

BỘ GIÁO DỤC VÀ 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: CNKT ĐIỆN TỬ - TRUYỀN THÔNG

ĐỀ TÀI:

**ỨNG DỤNG XỬ LÝ ẢNH TRONG HỆ
THỐNG PHÂN LOẠI SẢN PHẨM**

GVHD: TH.S NGUYỄN DUY THẢO

SVTH: NGUYỄN HIỀN MINH 15141209

PHAN THANH PHONG 15141238

TP. HỒ CHÍ MINH – 6/2019

BỘ GIÁO DỤC VÀ 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: CNKT ĐIỆN TỬ - TRUYỀN THÔNG

ĐỀ TÀI:

**ỨNG DỤNG XỬ LÝ ẢNH TRONG HỆ
THỐNG PHÂN LOẠI SẢN PHẨM**

GVHD: TH.S NGUYỄN DUY THẢO

SVTH: NGUYỄN HIỀN MINH 15141209

PHAN THANH PHONG 15141238

TP. HỒ CHÍ MINH – 6/2019

NHIỆM VỤ ĐO ÁN TỐT NGHIỆP

Họ tên sinh viên:	Phan Thanh Phong Nguyễn Hiền Minh	MSSV: 15141238 MSSV: 15141209
Chuyên ngành:	Điện tử Công nghiệp	Mã ngành: 141
Hệ đào tạo:	Đại học chính quy	Mã hệ: 15
Khóa:	2015	Lớp: 15141DT2A

I. TÊN ĐỀ TÀI: **ỨNG DỤNG XỬ LÝ ẢNH TRONG HỆ THỐNG PHÂN LOẠI SẢN PHẨM**

II. NHIỆM VỤ

1. Các số liệu ban đầu:

Một kit Raspberry Pi, Một kit Arduino, Một Camera Pi, Hệ thống băng tải, Cảm biến. Sản phẩm gồm các phôi với 3 màu sắc đỏ, xanh, vàng.

2. Nội dung thực hiện:

Tổng quan về xử lý ảnh; Tìm hiểu phương pháp nhận dạng và phân loại sản phẩm; Tìm hiểu kit Raspberry Pi; Tìm hiểu kit Arduino Uno và các linh kiện liên quan; Viết chương trình trên kit Raspberry Pi; Viết chương trình trên Arduino Uno; Thiết kế mô hình phân loại sản phẩm.

III. NGÀY GIAO NHIỆM VỤ: 26/02/2019

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 30/06/2019

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: Phan Thanh Phong

Lớp: 15141DT2AMSSV: 15141238

Họ tên sinh viên 1: Nguyễn Hiền Minh

Lớp: 15141DT1BMSSV: 15141209

Tên đề tài: Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm

<i>Tuần/ngày</i>	<i>Nội dung</i>	<i>Xác nhận GVHD</i>
1-2	Thực hiện chọn đề tài	
3	Nhận đề tài, Gặp GVHD để phổ biến quy định.	
4-5	Nghiêm cứu đề tài, tìm tài liệu về đề tài.	
6	Viết đề cương chi tiết.	
7-8	Tìm hiểu về cơ sở lý thuyết.	
9-11	Tiến hành thi công phần cứng.	
12-14	Viết chương trình.	
15	Nạp code và cân chỉnh hệ thống.	
16-17	Viết sách đồ án, Báo cáo đề tài tốt nghiệp.	

GV HUỐNG DÂN
(Ký và ghi rõ họ và tên)

Ths. Nguyễn Duy Thảo

LỜI CAM ĐOAN

Đề tài “Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm” là nhóm tôi tự thực hiện dựa vào tham khảo một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Nhóm thực hiện đề tài
Phan Thanh Phong Nguyễn Hiền Minh

LỜI CẢM ƠN



Để có thể thực hiện và hoàn thành đề tài này, nhóm xin gửi lời chân thành cảm ơn các thầy cô trong Khoa Điện-Điện Tử đã tạo những điều kiện tốt nhất cho em hoàn thành đề tài. Những kiến thức bổ ích mà các Thầy Cô dạy, nó được áp dụng vào đề tài Đồ Án Tốt Nghiệp rất nhiều, từ những kiến thức nhỏ nhặt cho tới những bài học lớn. Một lần nữa nhóm xin được gửi lời cảm ơn đến tất cả Thầy Cô, nếu không có Thầy Cô thì chắc giờ này nhóm sẽ khó có thể hoàn thành đề tài này.

Ngoài sự cố gắng của bản thân, nhóm em không thể nào không nhắc đến công lao đã vạch ra hướng đi cho đề tài và hướng dẫn từng yêu cầu của đề tài mà thầy **Th.S. Nguyễn Duy Thảo** đã truyền đạt cho nhóm em những kiến thức hết sức bổ ích và những ứng dụng thực tế. Thầy **Th.S. Nguyễn Duy Thảo** ân cần chỉ bảo tận tình. Giải thích rõ ràng những chỗ mà nhóm em chưa hiểu.

Tiếp theo nhóm cũng xin cảm ơn tới các Anh, Chị khóa trên cùng các bạn sinh viên đã tạo điều kiện giúp đỡ, từ những tài liệu liên quan tới đề tài cho tới những kinh nghiệm sống thực tế. Nhờ họ mà nhóm mới có thể phát triển được.

Cuối cùng là gửi lời cảm ơn đến Cha, Mẹ nêu không có hai đấng sinh thành thì ngày hôm nay cũng không có ai hiện diện ở đây để thực hiện những việc mình muốn, họ đã tạo mọi điều kiện để giúp con của mình hướng tới một tương lai tốt đẹp.

Mặc dù nhóm em đã cố gắng hoàn thành tốt đề tài này một cách hoàn chỉnh nhất, nhưng cũng không thể tránh những sai sót nhất định trong công tác nghiên cứu, tiếp cận thực tế, cũng như những hạn chế về kiến thức lẫn thời gian thực hiện. Rất mong nhận được sự góp ý của quý thầy cô và các bạn để đề tài này được hoàn chỉnh hơn.

Xin chân thành cảm ơn!

Nhóm thực hiện đề tài

Phan Thanh Phong Nguyễn Hiền Minh

Mục lục

TRANG BÌA	i
NHIỆM VỤ ĐỒ ÁN.....	ii
LỊCH TRÌNH	iii
CAM ĐOAN	iv
LỜI CÁM ƠN	v
MỤC LỤC	vi
LIỆT KÊ HÌNH VẼ.....	ix
LIỆT KÊ BẢNG.....	xii
TÓM TẮT	xiii
CHƯƠNG 1: TỔNG QUAN	1
1.1 ĐẶT VÂN ĐỀ	1
1.2 MỤC TIÊU.....	1
1.3 NỘI DUNG NGHIÊM CỨU.....	1
1.4 GIỚI HẠN	2
1.5 BỐ CỤC	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	4
2.1 TỔNG QUAN VỀ XỬ LÝ ẢNH	4
2.1.1 Thu nhận ảnh (Image Acquisition).....	5
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)	6
2.1.5 Nhận dạng và nội suy ảnh (Image Recognition and Interpretation)	6
2.1.6 Cơ sở tri thức (Knowledge Base)	6
2.1.7 Mô tả	7
2.2 NHỮNG VÂN ĐỀ TRONG XỬ LÝ ẢNH	7
2.2.1 Điểm ảnh (Picture Element).....	7
2.2.2 Ảnh số	7
2.2.3 Phân loại ảnh	7
2.2.4 Quan hệ giữa các điểm ảnh	8
2.2.5 Lọc nhiễu.....	8

2.2.6	Phương pháp phát hiện biên.....	9
2.2.7	Phân đoạn ảnh.....	11
2.2.8	Các phép toán hình thái Morphology	12
2.3	GIỚI THIỆU RASPBERRY PI 3	14
2.3.1	Giới thiệu.....	14
2.3.2	Thông tin cấu hình Raspberry Pi 3	15
2.3.3	Ứng dụng.....	16
2.4	GIỚI THIỆU VỀ CAMERA PI.....	16
2.4.1	Giới thiệu.....	16
2.4.2	Thông tin cấu hình Camera Pi v2.1	17
2.4.3	Ứng dụng.....	18
2.5	GIỚI THIỆU VỀ ARDUINO UNO R3	18
2.5.1	Giới thiệu.....	18
2.5.2	Thông tin cấu hình Arduino Uno R3	18
2.5.3	Ứng dụng.....	22
2.6	GIỚI THIỆU VỀ CẢM BIẾN HỒNG NGOẠI E18-D80NK	23
2.6.1	Giới thiệu.....	23
2.6.2	Thông số kỹ thuật	23
2.6.3	Ứng dụng.....	24
2.7	GIỚI THIỆU VỀ ĐỘNG CƠ DC.....	24
2.7.1	Giới thiệu.....	24
2.7.2	Cấu tạo và nguyên lý hoạt động	24
2.7.3	Ứng dụng.....	26
2.8	GIỚI THIỆU VỀ ĐỘNG CƠ SERVO MG996R	27
2.8.1	Tổng quan về động cơ servo	27
2.8.2	Giới thiệu động cơ servo MG996R	28
2.8.3	Ứng dụng.....	29
2.9	GIỚI THIỆU HỆ THỐNG BĂNG TẢI.....	29
2.9.1	Giới thiệu.....	29
2.9.2	Cấu tạo	30
2.9.3	Ứng dụng.....	31
2.10	GIỚI THIỆU VỀ LCD16X2	31
2.10.1	Giới thiệu.....	31

2.10.2 Cấu tạo	31
2.10.3 Ứng dụng.....	33
2.11 GIỚI THIỆU HỆ ĐIỆU HÀNH TRÊN RASPBERRY PI	33
2.12 GIỚI THIỆU NGÔN NGỮ PYTHON VÀ THƯ VIỆN OPENCV	34
2.12.1 Ngôn ngữ Python	34
2.12.2 Thư viện OPENCV	35
2.13 GIỚI THIỆU VỀ CÁC CHUẨN GIAO TIẾP	36
2.13.1 Chuẩn giao tiếp Uart	36
2.13.2 Chuẩn giao tiếp I2C	39
2.14 GIỚI THIỆU VỀ PHẦN MỀM ARDUINO IDE.....	42
2.15 PHƯƠNG PHÁP PHÂN LOẠI SẢN PHẨM THEO MÀU SẮC	44
2.15.1 Các màu sắc cơ bản của sản phẩm	44
2.15.2 Phương pháp nhận dạng màu sắc	44
CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ	50
3.1 GIỚI THIỆU.....	50
3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG	50
3.2.1 Thiết kế sơ đồ khói hệ thống	50
3.2.2 Sơ đồ kết nối hệ thống	51
3.2.3 Sơ đồ kết nối toàn mạch.....	64
CHƯƠNG 4: THI CÔNG HỆ THỐNG	65
4.1 GIỚI THIỆU.....	65
4.2 THI CÔNG HỆ THỐNG	65
4.2.1 Chuẩn bị phần cứng	65
4.2.2 Lắp ráp và kiểm tra	66
4.3 LẬP TRÌNH HỆ THỐNG.....	71
4.3.1 Lưu đồ giải thuật trên Arduino.....	71
4.3.2 Lưu đồ giải thuật xử lý ảnh trên Raspberry	73
4.3.3 Các bước tuy cập, lập trình trên Python	75
4.3.4 Hướng dẫn sử dụng và thao tác	78
CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ.....	85
5.1 KẾT QUẢ.....	85
5.2 NHẬN XÉT VÀ ĐÁNH GIÁ	97
CHƯƠNG 6: KẾT LUẬN VÀ PHÁT TRIỂN	99

6.1 KẾT LUẬN	99
6.1.1 Kết quả đạt được.....	99
6.1.2 Những mặt hạn chế.....	99
6.2 HƯỚNG PHÁT TRIỂN	99
TÀI LIỆU THAM KHẢO	100
PHỤ LỤC.....	101

LIỆT KÊ HÌNH ẢNH

Hình	Trang
Hình 2.1: Các bước cơ bản trong xử lý ảnh.....	5
Hình 2.2: Lân cận 4 lân cận 8	8
Hình 2.3: Hình tách biên.....	10
Hình 2.4: Phép giản	13
Hình 2.5: Phép co	13
Hình 2.6: Raspberry Pi 3	14
Hình 2.7: Sơ đồ Raspberry Pi 3 Module B	15
Hình 2.8: Camera Pi v2.1	17
Hình 2.9: Sơ đồ khối Camera Raspberry Pi	18
Hình 2.10: Arduino Uno R3	20
Hình 2.11: Sơ đồ chân ATMega 328P ứng với arduino Uno R3	22
Hình 2.12: Cảm biến hồng ngoại E18-D80NK.....	23
Hình 2.13: Động cơ DC	24
Hình 2.14: Pha 1 động cơ DC	25
Hình 2.15: Pha 2 động cơ DC	25
Hình 2.16: Pha 3 động cơ DC	26
Hình 2.17: Động cơ servo	27
Hình 2.18: Cấu tạo bên trong động cơ servo	28
Hình 2.19: Động cơ servo MG996R	29
Hình 2.20: Mô hình băng tải	30
Hình 2.21: LCD16x2	31
Hình 2.22: Sơ đồ chân LCD16x2	32
Hình 2.23: Hệ điều hành Raspbian	34
Hình 2.24: Giao tiếp Uart	36
Hình 2.25: Giao tiếp song song.....	37
Hình 2.26: Truyền thông Uart.....	38
Hình 2.27: Giao diện Uart	39
Hình 2.28: Hệ thống các thiết bị giao tiếp chuẩn I2C.....	40
Hình 2.29: Quá trình chủ ghi dữ liệu vào tớ.....	41
Hình 2.30: Quá trình chủ đọc dữ liệu từ tớ.....	42
Hình 2.31: Giao diện phần mềm Arduino IDE	43
Hình 2.32: Icon và chức năng trong Arduino IDE.....	43
Hình 2.33: Không gian màu RGB	45

Hình 2.34: Không gian màu CMYK	46
Hình 2.35: Không gian màu HSV	46
Hình 2.36: Sơ đồ các bước thực hiện phân loại màu	47
Hình 3.1: Sơ đồ khói hệ thống	50
Hình 3.2: Sơ đồ kết nối của Raspberry Pi	52
Hình 3.3: Sơ đồ kết nối thực tế của Raspberry Pi.....	52
Hình 3.4: Sơ đồ các cổng ngoại vi sử dụng	53
Hình 3.5: Sơ đồ kết nối thẻ nhớ của Raspberry	54
Hình 3.6: Gắn thẻ nhớ vào khe thực tế.....	54
Hình 3.7: Sơ đồ kết nối Arduino	55
Hình 3.8: Giao tiếp Uart giữa Raspberry Pi và Arduino Uno	56
Hình 3.9: Sơ đồ kết nối khói Camera	57
Hình 3.10: Kết nối Camera thực tế	57
Hình 3.11: Giao tiếp I2C giữa Arduino Uno và LCD16x2	58
Hình 3.12: Kết nối Arduino Uno và động cơ DC	59
Hình 3.13: Kết nối giữa Arduino Uno và động cơ servo	59
Hình 3.14: Kết nối giữa Arduino Uno và cảm biến E18-D80NK	60
Hình 3.15: Adapter 5VDC – 2A	60
Hình 3.16: Sơ đồ nguyên lý mạch hạ áp LM2596	61
Hình 3.17: Hình ảnh thực tế mạch hạ áp LM2596.....	61
Hình 3.18: Mạch cấp nguồn cho động cơ DC 24V	62
Hình 3.19: Mạch cấp nguồn cho Servo MG996, Relay 5V, Cảm biến E18-D80NK, LCD16x2	63
Hình 3.20: Nguồn tổ ong 24VDC – 5A.....	63
Hình 3.21: Sơ đồ kết nối toàn mạch.....	64
Hình 4.1: Băng tải và động cơ DC	67
Hình 4.2: Máng đưa sản phẩm vào	67
Hình 4.3: Máng đưa sản phẩm sau khi phân loại.....	68
Hình 4.4: Màn hình LCD16x2 hiển thị kết quả	68
Hình 4.5: Hình ảnh thực tế kết nối Raspberry và Arduino.....	70
Hình 4.6: Lắp ráp giữa Lm2596 với Servo Mg996r, Cảm biến	70
Hình 4.7: Lắp ráp giữa Arduino Uno với Servo Mg996r, Cảm biến, Relay 5V	71
Hình 4.8: Mô hình toàn hệ thống	71
Hình 4.9: Lưu đồ giải thuật trên Arduino Uno	72
Hình 4.10: Lưu đồ giải thuật xử lý ảnh trên Raspberry Pi	74
Hình 4.11: Tạo địa chỉ IP tĩnh.....	76

Hình 4.12: Cho phép chia sẻ mạng Lan	76
Hình 4.13: Phần mềm Advanced IP Scanner.....	77
Hình 4.14: Remote Desktop Connection	77
Hình 4.15: Giao diện đăng nhập vào Raspberry	78
Hình 4.16: Giao diện Raspberry	78
Hình 4.17: Cửa sổ Terminal.....	79
Hình 4.18: Gọi lệnh thực thi chương trình	80
Hình 4.19: Cửa sổ PuTTY	81
Hình 4.20: Cửa sổ Terminal.....	81
Hình 4.21: Tạo một file.sh trong đường dẫn home/pi.....	82
Hình 4.22: Đặt tên file với đuôi .sh.....	82
Hình 4.23: Lưu các lệnh cần thực thi	83
Hình 4.24: Lệnh chạy tự động	83
Hình 4.25: Thêm đường dẫn file cần chạy tự động	84
Hình 4.26: Công tắt nguồn và nút nhấn Reset	84
Hình 5.1: Mô hình hoàn thiện chụp từ bên trái.....	86
Hình 5.2: Mô hình hoàn thiện chụp từ bên phải	86
Hình 5.3: Mô hình hoàn thiện chụp từ trên xuống.....	87
Hình 5.4: Kết quả nhận dạng màu đỏ lần 1	87
Hình 5.5: Kết quả nhận dạng màu đỏ lần 2	88
Hình 5.6: Kết quả nhận dạng màu đỏ lần 3	88
Hình 5.7: Kết quả nhận dạng màu đỏ lần 4	88
Hình 5.8: Kết quả nhận dạng màu đỏ lần 5	89
Hình 5.9: Kết quả nhận dạng màu đỏ	89
Hình 5.10: Hình ảnh thực tế phân loại sản phẩm đỏ	90
Hình 5.11: Kết quả hiển thị sau 5 lần phân loại sản phẩm đỏ	90
Hình 5.12: Kết quả nhận dạng màu xanh lần 1.....	91
Hình 5.13: Kết quả nhận dạng màu xanh lần 2.....	91
Hình 5.14: Kết quả nhận dạng màu xanh lần 3.....	91
Hình 5.15: Kết quả nhận dạng màu xanh lần 4.....	92
Hình 5.16: Kết quả nhận dạng màu xanh lần 5.....	92
Hình 5.17: Kết quả nhận dạng màu xanh	93
Hình 5.18: Hình ảnh thực tế phân loại sản phẩm màu xanh.....	93
Hình 5.19: Kết quả hiển thị sau 5 lần phân loại sản phẩm xanh.....	94
Hình 5.20: Kết quả nhận dạng màu vàng lần 1	94

Hình 5.21: Kết quả nhận dạng màu vàng lần 2.....	94
Hình 5.22: Kết quả nhận dạng màu vàng lần 3.....	95
Hình 5.23: Kết quả nhận dạng màu vàng lần 4.....	95
Hình 5.24: Kết quả nhận dạng màu vàng lần 5.....	95
Hình 5.25: Kết quả nhận dạng màu vàng	96
Hình 5.26: Hình ảnh thực tế phân loại sản phẩm màu vàng.....	96
Hình 5.27: Kết quả hiển thị sau 5 lần phân loại sản phẩm vàng.....	97

LIỆT KÊ BẢNG

Bảng	Trang
Bảng 2.1: Bảng thông số Arduino Uno R3.....	18
Bảng 2.2: Thông số kỹ thuật: Ví điều khiển ATmega328P	21
Bảng 2.3: Chức năng các chân trên LCD	32
Bảng 3.1: Thông kê dòng tiêu thụ.....	55
Bảng 4.1: Danh sách các linh kiện	65
Bảng 5.1: Bảng đánh giá độ chính xác phân loại sản phẩm.....	97

TÓM TẮT

Đề tài “Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm” là mô hình phân loại sản phẩm theo màu sắc (đỏ, xanh, vàng). Dựa trên ngôn ngữ Python với thư viện chính là OpenCV và được thực hiện trên Kit Raspberry và Kit Arduino Uno. Ở đây sử dụng các đặc điểm riêng biệt của từng màu sắc để nhận dạng và sau đó phân loại từng sản phẩm. Kết quả thực hiện của đề tài đã nhận dạng được những sản phẩm có màu sắc (đỏ, xanh, vàng) cùng với việc đếm được sản phẩm theo màu sắc của từng sản phẩm.

CHƯƠNG 1: TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Ngày nay, xã hội ngày càng phát triển. Công nghiệp hóa, hiện đại hóa ngày càng được nâng cao để phát triển đất nước và cải thiện cuộc sống của người dân. Vì vậy việc ứng dụng khoa học kỹ thuật ngày càng rộng rãi, phổ biến và mang lại hiệu quả cao trong hầu hết các lĩnh vực kinh tế, kỹ thuật cũng như trong đời sống xã hội.

Xét điều kiện cụ thể ở nước ta trong công cuộc công nghiệp hóa, hiện đại hóa sử dụng ngày càng nhiều thiết bị hiện đại để điều khiển tự động các quá trình sản xuất, gia công và chế biến sản phẩm... Điều này dẫn đến việc hình thành các hệ thống sản xuất linh hoạt, cho phép tự động hóa ở mức độ cao trên cơ sở sử dụng các máy CNC, robot công nghiệp. Trong đó có một khâu quan trọng ảnh hưởng đến chất lượng sản phẩm là hệ thống phân loại sản phẩm. Hệ thống phân loại sản phẩm nhằm chia sản phẩm ra các nhóm có cùng thuộc tính với nhau để thực hiện đóng gói hay loại bỏ sản phẩm hỏng. Hiện nay để phân loại sản phẩm người ta thường sử dụng các loại cảm biến với các chức năng khác nhau để phân loại sản phẩm theo mong muốn như cảm biến phân loại theo màu sắc, cảm biến phân loại theo hình dáng... Những cảm biến này có ưu điểm chung là quá trình lắp đặt và vận hành tương đối đơn giản nhưng lại dễ gây nhiễu. Do đó dựa trên nền tảng kiến thức đã học, vốn hiểu biết về điện tử và công nghệ xử lý ảnh cùng với sự cho đồng ý của giáo viên hướng dẫn – thầy **Nguyễn Duy Thảo**, nhóm chúng em chọn đề tài: “**Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm**”.

1.2 MỤC TIÊU

Đề tài “**Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm**” với mục tiêu là phân loại sản phẩm theo màu sắc (đỏ, xanh, vàng). Dựa trên ngôn ngữ Python với thư viện chính là OpenCV và được thực hiện trên Kit Raspberry và kit Arduino Uno

1.3 NỘI DUNG NGHIÊM CỨU

Đề tài “**Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm**” Có những nội dung sau:

- **NỘI DUNG 1:** Tìm hiểu về Raspberry pi 3 và Arduino Uno
- **NỘI DUNG 2:** Tổng quan về xử lý ảnh.
- **NỘI DUNG 3:** Viết chương trình

CHƯƠNG 1: TỔNG QUAN

- NỘI DUNG 4: Thiết kế và thi công mô hình
- NỘI DUNG 5: Chạy thử nghiệm và cân chỉnh mô hình
- NỘI DUNG 6: Viết sách đồ án tốt nghiệp
- NỘI DUNG 7: Báo cáo đề tài đồ án tốt nghiệp

1.4 GIỚI HẠN

Thời gian và tốc độ xử lý còn chậm, cùng với Camera chụp ảnh không mong muốn ở điều kiện thiếu ánh sáng mà phải được che kín và chiếu thêm đèn led để tăng cường sáng. Đề tài chỉ xây dựng với mô hình nhỏ và gọn không làm thành dây chuyền sản xuất, sử dụng kit Raspberry Pi 3, Arduino, băng tải, cảm biến, động cơ Servo và hiển thị kết quả trên màn hình lcd.

1.5 BỐ CỤC

- ❖ Nội dung đề tài gồm các phần sau

Chương 1: Tổng quan

- Đặt vấn đề
- Mục tiêu
- Nội dung nghiên cứu
- Giới hạn
- Bố cục

Chương 2: Cơ sở lý thuyết

- Tổng quan về xử lý ảnh
- Những vấn đề cơ bản trong xử lý ảnh
- Giới thiệu về Raspberry pi 3
- Giới thiệu về Camera Pi
- Giới thiệu về Arduino Uno
- Giới thiệu về cảm biến
- Giới thiệu về động cơ DC
- Giới thiệu về động cơ servo
- Giới thiệu về hệ thống băng tải
- Giới thiệu hệ điều hành trên Raspberry pi 3
- Giới thiệu ngôn ngữ Python và thư viện OPENCV
- Giới thiệu về các chuẩn giao tiếp
- Phương pháp phân loại sản phẩm theo màu sắc

CHƯƠNG 1: TỔNG QUAN

Chương 3: Tính toán và thiết kế

- Giới thiệu
- Tính toán và thiết kế

Chương 4: Thi công hệ thống

- Giới thiệu
- Lập trình hệ thống
- Lưu đồ điều khiển
- Nạp code và chạy thử nghiệm

Chương 5: Kết quả, nhận xét, đánh giá

- Kết quả nhận dạng và phân loại sản phẩm
- Nhận xét, đánh giá mô hình hệ thống

Chương 6: Kết luận và hướng phát triển

- Kết luận
- Những hạn chế của đề tài
- Hướng phát triển

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 mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, 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:

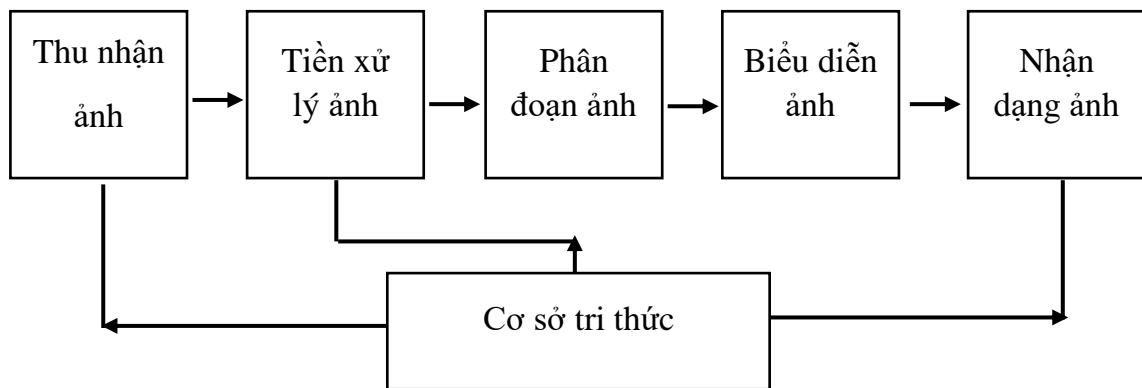
- Trong lĩnh vực quân sự: xử lý và nhận dạng ảnh quân sự.
- Trong lĩnh vực giao tiếp người máy: nhận dạng ảnh, xử lý âm thanh, đồ họa.
- Trong lĩnh vực an, bảo mật: nhận diện khuôn mặt người, nhận diện vân tay, mẫu mắt, ...
- Trong lĩnh vực giải trí: trò chơi điện tử.
- Trong lĩnh vực y tế: Xử lý ảnh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

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền từ Luân đôn đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh số được thuận lợi hơn. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ-ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu được nhiều kết quả khả quan hơn.^[1]

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

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

chụp ảnh). Trước đây, ảnh thu qua Camera là các ảnh tương tự (loại Camera ống kiểu 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. Một khía ảnh có thể được quét từ vè tinh chụp trực tiếp bằng máy quét ảnh^[1]. Hình 2.1 dưới đây mô tả các bước cơ bản trong xử lý ảnh.



Hình 2.1: Các bước cơ bản trong xử lý ảnh

Sơ đồ này bao gồm các thành phần sau:

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 chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã số hóa (như loại CCD – Change Coupled Device) là loại photodiode tạo cường độ sáng tại mỗi điểm ảnh.^[1]

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 nhiều 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.^[1]

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

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.^[1]

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 Extraction) 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.^[1]

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.

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 (Text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người,...^[1]

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.^[1]

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.^[1]

2.2 NHỮNG VẤN ĐỀ TRONG XỬ LÝ ẢNH

2.2.1 Điểm ảnh (Picture Element)

Là đơn vị cơ bản nhất để tạo nên một bức ảnh kỹ thuật số. Địa chỉ của điểm ảnh được xem như là một tọa độ (x,y) nào đó. Một bức ảnh kỹ thuật số, có thể được tạo ra bằng cách chụp hoặc bằng một phương pháp đồ họa nào khác, được tạo nên từ hàng ngàn hoặc hàng triệu pixel riêng lẻ. Bức ảnh càng chứa nhiều pixel thì càng chi tiết. Một triệu pixel thì tương đương với 1 megapixel.^[1]

2.2.2 Ảnh số

Ảnh số là tập hợp hữu hạn các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật. Số điểm ảnh xác định độ phân giải của ảnh. Ảnh có độ phân giải càng cao thì càng thể hiện rõ nét các đặt điểm của tấm hình càng làm cho tấm ảnh trở nên thực và sắc nét hơn. Một hình ảnh là một tín hiệu hai chiều, nó được xác định bởi hàm toán học $f(x, y)$ trong đó x và y là hai tọa độ theo chiều ngang và chiều dọc. Các giá trị của $f(x, y)$ tại bất kỳ điểm nào là cung cấp các giá trị điểm ảnh (pixel) tại điểm đó của một hình ảnh.^[1]

2.2.3 Phân loại ảnh

Mức xám của điểm ảnh là cường độ sáng, gán bằng một giá trị tại điểm đó. Các mức ảnh xám thông thường: 16, 32, 64, 128, 256. Mức được sử dụng thông dụng nhất là 256, tức là dùng 1 byte để biểu diễn mức xám. Trong đó:

- Ảnh nhị phân: Là ảnh có 2 mức trắng và đen, chỉ có 2 giá trị 0 và 1 và chỉ sử dụng 1 bit dữ liệu trên 1 điểm ảnh
- Ảnh đen trắng: Là ảnh có hai màu đen, trắng (không chứa màu khác) với mức xám ở các điểm ảnh có thể khác nhau.
- Ảnh màu: Là ảnh kết hợp của 3 màu cơ bản lại với nhau để tạo ra một thế giới màu sinh động. Người ta thường dùng 3byte để mô tả mức màu, tức là có

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

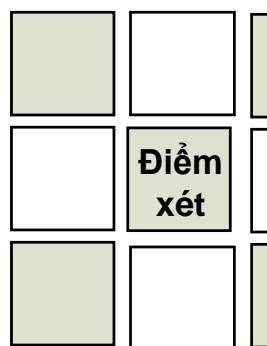
khoảng 16,7 triệu mức màu.^[1]

2.2.4 Quan hệ giữa các điểm ảnh

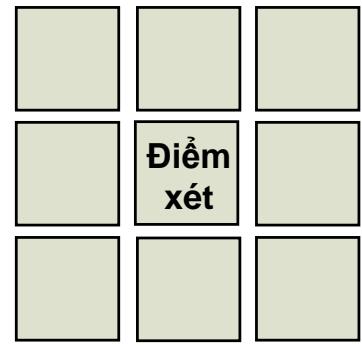
Lân cận điểm ảnh: được nói một cách hài hước như là hàng xóm của cái điểm ảnh. Có 2 loại lân cận cơ bản là lân cận 4 và lân cận 8.



Lân cận 4 hàng cột



Lân cận 4 chéo



Lân cận 8

Hình 2.2: Lân cận 4 lân cận 8

4 điểm ảnh lân cận 4 theo cột và hàng với tọa độ lần lượt là $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$ ký hiệu là tập $N4(p)$. 4 điểm ảnh lân cận 4 theo đường chéo có tọa độ lần lượt là $(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$ ký hiệu là tập $ND(p)$. Tập 8 điểm ảnh lân cận 8 là hợp của 2 tập trên:

$$N8(p) = N4(p) + ND(p) \quad (2.1)$$

Liên kết ảnh: Các mối liên kết của ảnh được xem như là mối liên kết của 2 điểm ảnh gần nhau, có 3 loại liên kết: liên kết 4, liên kết 8, liên kết m (liên kết hỗn hợp). Trong ảnh đa mức xám, ta có thể đặt V chứa nhiều giá trị như $V=\{\text{tập con}\}$. Cho p có tọa độ (x, y) .^[1]

Liên kết 4: hai điểm ảnh p và q có giá trị thuộc về tập V được gọi là liên kết 4 của nhau nếu q thuộc về tập $N4(p)$.

Liên kết 8: hai điểm ảnh p và q có giá trị thuộc về tập V được gọi là liên kết 8 của nhau nếu q thuộc về tập $N8(p)$.

