 4. 28: Số lượng kẹo đúng thì servo không tác động*

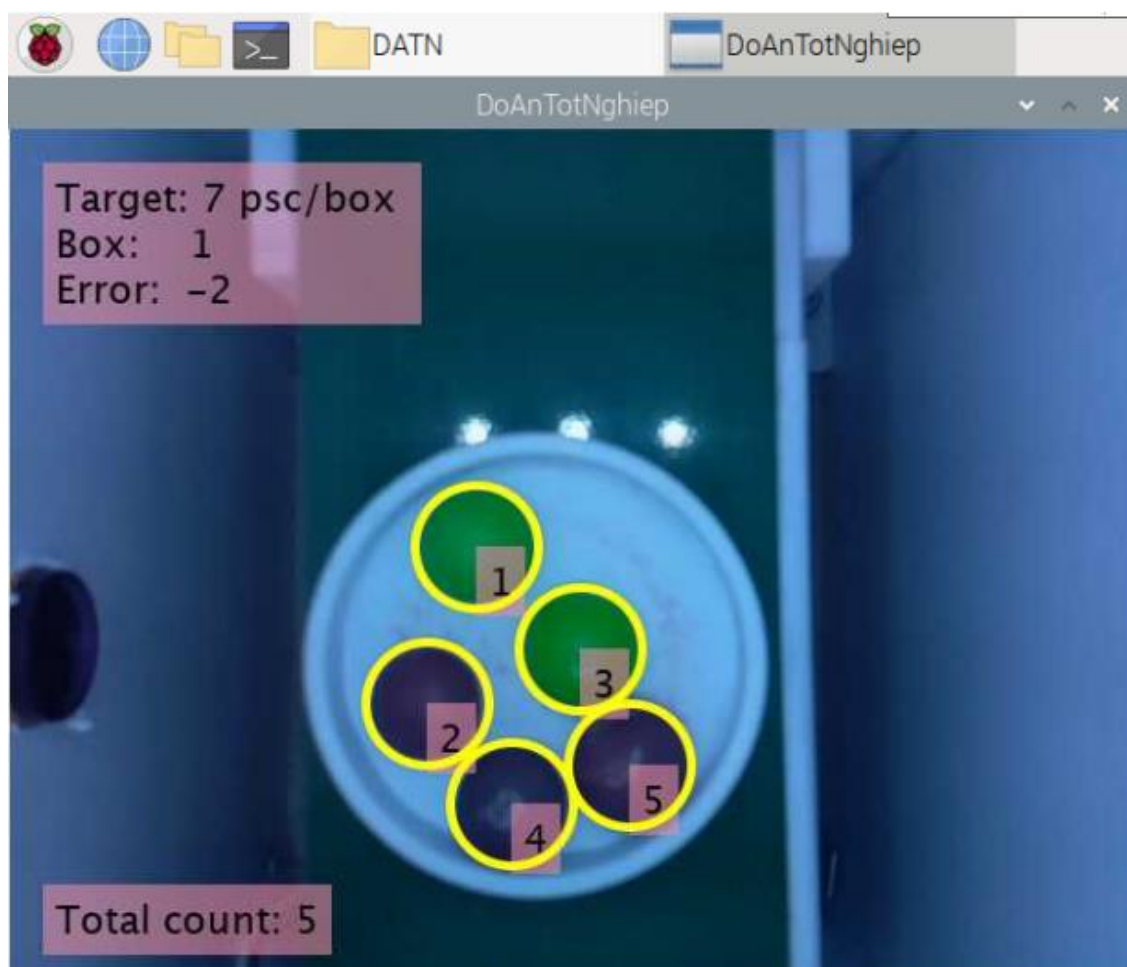
Tiếp theo khuôn kẹo thứ 2 sẽ được đưa vào buồng ảnh và được kiểm tra, xử lý theo trình tự như trên.



*Hình 4. 29: Khuôn kẹo tiếp theo trước khi đưa vào buồng chụp*



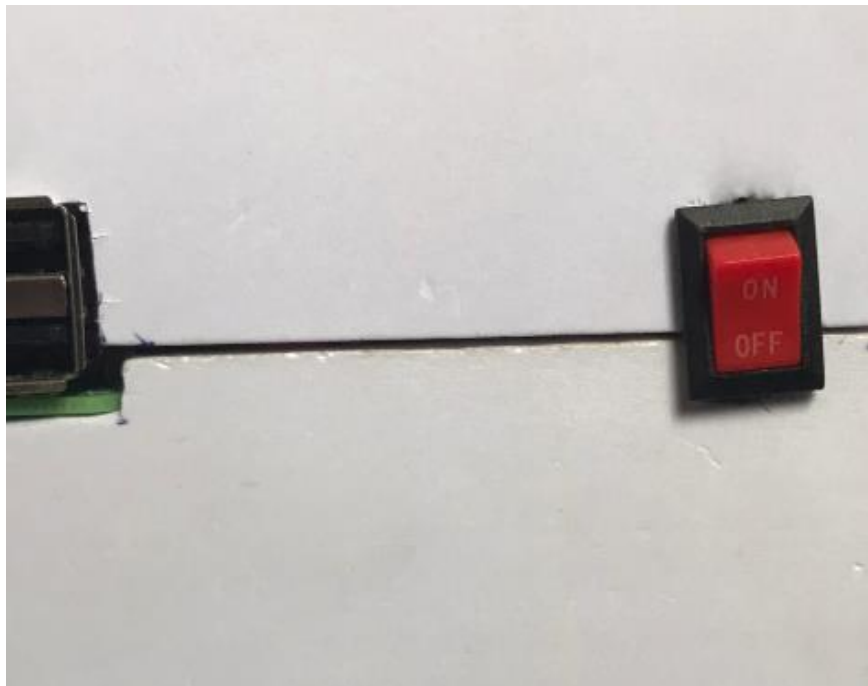
*Hình 4. 30: Khuôn kẹo tiếp theo được đưa vào buồng chụp*