Liên kết m: hai điểm ảnh p và q có giá trị thuộc về tập V được gọi là liên kết M của nhau nếu thỏa 1 trong 2 điều kiện sau: q thuộc về tập $N4(p)$, q thuộc về tập $ND(p)$ và giao của hai tập $N4(p)$, $N4(q)$ không chứa điểm ảnh nào có giá trị thuộc V .^[1]

2.2.5 Lọc nhiễu

Ảnh thu nhận được thường sẽ bị nhiễu nên cần phải loại bỏ nhiễu. Các toán tử không gian dùng trong kỹ thuật tăng cường ảnh được phân nhóm theo công dụng:

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

làm trơn nhiễu, nỗi biên. Để làm trơn nhiễu hay tách nhiễu, người ta sử dụng các bộ lọc tuyến tính (lọc trung bình, thông thấp) hoặc lọc phi tuyến (trung vị, giả trung vị, lọc đồng hình). Từ bản chất của nhiễu (thường tương ứng với tần số cao) và từ cơ sở lý thuyết lọc là: bộ lọc chỉ cho tín hiệu có tần số nào đó thông qua, để lọc nhiễu người ta thường dùng lọc thông thấp (theo quan điểm tần số không gian) hay lấy tổ hợp tuyến tính để san bằng (lọc trung bình). Để làm nỗi cạnh (ứng với tần số cao), người ta dùng các bộ lọc thông cao, lọc Laplace. Phương pháp lọc nhiễu. Chia làm 2 loại: lọc tuyến tính, lọc phi tuyến.^[1]

Làm trơn nhiễu bằng lọc tuyến tính: Khi chụp ảnh có thể xuất hiện nhiều loại nhiễu vào quá trình xử lý ảnh, nên ta cần phải lọc nhiễu. Gồm các phương pháp cơ bản lọc trung bình, lọc thông thấp,... Ví dụ lọc trung bình: Với lọc trung bình, mỗi điểm ảnh được thay thế bằng trung bình trọng số của các điểm lân cận.^[1]

Làm trơn nhiễu bằng lọc phi tuyến: Các bộ lọc phi tuyến cũng hay được dùng trong kỹ thuật tăng cường ảnh. Một số phương pháp lọc cơ bản bộ lọc trung vị, lọc ngoài,... Với lọc trung vị, điểm ảnh đầu vào sẽ được thay thế bởi trung vị các điểm ảnh còn lọc giả trung vị sẽ dùng trung bình cộng của hai giá trị “trung vị” (trung bình cộng của max và min).^[1]

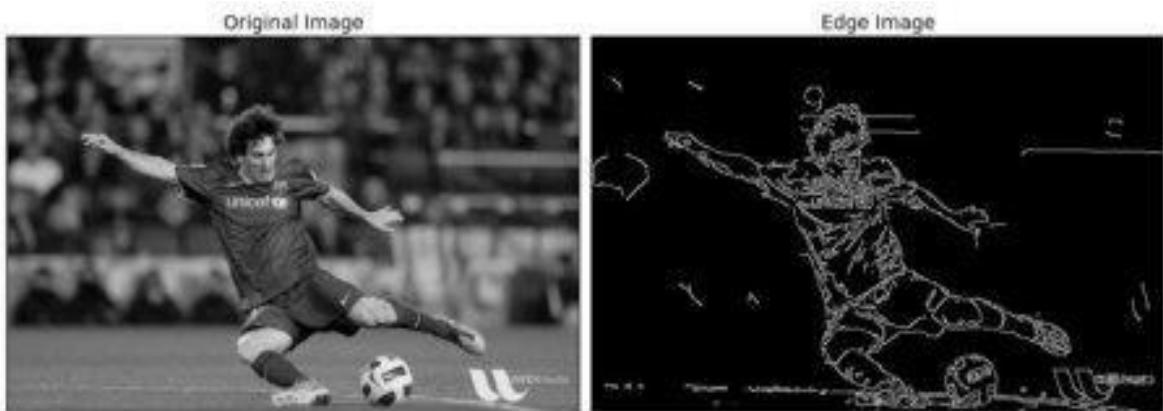
Lọc trung vị: Kỹ thuật này đòi hỏi giá trị các điểm ảnh trong cửa sổ phải xếp theo thứ tự tăng hay giảm dần so với giá trị trung vị. Kích thước cửa sổ thường được chọn sao cho số điểm ảnh trong cửa sổ là lẻ.^[1]

Lọc ngoài: Giả thiết có ngưỡng nào đó cho các mức nhiễu (có thể dựa vào lược đồ xám). Tiến hành so sánh giá trị độ xám của một điểm ảnh với trung bình số học 8 lân cận của nó. Nếu sai lệch lớn hơn ngưỡng, điểm ảnh này được coi như nhiễu. Trong trường hợp đó, thay thế giá trị của điểm ảnh bằng giá trị trung bình 8 lân cận vừa tính được.^[1]

2.2.6 Phương pháp phát hiện biên

Biên là một trong những vấn đề ta cần quan tâm trong xử lý ảnh. Vì ở giai đoạn phân đoạn ảnh chủ yếu dựa vào biên.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.3: Hình tách biên

Điểm biên: Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu). Ví dụ trong ảnh nhị phân, điểm đen gọi là điểm biên nếu lân cận nó có ít nhất một điểm trắng. **Đường biên (đường bao: boundary):** tập hợp các điểm biên liên tiếp tạo thành một đường biên hay đường bao.^[1]

Ý nghĩa của đường biên trong xử lý: ý nghĩa đầu tiên của đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh. Thứ hai, người ta sử dụng biên làm phân cách các vùng xám (màu) cách biệt. Ngược lại, người ta cũng sử dụng các vùng ảnh để tìm đường phân cách. Tầm quan trọng của biên: để thấy rõ tầm quan trọng của biên, xét ví dụ sau: khi người họa sỹ muốn vẽ một danh nhân, họa sỹ chỉ cần vẽ vài đường nứt tóc họa mà không cần vẽ một cách đầy đủ.^[1]

Như vậy, phát hiện biên một cách lý tưởng là phát hiện được tất cả các đường biên trong các đối tượng. Định nghĩa toán học của biên ở trên là cơ sở cho các kỹ thuật phát hiện biên. Điều quan trọng là sự biến thiên giữa các điểm ảnh thường nhỏ, trong khi đó biến thiên độ sáng của điểm biên thường là khá lớn khi qua biên. Xuất phát cơ sở này người ta thường sử dụng hai phương pháp phát hiện biên như sau: Tách biên theo đạo hàm bậc một: Có 2 phương pháp cơ bản là: một là tạo gradient của hai hướng và trực giao trong ảnh, hai là dùng tập đạo hàm có hướng. Tách biên theo đạo hàm bậc hai: được thực hiện trên một số dạng vi phân bậc 2 để làm xuất hiện biên. Có hai dạng của phương pháp đạo hàm bậc hai đã được nghiên cứu là: phương pháp Laplace và đạo hàm trực tiếp.^[1]

Bộ tách biên Canny: Phương pháp phát hiện này được sử dụng phổ biến vì nó có nhiều ưu điểm hơn các phương pháp khác. Các bước thực hiện: Làm phẳng dùng bộ lọc Gauss; Sau đó, Gradient cục bộ của biên độ và hướng được tính. Tìm điểm

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

ảnh có biên độ lớn nhất dùng kỹ thuật nonmaximal suppression; Các điểm ảnh đỉnh (tìm được từ bước 2) được chia làm hai ngưỡng T1 và T2, $T1 < T2$. Các điểm ảnh đỉnh có giá trị lớn hơn T2 được gọi là Strong và nằm trong khoảng T1 và T2 được gọi là Weak. Liên kết các điểm ảnh Weak có 8 kết nối đến điểm ảnh Strong.^[1]

Phương pháp gradient: Gradient là một vector có các thành phần biểu thị tốc độ thay đổi giá trị của điểm ảnh theo 2 hướng x và y, hay có thể nói là nó đại diện cho sự thay đổi về hướng và độ lớn của một vùng ảnh.^[1]

$$\frac{\partial(x,y)}{\partial x} = f'x = \Delta x \approx \frac{f(x+dx,y) - f(x,y)}{dx} \quad (2.2)$$

$$\frac{\partial(x,y)}{\partial y} = f'y = \Delta y \approx \frac{f(x,y+dy) - f(x,y)}{dy} \quad (2.3)$$

Với dx, dy lần lượt là khoảng cách giữa 2 điểm lân cận theo chiều x và chiều y. (Ta có thể xem dx, dy là số lượng điểm ảnh giữa 2 điểm).^[1]

$$\Delta x = f(i+1,j) - f(i,j) \quad (2.4)$$

$$\Delta y = f(i,j+1) - f(i,j) \quad (2.5)$$

Nếu định nghĩa g1 ($f'x$), g2 ($f'y$) là Gradient (vector gradient thành G(g1,g2)) theo hai hướng x, y tương ứng thì biên độ (tức độ lớn) g(m,n) và hướng của biên tại điểm (m,n) được tính như sau:

$$A_0 = g(m,n) = \sqrt{g_1^2(m,n) + g_2^2(m,n)} \quad (2.6)$$

$$\theta = \arctan \frac{g_2}{g_1} \quad (2.7)$$

Việc tính xấp xỉ đạo hàm bậc nhất ($f'x$ và $f'y$ ở đây là g1 và g2) theo các hướng x và y được thực hiện thông qua 2 mặt nạ nhân chập (toán tử đạo hàm), tương ứng sẽ cho ta các kỹ thuật phát hiện biên khác nhau tương ứng với mặt nạ mà nó sử dụng (Roberts, Sobel, Prewitt,...).^[1]

2.2.7 Phân đoạn ảnh

Phân đoạn ảnh là bước then chốt trong xử lý ảnh. Giai đoạn này nhằm phân tích ảnh thành các vùng có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng mức xám, cùng màu hay cùng độ nhón.^[1]

Quá trình phân đoạn ảnh nhằm tách đối tượng cần khảo sát ra khỏi phần nội

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

dung còn lại của ảnh, hay phân chia các đối tượng trong ảnh thành những đối tượng riêng biệt. Như vậy quá trình phân đoạn ảnh là quá trình giảm bớt số lượng thông tin trong ảnh và chỉ giữ lại những thông tin cần thiết cho ứng dụng. Do đó phân đoạn ảnh là quá trình loại bỏ các đối tượng không quan tâm trong ảnh. Có nhiều phương pháp phân đoạn ảnh khác nhau. Trong đó quá trình phân đoạn ảnh sử dụng một ngưỡng giá trị xám để phân đoạn ảnh ra thành các đối tượng và nền là phương pháp đơn giản nhất. Lúc này các điểm ở bên dưới ngưỡng giá trị xám thuộc về nền còn những điểm ảnh ở bên trên ngưỡng giá trị xám thuộc về đối tượng. Phương pháp phân đoạn ảnh này hiệu quả lớn đối với ảnh nhị phân, văn bản in hay đồ họa... [1] Dựa vào đặc tính vật lý của vùng ảnh, các kỹ thuật phân đoạn vùng có thể được chia làm 3 loại:

- Các kỹ thuật cục bộ: dựa trên các đặc tính cục bộ của các điểm ảnh và các lân cận của nó
- Các kỹ thuật tổng thể: phân đoạn một ảnh dựa trên cơ sở của thông tin lấy từ tổng thể như sử dụng biểu đồ mức xám histogram.
- Các kỹ thuật chia, nối và phát triển: dựa trên các khái niệm tương đồng về hình dạng và tính đồng nhất. Hai vùng có thể được nối lại với nhau và liền kề bên nhau. Các vùng không đồng nhất có thể được chia thành các vùng nhỏ. Một vùng có thể được phát triển bằng các nối các điểm ảnh sao cho nó đồng nhất với nhau.^[1]

2.2.8 Các phép toán hình thái Morphology

Khái niệm Morphology trong xử lý ảnh số khởi nguồn từ một ngành của sinh học, nghiên cứu về hình thể và cấu trúc của động thực vật. Đây là một công cụ giúp rút trích các thành phần trong ảnh nhị phân, biểu diễn và mô tả chúng dưới dạng các vùng hoặc dạng như các đường biên, xương và bao lồi. Kỹ thuật morphology cũng được áp dụng trên ảnh xám cho các công đoạn tiền/hậu xử lý ảnh (pre or post processing). Các phép toán đầu tiên trong Morphology được phát triển vào những năm 1964 bởi Georges Matheron (1930 - 2000) và Jean Serra (1940) tại trường đại học École des Mines de Paris, Pháp. Matheron là tiến sĩ hướng dẫn của Jean, họ cố gắng xác định số lượng đặc tính của khoáng sản thông qua “thin cross section” và công việc cho ra kết quả là một phương pháp tiếp cận mới, cũng như sự tiến bộ trong hình học tích phân và tô pô. Từ đó đến hết năm 1970, Morphology xử lý cơ bản với các ảnh nhị phân, tạo ra các phép toán và kỹ thuật như: Dilation, Erosion, Opening,

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Closing....^[1]

Phép giãn (Dilation): Phép toán Dilation là thao tác giãn nở/phình to các đối tượng ảnh đơn sắc.^[1]



Hình 2.4: Phép giãn

Phép co (Erosion): Phép toán Erosion là thao tác xói mòn/co hẹp các đối tượng ảnh đơn sắc. Nếu như phép dãn có thể nói là thêm điểm ảnh vào trong đối tượng ảnh, làm cho đối tượng ảnh trở nên lớn hơn thì phép co sẽ làm cho đối tượng ảnh trở nên nhỏ hơn, ít điểm ảnh.^[1]



Hình 2.5: Phép co

a. Những định dạng của ảnh

Ảnh thu nhận được sau quá trình số hóa thường được lưu lại cho các quá trình xử lý tiếp theo hay truyền đi. Trong quá trình truyền của kỹ thuật xử lý ảnh, tồn tại nhiều định dạng khác nhau từ ảnh đen trắng như định dạng IMG, ảnh đa cấp xám cho đến ảnh màu (BMP,JPEG,GIF).^[1]

b. Các phần mềm hỗ trợ xử lý ảnh

Hiện nay xử lý ảnh được giảng dạy trường đại học và ứng dụng vào thực tế rất nhiều như các phần mềm chỉnh sửa hình ảnh hay nhận biết khuôn mặt. Chính vì thế có rất nhiều công cụ để chúng ta lập trình ứng dụng vào thực tế, Như phải kể đến

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Matlap, hay ngôn ngữ Python,...^[1]

2.3 GIỚI THIỆU RASPBERRY PI 3

2.3.1 Giới thiệu



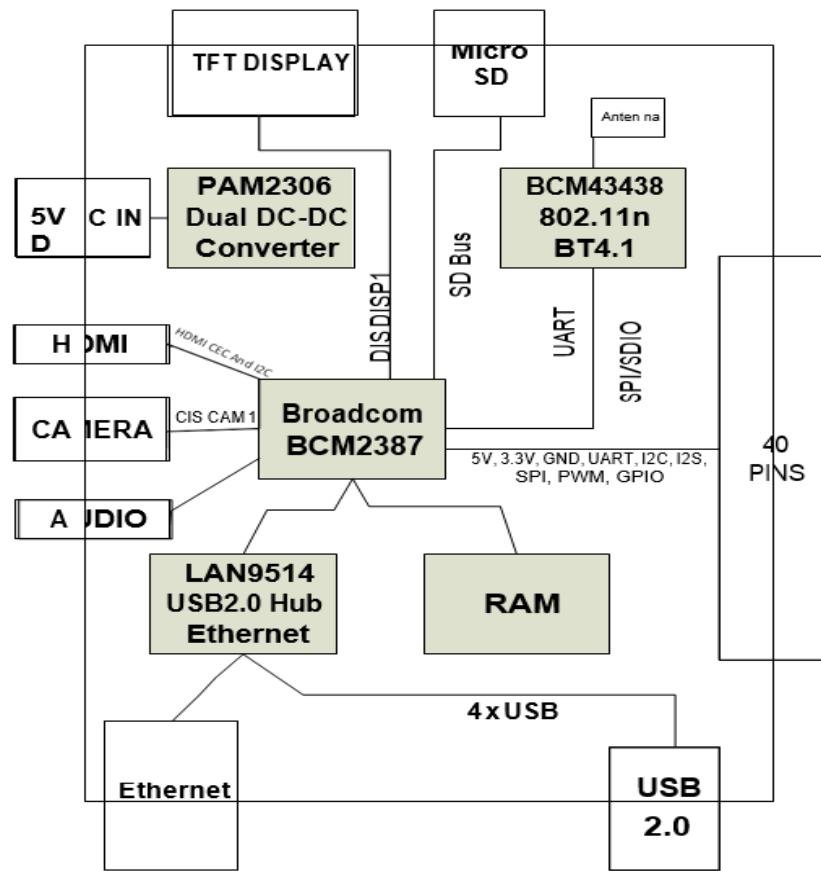
Hình 2.6: Raspberry Pi 3

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 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.^[1]

Raspberry Pi xây dựng xoay quanh bộ xử lý 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 đây tài liệu này sử dụng Kit Raspberry Pi 3 Model B vì thông dụng hơn.^[1]

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.3.2 Thông tin cấu hình Raspberry Pi 3



Hình 2.7: Sơ đồ Raspberry Pi 3 Module B

Thông tin cấu hình Raspberry Pi 3:

- 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
 - 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
 - x USB 2.0 ports
 - 40 GPIO pins
 - Chip antenna

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Kết nối hiển thị DSI
- Khe cắm thẻ nhớ microsd
- Kích thước: 85 x 56 x 17 mm^[1]

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,...

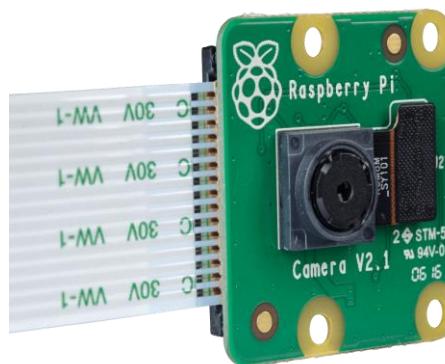
2.4 GIỚI THIỆU VỀ CAMERA PI

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 dài cần thiết trong khi vẫn đảm bảo tốc độ truyền hình ảnh từ module về RPi.^[1]

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).^[1]



Hình 2.8: Camera Pi v2.1

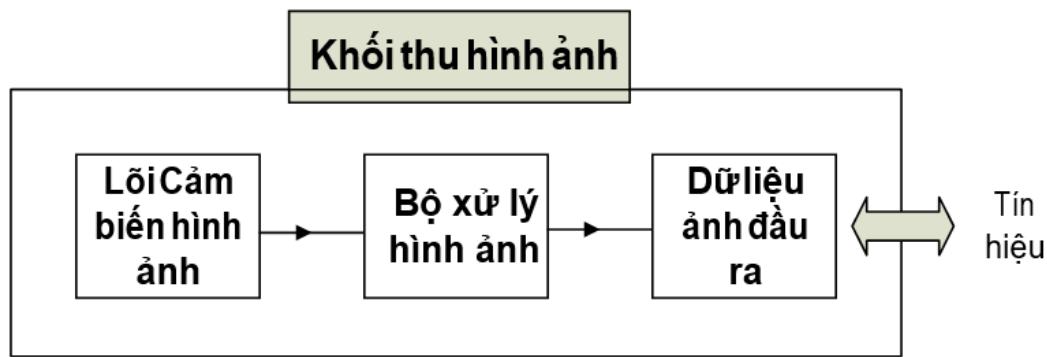
Camera Module có thể được sử dụng để quay video độ nét cao, cũng như chụp hình ảnh tĩnh. Nó khá dễ dàng để sử dụng cho người mới bắt đầu, nhưng cũng có rất nhiều giải pháp mở rộng để cung cấp cho người dùng yêu cầu cao. Có rất nhiều demo của người dùng về công dụng của Camera Module như chụp Time-Lapse, Slow-Motion và rất nhiều ứng dụng khác.^[1]

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.^[1]

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^[1]



Hình 2.9: Sơ đồ khái niệm 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.^[1]

2.5 GIỚI THIỆU VỀ ARDUINO UNO R3

2.5.1 Giới thiệu

Arduino Uno R3 là một board mạch vi xử lý tích hợp, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8 bit. Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau.^[2]

2.5.2 Thông tin cấu hình Arduino Uno R3

Bảng 2.1: Bảng thông số Arduino Uno R3

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	khoảng 30mA

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Điện áp vào khuyên dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

Cấu tạo chính của Arduino Uno R3 bao gồm các phần sau:

- Cổng USB: đây là loại cổng giao tiếp để ta nạp code từ PC lên vi điều khiển. Đồng thời nó cũng là giao tiếp serial để truyền dữ liệu giữa vi điều khiển và máy tính.
- Jack nguồn: để chạy Arduino thì có thể lấy nguồn từ cổng USB ở trên hoặc một nguồn từ 9V đến 12V. Với các chân điện như sau:
 - GND (Ground): cực âm của nguồn điện cấp cho Arduino UNO. Khi dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
 - 5V: cáp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
 - 3.3V: cáp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
 - Vin (Voltage Input): để cấp nguồn ngoài cho Arduino UNO, nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
 - IOREF: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này và nó luôn bằng 5V. Mặc dù vậy không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
 - RESET: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở $10K\Omega$.^[2]

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.10: Arduino Uno R3

Arduino UNO R3 có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).^[2]

Một số chân digital có các chức năng đặc biệt như sau:

- Chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này.
- Chân PWM (~): 3, 5, 6, 9, 10, và 11: cho phép xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 → 256-1 tương ứng với 0V → 5V) bằng hàm digitalWrite().
- Chân giao tiếp SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
- LED 13: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được sử dụng LED này sẽ sáng.^[2]

Arduino UNO có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit ($0 \rightarrow 2^{10}-1$) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân AREF trên board ta có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Nếu ta cấp điện áp 2.5V vào chân này thì ta có thể dùng các chân analog để đo điện áp trong khoảng

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

từ 0V → 2.5V với độ phân giải vẫn là 10bit.^[2]

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

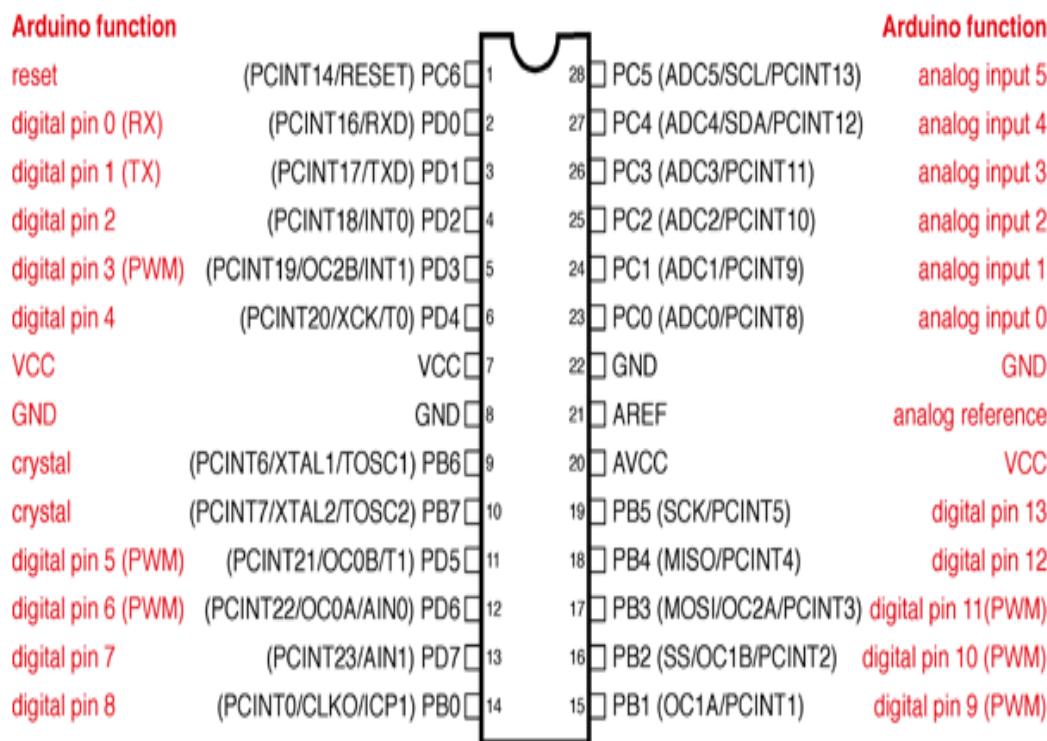
Bảng 2.2: Thông số kỹ thuật: Vi điều khiển ATmega328P

Điện áp hoạt động	5V
Điện áp đầu vào (khuyến nghị)	7-12V
Điện áp đầu vào (giới hạn)	6-20V
Số I/O	14(có 6 chân sử dụng như PWM)
Số ngõ vào Analog	6
Dòng DC trên mỗi I/O:	20mA
Dòng DC cho chân 3,3V	50mA
Bộ nhớ flash	32 KB (0.5 KB nạp bộ khởi động)
SRAM	2 KB
EEPROM	1 KB
Tốc độ xung	16 MHz
Chiều dài	68.6 mm
Chiều rộng	53,4 mm
Cân nặng	25g

Bộ nhớ vi điều khiển Atmega328:

- 32KB bộ nhớ Flash: những đoạn lệnh ta lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển.
- 2KB cho SRAM (Static Random Access Memory): giá trị các biến ta khai báo khi lập trình sẽ lưu ở đây. Khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
- 1KB cho EEPROM : đây giống như một chiếc ổ cứng mini – nơi có thể đọc và ghi dữ liệu vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.^[2]

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.11: Sơ đồ chân ATMega 328P ứng với arduino Uno R3

2.5.3 Ứng dụng

Arduino là một nền tảng mã nguồn mở được sử dụng để xây dựng các ứng dụng điện tử tương tác với nhau hoặc với môi trường được thuận lợi hơn.

Arduino giống như một máy tính nhỏ để người dùng có thể lập trình và thực hiện các dự án điện tử mà không cần phải có các công cụ chuyên biệt để phục vụ việc nạp code.^[2]

Một số ứng dụng của Arduino Uno trong đời sống:

- Làm Robot. Arduino có khả năng đọc các thiết bị cảm biến, điều khiển động cơ,... nên nó thường được dùng để làm bộ xử lý trung tâm của rất nhiều loại robot.
- Game tương tác: Arduino có thể được sử dụng để tương tác với Joystick, màn hình,... khi chơi các game như Tetrix, phá gach, Mario...
- Máy bay không người lái.
- Điều khiển đèn tín hiệu giao thông, làm hiệu ứng đèn Led nhấp nháy trên các biển quảng cáo...
- Điều khiển các thiết bị cảm biến ánh sáng, âm thanh.
- Làm máy in 3D

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Làm đèn bằng ánh sáng
- Làm lò nướng bánh biệt tweet để báo cho bạn khi bánh chín.

Arduino còn rất nhiều ứng dụng hữu ích khác tùy vào sự sáng tạo của người dùng.

2.6 GIỚI THIỆU VỀ CẢM BIẾN HỒNG NGOẠI E18-D80NK

2.6.1 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 báo mong muốn thông qua biến trớ, ngõ ra cảm biến ở dạng cực thu hở nên cần thêm 1 trớ treo lên nguồn ở chân tín hiệu khi sử dụng.^[3]



Hình 2.12: Cảm biến hồng ngoại E18-D80NK

2.6.2 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 trớ.
- Dòng kích ngõ ra: 300mA.
- Ngõ ra dạng NPN cực thu hở giúp biến được điện áp ngõ ra, trớ treo lên áp bao nhiêu sẽ tạo thành điện áp ngõ ra bấy nhiêu.
- Chất liệu sản phẩm: nhựa.
- Có led hiển thị ngõ ra màu đỏ.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

➤ Kích thước: 1.8cm (D) x 7.0cm (L).^[3]

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, cần phải có trở kéo để tạo thành mức cao.^[3]

2.6.3 Ứng dụng

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

2.7 GIỚI THIỆU VỀ ĐỘNG CƠ DC

2.7.1 Giới thiệu

Động cơ một chiều DC (DC là từ viết tắt của "Direct Current Motors") là Động cơ điều khiển bằng dòng có hướng xác định hay nói dễ hiểu hơn thì đây là loại động cơ chạy bằng nguồn điện áp DC- điện áp 1 chiều(Khác với điện áp AC xoay chiều). Đầu dây ra của động cơ thường gồm hai dây (dây nguồn- VCC và dây tiếp đất- GND). DC motor là một động cơ một chiều với cơ năng quay liên tục.^[4]



Hình 2.13: Động cơ DC

2.7.2 Cấu tạo và nguyên lý hoạt động

a. Cấu tạo

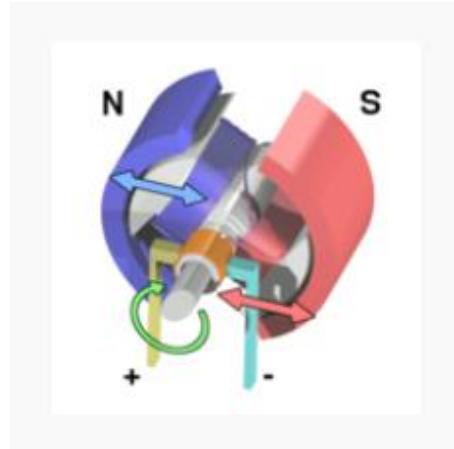
Gồm có 3 phần chính stator(phần cảm), rotor (phần ứng), và phần cỗ gop-

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

chỉnh lưu.

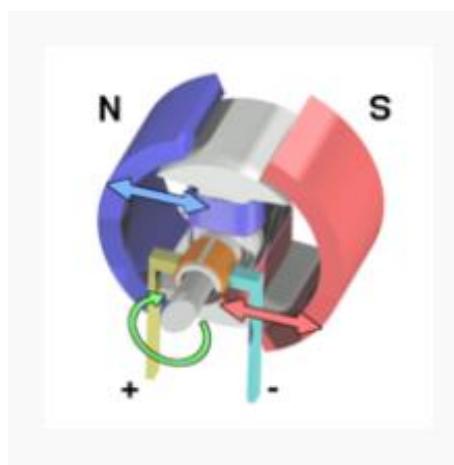
- Stator của động cơ điện 1 chiều thường là 1 hay nhiều cặp nam châm vĩnh cửu, hay nam châm điện.
- Rotor có các cuộn dây quấn và được nối với nguồn điện một chiều.
- Bộ phận chỉnh lưu, nó có nhiệm vụ là đổi chiều dòng điện trong khi chuyển động quay của rotor là liên tục. Thông thường bộ phận này gồm có một bộ cỗ góp và một bộ chổi than tiếp xúc với cỗ góp.^[4]

b. Nguyên lý hoạt động



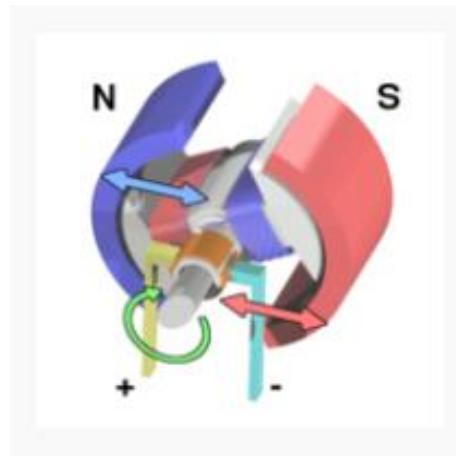
Hình 2.14: Pha 1 động cơ DC

Pha 1: Từ trường của rotor cùng cực với stator, sẽ đẩy nhau tạo ra chuyển động quay của rotor.



Hình 2.15: Pha 2 động cơ DC

Pha 2: Rotor tiếp tục quay



Hình 2.16: Pha 3 động cơ DC

Pha 3: Bộ phận chỉnh điện sẽ đổi cực sao cho từ trường giữa stator và rotor cùng dấu, trở lại pha 1

Nếu trục của một động cơ điện một chiều được kéo bằng 1 lực ngoài, động cơ sẽ hoạt động như một máy phát điện một chiều, và tạo ra một sức điện động cảm ứng Electromotive force (EMF). Khi vận hành bình thường, rotor khi quay sẽ phát ra một điện áp gọi là sức phản điện động counter-EMF (CEMF) hoặc sức điện động đối kháng, vì nó đối kháng lại điện áp bên ngoài đặt vào động cơ. Sức điện động này tương tự như sức điện động phát ra khi động cơ được sử dụng như một máy phát điện (như lúc ta nối một điện trở tải vào đầu ra của động cơ, và kéo trục động cơ bằng một ngẫu lực bên ngoài). Như vậy điện áp đặt trên động cơ bao gồm 2 thành phần: sức phản điện động, và điện áp giáng tạo ra do điện trở nội của các cuộn dây phản ứng. Dòng điện chạy qua động cơ được tính theo biểu thức sau:^[4]

$$I = (V_{\text{nguồn}} - V_{\text{Phản điện động}})/R_{\text{Phản cứng}} \quad (2.8)$$

Công suất cơ mà động cơ đưa ra được, được tính bằng:

$$P = I^*(V_{\text{phản điện động}}) \quad (2.9)$$

2.7.3 Ứng dụng

Ngày nay động cơ điện DC được dùng trong hầu hết mọi lĩnh vực, từ các động cơ nhỏ dùng trong lò vi sóng để chuyển động đĩa quay, hay trong các máy đọc đĩa (máy chơi CD hay DVD), đến các đồ nghề như máy khoan, hay các máy gia dụng như máy giặt, sự hoạt động của thang máy hay các hệ thống thông gió cũng dựa vào động cơ điện. Ở nhiều nước động cơ điện được dùng trong các phương tiện vận chuyển, đặc biệt trong các đầu máy xe lửa.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Trong công nghệ máy tính: Động cơ điện được sử dụng trong các ổ cứng, ổ quang (chúng là các động cơ bước rất nhỏ).

2.8 GIỚI THIỆU VỀ ĐỘNG CƠ SERVO MG996R

2.8.1 Tổng quan về động cơ servo

a. Tổng quan

Động cơ servo đã có từ rất lâu và được sử dụng trong nhiều ứng dụng. Chúng có kích thước nhỏ nhưng đóng gói một cú đấm lớn và rất tiết kiệm năng lượng. Những tính năng này cho phép chúng được sử dụng để vận hành xe ô tô đồ chơi, robot và máy bay điều khiển từ xa. Động cơ servo cũng được sử dụng trong các ứng dụng công nghiệp, robot, sản xuất nội tuyến, dược phẩm và dịch vụ thực phẩm.

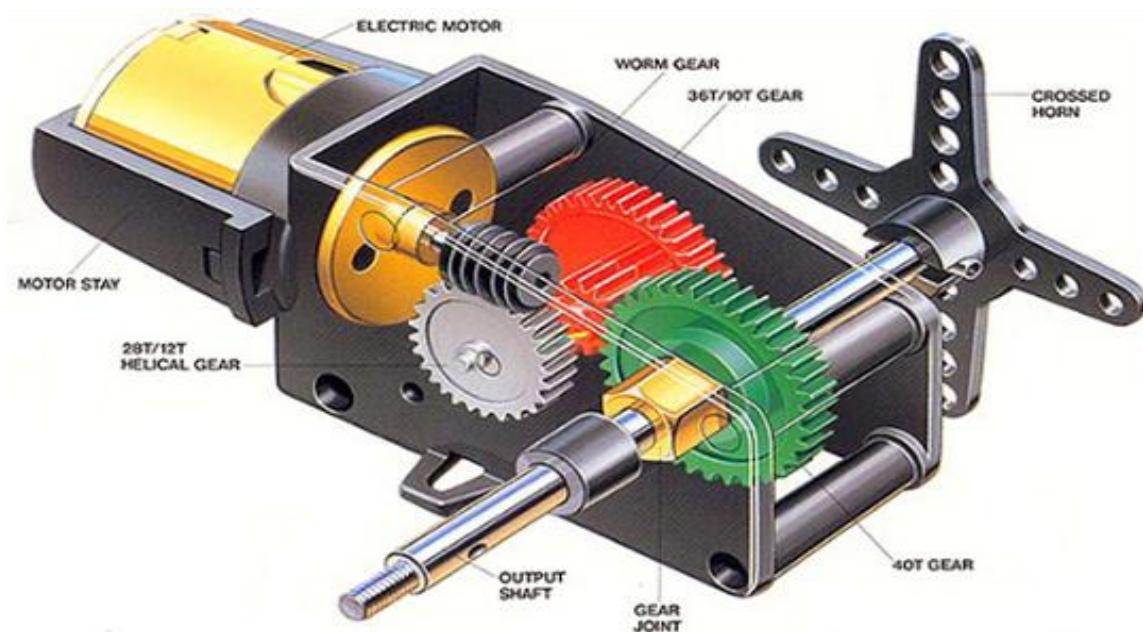


Hình 2.17: Động cơ servo

Động cơ servo có nhiều kiểu dáng và kích thước, được sử dụng trong nhiều máy khác nhau, từ máy tiện điều khiển bằng tay cho tới các mô hình máy bay và xe hơi. Ứng dụng mới nhất của động cơ servo là trong các robot...^[5]

b. Cấu tạo và nguyên lý hoạt động

Bên trong có một thiết lập khá đơn giản: một động cơ DC nhỏ, chiết áp và mạch điều khiển. Động cơ được gắn bởi bánh răng vào bánh xe điều khiển. Khi động cơ quay, điện trở của chiết áp thay đổi, do đó mạch điều khiển có thể điều chỉnh chính xác mức độ chuyển động có và theo hướng nào.^[5]



Hình 2.18: Cấu tạo bên trong động cơ servo

Khi trục của động cơ ở vị trí mong muốn, nguồn điện cung cấp cho động cơ sẽ dừng lại. Nếu không, động cơ được quay theo hướng thích hợp. Vị trí mong muốn được gửi qua các xung điện qua dây tín hiệu. Tốc độ của động cơ tỷ lệ thuận với sự khác biệt giữa vị trí thực tế và vị trí mong muốn. Vì vậy, nếu động cơ ở gần vị trí mong muốn, nó sẽ quay chậm, nếu không nó sẽ quay nhanh. Điều này được gọi là kiểm soát tỷ lệ.

Động cơ servo được điều khiển bằng cách gửi một xung điện có độ rộng thay đổi hoặc điều chỉnh độ rộng xung (PWM), thông qua dây điều khiển. Có một xung tối thiểu, một xung tối đa và tốc độ lặp lại. Một động cơ servo thường chỉ có thể quay 90° theo một trong hai hướng cho tổng chuyển động 180° . Vị trí trung tính của động cơ được xác định là vị trí mà servo có cùng số vòng quay tiềm năng theo cả chiều kim đồng hồ hoặc ngược chiều kim đồng hồ. PWM được gửi đến động cơ xác định vị trí của trục và dựa trên thời gian của xung được gửi qua dây điều khiển; các rotor sẽ chuyển sang vị trí mong muốn.^[5]

2.8.2 Giới thiệu động cơ servo MG996R

a. Giới thiệu

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

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

được tích hợp sẵn Driver điều khiển động cơ bên trong theo cơ chế phát xung - quay góc nên rất dễ sử dụng.^[6]



Hình 2.19: Động cơ servo MG996R

b. 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
- Lực kéo:
 - 3.5 kg-cm (180.5 ozin) at 4.8V-1.5A
 - 5.5 kg-cm (208.3 ozin) at 6V-1.5A
- Tốc độ quay:
 - 0.17sec / 60 degrees (4.8V no load)
 - 0.13sec / 60 degrees (6.0V no load)^[6]

2.8.3 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 sự chuyển đổi trơn tru và chính xác.

2.9 GIỚI THIỆU HỆ THỐNG BĂNG TẢI

2.9.1 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

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

lao động, số lượng nhân công, giảm thời gian và tăng năng suất lao động.^[7]

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.^[7]



Hình 2.20: Mô hình băng tải

2.9.2 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 dày 1.5mm.
- Động cơ chuyền động: Là động cơ giảm tốc công suất 0.2KW, 0.4KW, 0.75KW, 1.5KW, 2.2KW.
- Bộ điều khiển băng chuyền: Thường gồm có biến tần, sensor, timer, PLC...
- Cơ cấu truyền động gồm có: Rulo kéo, con lăn đỡ, nhông xích...
- Hệ thống bàn thao tác trên băng chuyền thường bằng gỗ, thép hoặc inox trên mặt có dán thảm cao su chống tĩnh điện.
- Hệ thống đường khí nén và đường điện có ổ cắm để lấy điện cho các máy dùng trên băng chuyền.
- Ngoài ra thường có thêm đường điện chiếu sáng.^[7]

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

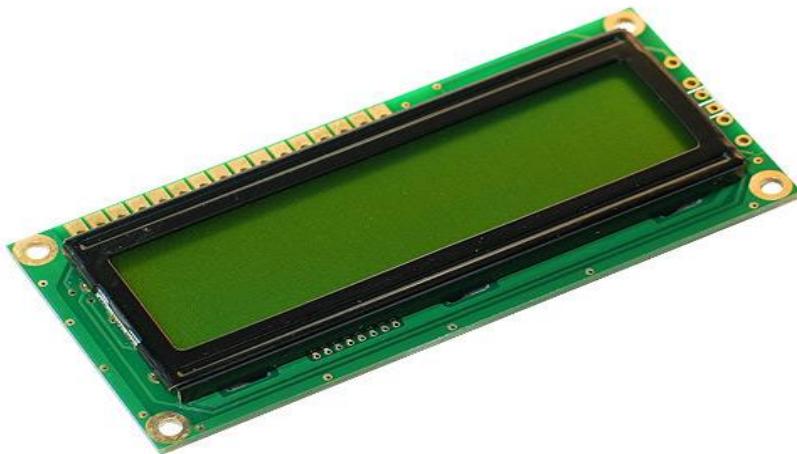
2.9.3 Ứng dụng

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.^[7]

2.10 GIỚI THIỆU VỀ LCD16X2

2.10.1 Giới thiệu

Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của VĐK. LCD có rất nhiều ưu điểm so với các dạng hiển thị khác: Nó có khả năng hiển thị ký tự đa dạng, trực quan (chữ, số và ký tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tốn rất ít tài nguyên hệ thống và giá thành rẻ ...^[8]

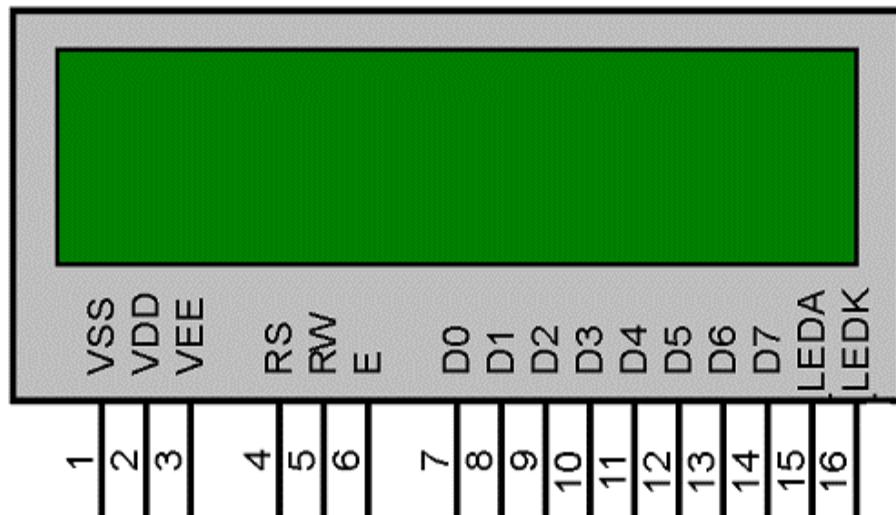


Hình 2.21: LCD16x2

2.10.2 Cấu tạo

LCD 16x2 có 2 hàng, mỗi hàng 16 ký tự.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.22: Sơ đồ chân LCD16x2

Bảng 2.3: Chức năng các chân trên LCD

Chân	Ký hiệu	Mô tả
1	Vss	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển
2	VDD	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển
3	VEE	Điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read) + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.
5	R/W	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.
7 - 14	DB0 - DB7	Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này : + Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

		+ Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7
15	-	Nguồn dương cho đèn nền
16	-	GND cho đèn nền

Trong 16 chân của LCD được chia làm 3 dạng tín hiệu như sau:

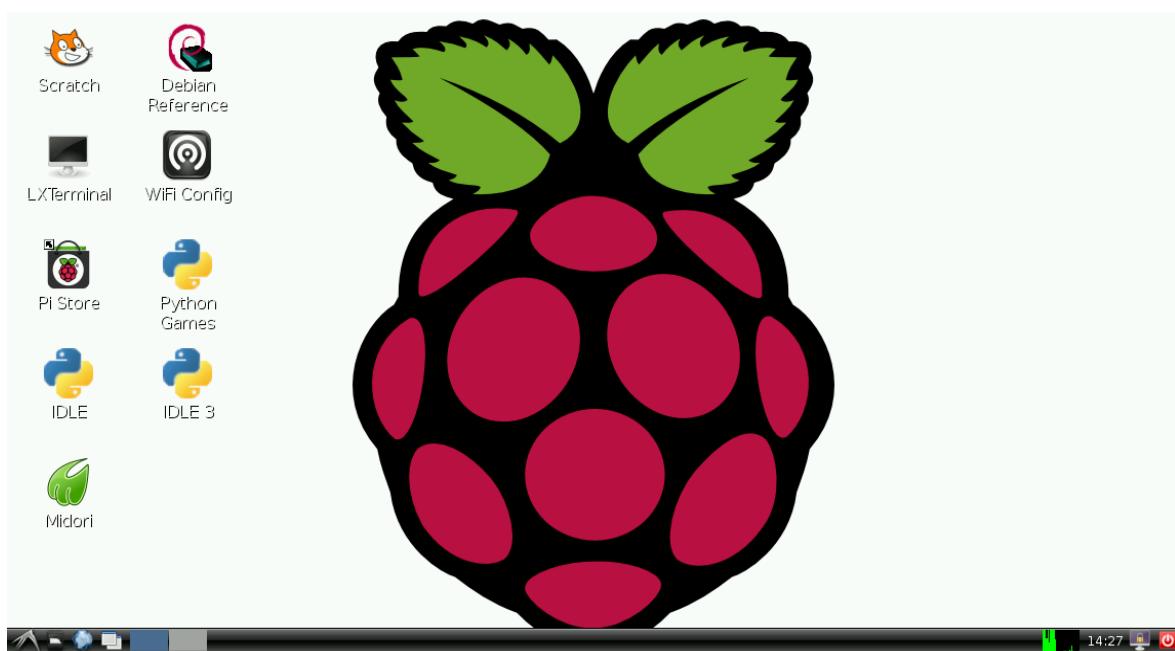
- Các chân cấp nguồn: chân số 1 nối mass (0V), chân số 2 là VDD nối với nguồn 5V, chân số 3 dùng để chỉnh contrast thường nối với biến trở.
- Các chân điều khiển: chân số 4 là chân RS dùng để điều khiển lựa chọn thanh ghi. Chân R/W dùng để điều khiển quá trình đọc và ghi. Chân E là chân cho phép dạng xung chốt.
- Các chân dữ liệu DB0-DB7: là chân từ số 7 đến 14 dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.
- Chân 15 nối nguồn +5V hoặc 4,2V đối với LED, chân 16 nối GND.^[8]

2.10.3 Ứng dụng

LCD thường được sử dụng trong các mạch điện tử, hiển thị thời gian thực, giá trị kết quả, hiệu ứng.

2.11 GIỚI THIỆU HỆ ĐIỀU HÀNH TRÊN RASPBERRY PI

Raspberry Pi có rất nhiều hệ điều hành hỗ trợ như Raspbian, Ubuntu Mate, PiNet, RiscOS,... nhưng trong đó Raspbian là hệ điều hành cơ bản, phổ biến nhất và do chính Raspberry Pi Foundation cung cấp. Nó cũng được hằng khuyến cáo sử dụng, nhất là cho người mới bắt đầu làm quen với Raspberry Pi.^[9]



Hình 2.23: Hệ điều hành Raspbian

Raspbian có dung lượng sau khi giải nén là khoảng gần 4GB, bạn cần tối thiểu 1 cái thẻ 4GB để có thể sử dụng Raspbian. Tuy nhiên, nên sử dụng thẻ tối thiểu 8GB vì bạn cần cài thêm các ứng dụng khác nữa.

Raspbian được hướng đến người dùng có mục đích:

- Sử dụng Raspberry Pi như máy tính văn phòng để lướt web, soạn văn bản, check mail và thi thoảng nghe nhạc/xem phim.
- Nghiên cứu phát triển các thiết bị điều khiển tự động.
- Sử dụng như một máy chủ cung cấp các dịch vụ như web, file server, printer server, ...

Theo đánh giá, Raspbian hoạt động rất ổn định, tốc độ nhanh (đặc biệt là trên Raspberry Pi 3). Nhược điểm của nó là giao diện đơn giản, cổ điển và rất không hào nhoáng. Nếu bạn không quá quan tâm tới giao diện mà hướng đến hiệu năng thì Raspbian rất phù hợp cho bạn.^[9]

2.12 GIỚI THIỆU NGÔN NGỮ PYTHON VÀ THU VIỆN OPENCV

2.12.1 Ngôn ngữ Python

a. Giới thiệu ngôn ngữ Python

Python là một ngôn ngữ lập trình được sử dụng phổ biến ngày nay từ trong môi trường học đường cho tới các dự án lớn. Ngôn ngữ phát triển nhiều loại ứng dụng, phần mềm khác nhau như các chương trình chạy trên desktop, server, lập trình các

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

ứng dụng web... Ngoài ra Python cũng là ngôn ngữ ưa thích trong xây dựng các chương trình trí tuệ nhân tạo trong đó bao gồm machine learning. Ban đầu, Python được phát triển để chạy trên nền Unix, nhưng sau này, nó đã chạy trên mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Python do Guido van Rossum tạo ra năm 1990. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.^[10]

b. Đặt điểm nổi bật của Python

Python là ngôn ngữ có hình thức đơn giản, cú pháp ngắn gọn, sử dụng một số lượng ít các từ khoá, do đó Python là một ngôn ngữ dễ học đối với người mới bắt đầu tìm hiểu. Python là ngôn ngữ có mã lệnh (source code hay đơn giản là code) không mấy phức tạp. Cả trường hợp bạn chưa biết gì về Python bạn cũng có thể suy đoán được ý nghĩa của từng dòng lệnh trong source code. Python có nhiều ứng dụng trên nhiều nền tảng, chương trình phần mềm viết bằng ngôn ngữ Python có thể được chạy trên nhiều nền tảng hệ điều hành khác nhau bao gồm Windows, Mac OSX và Linux.^[10]

2.12.2 Thư viện OPENCV

a. Giới thiệu

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở, nó là miễn phí cho những ai bắt đầu tiếp cận với các học thuật. OpenCV được ứng dụng trong nhiều lĩnh vực như cho thị giác máy tính hay xử lý ảnh và máy học. Thư viện được lập trình trên các ngôn ngữ cấp cao: C++, C, Python, hay Java và hỗ trợ trên các nền tảng Window, Linux, Mac OS, iOS và Android. OpenCV đã được tạo ra tại Intel vào năm 1999 bởi Gary Bradsky, và ra mắt vào năm 2000. Opencv có rất nhiều ứng dụng: Nhận dạng ảnh, xử lý hình ảnh, phục hồi hình ảnh/video, thực tế ảo,... Ở đề tài này thư viện OpenCV được chạy trên ngôn ngữ Python. OpenCV được dùng làm thư viện chính để xử lý hình ảnh đầu vào và sau đó đi nhận dạng ảnh.^[10]

b. Đặt điểm

OpenCV Là một thư viện mở nên sử dụng các thuật toán một cách miễn phí,

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

cùng với việc chúng ta cũng có thể đóng góp thêm các thuật toán giúp Thư viện thêm ngày càng phát triển.^[10]

Các tính năng của thư viện OpenCV:

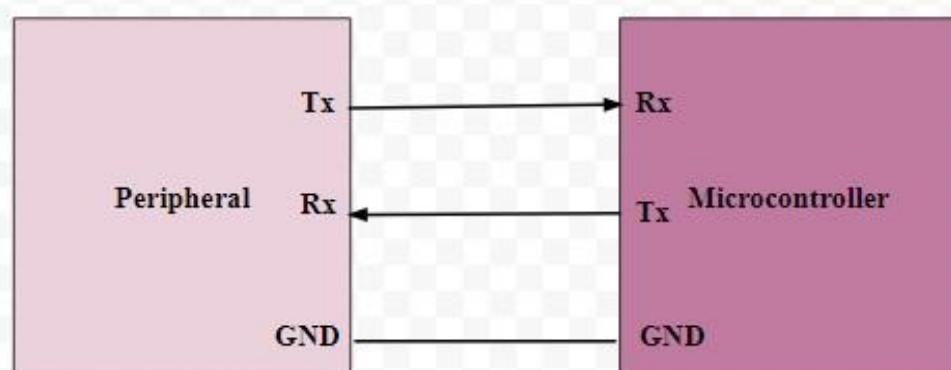
- Đối với hình ảnh, chúng ta có thể đọc và lưu hay ghi chúng.
- Về Video cũng tương tự như hình ảnh cũng có đọc và ghi.
- Xử lý hình ảnh có thể lọc nhiều cho ảnh, hay chuyển đổi ảnh.
- Thực hiện nhận dạng đặc điểm của hình dạng trong ảnh.
- Phát hiện các đối tượng xác định được xác định trước như khuôn mặt, mắt, xe trong video hoặc hình ảnh.
- Phân tích video,... ước lượng chuyển động của nó, trừ nền ra và theo dõi các đối tượng trong video.^[10]

2.13 GIỚI THIỆU VỀ CÁC CHUẨN GIAO TIẾP

2.13.1 Chuẩn giao tiếp Uart

a. Giới thiệu

Các tên đầy đủ UART là “Universal Asynchronous Receiver / Transmitter”, và nó là một vi mạch sẵn có trong một vi điều khiển nhưng không giống như một giao thức truyền thông (I2C & SPI). Chức năng chính của UART là truyền dữ liệu nối tiếp. Trong UART, giao tiếp giữa hai thiết bị có thể được thực hiện theo hai cách là giao tiếp dữ liệu nối tiếp và giao tiếp dữ liệu song song.



UART Communication

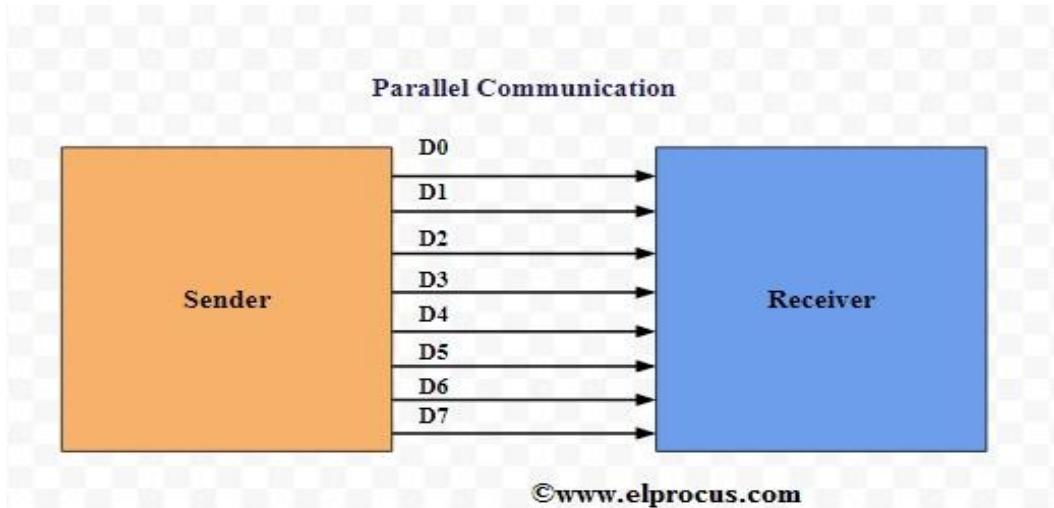
Hình 2.24: Giao tiếp Uart

Trong giao tiếp dữ liệu nối tiếp, dữ liệu có thể được truyền qua một cáp hoặc một đường dây ở dạng bit-bit và nó chỉ cần hai cáp. Truyền thông dữ liệu nối tiếp không đắt khi chúng ta so sánh với giao tiếp song song. Nó đòi hỏi rất ít mạch cũng

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

như dây. Vì vậy, giao tiếp này rất hữu ích trong các mạch ghép so với giao tiếp song song.^[11]

Trong giao tiếp dữ liệu song song, dữ liệu có thể được truyền qua nhiều cáp cùng một lúc. Truyền dữ liệu song song tốn kém nhưng rất nhanh, vì nó đòi hỏi phần cứng và cáp bổ sung. Các ví dụ tốt nhất cho giao tiếp này là máy in cũ, PCI, RAM, v.v.^[11]

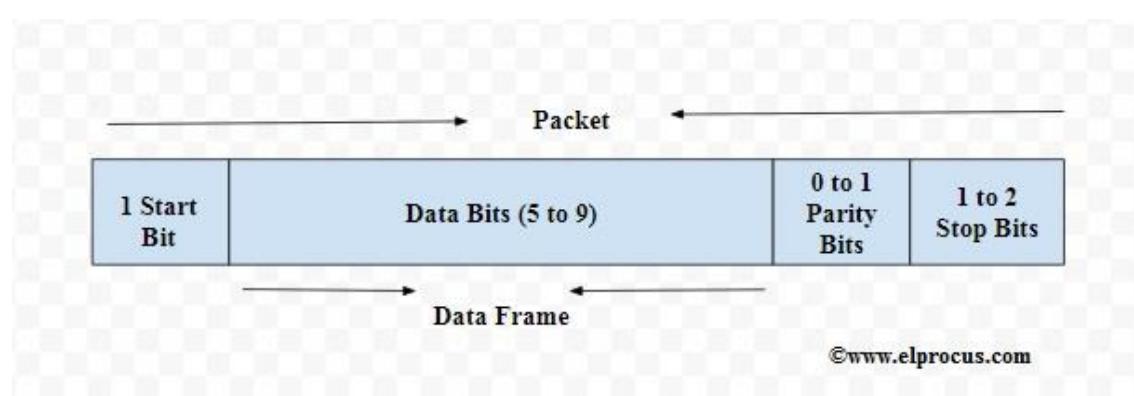


Hình 2.25: Giao tiếp song song

b. Truyền thông UART

Trong giao tiếp này, có hai loại UART có sẵn là truyền UART và nhận UART và giao tiếp giữa hai loại này có thể được thực hiện trực tiếp với nhau. Đối với điều này, chỉ cần hai cáp để giao tiếp giữa hai UART. Luồng dữ liệu sẽ từ cả hai chân truyền (Tx) và nhận (Rx) của UARTs. Trong UART, việc truyền dữ liệu từ Tx UART sang Rx UART có thể được thực hiện không đồng bộ (không có tín hiệu CLK để đồng bộ hóa các bit 0 / p).^[11]

Việc truyền dữ liệu của UART có thể được thực hiện bằng cách sử dụng bus dữ liệu ở dạng song song bởi các thiết bị khác như vi điều khiển, bộ nhớ, CPU, v.v. Sau khi nhận được dữ liệu song song từ bus, nó tạo thành gói dữ liệu bằng cách thêm ba bit như bắt đầu, dừng lại và trung bình. Nó đọc từng bit gói dữ liệu và chuyển đổi dữ liệu nhận được thành dạng song song để loại bỏ ba bit của gói dữ liệu. Tóm lại, gói dữ liệu nhận được bởi UART chuyển song song về phía bus dữ liệu ở đầu nhận.^[11]



Hình 2.26: Truyền thông Uart

Start-bit

Start-bit còn được gọi là bit đồng bộ hóa được đặt trước dữ liệu thực tế. Nói chung, một đường truyền dữ liệu không hoạt động được điều khiển ở mức điện áp cao. Để bắt đầu truyền dữ liệu, truyền UART kéo đường dữ liệu từ mức điện áp cao (1) xuống mức điện áp thấp (0). UART thu được thông báo sự chuyển đổi này từ mức cao sang mức thấp qua đường dữ liệu cũng như bắt đầu hiểu dữ liệu thực. Nói chung, chỉ có một start-bit.^[11]

Bit dừng

Bit dừng được đặt ở phần cuối của gói dữ liệu. Thông thường, bit này dài 2 bit nhưng thường chỉ sử dụng 1 bit. Để dừng sóng, UART giữ đường dữ liệu ở mức điện áp cao.^[11]

Bit chẵn lẻ

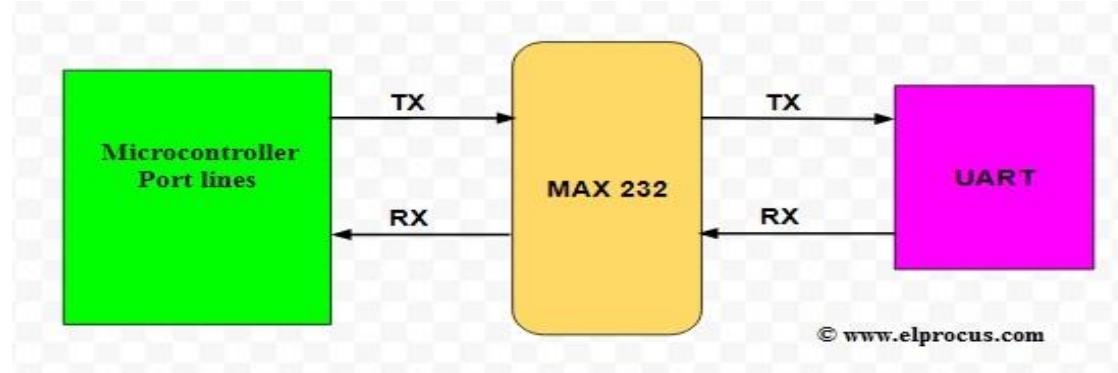
Bit chẵn lẻ cho phép người nhận đảm bảo liệu dữ liệu được thu thập có đúng hay không. Đây là một hệ thống kiểm tra lỗi cấp thấp & bit chẵn lẻ có sẵn trong hai phạm vi như Chẵn lẻ – chẵn lẻ cũng như Chẵn lẻ – lẻ. Trên thực tế, bit này không được sử dụng rộng rãi nên không bắt buộc.^[11]

Dữ liệu bit hoặc khung dữ liệu

Các bit dữ liệu bao gồm dữ liệu thực được truyền từ người gửi đến người nhận. Độ dài khung dữ liệu có thể nằm trong khoảng 5 & 8. Nếu bit chẵn lẻ không được sử dụng thì chiều dài khung dữ liệu có thể dài 9 bit. Nói chung, LSB của dữ liệu được truyền trước tiên sau đó nó rất hữu ích cho việc truyền.^[11]

Giao diện UART

Hình dưới đây cho thấy UART giao tiếp với vi điều khiển. Giao tiếp UART có thể được thực hiện bằng ba tín hiệu như TXD, RXD và GND.



© www.elprocus.com

Hình 2.27: Giao diện Uart

c. Ứng dụng của UART

UART thường được sử dụng trong các bộ vi điều khiển cho các yêu cầu chính xác và chúng cũng có sẵn trong các thiết bị liên lạc khác nhau như giao tiếp không dây, thiết bị GPS, mô-đun Bluetooth và nhiều ứng dụng khác.

Các tiêu chuẩn truyền thông như RS422 & TIA được sử dụng trong UART ngoại trừ RS232. Thông thường, UART là một IC riêng được sử dụng trong giao tiếp nối tiếp UART.^[11]

Ưu điểm và nhược điểm của UART

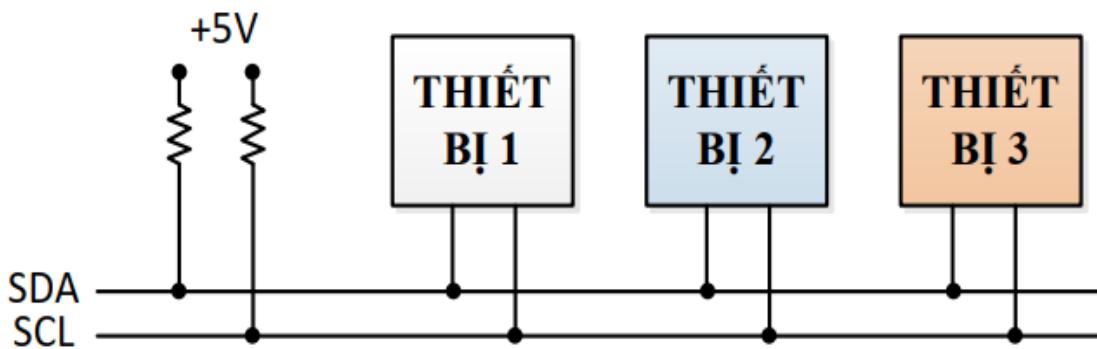
Những ưu và nhược điểm của UART bao gồm những điều sau đây:

- Nó chỉ cần hai dây để truyền dữ liệu
- Tín hiệu CLK là không cần thiết.
- Nó bao gồm một bit chẵn lẻ để cho phép kiểm tra lỗi
- Sắp xếp gói dữ liệu có thể được sửa đổi vì cả hai mặt được sắp xếp
- Kích thước khung dữ liệu tối đa là 9 bit
- Nó không chứa một số hệ thống phụ (hoặc)
- Tốc độ truyền của UART phải ở mức 10% của nhau^[11]

2.13.2 Chuẩn giao tiếp I2C

a. Giới thiệu

I2C viết tắt của từ Inter-Integrated Circuit là một chuẩn truyền thông do hãng điện tử Philips Semiconductor sáng lập cho phép giao tiếp một thiết bị chủ với nhiều thiết bị tớ với nhau như hình.^[12]



Hình 2.28: Hệ thống các thiết bị giao tiếp chuẩn I2C

Chuẩn giao tiếp I2C có 2 đường tín hiệu tên là SDA (serial data) có chức năng truyền tải dữ liệu và tín hiệu SCL (serial clock) truyền tải xung clock để dịch chuyển dữ liệu.

Trong hệ thống truyền dữ liệu I2C thì thiết bị nào cung cấp xung clock thì được gọi là chủ (master), thiết bị nhận xung clock được gọi là tớ (slave).

Thiết bị chủ chỉ có 1, thiết bị tớ thì có nhiều, mỗi thiết bị tớ sẽ có 1 địa chỉ độc lập, chuẩn truyền ban đầu dùng địa chỉ 7 bit nên có thể 1 chủ giao tiếp với 128 thiết bị tớ. Các thiết bị sau này tăng thêm số bit nên có thể giao tiếp nhiều hơn.

Giao diện I2C hỗ trợ tốc độ truyền chuẩn 100kHz hay tốc độ cao 400kHz. Ngoài ra còn hỗ trợ 7 hoặc 10 bit địa chỉ. Được thiết kế nhằm đơn giản hóa quá trình trao đổi với 2 kênh DMA cho truyền và nhận dữ liệu.^[12]

b. Quy trình truyền dữ liệu chuẩn I2C

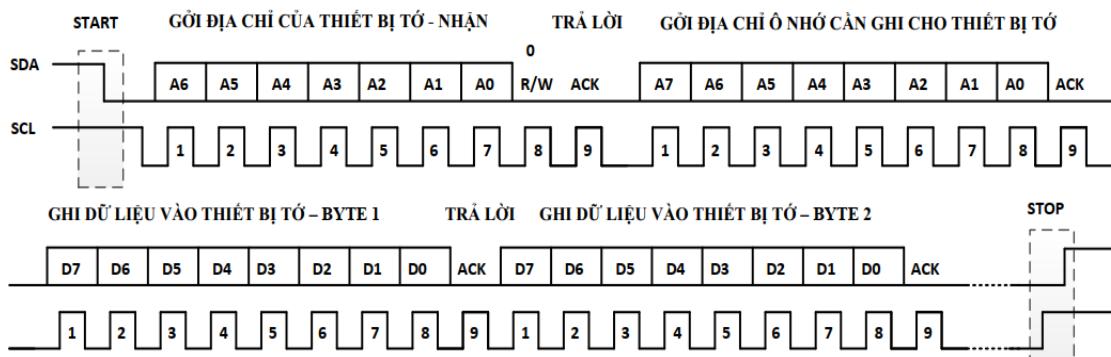
Quá trình thiết bị chủ ghi dữ liệu vào thiết bị tớ:

- Bước 1: Thiết bị chủ tạo trạng thái START để bắt đầu quá trình truyền dữ liệu – các thiết bị tớ sẽ ở trạng thái sẵn sàng nhận địa chỉ từ thiết bị chủ.
- Bước 2: Thiết bị chủ gửi địa chỉ của thiết bị tớ cần giao tiếp – khi đó tất cả các thiết bị tớ đều nhận địa chỉ và so sánh với địa chỉ của mình, các thiết bị tớ sau khi phát hiện không phải địa chỉ của mình thì chờ cho đến khi nào nhận trạng thái START mới.
- Trong dữ liệu 8 bit thì có 7 bit địa chỉ và 1 bit điều khiển đọc/ghi (R/W): thì bit này bằng 0 để báo cho thiết bị tớ sẽ nhận byte tiếp theo.
- Bước 3: Thiết bị chủ chờ nhận tín hiệu bắt tay từ thiết bị tớ. Thiết bị tớ nào đúng địa chỉ thì phát 1 tín hiệu trả lời cho chủ biết.
- Bước 4: Thiết bị chủ tiến hành gửi địa chỉ của ô nhớ bắt đầu cần ghi dữ liệu, bit

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

R/W ở trạng thái ghi.

- Bước 5: Thiết bị chủ chờ nhận tín hiệu trả lời từ thiết bị tớ.
- Bước 6: Thiết bị chủ tiến hành gửi dữ liệu để ghi vào thiết bị tớ, mỗi lần ghi 1 byte, sau khi gửi xong thì tiến hành chờ nhận tín hiệu trả lời từ thiết bị tớ, quá trình thực hiện cho đến byte cuối cùng xong rồi thì thiết bị chủ chuyển sang trạng thái STOP để chấm dứt quá trình giao tiếp với thiết bị tớ.^[12]



Hình 2.29: Quá trình chủ ghi dữ liệu vào tớ

Quá trình thiết bị chủ đọc dữ liệu vào thiết bị tớ:

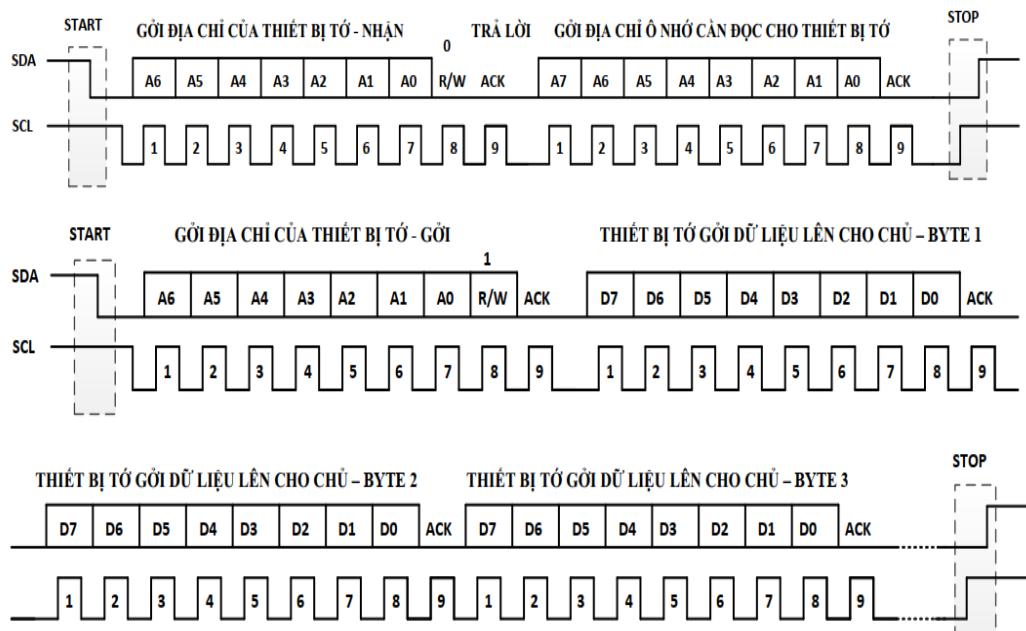
- Bước 1: Thiết bị chủ tạo trạng thái START để bắt đầu quá trình truyền dữ liệu – các thiết bị tớ sẽ ở trạng thái sẵn sàng nhận địa chỉ từ thiết bị chủ.
- Bước 2: Thiết bị chủ gửi địa chỉ của thiết bị tớ cần giao tiếp – khi đó tất cả các thiết bị tớ đều nhận địa chỉ và so sánh với địa chỉ mình, các thiết bị tớ sau khi phát hiện không phải địa chỉ của mình thì chờ cho đến khi nào nhận trạng thái START mới.

Trong dữ liệu 8 bit thì có 7 bit địa chỉ và 1 bit điều khiển đọc/ghi (R/W): thì bit này bằng 0 để báo cho thiết bị tớ sẽ nhận byte tiếp theo.

- Bước 3: Thiết bị chủ chờ nhận tín hiệu bắt tay từ thiết bị tớ. Thiết bị tớ nào đúng địa chỉ thì phát 1 tín hiệu trả lời cho chủ biết.
- Bước 4: Thiết bị chủ tiến hành gửi địa chỉ của ô nhớ bắt đầu cần đọc dữ liệu, bit R/W ở trạng thái đọc.
- Bước 5: Thiết bị chủ chờ nhận tín hiệu trả lời từ thiết bị tớ.
- Bước 6: Thiết bị chủ chuyển sang trạng thái STOP, bắt đầu lại trạng thái START, tiến hành gửi dữ liệu của thiết bị và bit R/W bằng 1 để yêu cầu tớ gửi dữ liệu nội dung ô nhớ của địa chỉ đã nhận.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Bước 7: Thiết bị chủ sau khi nhận sẽ báo tín hiệu trả lời, quá trình này thực hiện cho đến khi hết dữ liệu mong muốn thì thiết bị chủ tạo tín hiệu STOP để chấm dứt.^[12]



Hình 2.30: Quá trình chủ đọc dữ liệu từ tớ

c. Ứng dụng

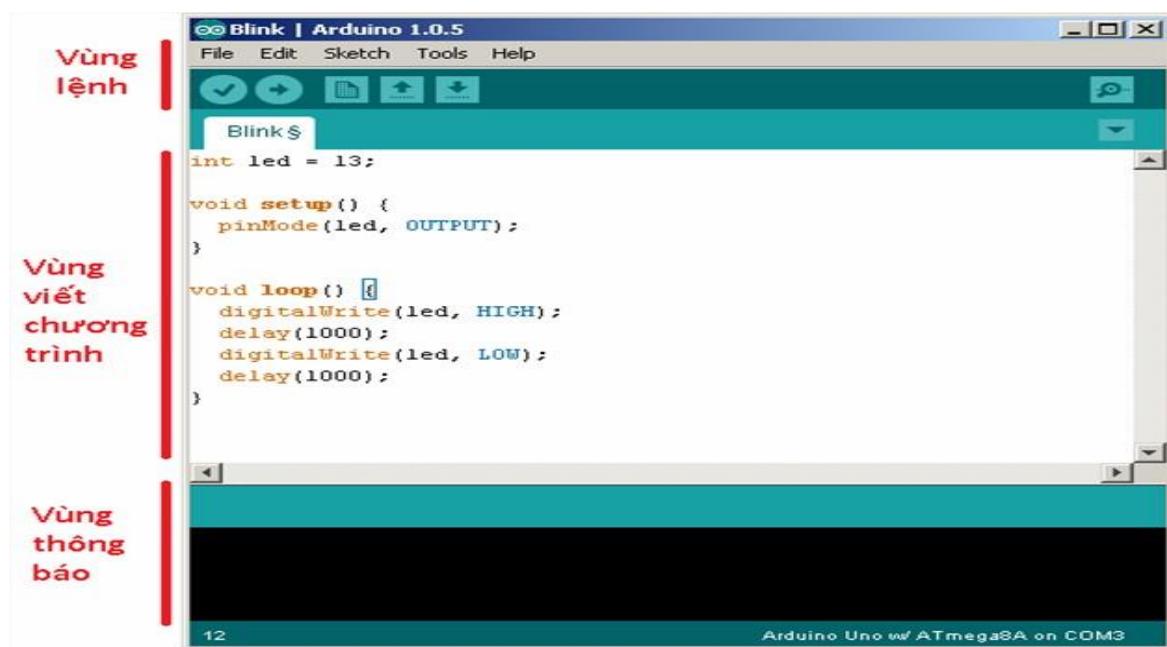
I₂C thường được dùng trong các khối truyền thông nối tiếp của vi điều khiển.

2.14 GIỚI THIỆU VỀ PHẦN MỀM ARDUINO IDE

Arduino IDE là phần mềm dùng để lập trình cho Arduino. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++. Arduino IDE có thể chạy trên 3 nền tảng phổ biến nhất hiện nay là Windows, MAC OS X và Linux. Hai hàm để tạo ra một chương trình vòng thực thi có thể chạy được:

- Setup(): hàm này chạy mỗi khi khởi động một chương trình, dùng để thiết lập các cài đặt
- Loop(): hàm này được gọi lặp lại cho đến khi tắt nguồn board mạch^[2]

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2.31: Giao diện phần mềm Arduino IDE

Vùng lệnh: bao gồm các nút lệnh menu (File, Edit, Sketch, Tools, Help). Phía dưới là các icon cho phép sử dụng nhanh các chức năng thường dùng của IDE được miêu tả như hình sau:

Icon	Chức năng
	Biên dịch chương trình đang soạn thảo để kiểm tra các lỗi lập trình.
	Biên dịch và upload chương trình đang soạn thảo.
	Mở một trang soạn thảo mới.
	Mở các chương trình đã lưu.
	Lưu chương trình đang soạn.
	Mở cửa sổ Serial Monitor để gửi và nhận dữ liệu giữa máy tính và board Arduino.

Hình 2.32: Icon và chức năng trong Arduino IDE

Vùng viết chương trình: Các đoạn code sẽ được viết trong vùng này. Tên chương trình được hiển thị ngay dưới dãy các icon, như ở hình 3.3 tên chương trình là “Blink”. Phía sau tên chương trình có một dấu “§” có nghĩa là đoạn chương trình chưa được lưu lại.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Vùng thông báo (Debug): Những thông báo từ IDE sẽ được hiển thị tại đây. Góc dưới cùng bên phải hiển thị loại board Arduino và cổng COM được sử dụng.^[2]

2.15 PHƯƠNG PHÁP PHÂN LOẠI SẢN PHẨM THEO MÀU SẮC

Với đề tài “Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm” Thiết kế mô hình phân loại sản phẩm (sử dụng sản phẩm là các phôi) theo các màu (đỏ, xanh, vàng). Ở phần này sẽ đi tìm hiểu chi tiết phương pháp nhận dạng của từng màu sắc của sản phẩm và phân loại sản phẩm theo từng màu sắc đã được nhận dạng.

2.15.1 Các màu sắc cơ bản của sản phẩm

Trong cuộc sống hiện nay, nhu cầu về màu sắc sản phẩm ngày càng được chú trọng, để đáp ứng nhu cầu thiết yếu đó nên trên thị trường ra nhiều sản phẩm có màu sắc đa dạng. Ở đây sản phẩm sử dụng màu sắc của các phôi, có hai nhóm màu sắc cơ bản: Màu sắc sản phẩm phổ biến, màu sắc sản phẩm ít phổ biến. Cách chia này dựa vào sự phổ biến của màu sắc sản phẩm ở ngoài cuộc sống, nhóm màu sắc chính (đỏ, xanh, vàng) và các màu sắc còn lại thì ít thông dụng hơn nên là thuộc nhóm còn lại. Ở đề tài để gần với thực tế, nhóm chọn sản phẩm phôi có màu sắc phổ biến.