*Hình 4. 31: Kết quả hiện thị trên màn hình*

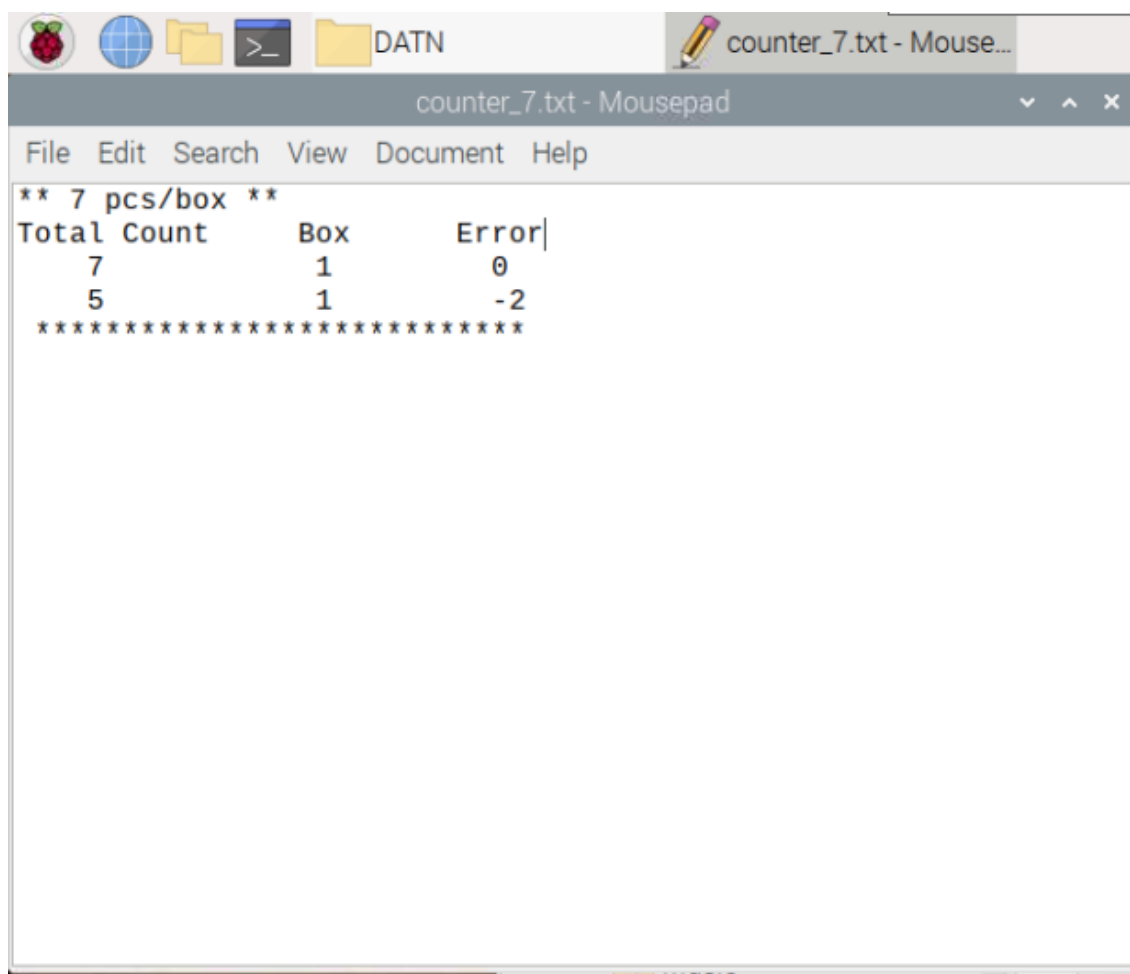


*Hình 4. 32: Số lượng kẹo sai thì servo tác động kẹo xuống máng*  
Hệ thống sẽ thoát khi công tắc chuyển sang trạng thái OFF.



*Hình 4. 33: Công tắc ở vị trí OFF*

Sau khi hệ thống dừng, kết quả của quá trình đếm sẽ lưu trong file.txt.



```
counter_7.txt - Mousepad
File Edit Search View Document Help
** 7 pcs/box **
Total Count      Box      Error|
      7          1          0
      5          1         -2
*****
*****
```

*Hình 4. 34: Kết quả của quá trình chạy hệ thống được lưu lại trong file text*

## **Chương 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

### **5.1 KẾT QUẢ**

#### **5.1.1 Tổng quan kết quả đạt được**

Mục tiêu ban đầu của đề tài “*Thiết kế và thi công hệ thống đếm số lượng trong khâu đóng gói bánh kẹo*” là đếm số lượng kẹo có trong khuôn. Dựa trên ngôn ngữ Matlab và được thực hiện trên Kit Raspberry.

Sau quá trình nghiên cứu và thực hiện đề tài, nhóm đã rút ra được nhiều vấn đề khác nhau, từ việc sử dụng phần mềm, các phương pháp giải thuật, cho tới sử dụng phần cứng. Thời gian thực hiện trong khoảng thời gian 15 tuần. Trong đó, gồm các vấn đề sau:

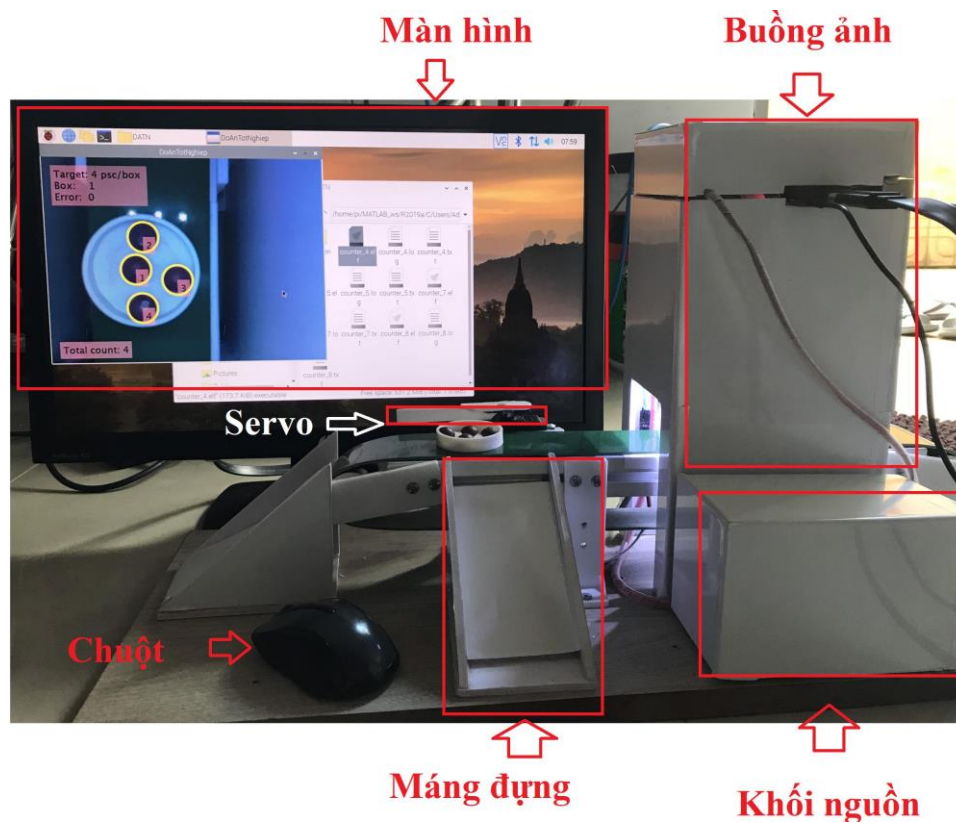
Đối với phần cứng: biết sử dụng Kit Raspberry, cài đặt hệ điều hành cho Raspberry Pi, biết sử dụng Camera Pi, hiểu biết sâu hơn về sử dụng các tính năng của Raspberry Pi như giao tiếp giữa Raspberry Pi với các module liên quan.

Đối với phần mềm: biết cách lập trình cơ bản trên Matlab, cũng như tiến hành chạy code Matlab trên Raspberry Pi độc lập và các giải thuật liên quan đến đề tài, tìm hiểu và nghiên cứu biết được các giải thuật xử lý ảnh để thực hiện xử lý ảnh.