2.15.2 Phương pháp nhận dạng màu sắc

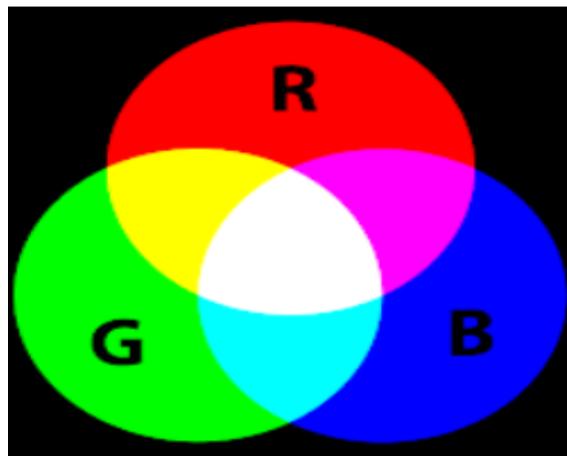
a. Giới thiệu về các không gian màu

Một bức ảnh chụp có thể được biểu diễn bằng các không gian màu khác nhau.

Không gian màu là một mô hình toán học dùng để mô tả các màu sắc trong thực tế được biểu diễn dưới dạng số học. Trên thực tế có rất nhiều không gian màu khác nhau được sử dụng vào những mục đích khác nhau. Trong phần này sẽ tìm hiểu qua về ba không gian màu cơ bản hay được nhắc tới và ứng dụng nhiều, đó là hệ không gian màu RGB, HSV và CMYK.^[13]

➤ Không gian màu RGB

RGB là không gian màu rất phổ biến được dùng trong đồ họa máy tính và nhiều thiết bị kỹ thuật số khác. Ý tưởng chính của không gian màu này là sự kết hợp của 3 màu sắc cơ bản : màu đỏ (R, Red), xanh lục (G, Green) và xanh lơ (B, Blue) để mô tả tất cả các màu sắc khác.^[13]



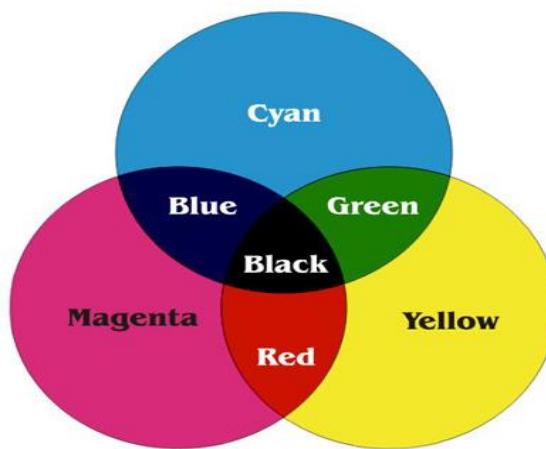
Hình 2.33: Không gian màu RGB

Nếu như một ảnh số được mã hóa bằng 24bit, nghĩa là 8bit cho kênh R, 8bit cho kênh G, 8bit cho kênh B, thì mỗi kênh màu này sẽ nhận giá trị từ 0-255. Với mỗi giá trị khác nhau của các kênh màu kết hợp với nhau ta sẽ được một màu khác nhau, như vậy ta sẽ có tổng cộng $255 \times 255 \times 255 = 1.66$ triệu màu sắc.

Ví dụ: màu đen là sự kết hợp của các kênh màu (R, G, B) với giá trị tương ứng (0, 0, 0) màu trắng có giá trị (255, 255, 255), màu vàng có giá trị (255, 255, 0), màu tím đậm có giá trị (64, 0, 128) ... Nếu ta dùng 16bit để mã hóa một kênh màu (48bit cho toàn bộ 3 kênh màu) thì dải màu sẽ trải rộng lên tới $3 \times 2^{16} = \dots$ Một con số rất lớn.^[13]

➤ Không gian màu CMYK

CMYK là không gian màu được sử dụng phổ biến trong ngành công nghiệp in ấn. Ý tưởng cơ bản của hệ không gian này là dùng 4 màu sắc cơ bản để phục vụ cho việc pha trộn mực in. Trên thực tế, người ta dùng 3 màu là C=Cyan: xanh lơ, M=Magenta: hồng xám, và Y=Yellow: vàng để biểu diễn các màu sắc khác nhau. Nếu lấy màu hồng xám cộng với vàng sẽ ra màu đỏ, màu xám kết hợp với xanh lơ sẽ cho xanh lam ... Sự kết hợp của 3 màu trên sẽ cho ra màu đen, tuy nhiên màu đen ở đây không phải là đen tuyệt đối và thường có độ tương phản lớn, nên trong ngành in, để tiết kiệm mực in người ta thêm vào màu đen để in những chi tiết có màu đen thay vì phải kết hợp 3 màu sắc trên. Và như vậy ta có hệ màu CMYK. chữ K ở đây là để kí hiệu màu đen (Black), có nhẽ chữ B đã được dùng để biểu diễn màu Blue nên người ta lấy chữ cái cuối K để biểu diễn màu đen?^[13]

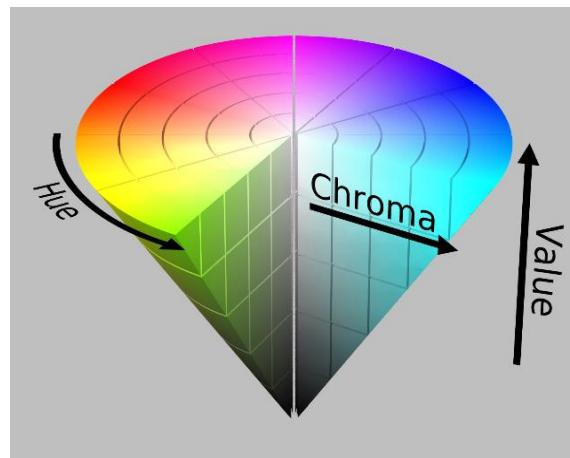


Hình 2.34: Không gian màu CMYK

Nguyên lý làm việc của hệ màu này như sau : Trên một nền giấy trắng, khi mỗi màu này được in lên sẽ loại bỏ dần đi thành phần màu trắng. 3 màu C, M, Y khác nhau in theo những tỉ lệ khác nhau sẽ loại bỏ đi thành phần đó một cách khác nhau và cuối cùng cho ta màu sắc cần in. Khi cần in màu đen, thay vì phải in cả 3 màu người ta dùng màu đen để in lên. Nguyên lý này khác với nguyên lý làm việc của hệ RGB ở chỗ hệ RGB là sự kết hợp của các thành phần màu, còn hệ CMYK là sự loại bỏ lẫn nhau của các thành phần màu.^[13]

➤ Không gian màu HSV

HSV và cũng gần tương tự như HSL là không gian màu được dùng nhiều trong việc chỉnh sửa ảnh, phân tích ảnh và một phần của lĩnh vực thị giác máy tính. Hệ không gian này dựa vào 3 thông số sau để mô tả màu sắc H = Hue: màu sắc, S = Saturation: độ đậm đặc, sự bảo hòa, V = value: giá trị cường độ sáng. Không gian màu này thường được biểu diễn dưới dạng hình trụ hoặc hình nón . Theo đó, Ứng dụng xử lý ảnh trong thực tế với thư viện OpenCV.



Hình 2.35: Không gian màu HSV

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Ví dụ: màu đen là sự kết hợp của các kênh màu (R, G, B) với giá trị tương ứng (0, 0, 0) màu trắng có giá trị (255, 255, 255), màu vàng có giá trị (255, 255, 0), màu tím đậm có giá trị (64, 0, 128) ... Nếu ta dùng 16bit để mã hóa một kênh màu (48bit cho toàn bộ 3 kênh màu) thì dải màu sẽ trải rộng lên tới $3 * 2^{16} = \dots$ Một con số rất lớn.^[13]

b. Phương pháp nhận dạng màu sắc

Với ảnh chụp thu thập được từ Camera sẽ cho không gian màu của ảnh là RGB, không gian màu RGB không thích hợp cho việc nhận dạng màu sắc, để phân biệt màu sắc người ta thường dùng không gian màu HSV. Vì vậy ta cần chuyển đổi ảnh gốc (ảnh có không gian màu RGB) sang ảnh có không gian màu HSV.

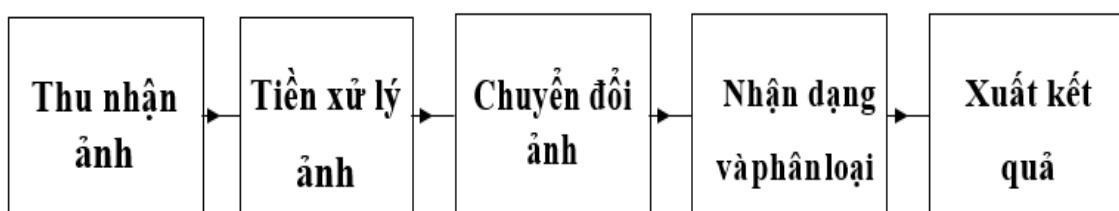
Thư viện OpenCV là một thư viện chuyên xử lý hình ảnh vì vậy nó hỗ trợ rất nhiều hệ màu như RGB, BRGA, YUV, HSV... và nó cũng có cung cấp cho chúng ta hàm cv2.cvtColor để thực hiện chuyển đổi giữa các hệ màu này. Trong OpenCV, H(Hue) có giá trị trong khoảng (0 – 179) mang thông tin về màu sắc, S(Saturation) có giá trị (0 - 255) cho thông tin về độ thuần khiết của màu, V(Value) có giá trị (0 - 255) cho biết độ sáng điểm ảnh. Chuyển đổi không gian màu từ RGB sang HSV:

```
hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV);[14]
```

Ví dụ: Màu đỏ thường thành phần H có giá trị trong khoảng (0 - 7) và (174 - 179), các thành phần S và V thường chọn trong lân cận 100 tùy theo điều kiện ánh sáng. Tương tự cho các màu còn lại.

Khi đã chuyển được ảnh sang ảnh HSV và có các giá trị HSV của các màu mong muốn, ta tiến hành so sánh để xác định được ảnh đang xét là ảnh màu gì.

Các bước thực hiện:



Hình 2.36: Sơ đồ các bước thực hiện phân loại màu

i. Thu nhận ảnh

Thu nhận ảnh là quá trình mà ảnh có thể thu nhận qua camera hoặc có thể lấy từ bộ nhớ của hệ thống.^[14]

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

ii. Tiền xử lý ảnh

Là bước chỉnh sửa chất lượng ảnh, như việc lọc nhiễu hay tăng cường độ sáng, để nâng cao chất lượng ảnh và chuyển sang ảnh xám để dễ dàng nhận dạng ảnh.^[14]

Chuyển đổi ảnh xám: Giả sử, hình ảnh của được lưu trữ dưới dạng RGB (Red-Green-Blue). Điều này có nghĩa là có ba ma trận xám tương ứng cho màu Red, Green, Blue. Công việc là tìm cách tổng hợp ba ma trận này về thành một ma trận duy nhất. Một trong số các công thức phổ biến để thực hiện việc đó là

$$Y = 0.2126R + 0.7152G + 0.0722B \quad (2.10)$$

Trong đó: Y: ma trận xám cần tìm; R: ma trận xám đỏ của ảnh; G: ma trận xám lục của ảnh; B: ma trận xám lam của ảnh.^[14]

Lọc trung bình: Giả sử có một ảnh đầu vào với $I(x,y)$ là giá trị điểm ảnh tại một điểm (x,y) và một ngưỡng θ .

Bước 1: Tính tổng các thành phần trong ma trận lọc (Kernel).

Bước 2: Chia lấy trung bình của tổng các thành phần trong ma trận được tính ở trên với số lượng các phần tử của cửa sổ lọc ra một giá trị $Itb(x, y)$.

Bước 3: Hiệu chỉnh:

$$\text{Nếu } I(x,y) - Itb(x,y) > \theta \text{ thì } I(x,y) = Itb(x,y) \quad (2.11)$$

$$\text{Nếu } I(x,y) - Itb(x,y) \leq \theta \text{ thì } I(x,y) = I(x,y) \quad (2.12)$$

Chú ý: θ là một giá trị cho trước và có thể có hoặc không tùy thuộc vào mục đích.^[14]

iii. Chuyển đổi ảnh

Không gian màu RGB không thích hợp cho việc nhận dạng màu sắc, để phân biệt màu sắc người ta thường dùng không gian màu HSV. Vì vậy ta cần chuyển đổi ảnh có không gian màu RGB sang ảnh có không gian màu HSV.^[14]

Công thức chuyển đổi giữa RGB và HSV

➤ Chuyển từ RGB sang HSV :

Giả sử ta có một điểm màu có giá trị trong hệ RGB là (R, G, B) . ta chuyển sang không gian HSV như sau:

Đặt $M = \text{Max}(R, G, B)$, $m = \text{Min}(R, G, B)$ và $C = M - m$.

Nếu $M = R$, $H' = (G - B)/C \bmod 6$. Nếu $M = G$, $H' = (B - R)/C + 2$. Nếu $M = B$, $H' = (R - G)/C + 4$. Và $H = H' \times 60$. Trong trường hợp $C = 0$, $H = 00$ $V = M$.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

$S = C/V$. Trong trường hợp V hoặc C bằng 0, $S = 0$.

➤ Chuyển từ HSV sang RGB :

Giả sử ta có không gian màu HSV với $H = [0, 360]$, $S = [0, 1]$, $V = [0, 1]$. Khi đó, ta tính $C = VxS$. $H' = H/60$.

$X = C(1 - |H' \bmod 2 - 1|)$. Ta biểu diễn hệ $(R1, G1, B1)$ như sau: $(1, 1, 1) = \{$

- $(0, 0, 0)$ nếu H chưa được xác định
- $(C, X, 0)$ nếu $0 < H' < 1$
- $(X, C, 0)$ nếu $1 < H' < 2$
- $(0, C, X)$ nếu $2 < H' < 3$
- $(0, X, C)$ nếu $3 < H' < 4$
- $(X, 0, C)$ nếu $4 < H' < 5$
- $(C, 0, X)$ nếu $5 < H' < 6$

$\}^{[14]}$

iv. Nhận dạng và phân loại

Ta nhận dạng màu sắc dựa vào các đặc điểm riêng biệt của từng màu, khi có được các đặc điểm riêng biệt của từng màu, ta tiến hành phân loại từng màu sắc sản phẩm một nhau nhờ vào việc so sánh các đặc điểm riêng biệt của chúng với các giá trị đã định trước. Ta lần lượt đi so sánh từng màu một, từ màu đỏ, đến màu xanh, đến màu vàng. Nếu các đặc điểm trùng khớp với đặc điểm được đề ra trước ở màu nào thì ta kết luận ngay đó là sản phẩm có màu đó. Ngược lại nếu không có đặc điểm nào khớp ta kết luận đó là một sản phẩm nào đó khác mà không phải sản phẩm đã định sẵn.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

3.1 GIỚI THIỆU

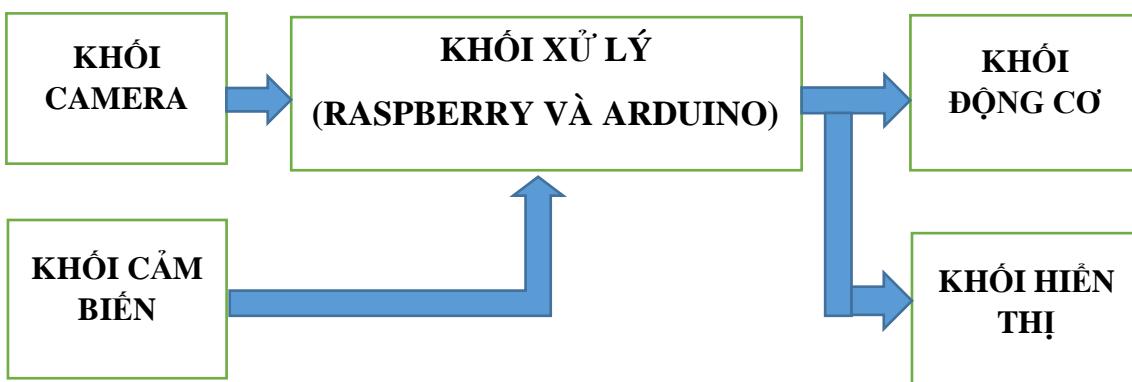
Ngày nay, phần lớn các thiết bị điện tử đều phát triển theo xu hướng tự động hóa, thông minh, càng hiểu ý con người, chúng giao tiếp được với con người và hoạt động theo ý muốn con người, để làm được điều đó thì các thiết bị cảm biến, thuật toán hỗ trợ ra đời. Trong số đó bài toán “phân loại sản phẩm theo màu sắc” là một đề tài đáng để cho ta tìm hiểu.

Đề tài “Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm” với mục tiêu là phân loại sản phẩm theo màu sắc (đỏ, xanh, vàng). Dựa trên ngôn ngữ Python với thư viện chính là OpenCV, được thực hiện với bộ xử lý trung tâm là kit Raspberry và kit Arduino Uno, khói thu tín hiệu hình ảnh từ bên ngoài là Camera Raspberry Pi, điều khiển cơ cấu chấp hành, sau đó hiển thị kết quả trên màn hình.

Đề tài chỉ xây dựng với mô hình nhỏ và gọn không làm thành dây chuyền sản xuất. Thời gian và tốc độ xử lý còn chậm, cùng với Camera chụp ảnh không mong muốn ở điều kiện thiếu ánh sáng mà phải được che kín và chiếu thêm đèn led để tăng cường sáng.

3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

3.2.1 Thiết kế sơ đồ khái niệm hệ thống



Hình 3.1: Sơ đồ khái niệm hệ thống

Đề tài xây dựng với mô hình nhỏ và gọn không làm thành dây chuyền sản xuất. Mô hình gồm bộ điều khiển trung tâm là Kit Raspberry và Arduino Uno, khói thu tín hiệu hình ảnh từ bên ngoài là Camera Raspberry Pi, sau đó điều khiển hoạt động của các động cơ và hiển thị kết quả trên màn hình.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

Chức năng từng khối:

Khối xử lý:

• Raspberry Pi: có chức năng nhận tín hiệu hình ảnh từ khói thu tín hiệu hình ảnh khi có mệnh lệnh của Arduino Uno, tiến hành xử lý hình ảnh và phân tích nhận dạng, sau đó trả lại kết quả cho Arduino Uno, tiếp tục chờ mệnh lệnh tiếp theo của Arduino Uno. Sử dụng Kit Raspberry Pi 3 Model B.

• Arduino Uno: Có chức năng nhận tín hiệu từ khói cảm biến sau đó ra lệnh cho Raspberry Pi hoạt động, rồi nhận dữ liệu trả về từ Raspberry Pi, tiến hành phân tích xử lý, điều khiển hoạt động của khói động cơ và gửi dữ liệu sang khói hiển thị. Ở đây sử dụng Kit Arduino Uno R3.

Khối camera: Có chức năng thu thập tín hiệu hình ảnh từ thực tế chuyển về tín hiệu điện và gửi dữ liệu cho khói Raspberry Pi. Ở đây sử dụng Camera Raspberry Pi.

Khối cảm biến: Có chức năng gửi tín hiệu điện cho khói điều khiển khi có vật cản đi qua. Sử dụng cảm biến hồng ngoại E18-D80NK.

Khối động cơ: Được vận hành bởi khói điều khiển. Sử dụng động cơ DC và động cơ Servo MG996R.

Khối hiển thị: Có chức năng hiển thị kết quả nhận được từ khói điều khiển. Sử dụng màn hình LCD16x2.

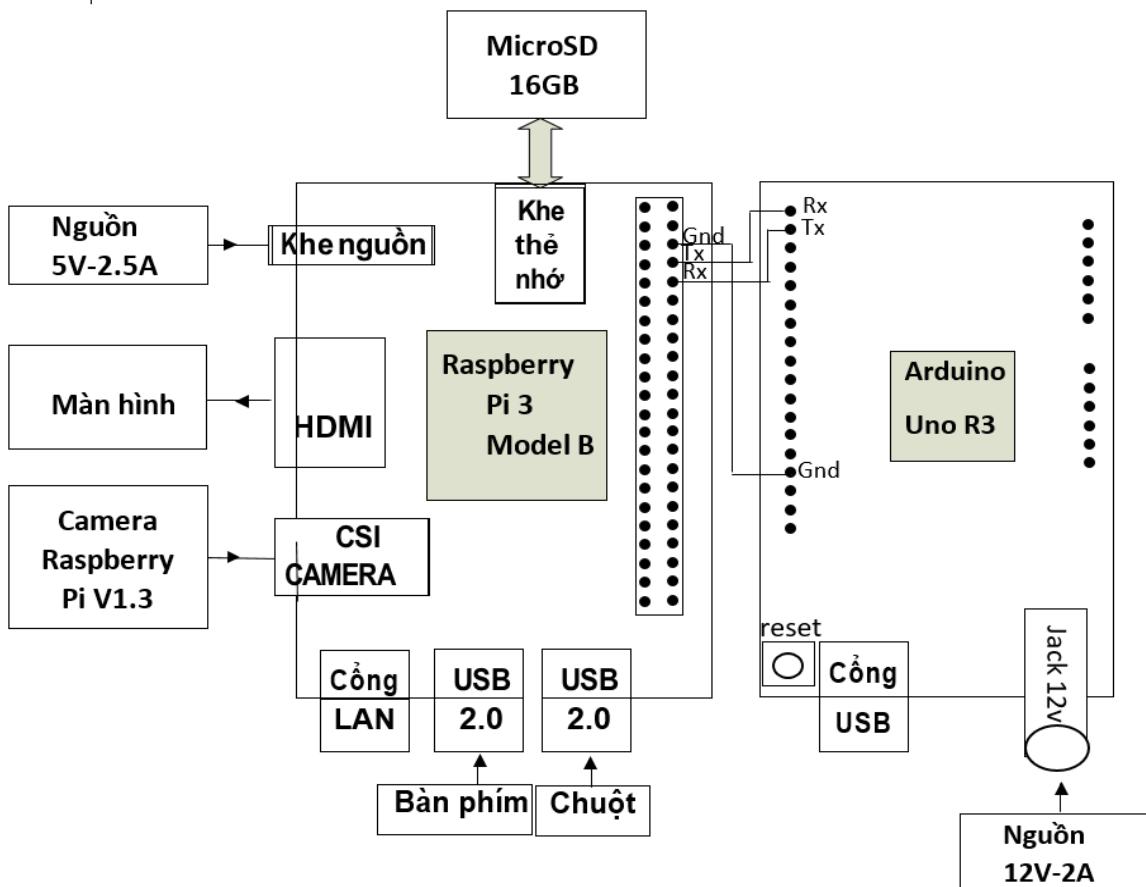
3.2.2 Sơ đồ kết nối hệ thống

Hệ thống được kết nối dựa trên phần cứng có sẵn của nhà sản xuất, nên không có phần thiết kế mạch in cũng như sơ đồ nguyên lý chi tiết.

a. **Khối xử lý**

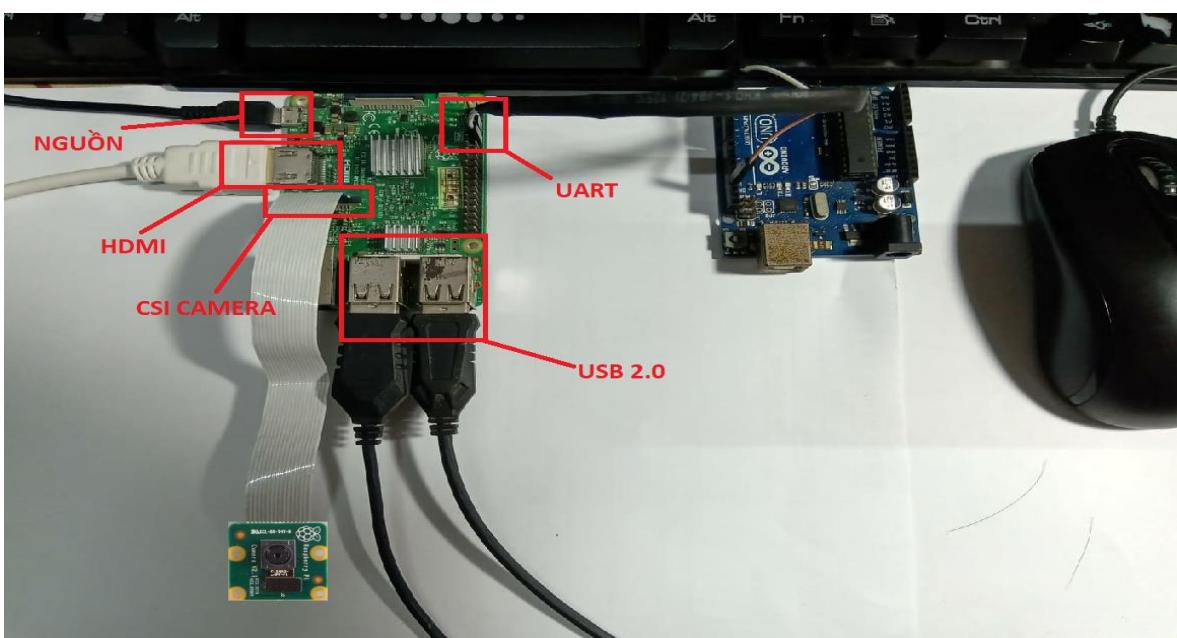
- **Raspberry Pi**

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ



Hình 3.2: Sơ đồ kết nối của Raspberry Pi

Trên đây là sơ đồ kết nối của Raspberry Pi, khói thu tín hiệu hình ảnh từ Camera Raspberry Pi được kết nối vào cổng CIS Camera, Raspberry Pi và Arduino truyền nhận dữ liệu theo chuẩn Uart thông qua 2 chân Rx, Tx. Ngoài ra sử dụng thêm các thiết bị ngoại vi để hỗ trợ lập trình như: bàn phím, chuột, màn hình, thẻ nhớ...



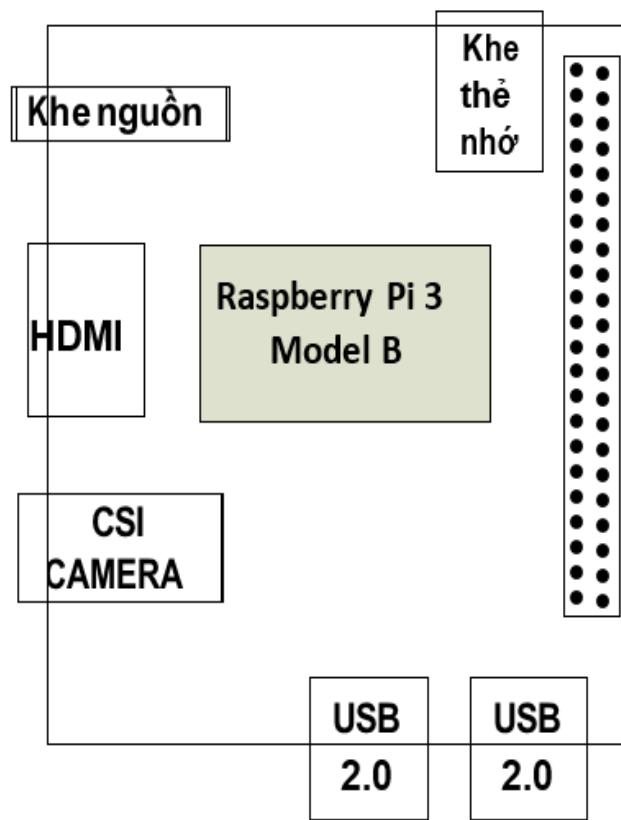
Hình 3.3: Sơ đồ kết nối thực tế của Raspberry Pi

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

Chức năng của Kit Raspberry Pi là nhận tín hiệu hình ảnh từ khói Camera, sau đó xử lý ảnh và phân tích để đưa kết quả sang Arduino Uno. 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 đã sử dụng hệ thống nhúng Raspberry Pi 3 Model B làm nhiệm vụ này.

Máy tính nhúng Raspberry có CPU 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 model từ bên ngoài. Raspberry có nhiều cổng giao tiếp nhưng trong mô hình chỉ sử dụng một số cổng sau:

Giao tiếp ngoại vi sử dụng:



Hình 3.4: Sơ đồ các cổng ngoại vi sử dụng

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: dụng 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

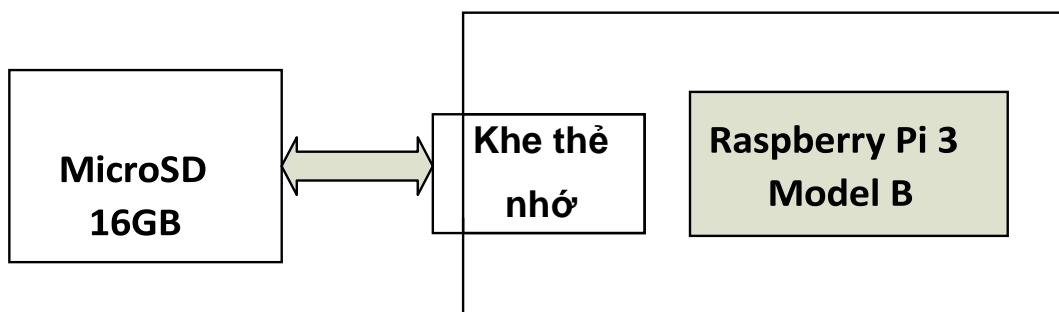
CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

hình Ảnh KTS, hỗ trợ âm thanh KTS 8 kênh.

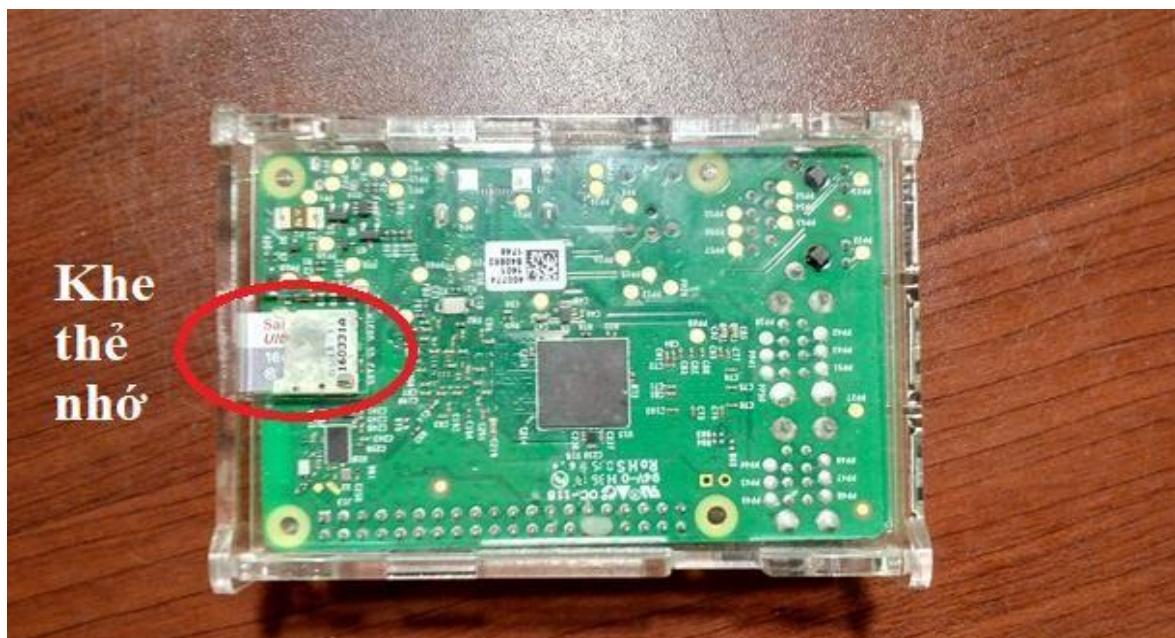
Cổng CSI Camera: Mắt của mô hình này chính là Camera được kết nối thông qua cổng CSI Camera có 15 chân.

Nguồn: Dòng hoạt động của Raspberry 500-1000mA, cổng HDMI sử dụng 50mA, module Camera sử dụng 250mA, bàn phím và chuột sử dụng từ 100mA-1000mA. Tổng dòng tiêu thụ được khoảng 2300mA. Chính vì vậy ta sử dụng Adapter 5V và 2.5A để cấp cho máy tính nhúng cho Raspberry.

Chọn thẻ nhớ lưu dữ liệu: Muốn chạy được chương trình trên máy tính nhúng 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



Hình 3.5: Sơ đồ kết nối thẻ nhớ của Raspberry 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, tốc độ đọc cũng ảnh hưởng tới tốc độ xử lý dữ liệu của các chương trình chính vì vậy với tốc độ 48MB/s là tốc độ hợp lý.^[1]



Hình 3.6: Gắn thẻ nhớ vào khe thực tế

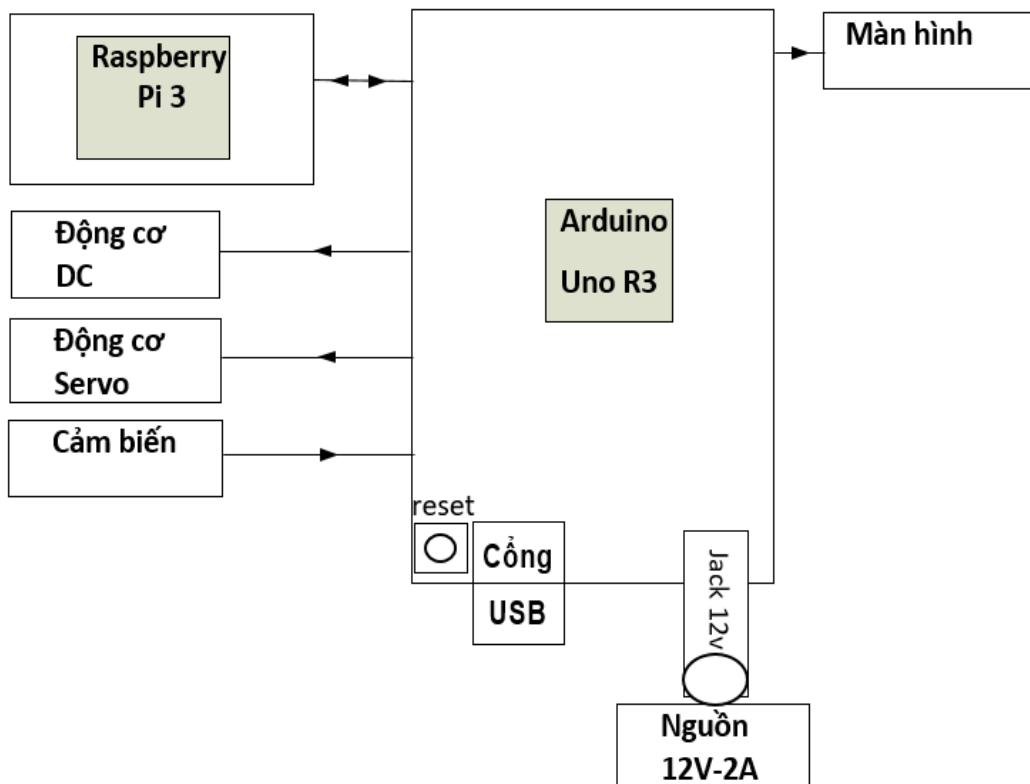
CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

Bảng 3.1: Thống kê dòng tiêu thụ

Thiết bị	Dòng tiêu thụ	Ghi chú
Raspberry Pi	500-1000mA	Khi Kit hoạt động mình nó
Module Camera	250mA	
Cổng HDMI	50mA	
Bàn phím và chuột	100mA - 1000mA	
Tổng	900mA – 2300mA	Chọn nguồn 2.5A

Từ bảng thống kê trên cho ta thấy, giao tiếp ngoại vi của Raspberry Pi là rất tốn kém năng lượng, chính vì thế để hệ thống hoạt động ổn định thì nên đầu tư mua một bộ nguồn cung cấp cho Raspberry Pi.

• Arduino Uno



Hình 3.7: Sơ đồ kết nối Arduino

Chức năng của Arduino Uno là nhận tín hiệu từ cảm biến, ra lệnh cho Raspberry hoạt động, nhận tín hiệu từ Raspberry Pi tiến hành phân tích xử lý, điều khiển hoạt động của các động cơ và gửi dữ liệu sang khối hiển thị. Với yêu cầu trên ta đều có thể sử dụng nhiều loại vi điều khiển làm khối xử lý. Nhưng ở đây ta chọn Kit Arduino Uno R3 vì nó là một trong những Kit phổ biến với nhiều chức năng, giá thành rẻ hơn

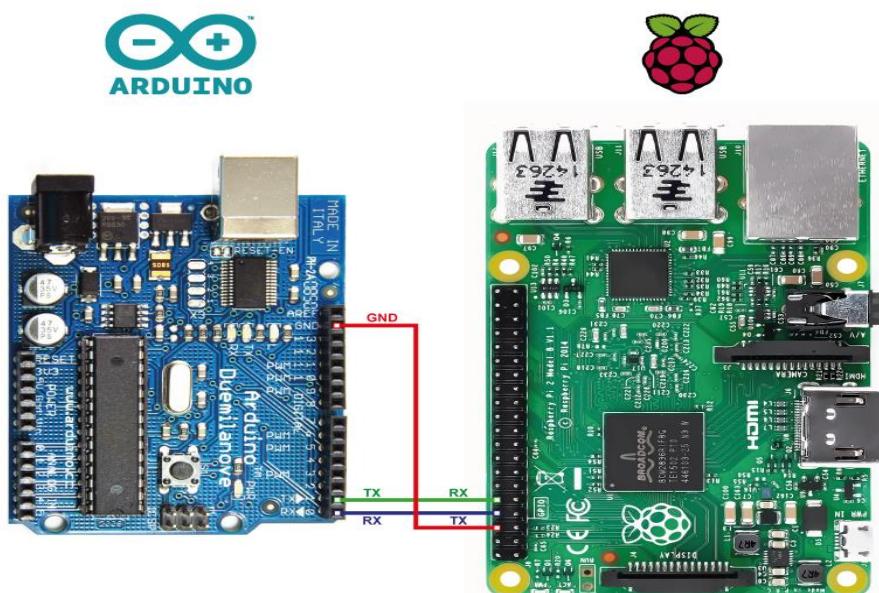
CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

các vi điều khiển khác.

❖ Giao tiếp giữa Raspberry Pi và Arduino Uno

Trong suốt quá trình hoạt động của hệ thống, Raspberry Pi và Arduino Uno cần truyền nhận dữ liệu một cách liên tục, ở đây nhóm chọn kiểu truyền Uart vì trên cả 2 Kit điều khiển đều được tích hợp sẵn chuẩn truyền này nên quá trình thiết lập khá dễ dàng, đồng thời chuẩn Uart với tốc độ truyền nhận nhanh và ổn định, đáp ứng tốt được yêu cầu của hệ thống.

- Chân RX của Arduino Uno nối với chân TX của Raspberry Pi 3.
- Chân TX của Arduino Uno nối với chân RX của Raspberry Pi 3.
- Nối chung 2 chân GND.



Hình 3.8: Giao tiếp Uart giữa Raspberry Pi và Arduino Uno

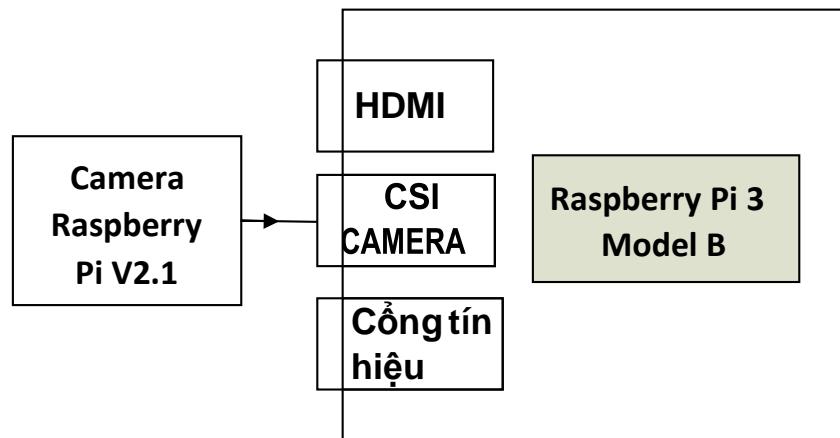
b. Khối Camera

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

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

gọn megapixel sẽ quy định kích thước bức ảnh chụp. Một MP tương ứng với một triệu điểm ảnh (1.000×1.000 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: $25 \times 23 \times 9$ mm. Camera Pi được kết nối trực tiếp với máy tính nhúng Raspberry thông qua cổng giao tiếp ngoại vi CSI Camera có 15 chân. Để tăng cường độ sáng cũng như 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.^[1]



Hình 3.9: Sơ đồ kết nối khối Camera



Hình 3.10: Kết nối Camera thực tế

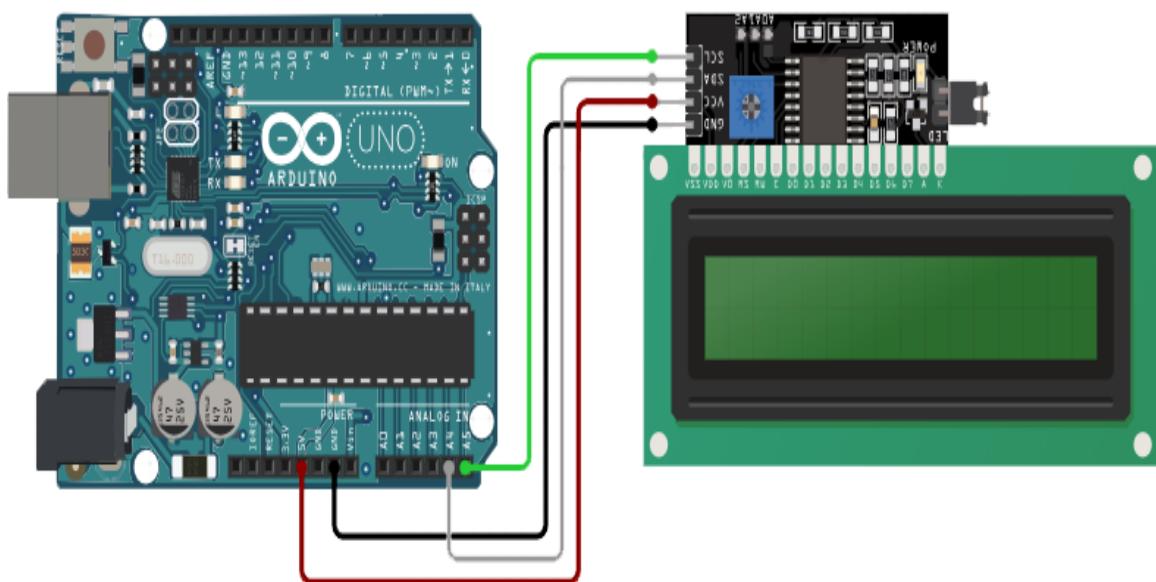
c. Khối hiển thị

Ở đây nhóm sử dụng màn hình LCD16x2 để hiển thị kết quả, LCD16x2 có các chuẩn giao tiếp với Arduino Uno: Giao tiếp chuẩn 4bit, chuẩn 8bit, chuẩn I2C. Nhóm

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

quyết định chọn chuẩn I2C vì với chuẩn này việc lắp đặt khá dễ dàng, chỉ cần sử dụng 2 chân SDA, SCL.

- Chân SDA của I2C trên LCD được nối vào chân A4 trên Arduino Uno (chân SDA trên Arduino Uno)
- Chân SCL của I2C trên LCD được nối vào chân A5 trên Arduino Uno (chân SCL trên Arduino Uno)



Hình 3.11: Giao tiếp I2C giữa Arduino Uno và LCD16x2

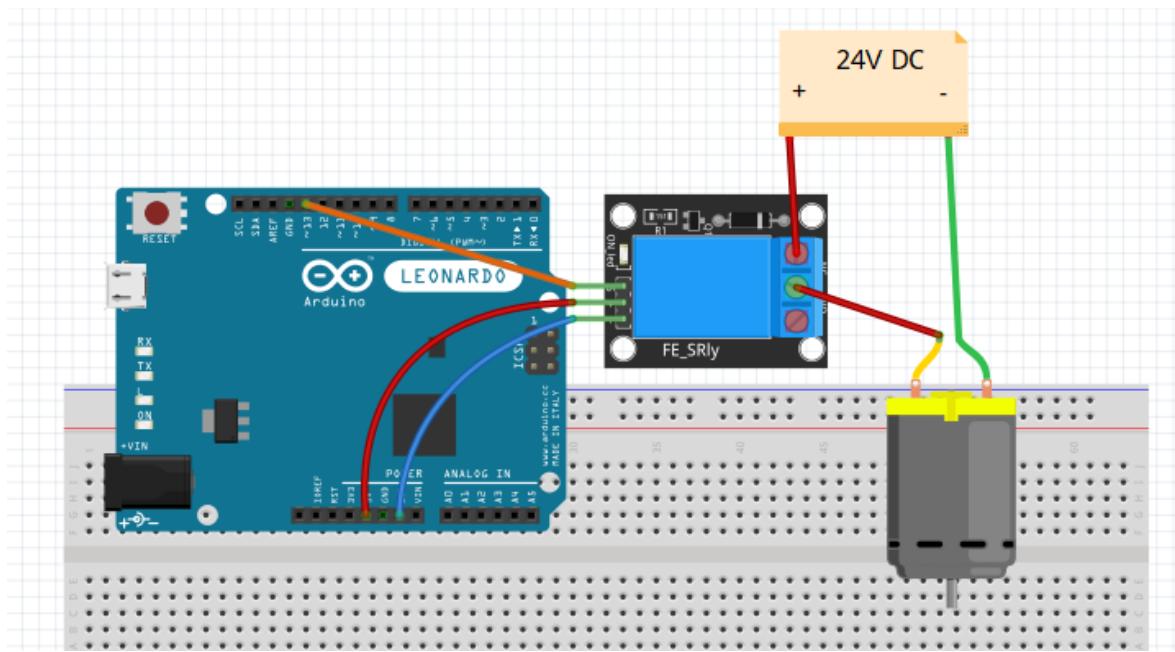
d. Khối động cơ

➤ Giao tiếp động cơ DC:

Động cơ DC được sử dụng để điều khiển cho băng tải chạy, sử dụng động cơ DC 24V vì thế nhóm sử dụng Arduino Uno điều khiển Relay 5V để đóng ngắt quá trình chạy.

- Chân kích của Relay 5V được nối vào chân 13 của Arduino Uno.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

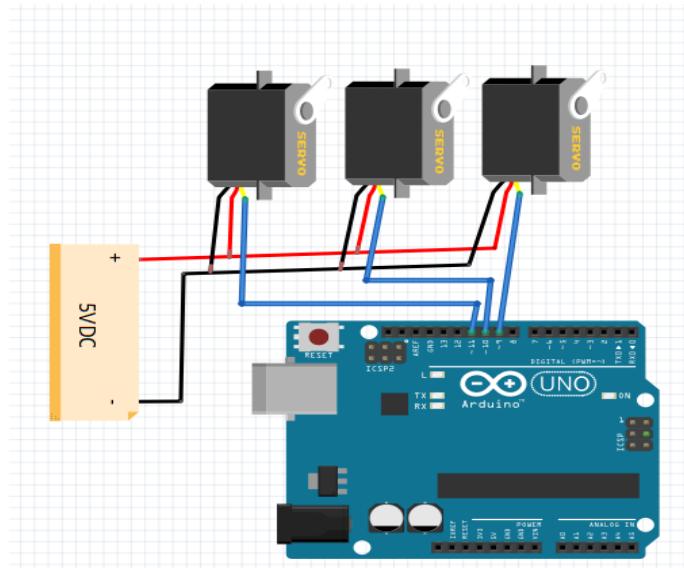


Hình 3.12: Kết nối Arduino Uno và động cơ DC

- Giao tiếp động cơ Servo MG996R

Động cơ Servo MG996R được sử dụng để gạt sản phẩm nhằm phân loại theo cùng màu sắc. Sử dụng 3 servo tương ứng với 3 sản phẩm đỏ, xanh, vàng.

- 3 chân điều khiển của 3 servo được nối tương ứng vào 3 chân 9, 10, 11 trên Arduino Uno.



Hình 3.13: Kết nối giữa Arduino Uno và động cơ servo

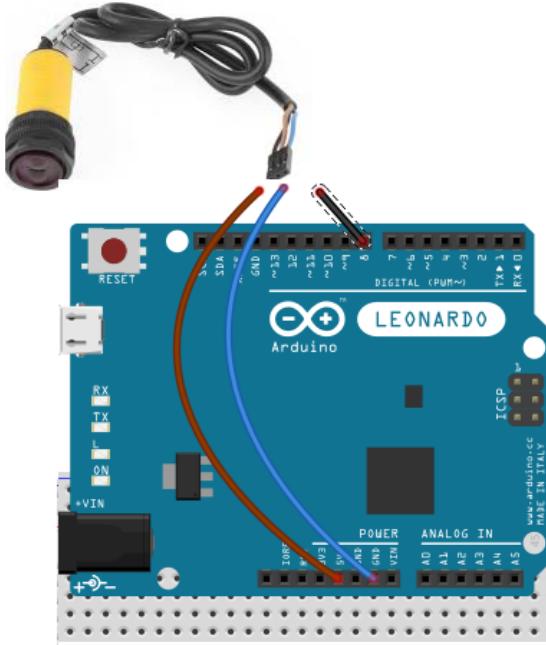
e. Khối cảm biến

Đè tài sử dụng cảm biến vật cản hồng ngoại E18-D80NK để phát hiện sản phẩm

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

đi qua, gửi tín hiệu điện về cho Arduino Uno để thực hiện các quá trình tiếp theo.

- Chân tín hiệu của cảm biến được nối vào chân số 8 của Arduino Uno



Hình 3.14: Kết nối giữa Arduino Uno và cảm biến E18-D80NK

f. Khối nguồn

- ❖ Nguồn cấp cho Raspberry Pi 3 và camera Pi

Raspberry Pi 3 hoạt động ổn định với nguồn 5VDC-2A, vì vậy ở đây nhóm sử dụng Adapter chuyên dụng cho Raspberry Pi.

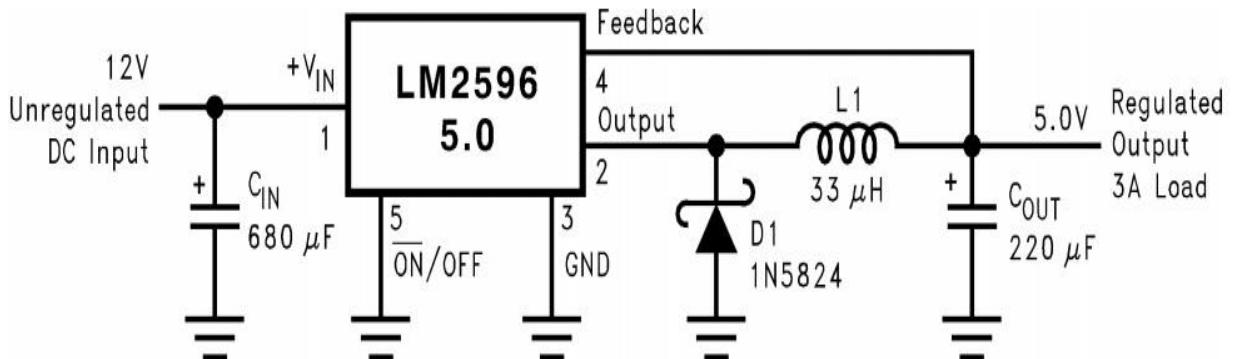


Hình 3.15: Adapter 5VDC – 2A

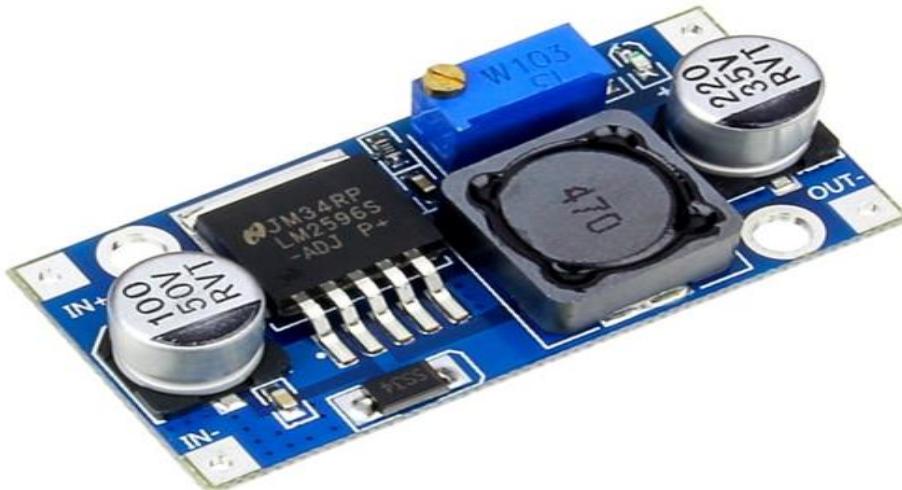
- ❖ Nguồn cấp cho Arduino Uno

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

Nguồn cung cấp cho Arduino Uno R3 ở mức điện áp 7-12VDC, dòng điện max $I_{max}=50mA$, do đó nhóm sử dụng mạch hạ áp LM2596 để hạ điện áp 24VDC từ nguồn tổ ong còn 12VDC cấp cho Arduino Uno.



Hình 3.16: Sơ đồ nguyên lý mạch hạ áp LM2596

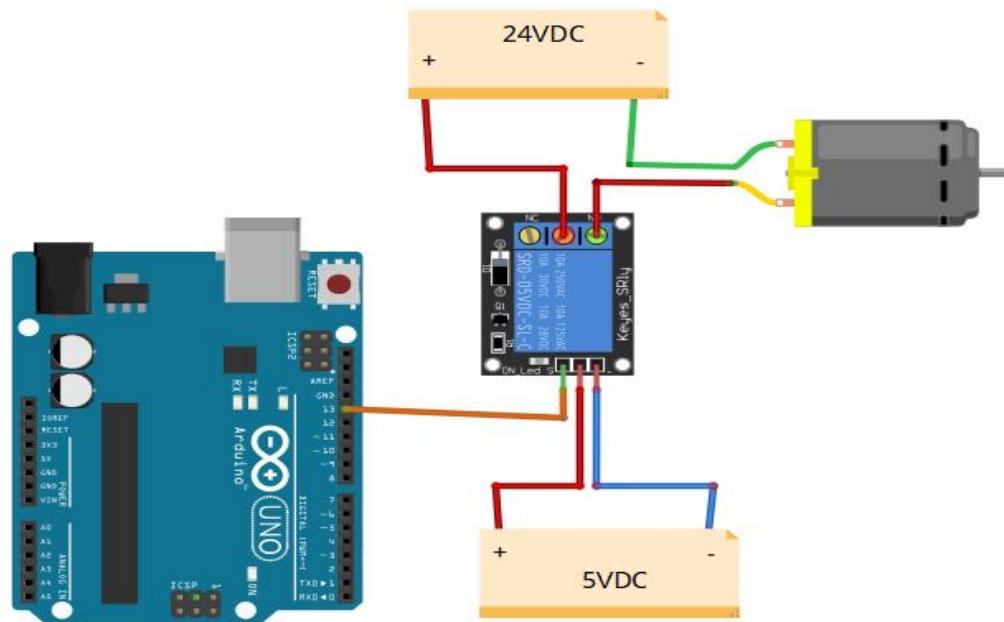


Hình 3.17: Hình ảnh thực tế mạch hạ áp LM2596

❖ Nguồn cấp cho động cơ DC

Ở đây nhóm sử dụng nguồn tổ ong cấp cho động cơ DC 24VDC để điều khiển hoạt động của băng tải, do điện áp hoạt động quá lớn so với nguồn hoạt động của Arduino Uno, nên sử dụng thêm module Relay 5VDC để Arduino Uno có thể đóng ngắt quá trình hoạt động của băng tải.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ



Hình 3.18: Mạch cấp nguồn cho động cơ DC 24V

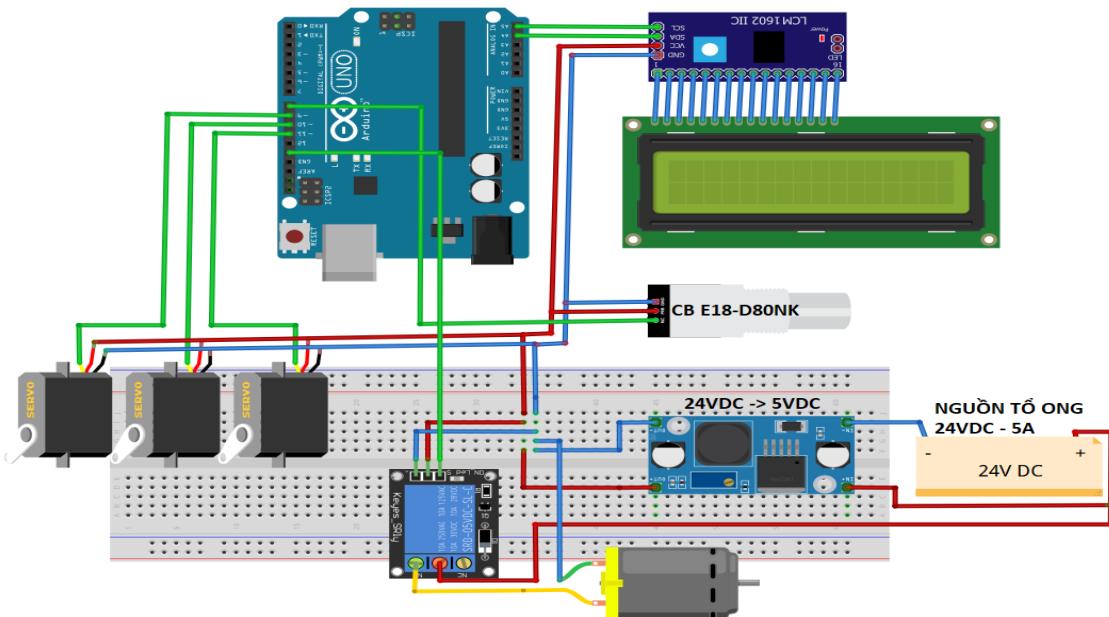
- ❖ Nguồn cấp cho cảm biến E18-D80NK, động cơ Servo MG996R, LCD 16x2, Relay 5V

Các thiết bị này có điểm chung là điều hoạt động tốt với điện áp 5VDC, chỉ khác dòng điện hoạt động.

- Cảm biến E18 – D80NK hoạt động với dòng điện $I_{max} = 300mA$.
- LCD 16x2 hoạt động với dòng $max = 2.5mA$.
- Động cơ Servo MG996R hoạt động với dòng điện $I_{max} = 900mA$.
- Relay 5V hoạt động với dòng điện cho phép qua tối đa $I_{max} = 10A$, dòng kích $I_{kích} = 5mA$.

Dòng điện tổng khoảng 1203 mA, vì vậy nhóm quyết định sử dụng mạch hạ áp LM2596 để chuyển điện áp 24VDC – 5A từ nguồn tổ ong còn 5VDC – 5A để cấp cho các thiết bị này.

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ



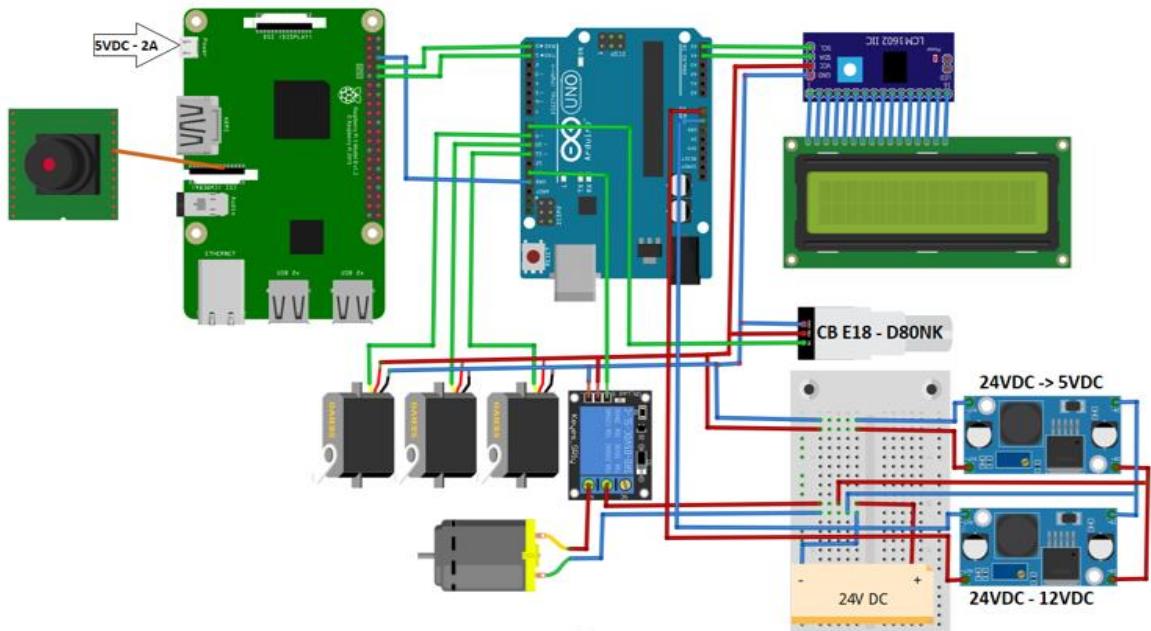
Hình 3.19: Mạch cấp nguồn cho Servo MG996, Relay 5V, Cảm biến E18-D80NK, LCD16x2



Hình 3.20: Nguồn tố ống 24VDC – 5A

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

3.2.3 Sơ đồ kết nối toàn mạch



Hình 3.21: Sơ đồ kết nối toàn mạch

CHƯƠNG 4: THI CÔNG HỆ THỐNG

4.1 GIỚI THIỆU

Đề tài xử lý ảnh hiện nay đã và đang đi vào trong giảng đường đại học từ sơ cấp đến nâng cao. Vì vậy, đề tài “Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm” được ứng dụng thông qua việc xử lý hình ảnh, nhận dạng hình ảnh và phân loại sản phẩm theo màu sắc. Ở phần này sẽ trình bày quá trình thi công hệ thống.

Với đề tài “Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm” bao gồm: Mô hình phân loại sản phẩm (sử dụng sản phẩm là các phôi) theo màu sắc (đỏ, xanh, vàng). Có thể phát hiện những sản phẩm không thuộc 3 màu trên.

Đề tài chỉ xây dựng với mô hình nhỏ và gọn không làm thành dây chuyền sản xuất. Mô hình được kết nối qua các dây tín hiệu nên sẽ không có phần thiết kế và thi công bo mạch.

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: đè, ròng rọc, dây băng tải
2	LCD	Điện áp hoạt động: 5V Dòng: 2.5mA	LCD16x2N	
3	Module I2C	Điện áp hoạt động: 5V	Module I2C	
4	Kit Arduino	Điện áp hoạt động: 5V Dòng: 50mA	Arduino Uno R3	
5	Kit Raspberry	Điện áp hoạt động: 5V Dòng: 1A	Kit Raspberry	Có quạt tản nhiệt
6	Camera Pi	Điện áp hoạt động: 5V Dòng: 250mA	Camera Pi	6

CHƯƠNG 4: THI CÔNG HỆ THỐNG

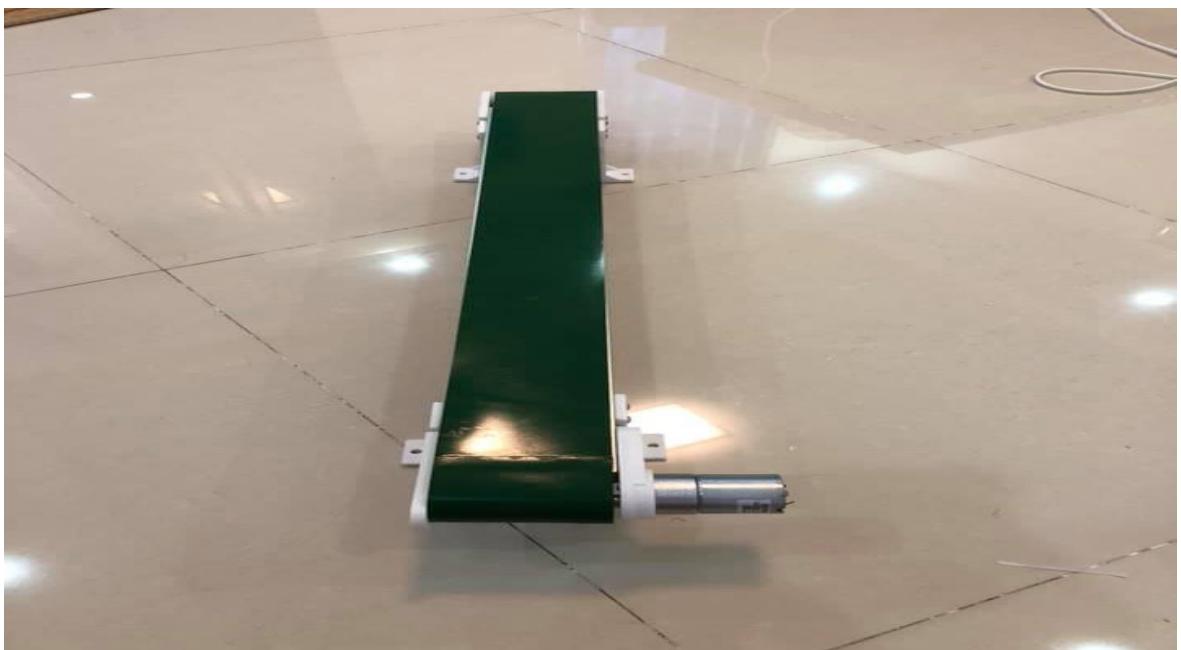
7	Servo MG996R	Điện áp hoạt động: 5V Dòng: 900mA	Nhựa cứng	
8	Cảm biến	Điện áp hoạt động: 5V Dòng: 300mA	Nhựa	
9	Mạch hạ áp LM2596	Đầu vào: 3V-40V Đầu ra: 1V-30V Dòng: 3A	LM2596	
10	Relay	Điện áp hoạt động: 5V Dòng cho phép qua tối đa: 10A Dòng kích: 5mA	Nhựa	
11	Domino		Nhựa cứng	
12	Led đơn	Điện áp hoạt động: 3V Dòng: 15mA	Led	
13	Bo mạch		Nhựa cứng	
14	Nguồn tổ ong	Điện áp vào: 220 VAC Điện áp ra: 24VDC Dòng ra: 10A	Thép	
15	Adapter	Điện áp vào: 220 VAC Điện áp ra: 5VDC Dòng ra: 2A	Nhựa cứng	
16	Động cơ DC	Điện áp hoạt động: 24V Dòng: 900mA	Hợp kim	

4.2.2 Lắp ráp và kiểm tra

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

a. Phản Băng tải

Lắp băng tải và động cơ thông qua chi tiết in 3D.

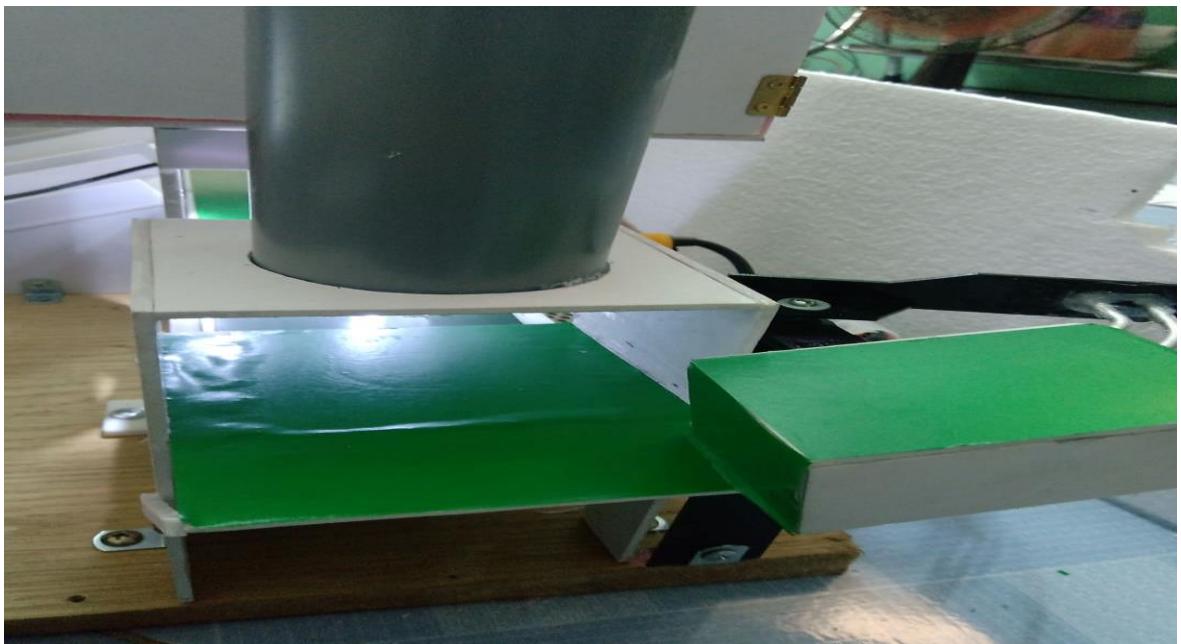


Hình 4.1: Băng tải và động cơ DC

b. Phần máng trượt

❖ Máng trượt chia làm hai phần:

- Phần thứ nhất là máng đẩy sản phẩm vào.



Hình 4.2: Máng đưa sản phẩm vào

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Phần thứ hai là máng đưa sản phẩm sau khi được phân loại màu.

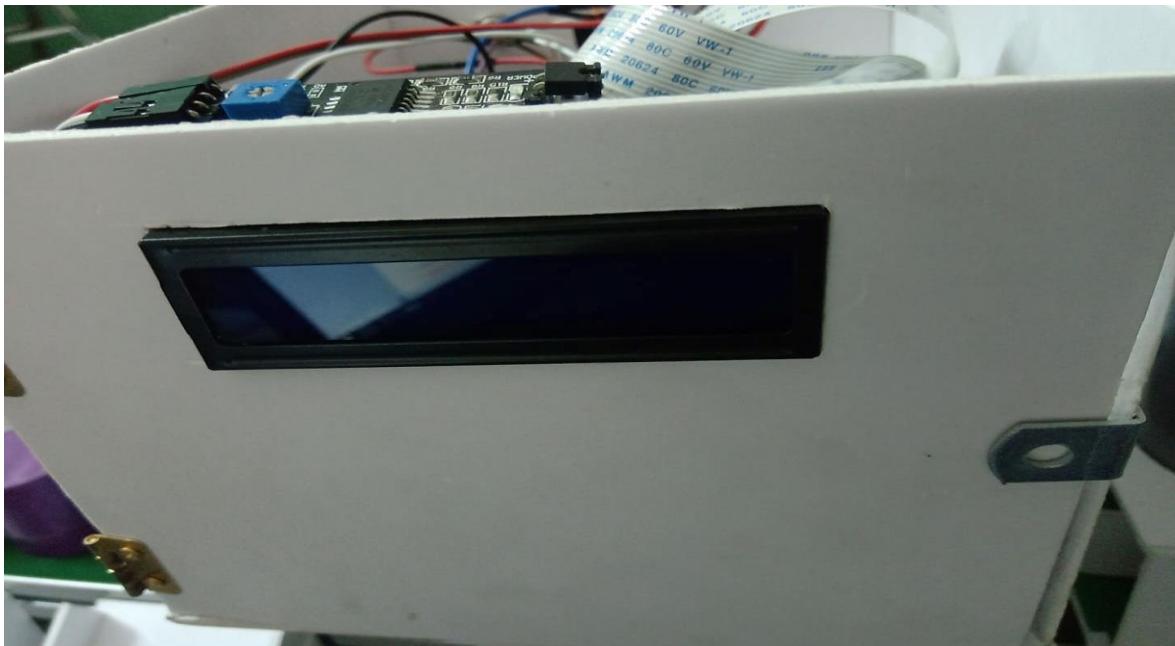


Hình 4.3: Máng đưa sản phẩm sau khi phân loại

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

c. Phần hiển thị

Kết nối module LCD16X2 I2C vào khói điều khiển và được đặt bên ngoài hộp điều khiển.



Hình 4.4: Màn hình LCD16x2 hiển thị kết quả

d. Phần module điều khiển

- ❖ Lắp ráp Raspberry với camera và Arduino Uno

CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Kết nối thông qua giao tiếp Uart giữa Raspberry và Arduino Uno bao gồm chân Tx của Raspberry nối với Rx của Arduino Uno và Rx của Raspberry nối với Tx của Arduino Uno.

- Kết nối Raspberry với Camera thông qua socket CSI Camera có sẵn trên kit Raspberry bằng cáp 15pin ribbon.

❖ Lắp ráp Cảm biến, Servo Mg996R với Arduino Uno

- Kiểm tra chính xác màu sắc và chức năng từng loại dây để việc kết nối trở nên thuận tiện.

- Cảm biến bao gồm 3 dây: Đen là dây tín hiệu được kết nối vào pin 8 của Arduino và dây nâu được nối vào nguồn 5v, dây xanh được nối GND.

- Servo Mg996R bao gồm 3 dây: Cam là dây tín hiệu được nối vào pin 6,9,10,11 của Arduino và dây đỏ được nối vào nguồn 5v, dây nâu được nối GND.

❖ Lắp ráp Relay 5V với Động cơ giảm tốc DC và Arduino Uno

- Khi kết nối Relay 5v với động cơ giảm tốc Dc bao gồm hai loại kết nối: thường đóng và thường mở.

- Chọn kết nối phù hợp và tiện lợi cho việc lập trình trên kit Arduino. Tiếp điểm thường đóng khi lập trình trạng thái mức 1 là động cơ hoạt động và mức 0 là động cơ ngưng và ngược lại đối với tiếp điểm thường mở .

❖ Lắp ráp Lm2596 với Cảm biến, Servo Mg996R, Lcd, Relay

- Hiệu chỉnh điện áp ngõ ra của Lm2596 từ 24V xuống 5V cấp cho Cảm biến, Servo, Lcd, Relay thông qua việc vẫn biến trở.

- Kết nối các dây nguồn và GND của các module với module Lm2596.

❖ Nguồn Adapter 5V kết nối với Raspberry pi 3

- Kiểm tra nguồn có kết nối được với Raspberry.

- Kiểm tra Raspberry có lên nguồn chưa. Nếu chưa thì kiểm tra lại kết nối.