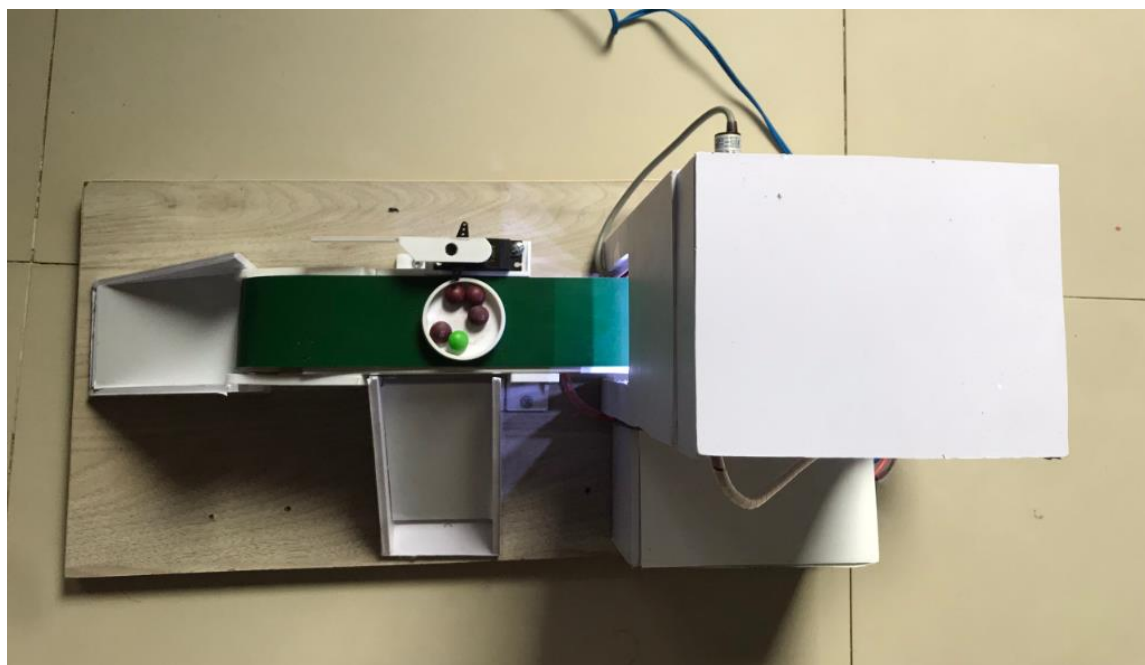
### **5.2 KẾT QUẢ THỰC NGHIỆM**

#### **5.2.1 Mô hình sản phẩm thực tế**

Mô hình sản phẩm hoàn thiện bao gồm băng chuyền, servo SG996, buồng sáng, camera.



*Hình 5.1: Mô hình thực tế nhìn từ phía trước*



*Hình 5.2: Mô hình thực tế nhìn từ trên xuống*

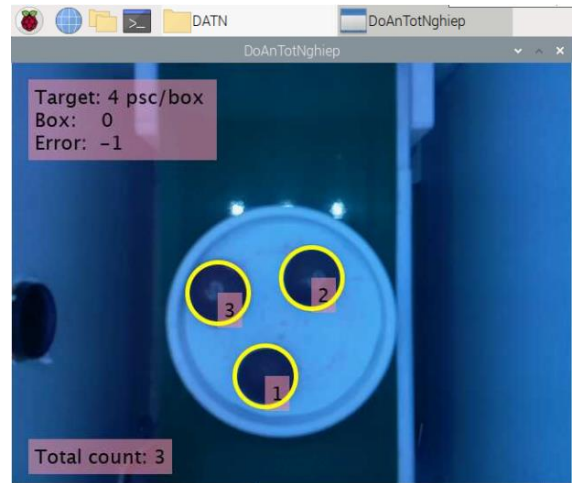
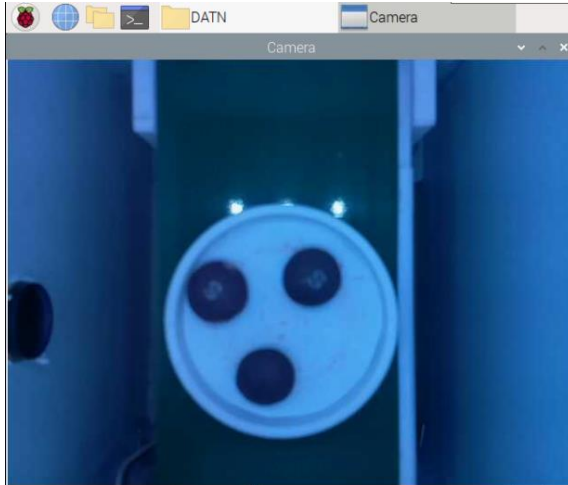


## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

### 5.2.2 Kết quả thực nghiệm từ việc đếm số lượng

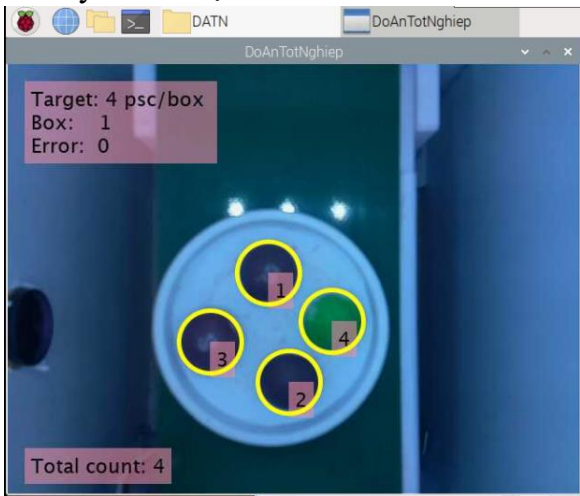
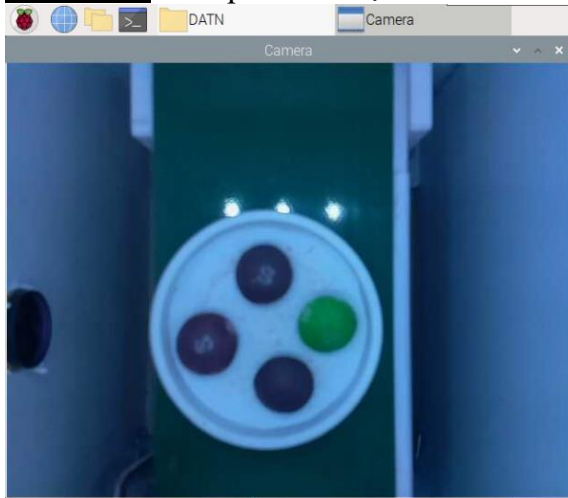
Khởi động phần mềm trên máy tính và tiến hành kiểm tra ảnh của nhiều trường hợp khác nhau

- **Đếm 4 viên/hộp.**



Hình 5.3: Kết quả chụp từ camera (3 viên) Hình 5.4: Kết quả sau khi xử lý (3 viên)