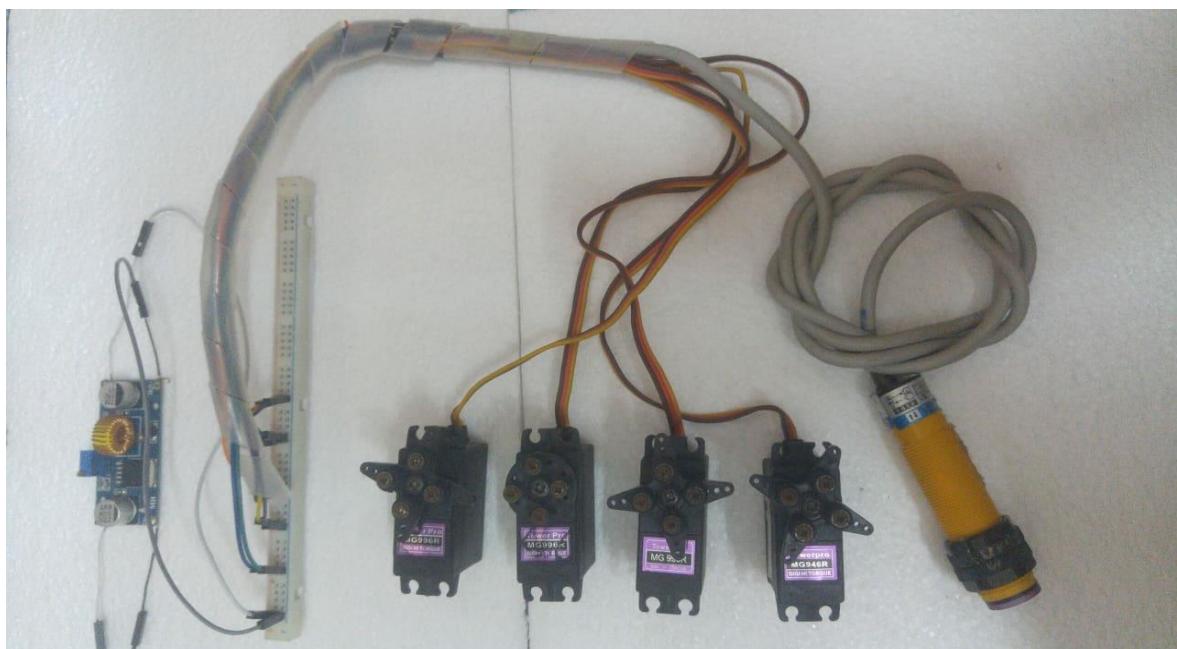
❖ Hình ảnh thực tế sau khi kết nối giữa Raspberry với Arduino Uno.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.5: Hình ảnh thực tế kết nối Raspberry và Arduino

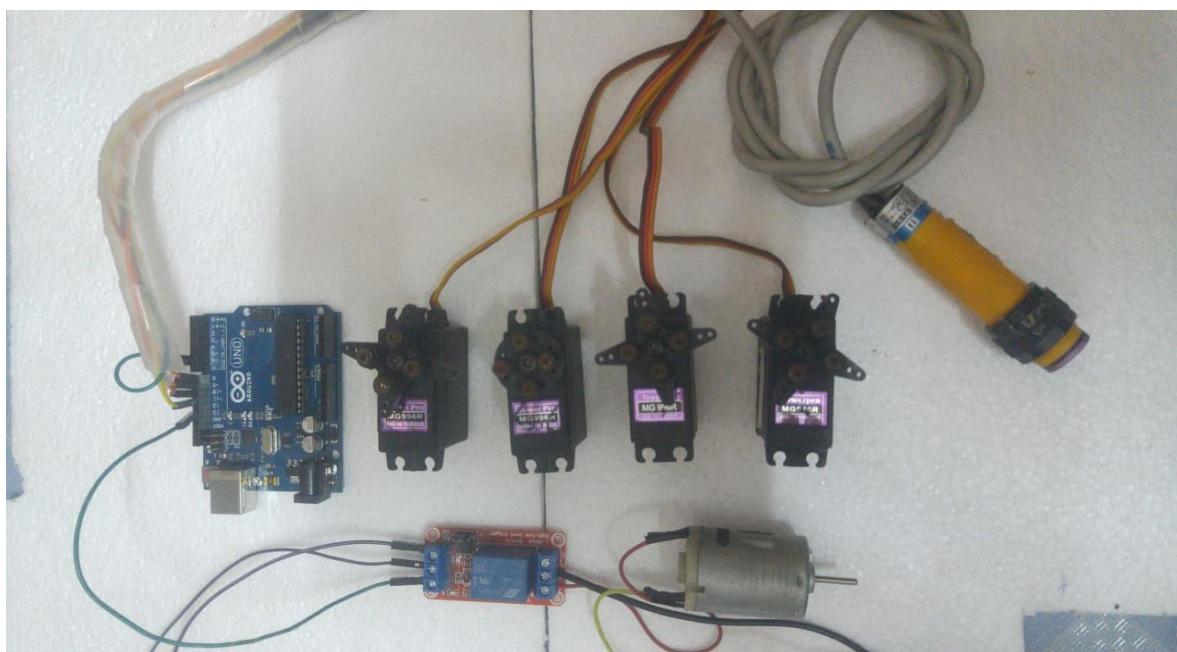
- ❖ Hình ảnh thực tế sau khi lắp ráp giữa Lm2596 với Servo Mg996r, Cảm biến.



Hình 4.6: Lắp ráp giữa Lm2596 với Servo Mg996r, Cảm biến

- ❖ Hình ảnh thực tế sau khi lắp ráp giữa Arduino Uno với Servo Mg996r, Cảm biến, Relay 5V.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.7: Lắp ráp giữa Arduino Uno với Servo Mg996r, Cảm biến, Relay 5V

- ❖ Hình ảnh thực tế lắp ráp thành công toàn mô hình .

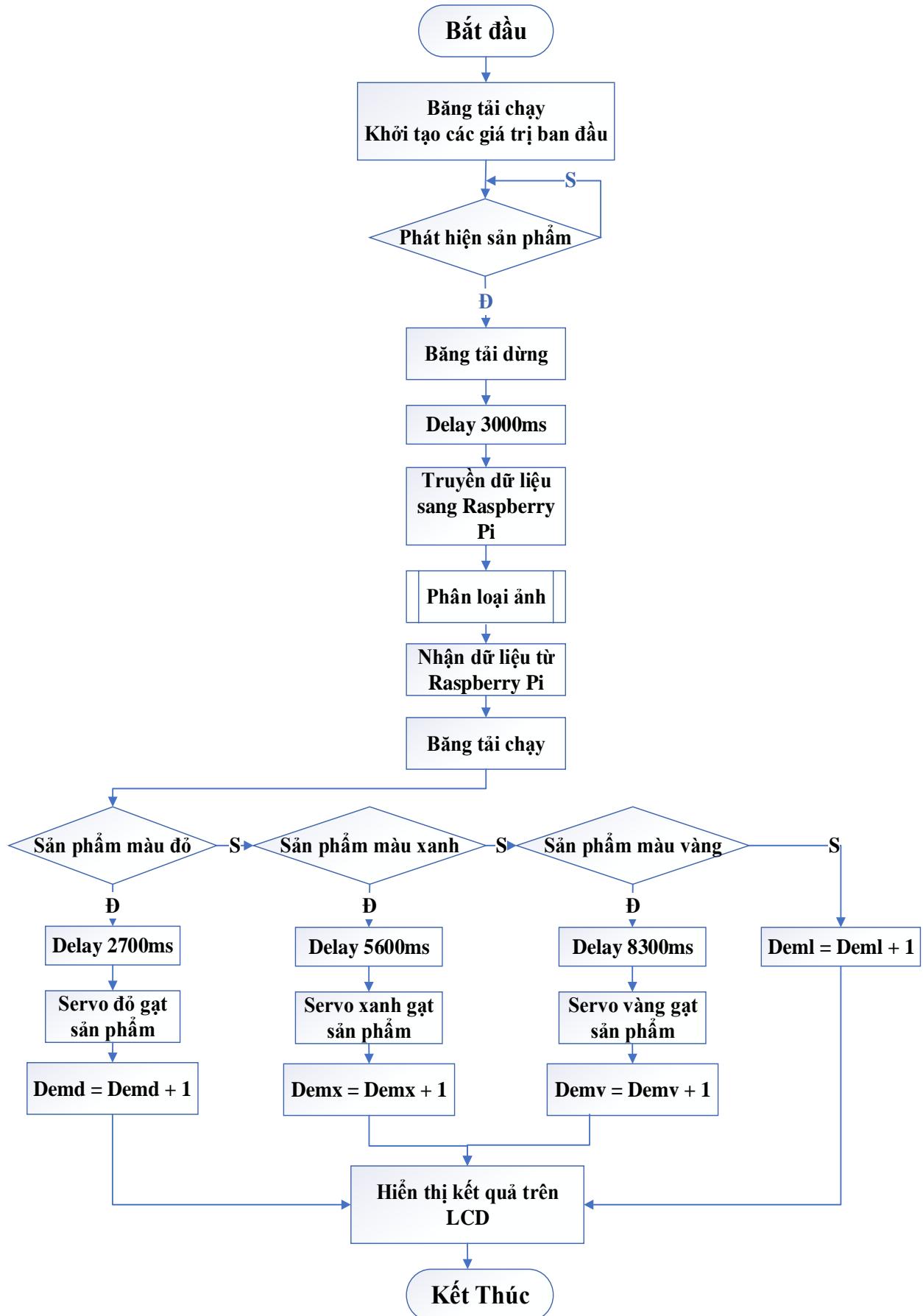


Hình 4.8: Mô hình toàn hệ thống

4.3 LẬP TRÌNH HỆ THỐNG

4.3.1 Lưu đồ giải thuật trên Arduino

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.9: Lưu đồ giải thuật trên Arduino Uno

CHƯƠNG 4: THI CÔNG HỆ THỐNG

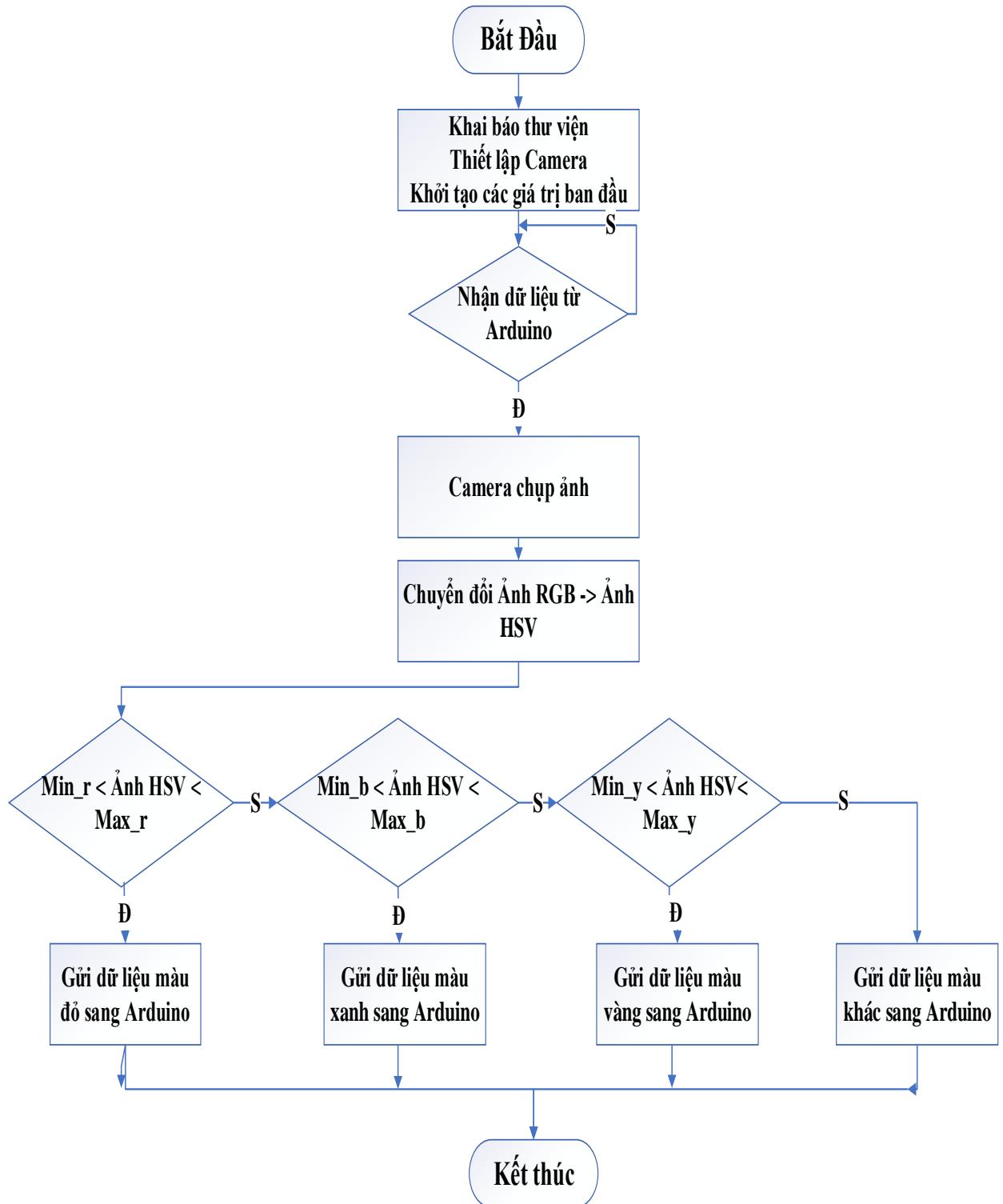
❖ Giải thích lưu đồ

- Bắt đầu
- Động cơ băng tải hoạt động và gán các tín hiệu đếm sản phẩm của từng màu và sản phẩm khác bằng giá trị ban đầu là 0.
- Chờ cảm biến hồng ngoại phát hiện được sản phẩm, nếu đúng thì xử lý chương trình kế tiếp. Ngược lại là sai thì tiếp tục chờ đến khi nào cảm biến phát hiện được sản phẩm đi qua.
- Khi cảm biến phát hiện có vật, băng tải dừng lại 3s, lúc đó Arduino sẽ gửi tín hiệu ra lệnh qua Raspberry Pi, Raspberry Pi tiến hành chụp ảnh thông qua Camera Pi.
- Chương trình bên kit Raspberry sẽ xử lý hình ảnh chụp được, nhận diện được màu sắc và truyền ngược lại cho kit Arduino bằng chuỗi dữ liệu đã cài đặt, động cơ tiếp tục chạy sau 3s trì hoãn.
- Nếu sản phẩm là màu đỏ thì sau 2,7s servo đỏ sẽ gạt sản phẩm màu đỏ vào máng trượt màu đỏ và đếm giá trị demd cộng lên 1 và hiển thị ra màn hình lcd.
- Nếu sản phẩm là màu xanh thì sau 5,6s servo xanh sẽ gạt sản phẩm màu xanh vào máng trượt màu xanh và đếm giá trị demx cộng lên 1 và hiển thị ra màn hình lcd.
- Nếu sản phẩm là màu vàng thì sau 8,3s servo vàng sẽ gạt sản phẩm màu vàng vào máng trượt màu vàng và đếm giá trị demv cộng lên 1 và hiển thị ra màn hình lcd.
- Nếu cả ba điều kiện trên điều sai thì sản phẩm còn lại sẽ được đi thẳng ra máng trượt còn lại và đếm giá trị deml cộng lên 1.
- Kết thúc quá trình và tiếp tục chờ khi có sản phẩm tiếp theo.

4.3.2 Lưu đồ giải thuật xử lý ảnh trên Raspberry

Là chương trình con “Phân loại ảnh” của lưu đồ ở trên.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.10: Lưu đồ giải thuật xử lý ảnh trên Raspberry Pi

❖ Giải thích lưu đồ

- Bắt đầu
- Khai báo các thư viện cần sử dụng trong lập trình, thiết lập các lệnh cần thiết cho camera, gán các giá trị min max HSV của 3 màu.
- Sau đó chờ tín hiệu ra lệnh từ kit Arduino Uno truyền qua, nếu đúng thì xử lý

CHƯƠNG 4: THI CÔNG HỆ THỐNG

tiếp chương trình. Ngược lại là sai thì tiếp tục chờ.

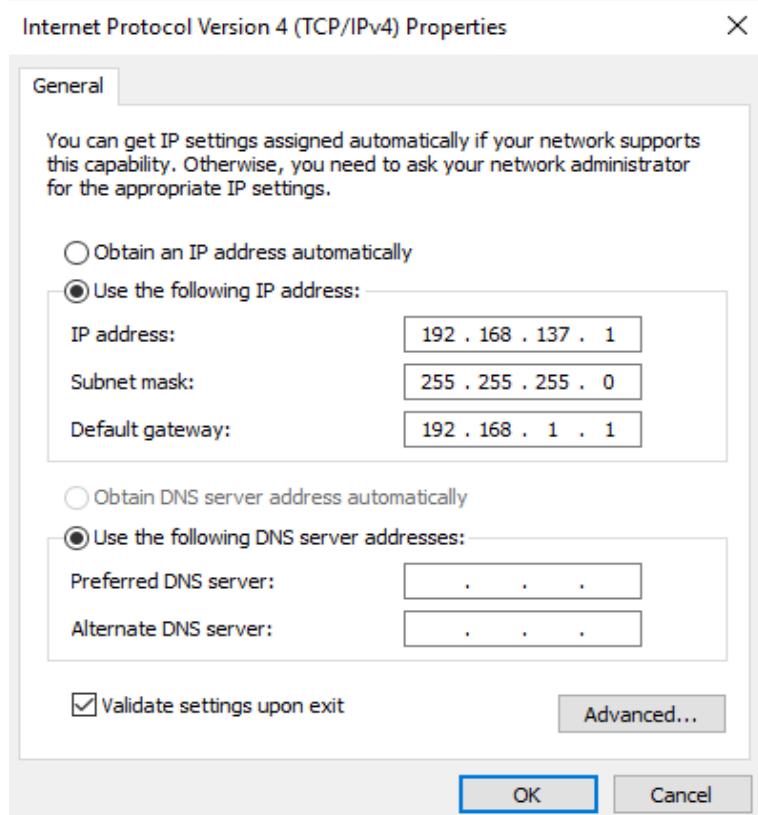
- Khi có tín hiệu ra lệnh, Camera sẽ tiến hành chụp ảnh và xử lý chuyển đổi ảnh màu chụp được từ RGB sang HSV.
- Tiến hành so sánh ảnh HSV vừa chuyển đổi với các giá trị HSV mặc định ở trên.
 - Nếu giá trị nằm trong khoảng Max_r và Min_r thì sẽ tiếp hành gửi dữ liệu màu đỏ sang kit Arduino Uno.
 - Nếu giá trị nằm trong khoảng Max_b và Min_b thì sẽ tiếp hành gửi dữ liệu màu xanh sang kit Arduino Uno.
 - Nếu giá trị nằm trong khoảng Max_y và Min_y thì sẽ tiếp hành gửi dữ liệu màu vàng sang kit Arduino Uno.
 - Nếu giá trị không nằm trong các khoảng Max và Min nào thì sẽ tiếp hành gửi dữ liệu không có màu sang kit Arduino Uno.
- Kết thúc quá trình, Raspberry tiếp tục chờ thực hiện mệnh lệnh tiếp theo.

4.3.3 Các bước tuy cập, lập trình trên Python

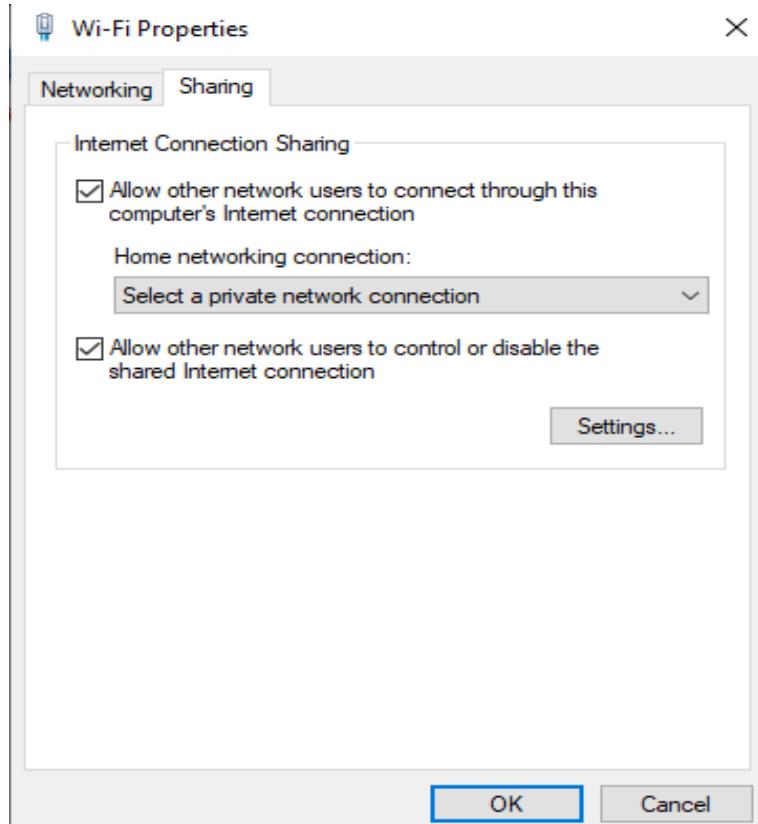
Ngay sau khi cài đặt xong Raspbian lên Raspberry Pi 3, có hai cách điều khiển Raspberry Pi 3 thông dụng:

- ❖ Cách 1: Điều khiển trực tiếp: Dùng chuột, bàn phím màn hình gắn vào Raspberry Pi điều khiển trực tiếp thông qua các cổng giao tiếp có sẵn.
- ❖ Cách 2: Điều khiển qua laptop: Sử dụng dây LAN để kết nối giữa Raspberry và máy tính. Dùng phần mềm Remote Desktop Connection để truy cập vào địa chỉ IP của Raspberry Pi 3. Gồm các thao tác sau:
 - Tạo địa chỉ IP tĩnh cho Raspberry và máy tính, cho phép máy tính chia sẻ mạng cho Raspberry.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



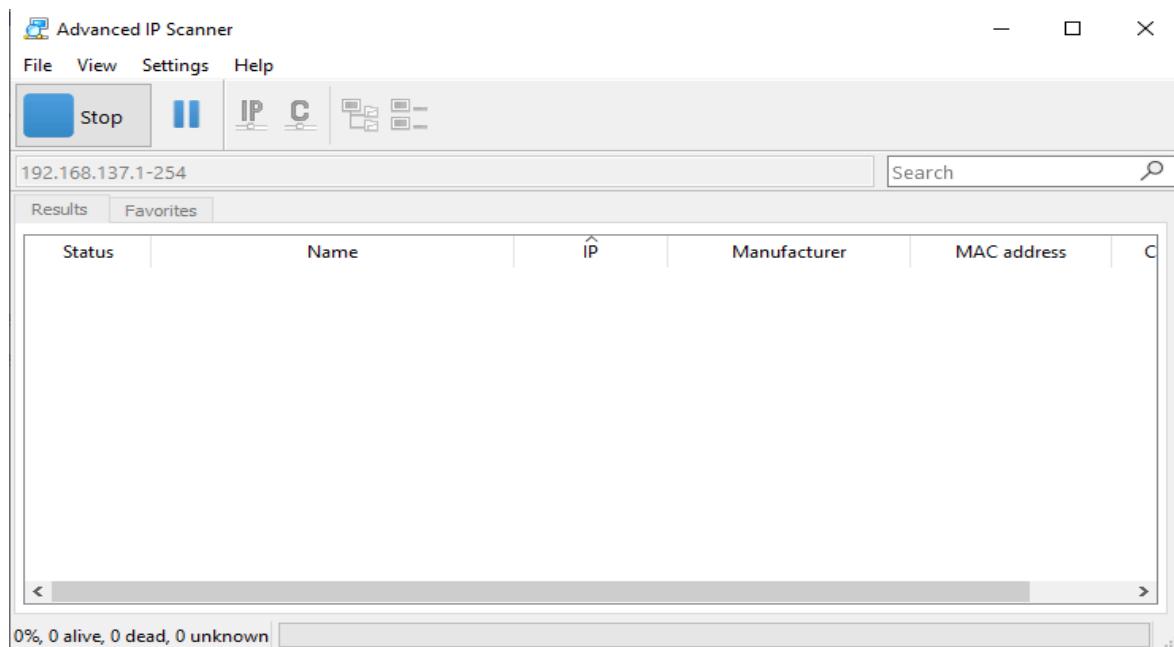
Hình 4.11: Tạo địa chỉ IP tĩnh



Hình 4.12: Cho phép chia sẻ mạng Lan

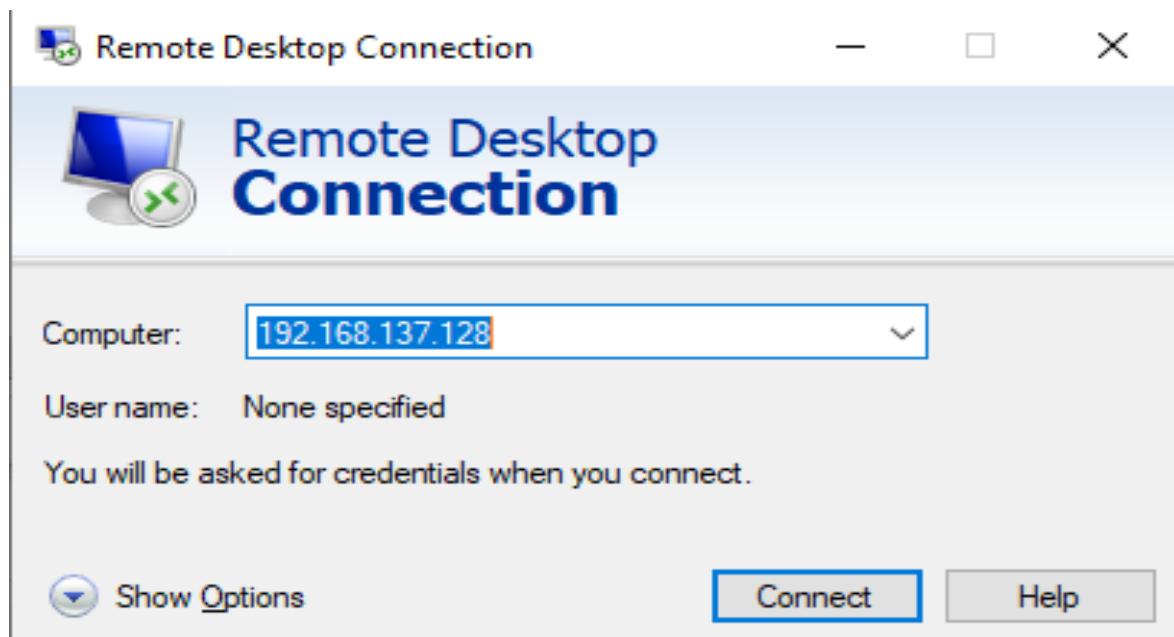
CHƯƠNG 4: THI CÔNG HỆ THỐNG

Sau đó sử dụng phần mềm Advanced IP Scanner để tìm địa chỉ IP chính xác và kiểm tra xem Raspberry có hoạt động hay không.



Hình 4.13: Phần mềm Advanced IP Scanner

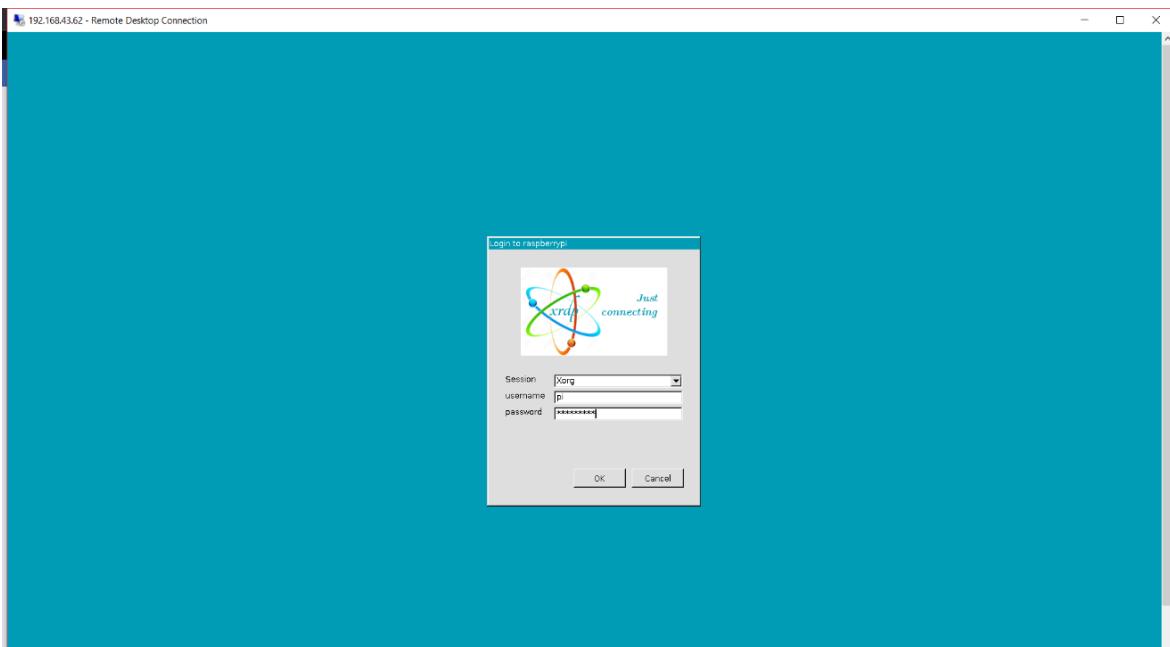
Dùng phần mềm Remote Desktop Connection để truy cập vào hệ điều hành Raspbian của Raspberry Pi 3.



Hình 4.14: Remote Desktop Connection

Sau đó nhập tài khoản, mật khẩu để truy cập vào hệ điều hành Raspbian, mặc định tài khoản: “pi”, mật khẩu: “raspbian”.

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.15: Giao diện đăng nhập vào Raspberry

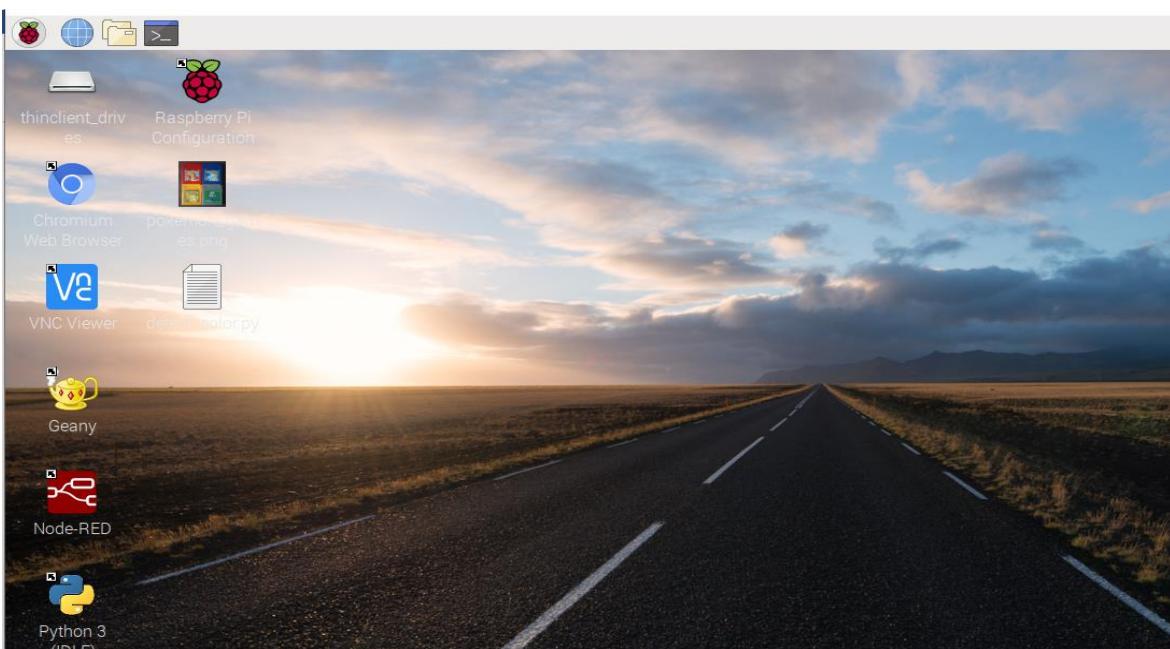
Trong cách 1 có nhiều giới hạn và bất tiện do phải gắng màn hình, chuột, bàn phím vào RPI. Vì vậy nhóm quyết định sử dụng cách 2 để lập trình trên Raspberry Pi.

4.3.4 Hướng dẫn sử dụng và thao tác

Có 2 cách để cho Raspberry hoạt động: Hoạt động thủ công và hoạt động tự động.

❖ Hoạt động thủ công:

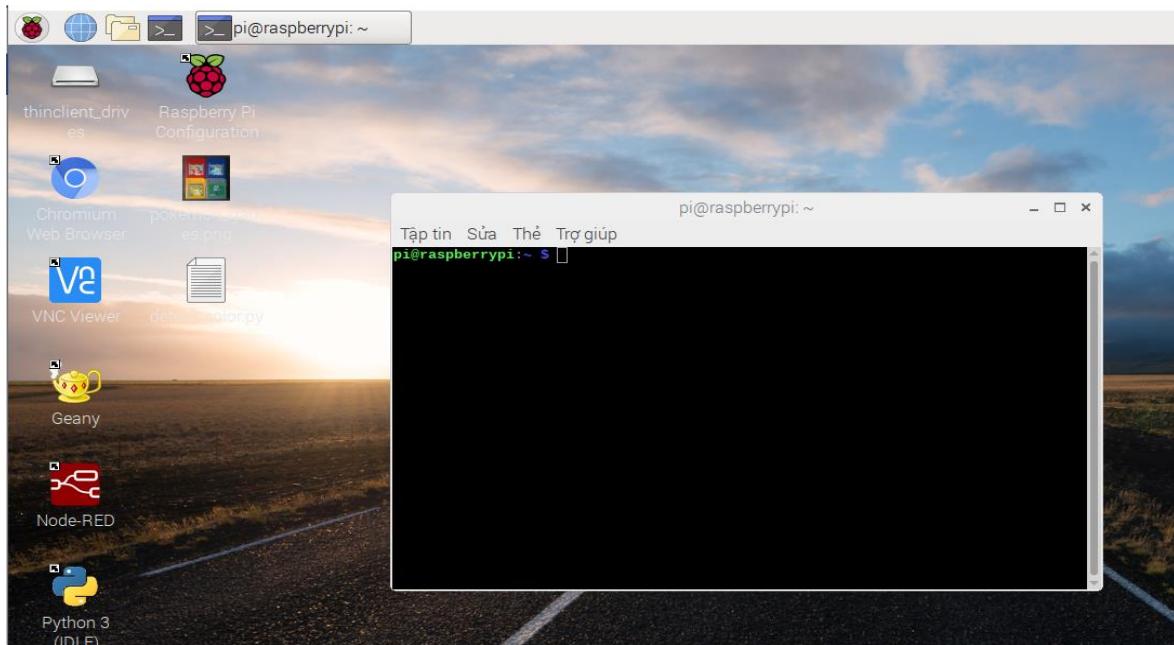
Bước 1: Sau khi cấp nguồn cho Raspberry, ta kết nối giữa máy tính và Raspberry thông qua dây LAN, tiến hành truy cập vào hệ điều hành của Raspberry như đã trình bày ở phần 4.3.3, giao diện hệ điều hành Raspbian xuất hiện.



Hình 4.16: Giao diện Raspberry

CHƯƠNG 4: THI CÔNG HỆ THỐNG

Bước 2: Truy cập và cửa sổ Terminal để thực thi chương trình đã viết.



Hình 4.17: Cửa sổ Terminal

Bước 3: Tại cửa sổ Terminal, ta tiến hành gọi tên chương trình đã viết nhưng ở đây do sử dụng thư viện OpenCV nên ta cần phải gọi thư viện ra trước.

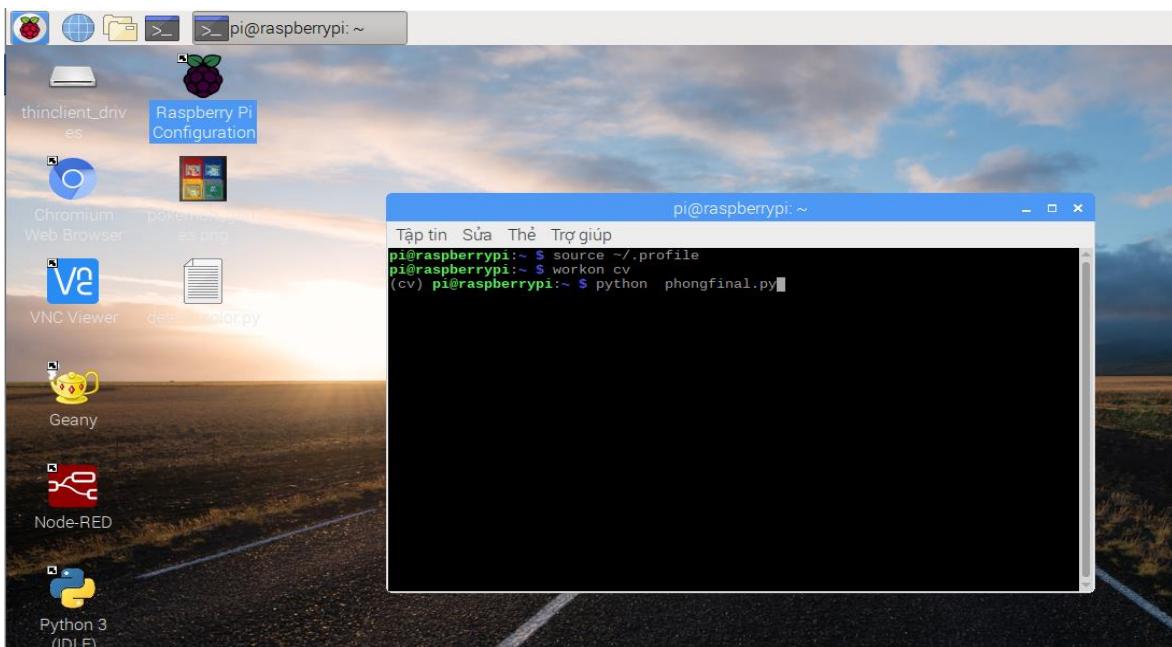
Lệnh gọi OpenCV:

- Source ~/.profile
- workonCV

Sau khi đã gọi thư viện OpenCV, tiếp tục gọi chương trình cần thực thi, ở đây nhóm đặt tên chương trình là: “phongfinal.py”.

- Python phongfinal.py

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.18: Gọi lệnh thực thi chương trình

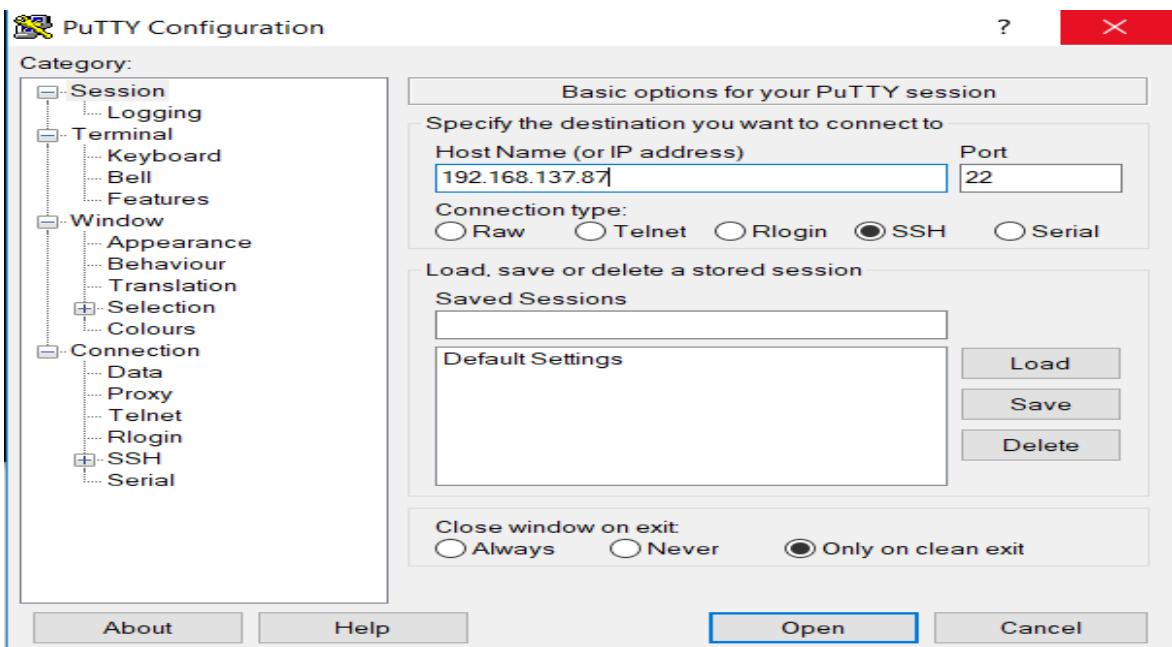
❖ **Hoạt động tự động**

Hoạt động tự động nghĩa là sau khi cấp nguồn vào Raspberry, Raspberry sẽ tự động chỉ đến đường dẫn và thực thi chương trình mà không cần phải trải qua các bước như ở hoạt động thủ công. Để thực hiện được điều đó:

Bước 1: Sau khi cấp nguồn cho Raspberry, ta kết nối giữa máy tính và Raspberry thông qua dây LAN, tiến hành truy cập đến Terminal của Raspberry bằng phần mềm PuTTY.

Ở đây ta nhập địa chỉ ID của Raspberry là: 192.168.137.87

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.19: Cửa sổ PuTTY

Bước 2: Truy cập cửa sổ Terminal và đăng nhập vào raspbian để thực thi chương trình đã viết.

Tài khoản: pi

Mật khẩu: raspbian

```
pi@raspberrypi: ~
login as: pi
pi@192.168.137.87's password:
Linux raspberrypi 4.19.23-v7+ #1203 SMP Tue Feb 19 23:26:17 GMT 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

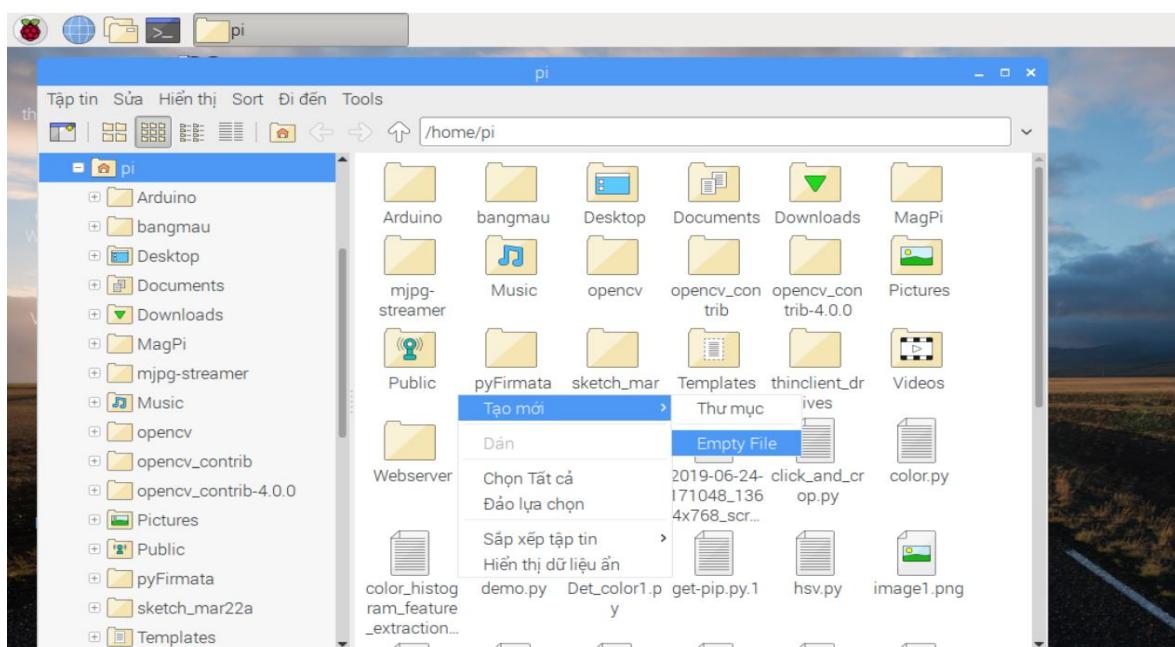
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 18 21:45:05 2019 from 192.168.137.1
pi@raspberrypi:~ $ nano start-final.sh
pi@raspberrypi:~ $
```

Hình 4.20: Cửa sổ Terminal

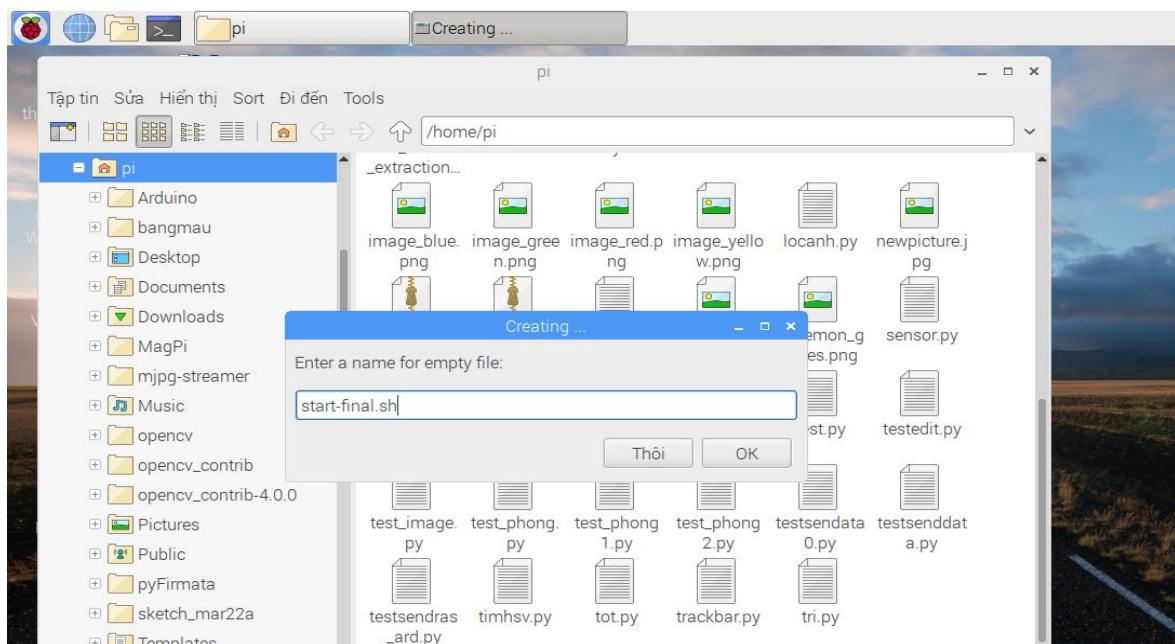
Bước 3: Tạo một file để lưu các lệnh cần thực thi

Ta cần truy cập vào hệ điều hành Raspberry để tạo 1 file với đuôi .sh trong đường dẫn home/pi lưu các lệnh cần thực thi.

CHƯƠNG 4: THI CÔNG HỆ THỐNG

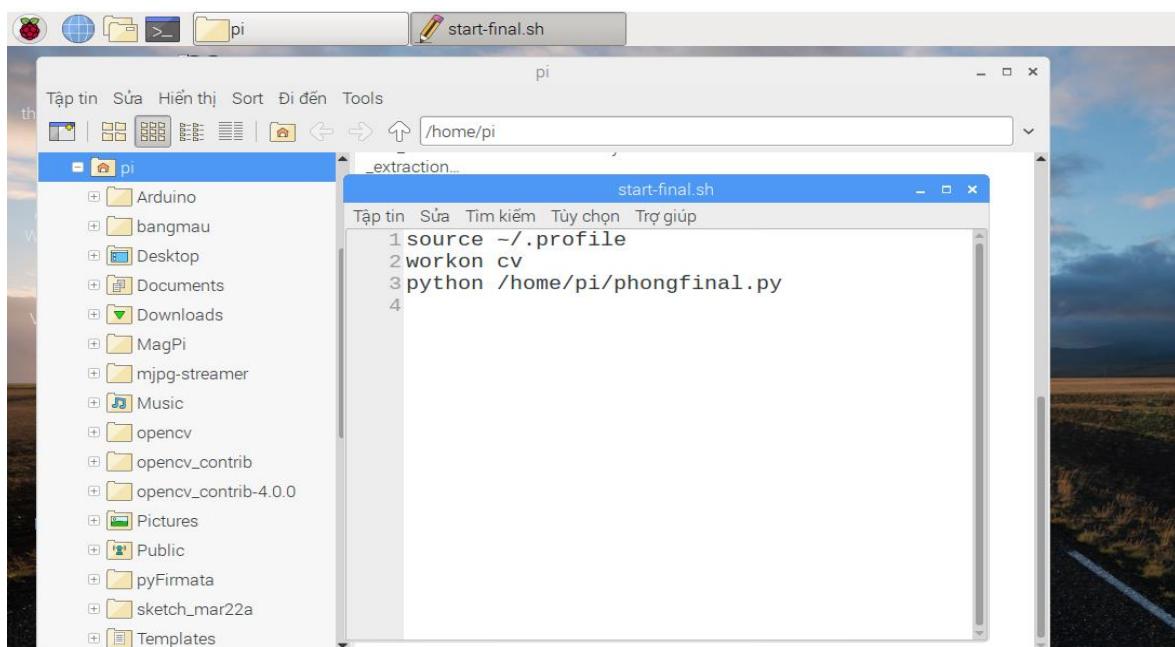


Hình 4.21: Tạo một file.sh trong đường dẫn home/pi



Hình 4.22: Đặt tên file với đuôi .sh

CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4.23: Lưu các lệnh cần thực thi

Bước 4: Mở tất cả các file đã có trong Raspberry và tiến hành mở file chạy tự động

Tiến hành gọi lệnh chạy tự động tại màn hình Terminal

- Nano start_final.hs
- Export VISUAL=nano; crontab -e

A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~'. The terminal shows the following command being run:

```
pi@raspberrypi:~ $ ls
pi@raspberrypi:~ $ nano start-final.sh
pi@raspberrypi:~ $ export VISUAL=nano; crontab -e
```

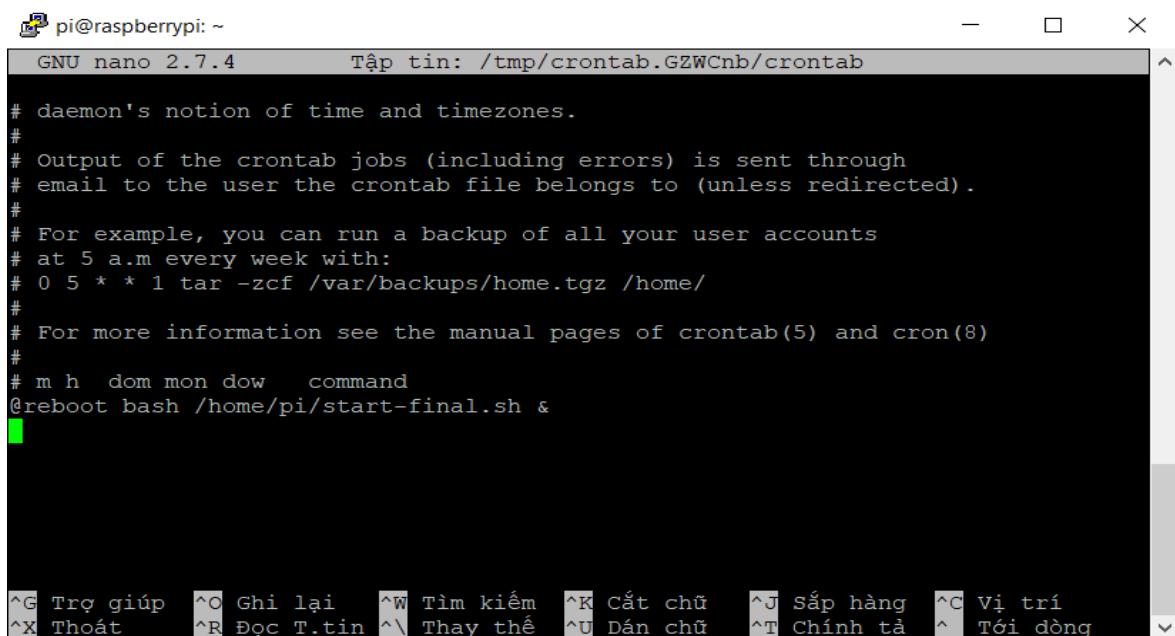
The terminal lists numerous files and folders including: 2019-03-30-122822_1364x768_scrot.png, 2019-06-15-162047_1920x1080_scrot.png, Arduino, bangmau, click_and_crop.py, color_histogram_feature_extraction.py, color.py, demo.py, Desktop, Det_color1.py, Documents, Downloads, get-pip.py.1, hsv.py, image1.png, image_blue.png, image_green.png, image_red.png, image_yellow.png, locanh.py, MagPi, mjpg-streamer, Music, newpicture.jpg, opencv, opencv_contrib, opencv_contrib-4.0.0, opencv_contrib.zip, opencv.zip, phongfinal.py, PictureName.jpg, Pictures, pokemon_games.png, Public, pyFirmata, sensor.py, sketch_mar22a, start-final.sh, start.sh, stop.sh, take_pic1.py, take_pic.py, Templates, testedit.py, test_image.py, test_phong1.py, test_phong2.py, test_phong.py, test.py, testsendata0.py, testsenddata.py, testsendras_ard.py, thinclient_drives, timhsv.py, tot.py, trackbar.py, tri.py, Videos, Webserver.

Hình 4.24: Lệnh chạy tự động

Thêm đường dẫn của file cần chạy tự động:

- /home/pi/start-final.sh &

CHƯƠNG 4: THI CÔNG HỆ THỐNG



```
pi@raspberrypi: ~
GNU nano 2.7.4          Tập tin: /tmp/crontab.GZWCnb/crontab

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
@reboot bash /home/pi/start-final.sh &
```

^G Trợ giúp ^O Ghi lại ^W Tìm kiếm ^K Cắt chữ ^J Sắp hàng ^C Vị trí
^X Thoát ^R Đọc T.tin ^\ Thay thế ^U Dán chữ ^T Chính tả ^_ Tới dòng

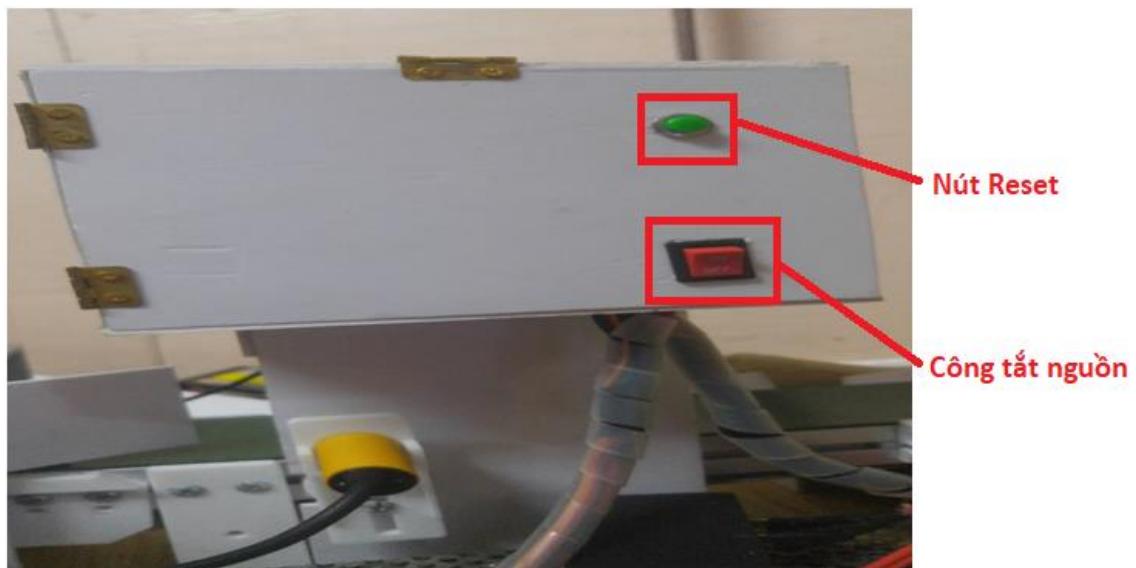
Hình 4.25: Thêm đường dẫn file cần chạy tự động

❖ Hướng dẫn sử dụng phần cứng

Sau khi đã cấp nguồn, tiến hành bật công tắc nguồn (công tắc màu đỏ) trên hộp điều khiển để khởi động hệ thống. Ở đây nhóm đã thiết lập chế độ chạy tự động nên khi mở công tắc nguồn đợi khoảng 15 giây khởi động, hệ thống sẽ tiến hành phân loại.

Đối với trường hợp chưa thiết lập chế độ chạy tự động, sau khi bật công tắc nguồn, ta tiến hành thực hiện các bước của hoạt động thủ công đã nêu ở trên.

Để thiết lập lại toàn bộ giá trị như ban đầu, ta nhấn nhả nút Reset (nút màu xanh lá) trên hộp điều khiển.



Hình 4.26: Công tắc nguồn và nút nhấn Reset

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

Mục tiêu ban đầu của đề tài “Ứng dụng xử lý ảnh trong hệ thống phân loại sản phẩm” là phân loại sản phẩm theo màu sắc (màu đỏ, màu xanh, màu vàng). Dựa trên ngôn ngữ Python với thư viện chính là OpenCV và được thực hiện trên Kit Raspberry và Arduino Uno. 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 14 tuần. Trong đó, gồm các vấn đề sau: Đối với phần cứng: Biết sử dụng Kit Raspberry, Kit Arduino Uno, cách điều khiển động cơ Servo, cài đặt hệ điều hành cho raspberry, biết sử dụng Camera Pi. Đối với phần mềm: Biết cách lập trình cơ bản trên Python cùng với thư viện OpenCV, lập trình trên Arduino IDE, và các giải thuật liên quan đến đề tài như: phương pháp nhận dạng, phân loại.

5.1 KẾT QUẢ

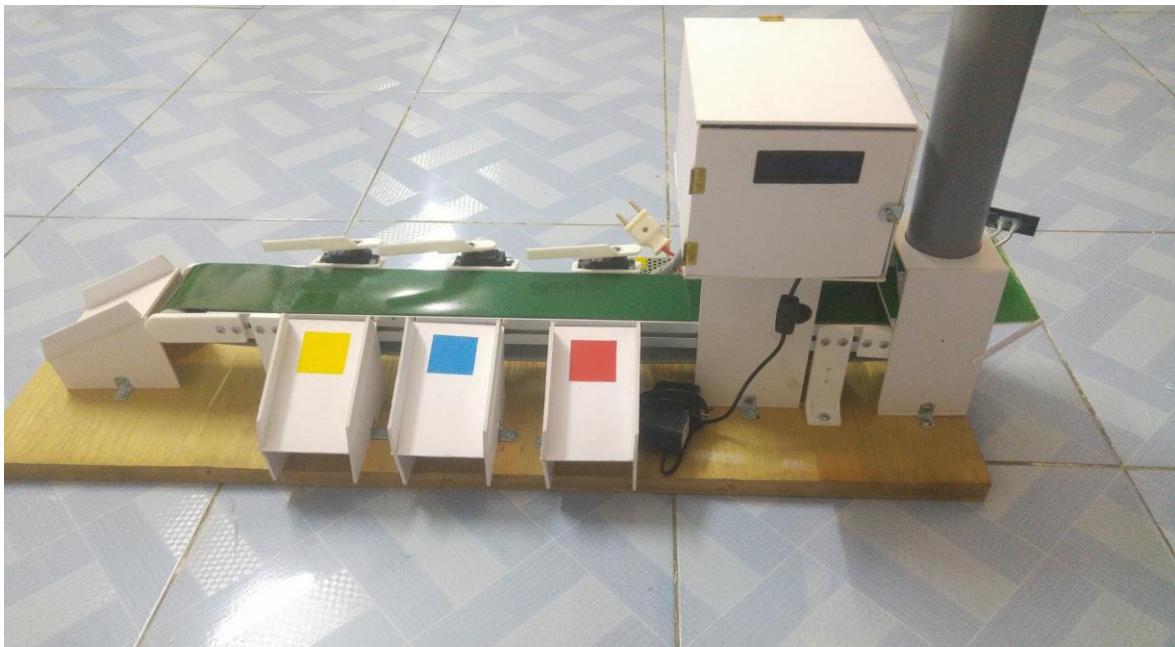
❖ Kết quả đạt được là khá tốt

- Hệ thống nhận dạng và phân loại được sản phẩm theo màu sắc bằng ngôn ngữ Python dựa vào thư viện OpenCV.
- Giao tiếp được Raspberry Pi với Camera Pi.
- Giao tiếp được Raspberry Pi với Arduino Uno.
- Giao tiếp được Arduino với LCD, Cảm biến hồng ngoại, Servo, Relay.
- Thiết kế thành công mô hình phân loại sản phẩm theo màu sắc.

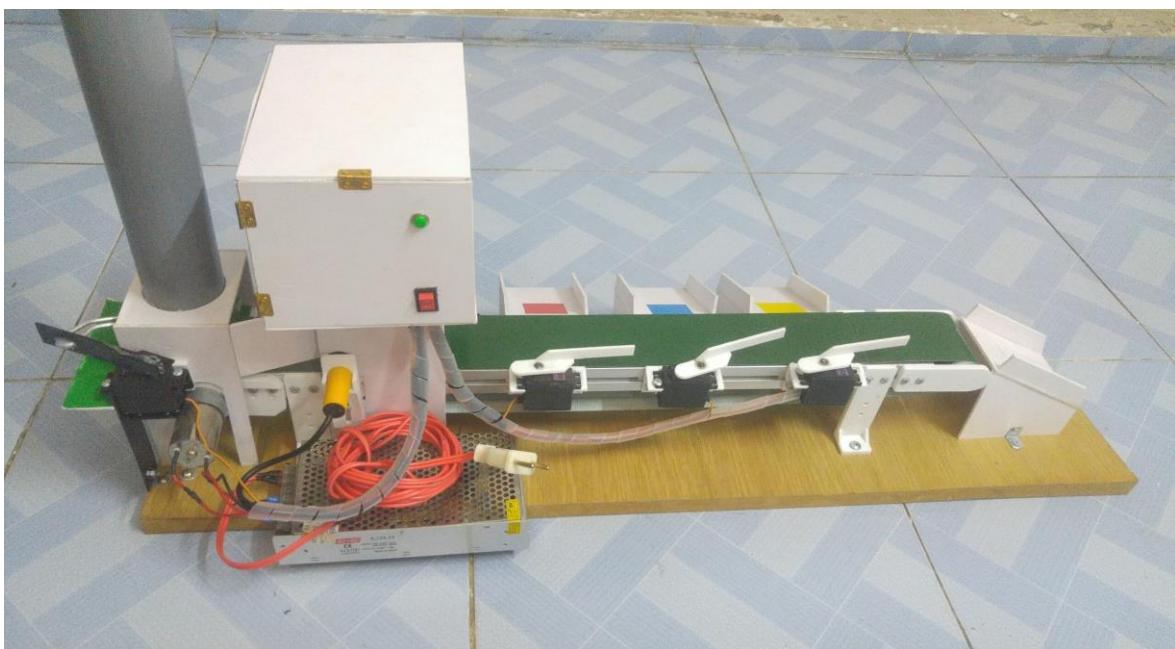
Mô hình hoạt động khá ổn định. Bộ xử lý trung tâm xử lý tín hiệu khá tốt. Phân loại thành công các sản phẩm đỏ, xanh, vàng, nếu không thuộc một trong các màu sắc kể trên thì được xem là sản phẩm khác, hệ thống máng cấp sản phẩm hoạt động ổn định.

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

- ❖ Những hình ảnh kết quả mô hình

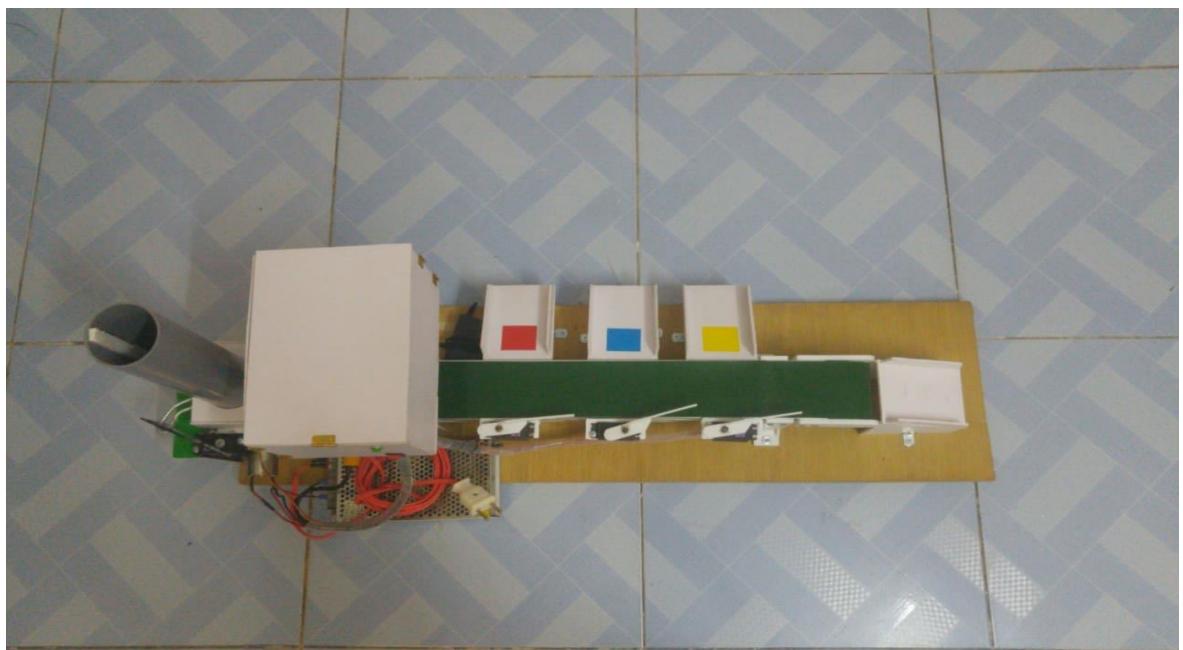


Hình 5.1: Mô hình hoàn thiện chụp từ bên trái



Hình 5.2: Mô hình hoàn thiện chụp từ bên phải

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

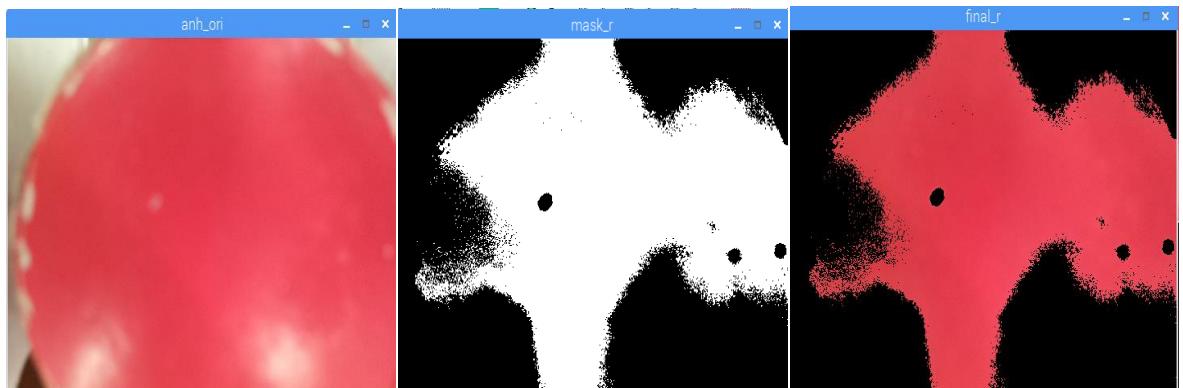


Hình 5.3: Mô hình hoàn thiện chụp từ trên xuống

Để có một kết quả khách quan nhất, nhóm thực hiện xử lý liên tiếp mỗi màu 5 sản phẩm. Điều kiện được chụp trên nền xanh lá (màu của băng tải) và có bóng đèn led để tăng cường cường độ sáng tốt nhất.

- Kết quả của 5 lần chụp liên tiếp sản phẩm màu đỏ:

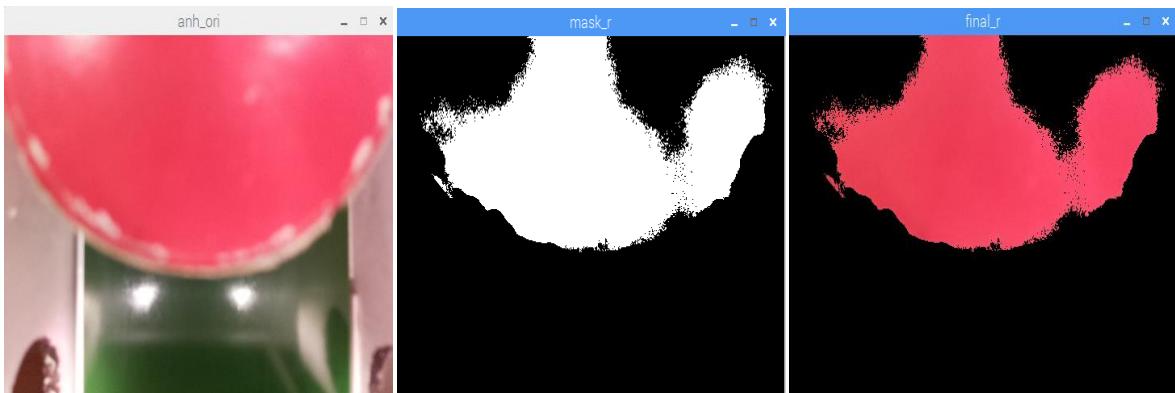
Lần 1:



Hình 5.4: Kết quả nhận dạng màu đỏ lần 1

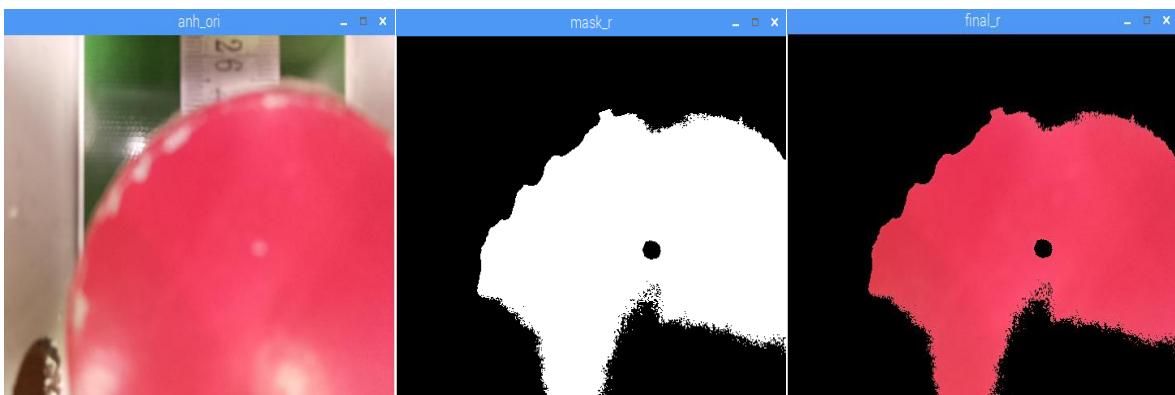
CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

Lần 2:



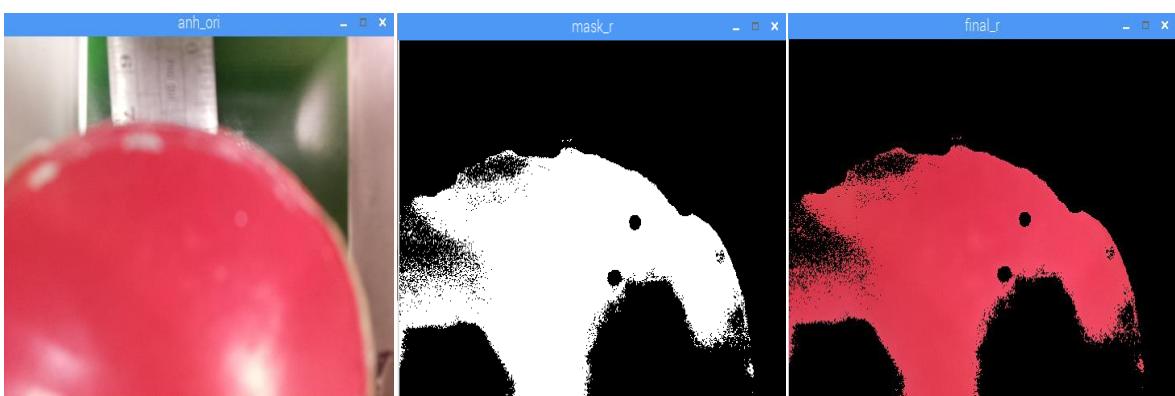
Hình 5.5: Kết quả nhận dạng màu đỏ lần 2

Lần 3:



Hình 5.6: Kết quả nhận dạng màu đỏ lần 3

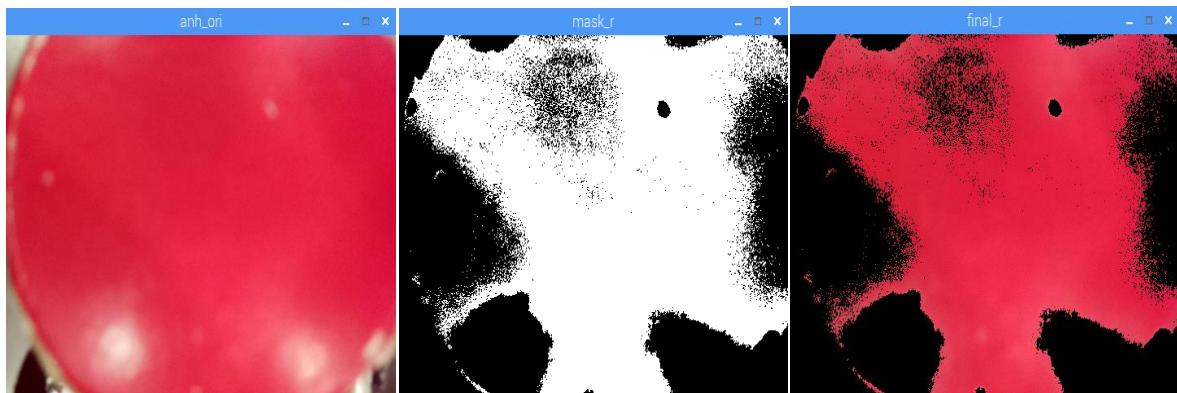
Lần 4:



Hình 5.7: Kết quả nhận dạng màu đỏ lần 4

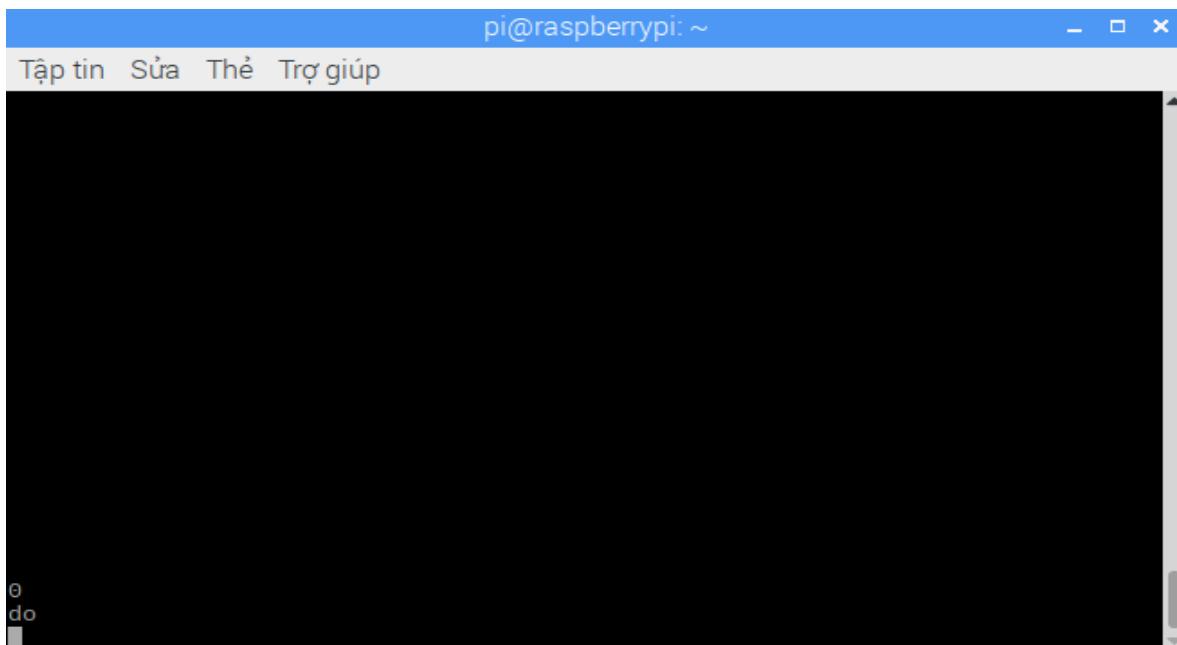
CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

Lần 5:



Hình 5.8: Kết quả nhận dạng màu đỏ lần 5

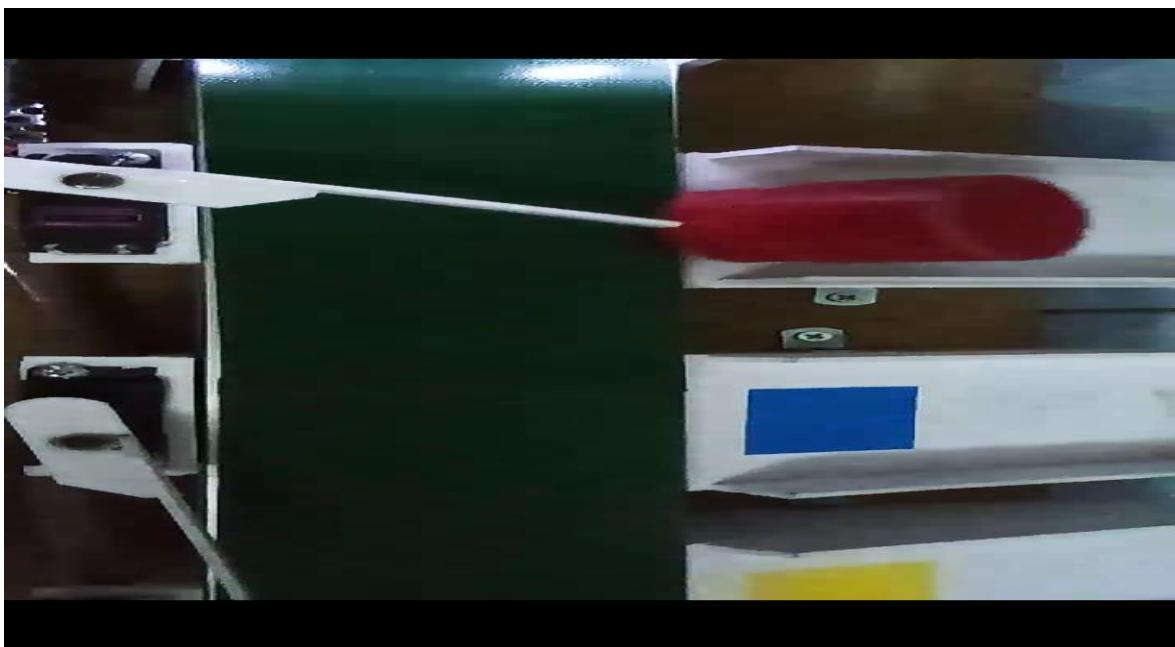
Sau khi nhận dạng được màu đỏ, Raspberry Pi sẽ tiến hành xuất ra kết quả và gửi chuỗi dữ liệu “do” cho Arduino.