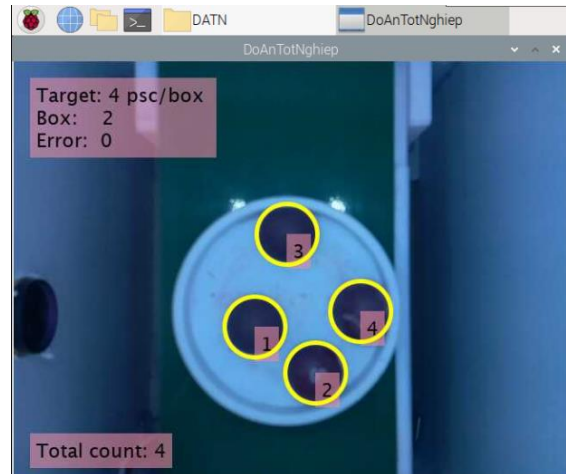
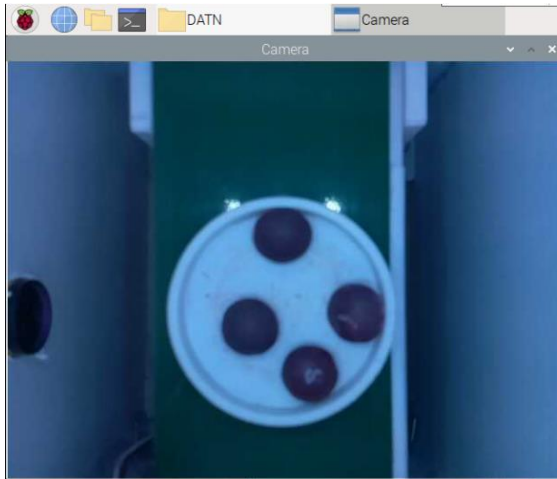
**Nhận xét:** Kết quả thu được thiếu 1 viên so với yêu cầu đặt ra.



Hình 5.5: Kết quả chụp từ camera (4 viên) Hình 5.6: Kết quả sau khi xử lý (4 viên)

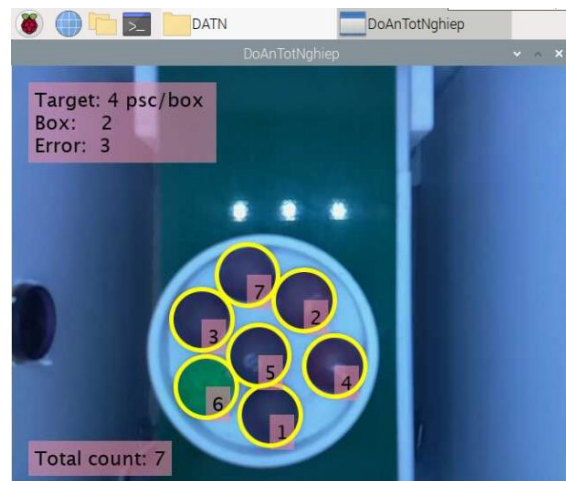
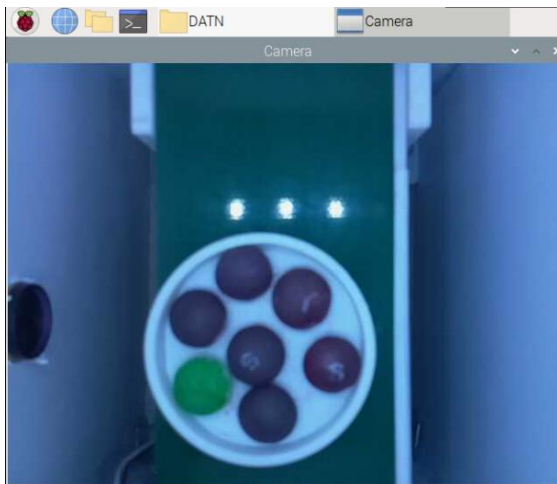
**Nhận xét:** Kết quả thu được đúng so với yêu cầu đặt ra, số Box tăng thêm 1.

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ



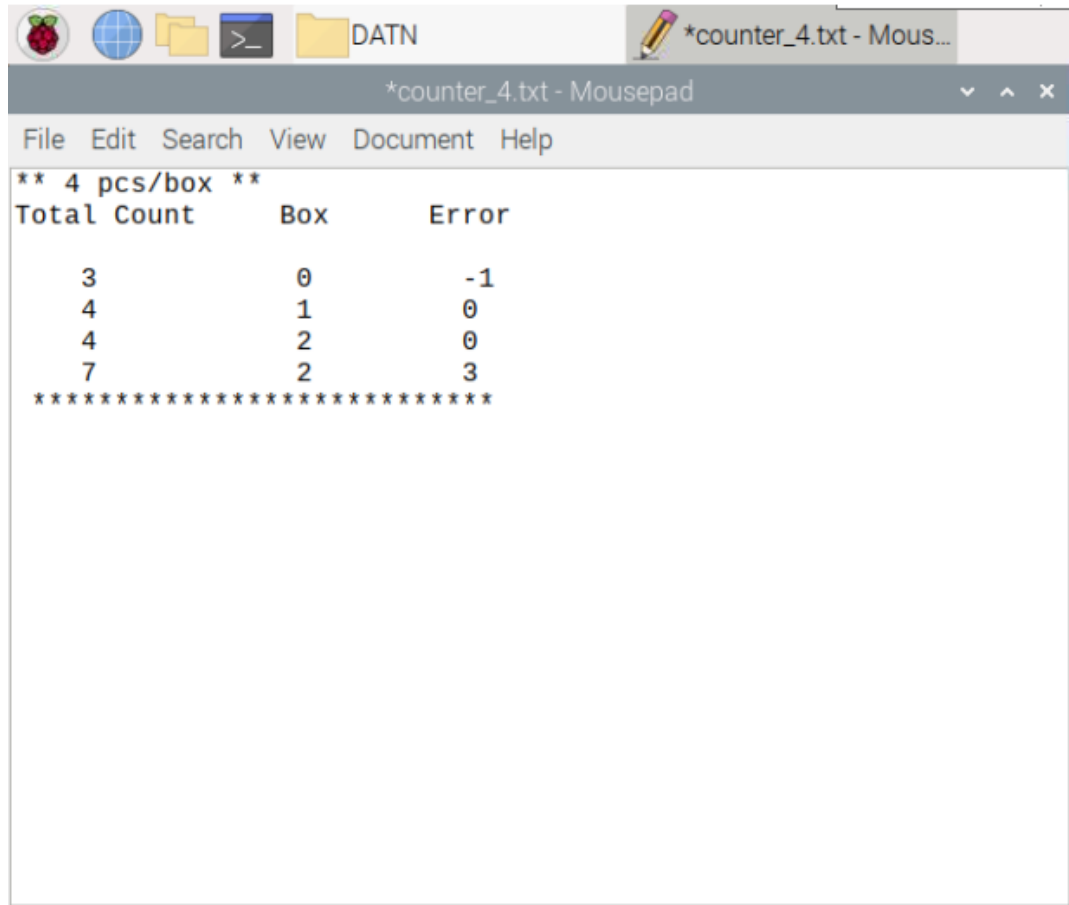
Hình 5.7: Kết quả chụp từ camera (4 viên) Hình 5.8: Kết quả sau khi xử lý (4 viên)

**Nhận xét:** Kết quả thu được đúng số với yêu cầu đặt ra, số Box tăng thêm 1.



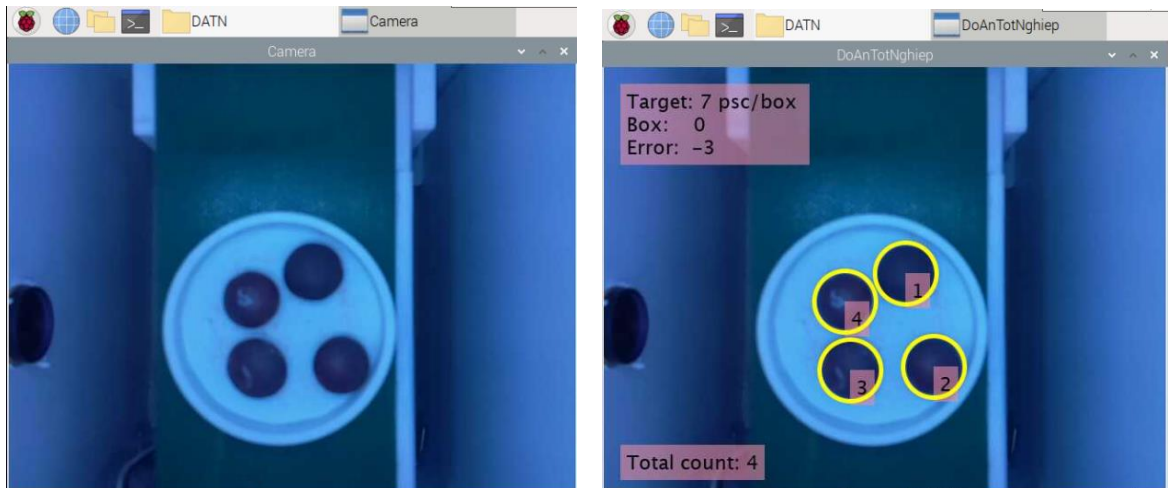
Hình 5.9: Kết quả chụp từ camera (7 viên) Hình 5.10: Kết quả sau khi xử lý (7 viên)

**Nhận xét:** Kết quả thu được thừa 3 viên so với yêu cầu đặt ra



Hình 5. 11: Kết quả của quá trình chạy lưu trong file text.

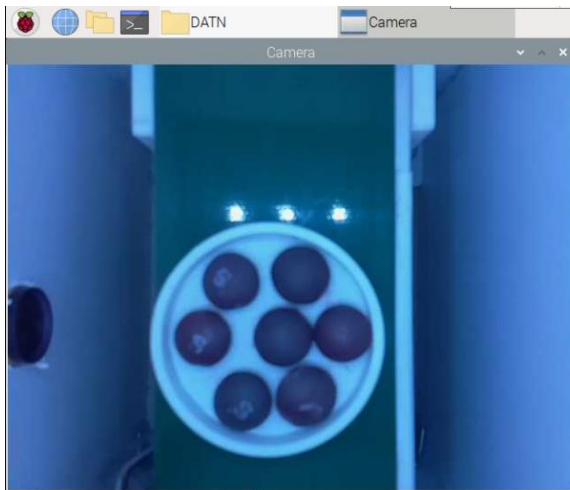
- **Đếm 7 viên/hộp.**



Hình 5.12: Kết quả chụp từ camera (4 viên) Hình 5.13: Kết quả sau khi xử lý (4 viên)

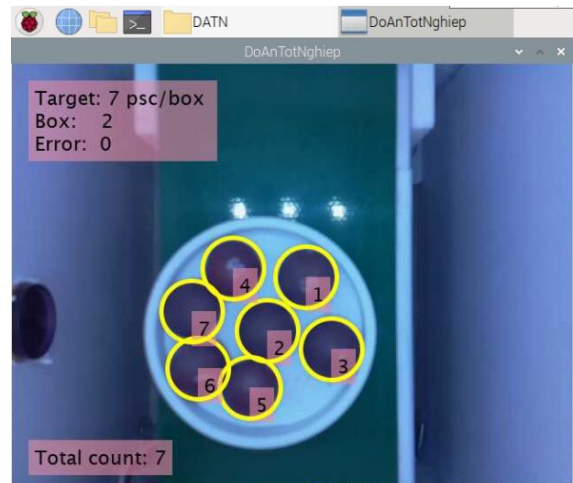
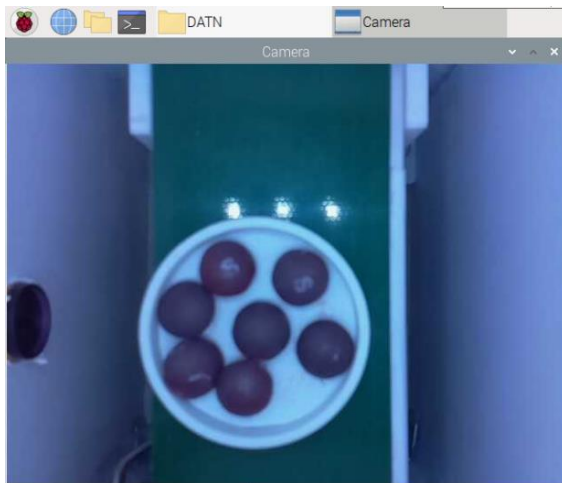
**Nhận xét:** Kết quả thu được thiếu 3 viên so với yêu cầu đặt ra.

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ



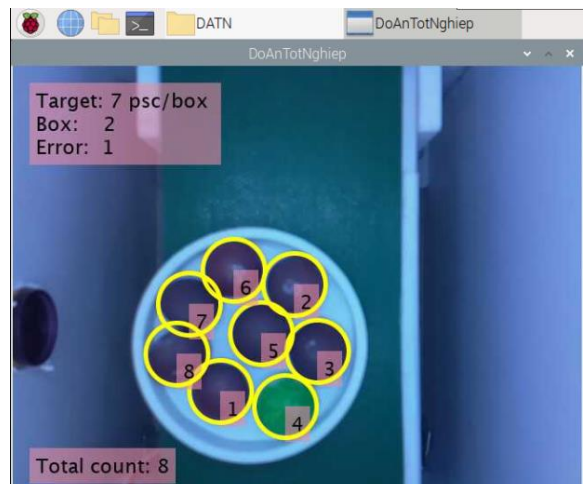
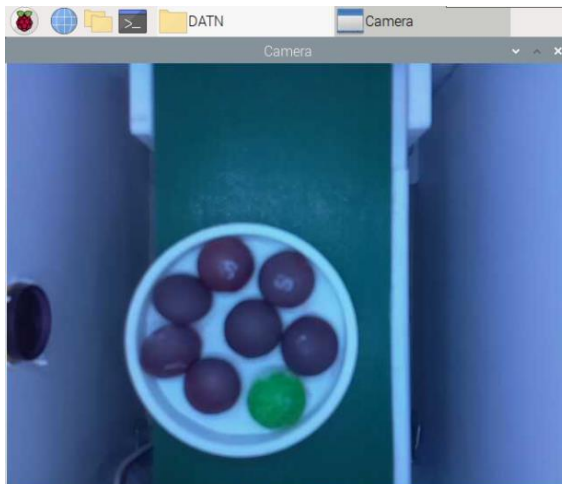
Hình 5.14: Kết quả chụp từ camera (7 viên) Hình 5.15: Kết quả sau khi xử lý (7 viên)

**Nhận xét:** Kết quả thu được đúng số với yêu cầu đặt ra, số Box tăng thêm 1.



Hình 5.16: Kết quả chụp từ camera (7 viên) Hình 5.17: Kết quả sau khi xử lý (7 viên)

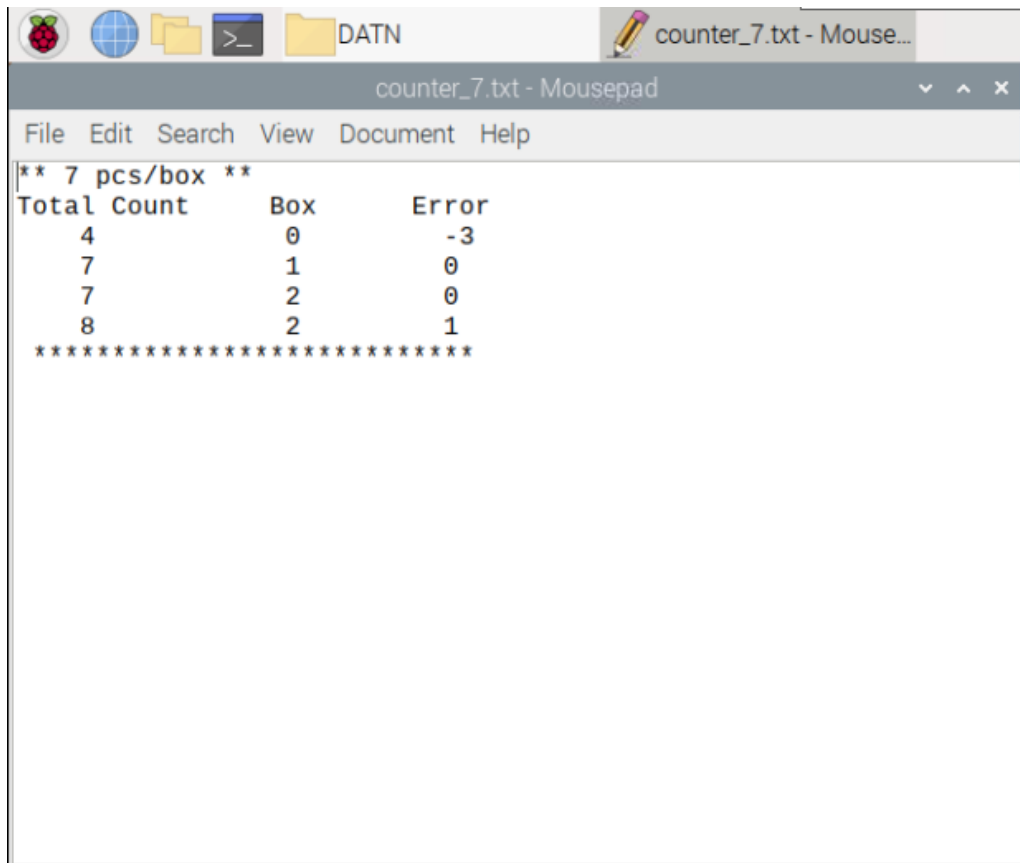
**Nhận xét:** Kết quả thu được đúng số với yêu cầu đặt ra, số Box tăng thêm 1.



Hình 5.18: Kết quả chụp từ camera (8 viên) Hình 5.19: Kết quả sau khi xử lý (8 viên)

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

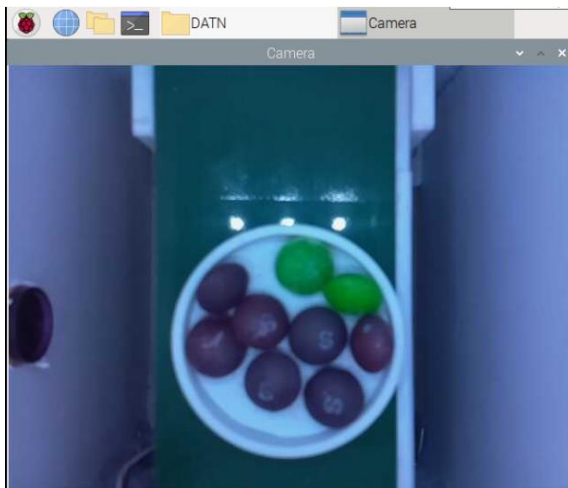
**Nhận xét:** Kết quả thu được thừa 1 so với yêu cầu đặt ra.



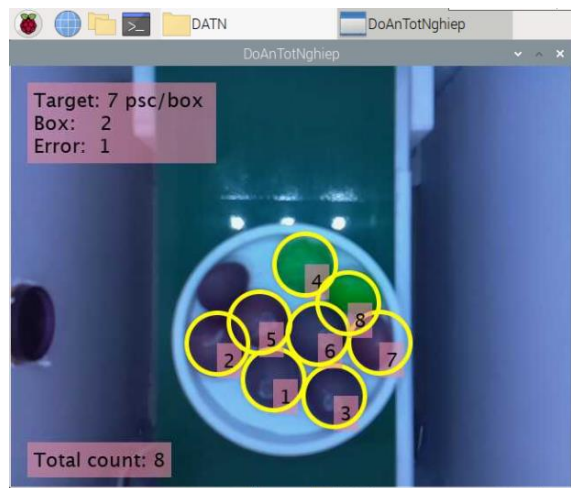
```
counter_7.txt - Mousepad
File Edit Search View Document Help
** 7 pcs/box **
Total Count      Box      Error
4                0        -3
7                1         0
7                2         0
8                2         1
*****
```

Hình 5.20: Kết quả của quá trình chạy lưu trong file text

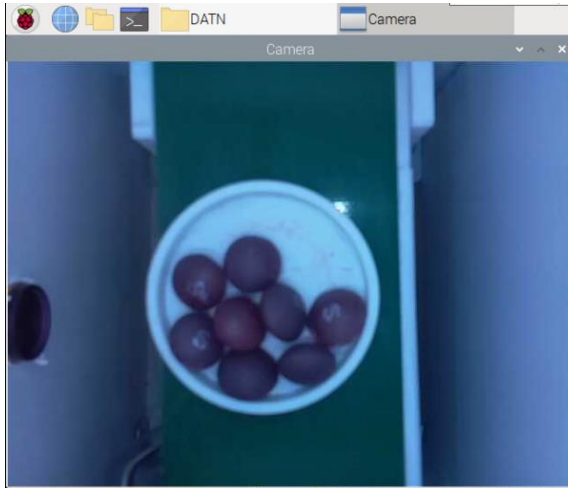
- Những trường hợp đếm lỗi



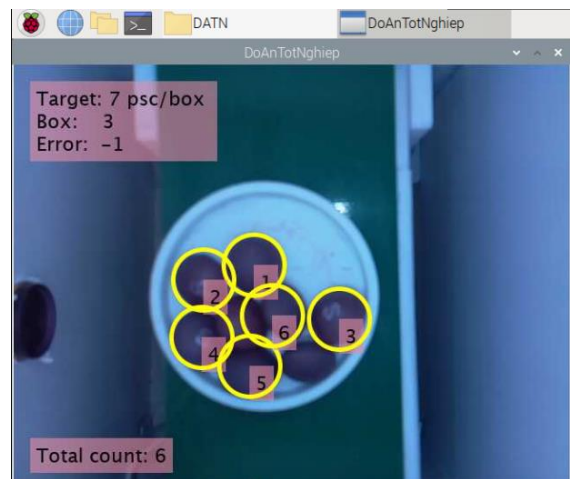
Hình 5.21: Kết quả chụp từ camera



Hình 5.22: Kết quả sau khi xử lý



Hình 5. 23: Kết quả chụp từ camera



Hình 5. 24: Kết quả sau khi xử lý

**Kết luận:** Các trường hợp đếm sai chủ yếu do việc sắp xếp vị trí của các đối tượng trong ảnh (xếp chồng, sát nhau,..) dẫn đến sai sót trong quá trình xử lý ảnh.

### 5.3 NHẬN XÉT – ĐÁNH GIÁ

#### 5.3.1 Nhận xét

Sau 15 tuần nghiên cứu và thực hiện đề tài, hệ thống đã đáp ứng được những mục tiêu và yêu cầu thiết kế.

#### Ưu điểm

- Hệ thống nhỏ gọn, dễ sử dụng, dễ thao tác.
- Lắp đặt thuận tiện cho việc sử dụng.
- Hệ thống có thể phát hiện và đếm chính xác số lượng kẹo.
- Hệ thống sử dụng ngôn ngữ Matlab chạy độc lập trên Raspberry Pi.

#### Nhược điểm

- Cảm biến tiệm cận còn bị nhiễu.
- Còn phụ thuộc vào điều kiện môi trường: độ sáng, không gian, cách sắp xếp của sản phẩm.

#### So sánh với đề tài trước đây

Trong công nghiệp hệ thống được sử dụng rộng rãi nên nhóm thực hiện so sánh với những ưu và nhược điểm sau đây.

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

### Ưu điểm

- Sử dụng ngôn ngữ mới là Matlab, và chạy hoàn tập độc lập như một máy tính nhưng vẫn đáp ứng đủ yêu cầu đặt ra.
- Có thể đếm được các đối tượng sát nhau, hoặc xếp chồng lên nhau.
- Giá thành ít hơn, giá trị đầu tư ít hơn nhưng vẫn đáp ứng đủ tiêu chí cho người tiêu dùng.

### Nhược điểm

- Chưa có giao diện để điều chỉnh thông số.
- Khó áp dụng ngoài công nghiệp.
- Hình ảnh thu được chưa được rõ nét.

### 5.3.2. Đánh giá

Sau khi vận hành hệ thống để tiến hành kiểm tra và giám sát. Quá trình đánh giá được thực hiện ở lúc đếm số lượng kẹo, khoảng thời gian thực hiện từ lúc chụp ảnh đến khi xử lý xong là khoảng 20 giây. Nhóm thực hiện việc đánh giá số lượng và vị trí kẹo trong khuôn là ngẫu nhiên. Dưới đây là bảng giám sát quá trình đánh giá số lượng.

Bảng 5. 1: Bảng đánh giá

Số viên Số lần	0	1	2	3	4	5	6	7	8	9	10
1	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	○	X	X	X	X	X
4	X	X	X	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X	X	X	○
6	X	X	X	X	○	X	X	X	X	○	○
7	X	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	○	X	○	X	X
9	X	X	X	X	X	X	X	○	X	X	X
10	X	X	X	X	X	X	X	X	X	○	X

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---

### **Ghi chú:**

Dấu X: Đếm đúng

Dấu O: Đếm sai

Lí do là vị trí sắp xếp các viên kẹo dính xác nhau nên khi xử lý vẫn xảy ra sai số.



## **CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **6.1 KẾT LUẬN**

#### **6.1.1 Kết quả thu được**

Sau khi tổng hợp các kết quả đạt được và đem so sánh với những yêu cầu và mục tiêu thiết kế cho thấy hệ thống đáp ứng tương đối đầy đủ, chính xác với những kết quả như sau:

- Mô hình phần cứng hoạt động tốt, thiết bị nhỏ gọn, lắp đặt dễ dàng, thuận tiện cho việc nghiên cứu và học tập, tuy nhiên để áp dụng vào môi trường sản xuất thì còn khó.
- Có thể đếm sản phẩm tương đối chính xác.
- Chương trình được viết bằng ngôn ngữ Matlab, sau đó được nhúng lên kit Raspberry Pi và hoàn toàn chạy độc lập, đó chính là cái mới mà những đề tài trước kia chưa làm.
- Sản phẩm đếm được ở dạng hình tròn, có thể xác định được khi xếp dính nhau từ 1 – 2 viên, tuy nhiên ở mức độ xếp dính nhau nhiều thì ko thể xác định được

#### **6.1.2 Những mặt hạn chế**

Ngoài những kết quả đạt được thì hệ thống vẫn còn những hạn chế sau:

- Khi chạy hoàn toàn độc lập trên Raspberry Pi thì tốc độ xử lý chương trình còn chậm so với việc chạy trên máy tính/laptop.
- Chưa đếm chính xác hoàn toàn những trường hợp kẹt bị xếp chồng.
- Chưa có nguồn dự trữ cho để duy trì hoạt động cho hệ thống trong trường hợp mất điện.
- Chưa có giao diện để điều chỉnh thông số theo yêu cầu.
- Các điều kiện về kinh phí cũng như thời gian thực hiện mà tính thẩm mỹ của hệ thống không cao (các đường dán keo thô và không đều).

### **6.2 HƯỚNG PHÁT TRIỂN**

Hệ thống có thể phát triển thêm theo hướng đếm được nhiều kẹo hơn, phát triển thêm công đoạn đóng gói, bao bì, đưa ra thành phẩm,...

## **CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

---

Ngoài ra cũng có thể nâng cao khả năng phát hiện sản phẩm lỗi và đếm kẹo bằng phương pháp nhận dạng bằng camera kép, một camera phát hiện sản phẩm lỗi và một camera đếm số lượng để nâng cao khả năng nhận dạng nhiều góc độ cũng như là tăng độ tin cậy của hệ thống lên cao.

Thiết kế giao diện người dùng để dễ dàng điều chỉnh thông số sản phẩm, tìm hiểu và phát triển thêm khả năng tự học hình dạng sản phẩm mới.

# TÀI LIỆU THAM KHẢO

## Sách tham khảo

- [1] Nguyễn Thanh Hải, “*Giáo Trình Xử lý ảnh*”, Nhà xuất bản Đại Học Quốc Gia, Tp. HCM, 2014.
- [2] Hồ Hữu Long, “*Kỹ thuật sản xuất kẹo*”, Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội, 1983.
- [3] Phạm Thị Thanh Thảo – Phan Trần Hoài Vũ, “*Nghiên cứu, thiết kế và thi công mô hình đếm và phân loại sản phẩm theo cân nặng, màu sắc*”, Đồ án tốt nghiệp, Trường ĐHSPKT Tp. HCM, 2019
- [4] Võ Sĩ Nguyên, “*Thiết kế và thi công hệ thống đếm số lượng cá giống*”, Đồ án tốt nghiệp, Trường ĐHSPKT Tp. HCM, 2019.
- [5] Prof. Irek Defee, “SIRIPHAT POMYEN: Signal and Image Processing with Matlab on Raspberry Pi Platform”, TAMPERE UNIVERSITY OF TECHNOLOGY, 2015.
- [6] Agus Kurniawan, “Getting Started with Matlab Simulink and Raspberry Pi”, 1st Edition, 2013.
- [7] Theo tạp chí STINFO, “*Hấp dẫn thị trường bánh kẹo Việt Nam*”  
<http://www.cesti.vn/the-gioi-du-lieu/hap-dan-thi-truong-banh-k-o-viet-nam.html>.
- [8] <https://www.mathworks.com/>

## PHỤ LỤC

Code chương trình điều khiển:

- Code chương trình đếm 4 viên/hộp.

```
function counter_4()

count = 0;
done = 0;
box = 0;
err = 0;
l = 0;

% ket noi raspberry
r=raspi;
w = cameraboard(r,'Resolution','640x480');% kích thước
khung hình
configurePin(r,22, 'PWM'); % cấu hình và kết nối servo
configurePin(r,27,'DigitalOutput'); % cấu hình và kết
nối relay
configurePin(r,17,'DigitalInput'); % cấu hình và kết
nối cảm biến
configurePin(r,4,'DigitalInput'); % cấu hình và kết nối
công tắc

    a = readDigitalPin(r,4);
    s = servo(r,22);
    writeDigitalPin(r,27,0);
    fid = fopen('counter_4.txt','a');
while( a == 0) % kiểm tra công tắc
    a = readDigitalPin(r,4);
    b = readDigitalPin(r,17);

while (b == 0)&&( count < 40) % kiểm tra cảm biến
    a = readDigitalPin(r,4);
    writeDigitalPin(r,27,1);

    count = count + 1;
    if count == 30
        cam.ImageEffect = 'sketch'; % quay hình
        for i = 1:20
            I = snapshot(w);
            displayImage(r,I,'Title','Camera');
        end

        img = rgb2gray(I); % chuyển thành ảnh xám
```

## TÀI LIỆU THAM KHẢO

```

        BW = imbinarize(img); % chuyen thanh anh nhi
phan
        BW = bwareaopen(~BW,60);% lay nguong
        input_edges = BW;
        [centres, radii] =
imfindcircles(input_edges, [30,
55], 'ObjectPolarity', 'bright', 'Sensitivity', 0.90, 'EdgeTh
reshold', 0.1); % tim tam va ban kinh
        [l,c] = size(radii);
        numcount = int32(1);
        lablecount = numcount;

        if l == 4 % kiem tra so luong
            box = box + 1;
            err = 0;
        else
            err = l - 4;
        end
        box1 = int32(box);
        err1 = int32(err);
        fprintf(fid, '\n      %d          %d
%d ', numcount, box1, err1);
        txt = sprintf('Total count: %d', numcount);

        I = insertText(I, [20
430], txt, 'FontSize', 22, 'BoxColor', [231 142 165]); % hien
thi ket qua
        target = sprintf('Target: 4 psc/box \nBox:
%d \nError:  %d', box1, err1);

        for numcount = 1:1:lablecount % danh nhan
doi tuong
            ht = sprintf('%d', numcount);
            I = insertText(I, [centres(numcount)
centres(numcount+1)], ht, 'FontSize', 22, 'BoxColor', [231
142 165]);
            I =
insertShape(I, 'circle', [centres(numcount)
centres(numcount+1) 35], 'LineWidth', 5);
        end
        I = insertText(I, [20
20], target, 'FontSize', 22, 'BoxColor', [231 142 165]);
        displayImage(r, I, 'Title', 'DoAnTotNghiep'); %
hien thi ket qua cuoi cung
        done = 1;
        count = 40;

    end
end
```

## PHỤ LỤC

```
end
    if l ~= 4 % kiem tra dieu kien servo
        writePosition(s, 120);
        pause(0.1);
    else
        writePosition(s, 0);
        pause(0.1);
    end
    if (done == 1)
        writeDigitalPin(r,27,0);
        pause(5);
        done = 0;
        count = 0;
    end
end
fprintf(fid,'\n *****');
fclose(fid);
writePosition(s, 0);
writeDigitalPin(r,27,1);
release(w);
end
```

- Code chương trình đếm 7 viên/hộp.

```
function counter_7()
count = 0;
done = 0;
box = 0;
err = 0;
l = 0;

% ket noi raspberry
r=raspi;
w = cameraboard(r,'Resolution','640x480');% kich thuc
khung hinh
configurePin(r,22, 'PWM'); % cau hinh va ket noi servo
configurePin(r,27,'DigitalOutput'); % cau hinh va ket
noi relay
configurePin(r,17,'DigitalInput'); % cau hinh va ket
noi cam bien
configurePin(r,4,'DigitalInput'); % cau hinh va ket noi
cong tac

    a = readDigitalPin(r,4);
    s = servo(r,22);
    writeDigitalPin(r,27,0);
    fid = fopen('counter_7.txt','a');
while( a == 0) % kiem tra cong tac
```

## PHỤ LỤC

```
a = readDigitalPin(r,4);
b = readDigitalPin(r,17);

while (b == 0)&&( count < 40) % kiem tra cam bien
    a = readDigitalPin(r,4);
    writeDigitalPin(r,27,1);

    count = count + 1;
    if count == 30
        cam.ImageEffect = 'sketch'; % quay hình
        for i = 1:20
            I = snapshot(w);
            displayImage(r,I,'Title','Camera');
        end

        img = rgb2gray(I); % chuyen thanh anh xam
        BW = imbinarize(img); % chuyen thanh anh nhi
phan
        BW = bwareaopen(~BW,60);% lay nguong
        input_edges = BW;
        [centres, radii] =
imfindcircles(input_edges, [30,
55],'ObjectPolarity','bright','Sensitivity',0.90,'EdgeTh
reshold',0.1); % tim tam va ban kinh
        [l,c] = size(radii);
        numcount = int32(l);
        lablecount = numcount;

        if l == 7 % kiem tra so luong
            box = box + 1;
            err = 0;
        else
            err = l - 7;
        end
        box1 = int32(box);
        err1 = int32(err);
        fprintf(fid,'\n      %d          %d
%d ', numcount, box1, err1);
        txt = sprintf('Total count: %d', numcount);

        I = insertText(I,[20
430],txt,'FontSize',22,'BoxColor',[231 142 165]); % hien
thi ket qua
        target = sprintf('Target: 4 psc/box \nBox:
%d \nError:  %d', box1, err1);

        for numcount = 1:1:lablecount % danh nhan
doi tuong
            ht = sprintf('%d',numcount);
```

## PHỤ LỤC

```
        I = insertText(I,[centres(numcount)
centres(numcount+1)],ht,'FontSize',22,'BoxColor',[231
142 165]);
        I =
insertShape(I,'circle',[centres(numcount)
centres(numcount+1) 35],'LineWidth',5);
        end
        I = insertText(I,[20
20],target,'FontSize',22,'BoxColor',[231 142 165]);
        displayImage(r,I,'Title','DoAnTotNghiep'); %
hien thi ket qua cuoi cung
        done = 1;
        count = 40;
    end
end

    if l ~= 7 % kiem tra dieu khien servo
        writePosition(s, 120);
        pause(0.1);
    else
        writePosition(s, 0);
        pause(0.1);
    end
    if (done == 1)
        writeDigitalPin(r,27,0);
        pause(5);
        done = 0;
        count = 0;
    end
end

    fprintf(fid,'\n *****');
    fclose(fid);
    writePosition(s, 0);
    writeDigitalPin(r,27,1);
    release(w);
end
```

- Code chương trình nạp

```
r=raspi('192.168.1.5','pi','raspberrry')
board = targetHardware('Raspberry Pi')
deploy(board,'counter_4')
deploy(board,'counter_7')
```