Hình 5.9: Kết quả nhận dạng màu đỏ

Với giá trị 0 là tín hiệu ra lệnh từ Arduino, ”do” là chuỗi dữ liệu Raspberry Pi truyền qua Arduino. Arduino khi nhận được tín hiệu “do” sẽ điều khiển cơ cấu chấp hành tiến hành phân loại, và hiển thị kết quả ra màn hình LCD.

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ



Hình 5.10: Hình ảnh thực tế phân loại sản phẩm đỏ

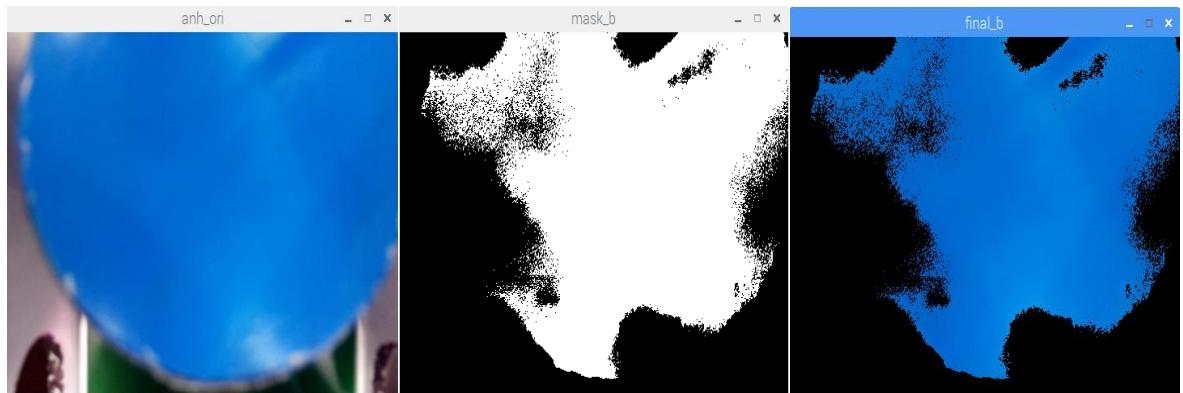


Hình 5.11: Kết quả hiển thị sau 5 lần phân loại sản phẩm đỏ

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

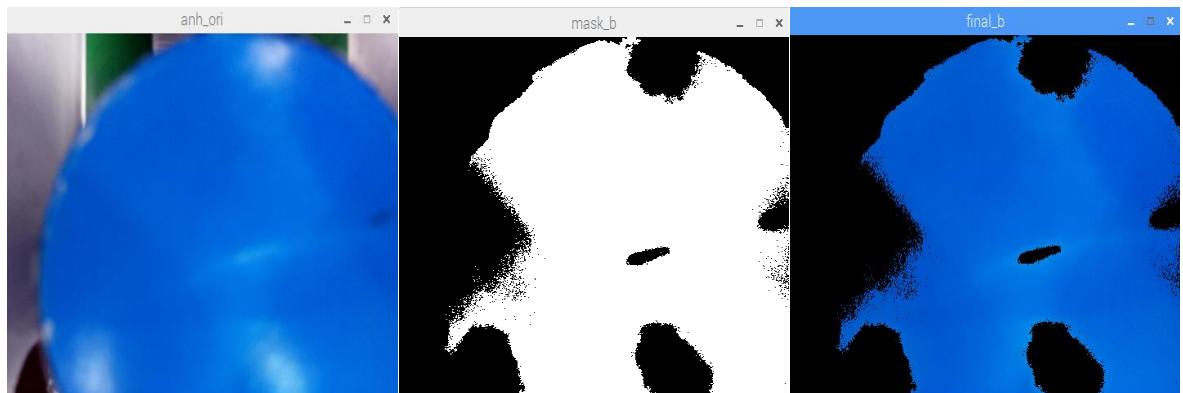
- Kết quả của 5 lần chụp liên tiếp sản phẩm màu xanh:

Lần 1:



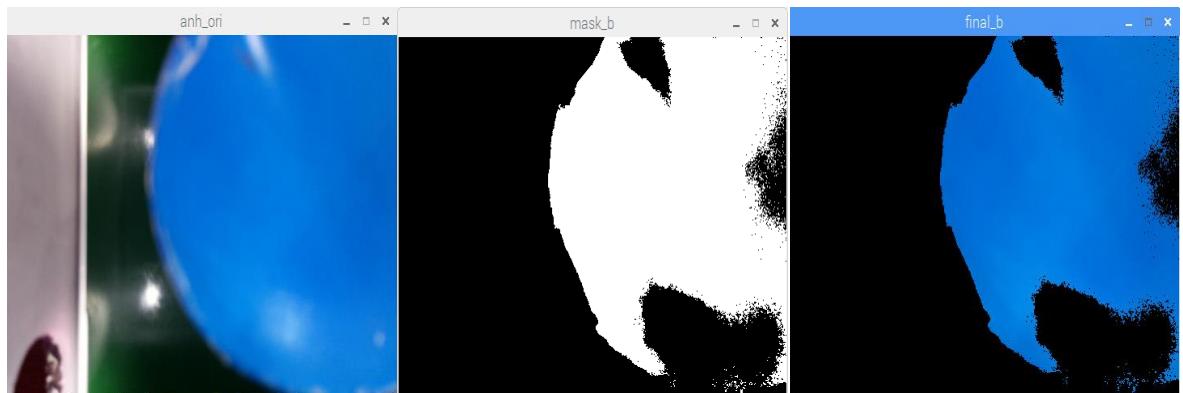
Hình 5.12: Kết quả nhận dạng màu xanh lần 1

Lần 2:



Hình 5.13: Kết quả nhận dạng màu xanh lần 2

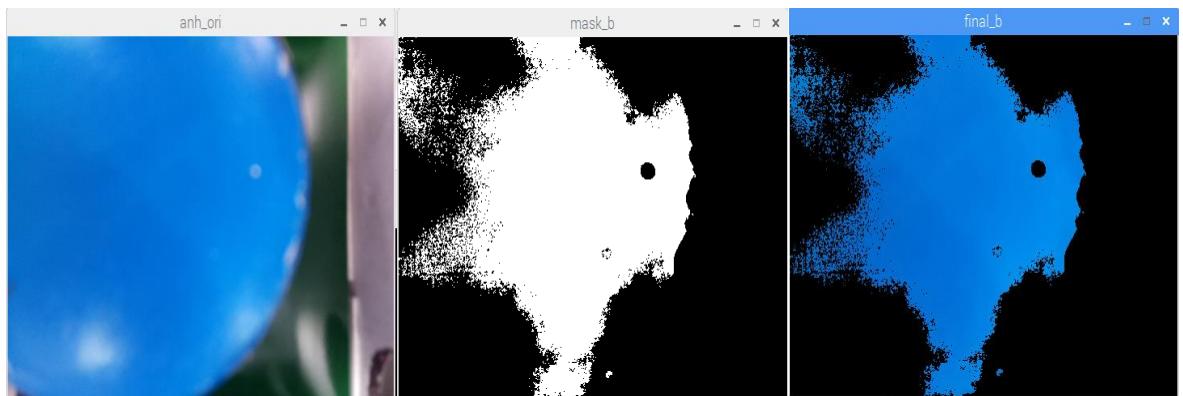
Lần 3:



Hình 5.14: Kết quả nhận dạng màu xanh lần 3

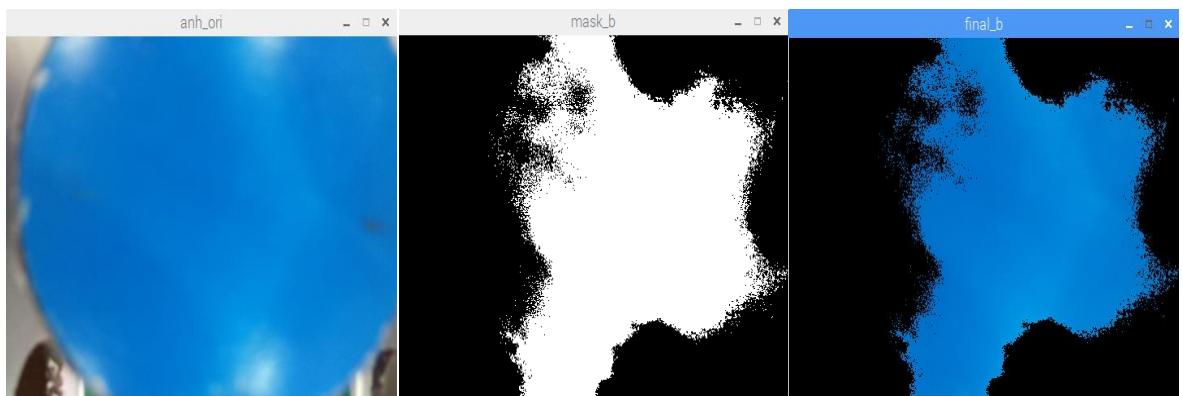
CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

Lần 4:



Hình 5.15: Kết quả nhận dạng màu xanh lần 4

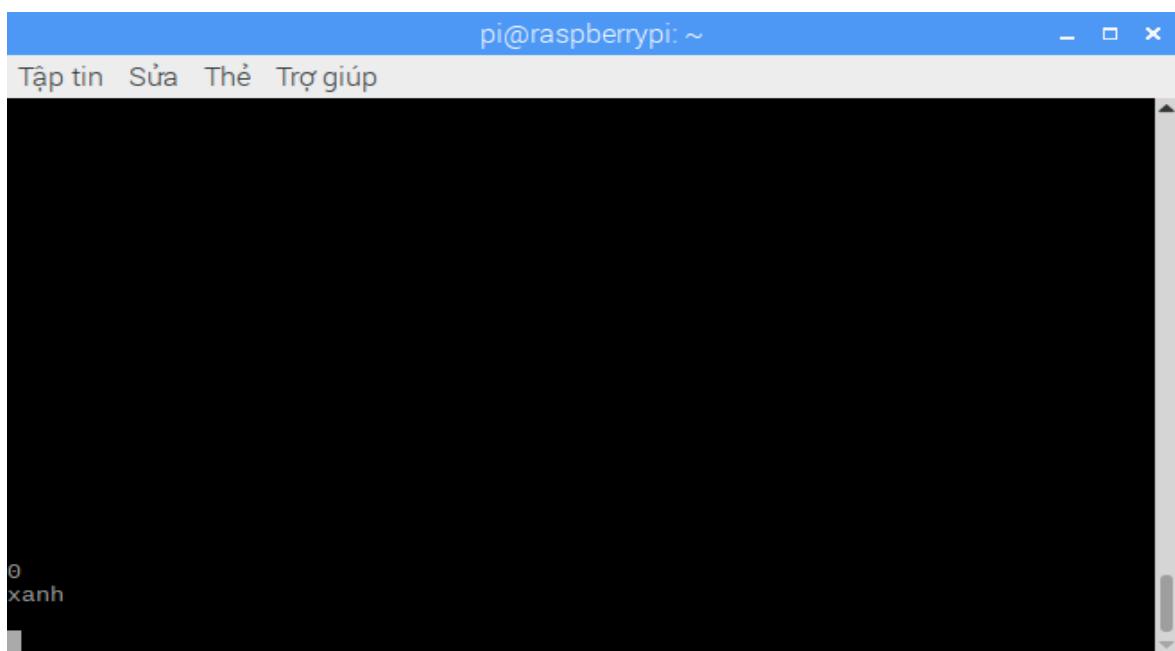
Lần 5:



Hình 5.16: Kết quả nhận dạng màu xanh lần 5

Sau khi nhận dạng được màu xanh, Raspberry Pi sẽ tiến hành xuất ra kết quả và gửi chuỗi dữ liệu “xanh” cho Arduino.

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ



Hình 5.17: Kết quả nhận dạng màu xanh



Hình 5.18: Hình ảnh thực tế phân loại sản phẩm màu xanh

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ



Hình 5.19: Kết quả hiển thị sau 5 lần phân loại sản phẩm xanh

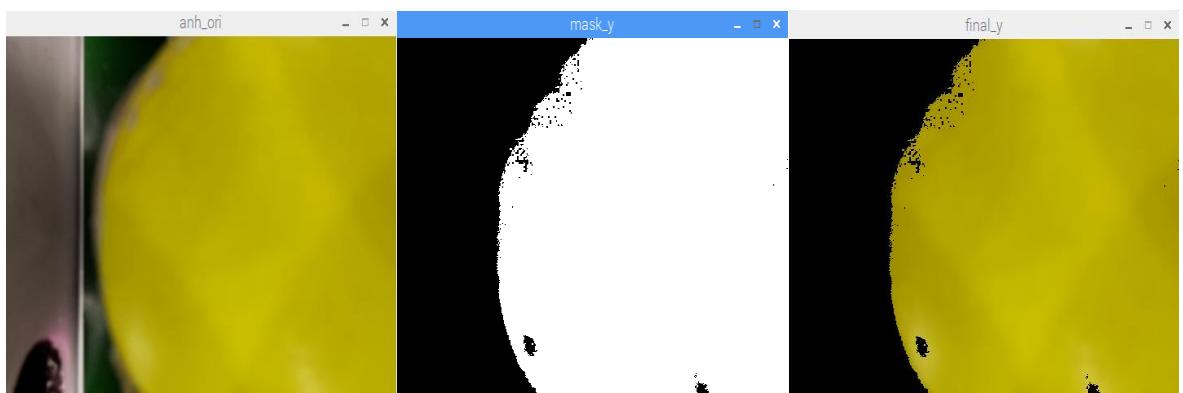
- Kết quả của 5 lần chụp liên tiếp sản phẩm màu vàng:

Lần 1:



Hình 5.20: Kết quả nhận dạng màu vàng lần 1

Lần 2:



Hình 5.21: Kết quả nhận dạng màu vàng lần 2

CHƯƠNG 5: KẾT QUẢ_NHẬN DẠNG MÀU VÀNG

Lần 3:



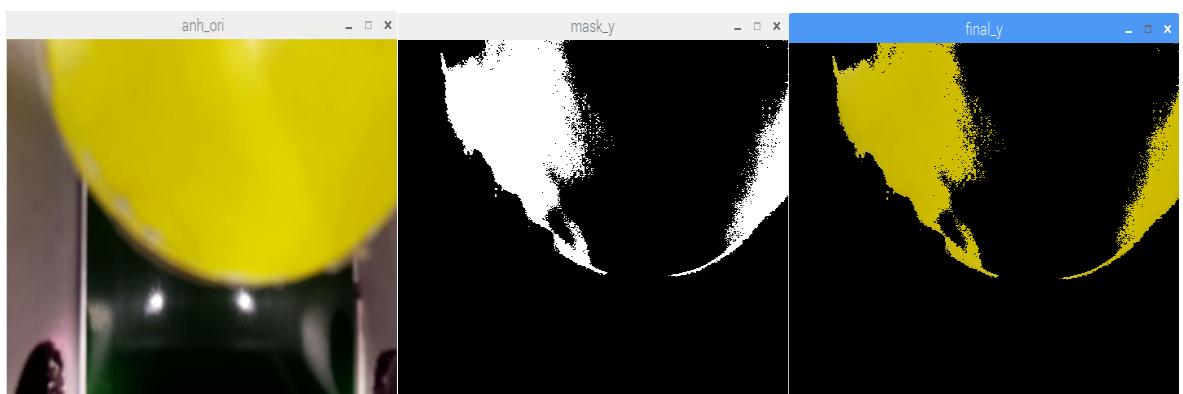
Hình 5.22: Kết quả nhận dạng màu vàng lần 3

Lần 4:



Hình 5.23: Kết quả nhận dạng màu vàng lần 4

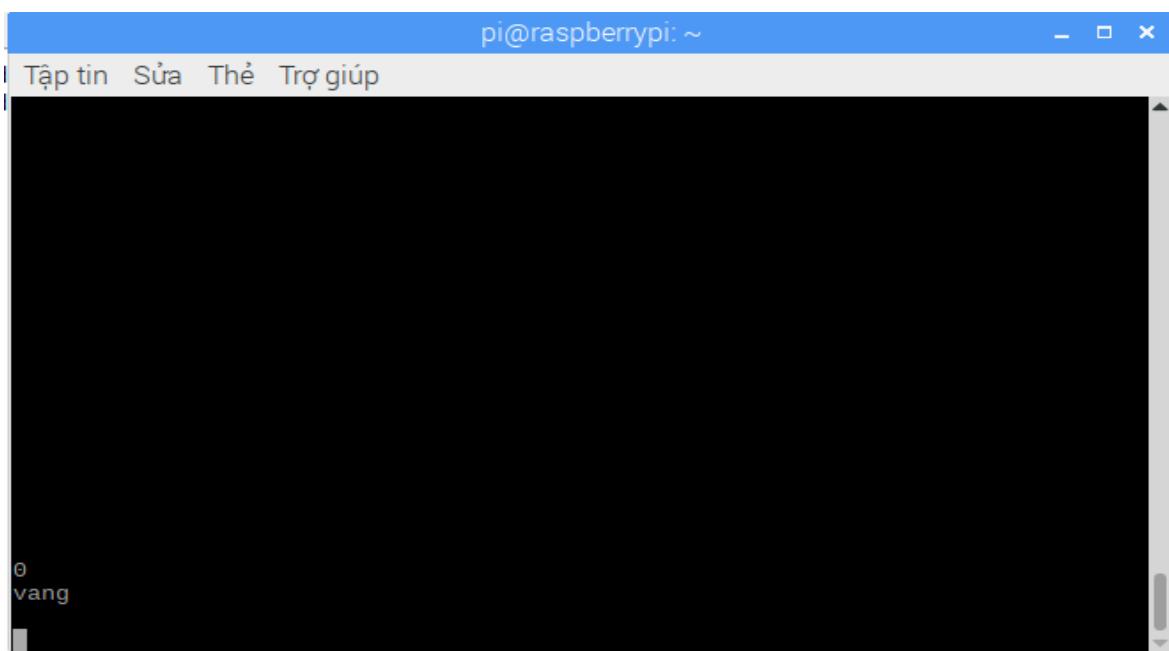
Lần 5:



Hình 5.24: Kết quả nhận dạng màu vàng lần 5

Sau khi nhận dạng được màu vàng, Raspberry Pi sẽ tiến hành xuất ra kết quả và gửi chuỗi dữ liệu “vàng” cho Arduino.

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ



Hình 5.25: Kết quả nhận dạng màu vàng



Hình 5.26: Hình ảnh thực tế phân loại sản phẩm màu vàng

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ



Hình 5.27: Kết quả hiển thị sau 5 lần phân loại sản phẩm vàng

Sau đây là bảng đánh giá độ chính xác sau khi chạy thực nghiệm:

Bảng 5.1: Bảng đánh giá độ chính xác phân loại sản phẩm

Sản phẩm	Đỏ	Xanh	Vàng
Kết quả	5/5	5/5	5/5
Độ chính xác (%)	100	100	100

Nhận xét:

Sau 5 lần thực nghiệm trên mỗi sản phẩm, hệ thống phân loại sản phẩm cho thấy hoạt động tương đối ổn định với độ chính xác tuyệt đối, không bị nhầm lẫn trong việc phân loại giữa các màu với nhau và phân loại rõ ràng từng sản phẩm. Tuy nhiên, ở đây chỉ thực hiện mỗi màu 5 sản phẩm liên tiếp, khi thực hiện liên tục nhiều sản phẩm hơn sẽ có xuất hiện những sai sót do ảnh hưởng của nhòe ảnh, nhiễu, cường độ sáng,... gây nhận diện không chính xác được sản phẩm.

5.2 NHẬN XÉT VÀ ĐÁNH GIÁ

Qua quá trình thực hiện đề tài, nhóm đã biết sử dụng Rasberry các chức năng cơ bản chiếc máy tính nhúng, như việc tạo file mới, tạo project mới, tìm hiểu Python trên Raspberry, cách chạy chương trình Python,... Ngoài ra nhóm còn tự cài đặt hệ điều hành cho máy và kết nối với Camera Pi chụp ảnh từ Raspberry,... Đánh giá máy

CHƯƠNG 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ

tính nhúng Raspberry một cách khách quan, máy tính được thiết kế nhỏ gọn, thuận tiện cho các mô hình không thích sự cồng kềnh, đáp ứng đủ các chức năng cơ bản như một chiếc máy tính bình thường, tốc độ xử lý tạm ổn cho các dự án nhỏ, hệ thống chạy ổn định, nhưng các dự án lớn cần cấu hình máy cao thì Raspberry chưa đáp ứng được nhu cầu. Vì thời gian thực hiện đè tài giới hạn, nên nhóm không đi sâu về các giao tiếp ngoại vi của Rasberry, như Wifi hay các chân GPIO,... Chỉ dừng lại ở việc sử dụng các chức năng cơ bản của máy. Đối với Model Camera Raspberry Pi có độ phân giải không quá cao nhưng ngược lại nó có giá thành rẻ, cùng với chất lượng ảnh phù hợp với việc xử lý của đè tài. Tìm hiểu chi tiết về Arduino Uno cũng như cách giao tiếp giữa Arduino với máy tính nhúng Raspberry Pi, điều khiển được hoạt động của Servo, Relay, Cảm biến và LCD.

Sau khi qua thời gian tìm hiểu phương pháp nhận dạng, phân loại sản phẩm, nhóm đã phân loại thành công được ba màu cơ bản (đỏ, xanh, vàng), nếu không thuộc một trong ba màu kể trên thì sản phẩm đó được xem là sản phẩm khác, cùng với đó là đếm sản phẩm theo từng màu và được hiển thị trên giao diện lcd.

CHƯƠNG 6: KẾT LUẬN VÀ PHÁT TRIỂN

6.1 KẾT LUẬN

6.1.1 Kết quả đạt 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.

Mô hình phần cứng hoạt động tốt, thiết bị nhỏ gọn, lắp đặt dễ dàng, có tính kinh tế, thuận tiện cho việc học tập và cũng có thể phát triển thành dây chuyền sản xuất.

Nhận dạng tốt được hầu hết các màu sắc đề ra của sản phẩm, có đếm sản phẩm.

6.1.2 Những mặt hạn chế

Ngoài những kết quả đạt được thì hệ thống vẫn còn những hạn chế sau: Tốc độ xử lý còn chậm, hệ thống buồng ảnh chưa đáp ứng được ánh sáng tốt nhất, chỉ phân loại được sản phẩm theo màu sắc, không phát hiện được sản phẩm bị lỗi.

6.2 HƯỚNG PHÁT TRIỂN

Từ những mặt hạn chế của đề tài, để đề tài hoạt động tốt hơn và có thể áp dụng vào thực tế sau này nhóm đã đề ra những hướng phát triển như sau: Làm thành dây chuyền sản xuất với cấu hình mạnh hơn, cải tiến thêm chức năng phát hiện được sản phẩm lỗi, tìm hiểu và phát triển thêm chức năng vận hành và giám sát từ xa.

TÀI LIỆU THAM KHẢO

- [1] Hà Duy Khánh, Đồ án Phân loại sản phẩm dùng Kit Raspberry, Đồ Án Tốt Nghiệp Đại Học, Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh, 2018.
- [2] Lê Thành Luân, Đồ án Đo nhiệt độ độ ẩm hiển thị trên led 7 đoạn có báo động khi quá nhiệt, Đồ án 1 Trường Đại Học Sư Phạm Kỹ Thuật Tp. Hồ Chí Minh, 2017.
- [3] Proximity Sensor/Switch E18-D80NK, Datasheet.
- [4] Sơ lược về động cơ DC, khái niệm – phân loại và điều khiển tốc độ. <http://motor2hand.com/>, 2017.
- [5] Động cơ Servo ứng dụng và nguyên lý hoạt động, <https://www.vincss.org/>, 2018.
- [6] MG996R_Tower-Pro, Datasheet.
- [7] Cấu tạo phân loại ứng dụng băng tải, <http://www.cokhimha.com/>, 2018.
- [8] Lê Thành Luân, Đồ án Đo nhiệt độ độ ẩm hiển thị trên LCD có báo động khi quá nhiệt, Đồ án 2 Trường Đại Học Sư Phạm Kỹ Thuật Tp. Hồ Chí Minh, 2018.
- [9] Wolfram Donat, Learn Raspberry Pi Programming with Python, Paul Manning, 2014.
- [10] Doxygen, “OpenCV – Python Tutorials”, <https://docs.opencv.org/>, 2018.
- [11] Khái niệm cơ bản về truyền thông UART, <https://advancecad.edu.vn>, 2019.
- [12] Nguyễn Đình Phú, Giáo Trình Thực Hành Vi Điều Khiển PIC, Đại Học Sư Phạm Kỹ Thuật TP. Hồ Chí Minh, 2016.
- [13] OpenCV xử lý màu chuyển hệ màu, <https://viblo.asia/>, 2019.
- [14] OpenCV xử lý ảnh xử lý màu chuyển hệ màu, <https://techblog.vn/>, 2018.

PHỤ LỤC

❖ Code trên Arduino Uno

```
#define sensor 8 //chan sensor
#include <Servo.h> //thu vien servo
#include <Wire.h> //thu vien i2c
#include <LiquidCrystal_I2C.h> //thu vien i2c
LiquidCrystal_I2C lcd(0x27,16,2);

Servo myservo; //sp do
Servo myservo1;//sp xanh
Servo myservo2;//sp vang
Servo myservo3;//day sp
int pos = 90; // biến pos dùng để lưu tọa độ các Servo do xanh vang
int pos1 = 0; // biến pos1 dùng để lưu tọa độ Servo day sp
int demd,demx,demv,deml;
void setup() {
    pinMode(sensor,INPUT); // sensor
    pinMode(13, OUTPUT); // bang tai
    Serial.begin(9600);
    myservo.attach(11); //servo do
    myservo1.attach(10); //servo xanh
    myservo2.attach(9); //servo vang
    myservo3.attach(6); //servo day sanpham
//LCD
    lcd.init(); //Khởi động màn hình. Bắt đầu cho phép Arduino sử dụng màn hình
    lcd.backlight(); //Bật đèn nền
    lcd.print("Do:");
    lcd.setCursor(0,1);
    lcd.print("vang:");
    lcd.setCursor(9,0);
    lcd.print("xanh:");
    lcd.setCursor(10,1);
    lcd.print("loi:");
//khoi tao cac gia tri ban dau
```

PHỤ LỤC

```
demd=0;
demv=0;
demx=0;
deml=0;

for(pos1 = 0; pos1 < 70; pos1 += 1) // cho servo quay từ 0->70 độ
{
    // mỗi bước của vòng lặp tăng 1 độ
    myservo3.write(pos1);           // xuất tọa độ ra cho servo
    delay(20);                   // đợi 20 ms cho servo quay đến góc đó rồi tới bước
tiếp theo
}
delay(1200);

for(pos1 = 70; pos1>=0; pos1-=1) // cho servo quay từ 70-->0 độ
{
    myservo3.write(pos1);           // xuất tọa độ ra cho servo
    delay(20);
}
}

void loop() {
    digitalWrite(13, LOW); //bang tai chay
    int gt = digitalRead(sensor);
    if(gt == 0){
        digitalWrite(13, LOW);
        delay(1000);
        Serial.println(gt);
        digitalWrite(13, HIGH); //bang tai dung 3s
        delay(3000);
        digitalWrite(13, LOW); //bang tai tt chay
        if (Serial.available() > 0) { //khi nhan dc gia tri tu ras
            String str = Serial.readString();
            Serial.println(str);
            if(str == String("do")){ //neu sp do
                delay (2700); //bang tai chay 2700ms
                for(pos = 90; pos < 170; pos += 1) // cho servo do quay từ 90-->170 độ
                {
                    // mỗi bước của vòng lặp tăng 1 độ
                    myservo.write(pos);           // xuất tọa độ ra cho servo do
```

PHỤ LỤC

```
delay(5); // đợi 5 ms cho servo do quay đến góc đó rồi
tới bước tiếp theo
}

for(pos = 170; pos>=90; pos-=1) // cho servo do quay từ 170-->90 độ
{
    myservo.write(pos); // xuất tọa độ ra cho servo do
    delay(1);
}

demd=demd+1; //tang gia tri sp do len 1
lcd.setCursor(3,0); //hien thi ra lcd
lcd.print(round(demd));
//day sp
for(pos1 = 0; pos1 < 70; pos1 += 1) // cho servo quay từ 0->70 độ
{
    // mỗi bước của vòng lặp tăng 1 độ
    myservo3.write(pos1); // xuất tọa độ ra cho servo
    delay(20); // đợi 20 ms cho servo quay đến góc đó rồi tới
bước tiếp theo
}

delay(1200);
for(pos1 = 70; pos1>=0; pos1-=1) // cho servo quay từ 70-->0 độ
{
    myservo3.write(pos1); // xuất tọa độ ra cho servo
    delay(20);
}

}

else if(str == String("xanh")){ // neu sp xanh
    delay (5900); //bang tai chay 5900ms
    for(pos = 90; pos < 170; pos += 1) // cho servo xanh quay từ 90->170 độ
    {
        // mỗi bước của vòng lặp tăng 1 độ
        myservo1.write(pos); // xuất tọa độ ra cho servo
        delay(5); // đợi 5 ms cho servo xanh quay đến góc đó rồi
tới bước tiếp theo
    }

    for(pos = 170; pos>=90; pos-=1) // cho servo xanh quay từ 170-->90
độ
    {
        myservo1.write(pos); // xuất tọa độ ra cho servo xanh
```

```
delay(1);
}

demx=demx+1; //tang gia tri xanh them 1
lcd.setCursor(14,0); //hien thi lcd
lcd.print(round(demx));
//day sp

for(pos1 = 0; pos1 < 70; pos1 += 1) // cho servo quay từ 0->70 độ
{
    // mỗi bước của vòng lặp tăng 1 độ
    myservo3.write(pos1); // xuất tọa độ ra cho servo
    delay(20); // đợi 20 ms cho servo quay đến góc đó rồi tới
bước tiếp theo
}
delay(1200);

for(pos1 = 70; pos1>=0; pos1-=1) // cho servo quay từ 70-->0 độ
{
    myservo3.write(pos1); // xuất tọa độ ra cho servo
    delay(20);
}
}

else if(str == String("vang")){//neu sp vang
delay (8300);
for(pos = 90; pos < 170; pos += 1) // cho servo vang quay từ 90->170 độ
{
    // mỗi bước của vòng lặp tăng 1 độ
    myservo2.write(pos); // xuất tọa độ ra cho servo
    delay(5); // đợi 5 ms cho servo vang quay đến góc đó rồi
tới bước tiếp theo
}
for(pos = 170; pos>=90; pos-=1) // cho servo vang quay từ 170-->90
độ
{
    myservo2.write(pos); // xuất tọa độ ra cho servo vang
    delay(1);
}
demv=demv+1; // tang gia tri vang them 1
lcd.setCursor(5,1); //hien thi lcd
lcd.print(round(demv));
//day sp
```

PHỤ LỤC

```
for(pos1 = 0; pos1 < 70; pos1 += 1) // cho servo quay từ 0->70 độ
{
    // mỗi bước của vòng lặp tăng 1 độ
    myservo3.write(pos1);           // xuất tọa độ ra cho servo
    delay(20);                    // đợi 20 ms cho servo quay đến góc đó rồi tới
bước tiếp theo
}
delay(1200);
for(pos1 = 70; pos1>=0; pos1-=1) // cho servo quay từ 70-->0 độ
{
    myservo3.write(pos1);           // xuất tọa độ ra cho servo
    delay(20);
}
}

else{//khong phai sp do xanh vang
    deml=deml+1; //tang gia tri loi them 1
    lcd.setCursor(14,1); //hien thi ra lcd
    lcd.print(round(deml));
    delay(12300);
    //day sp
    for(pos1 = 0; pos1 < 70; pos1 += 1) // cho servo quay từ 0->70 độ
    {
        // mỗi bước của vòng lặp tăng 1 độ
        myservo3.write(pos1);           // xuất tọa độ ra cho servo
        delay(20);                    // đợi 20 ms cho servo quay đến góc đó rồi tới
bước tiếp theo
    }
    delay(1200);
    for(pos1 = 70; pos1>=0; pos1-=1) // cho servo quay từ 70-->0 độ
    {
        myservo3.write(pos1);           // xuất tọa độ ra cho servo
        delay(20);
    }
}
}
```

❖ Code trên Raspberry Pi

```
# import the necessary packages
import numpy as np
import argparse
import cv2
import RPi.GPIO as GPIO
import time
import serial
import picamera

ser = serial.Serial(
    port = '/dev/ttyAMA0',
    baudrate = 9600,
    parity = serial.PARITY_NONE,
    stopbits = serial.STOPBITS_ONE,
    bytesize = serial.EIGHTBITS,
    timeout = 1
)
while True:
    s = ser.readline()
    data = s.decode()
    data = data.rstrip()
    print(data)
    with picamera.PiCamera() as camera:
        if(data == "0"):
            camera.resolution = (720,480)
            camera.capture("/home/pi/newpicture.jpg")
            img2 = cv2.imread("newpicture.jpg",1);
            hsv_img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2HSV);
            min_mau_y = np.array([15,111,144]) #yellow
            max_mau_y = np.array([18,123,155])
            min_mau_r = np.array([173,138,123]) #red
            max_mau_r = np.array([176,176,157])
            min_mau_b = np.array([126,91,48]) #blue
            max_mau_b = np.array([136,142,73])
```

```
mask_y = cv2.inRange(hsv_img2, min_mau_y, max_mau_y);
mask_r = cv2.inRange(hsv_img2, min_mau_r, max_mau_r);
mask_b = cv2.inRange(hsv_img2, min_mau_b, max_mau_b);

final_y = cv2.bitwise_and(img2,img2, mask= mask_y);
final_b = cv2.bitwise_and(img2,img2, mask= mask_b);
final_r = cv2.bitwise_and(img2,img2, mask= mask_r);

if np.any(final_r):
    print("1");
    cv2.imshow("anh_r", final_r);# red
    ser.write(b'do')
    ser.flush()
    time.sleep(3)
elif np.any(final_b):
    print("2");
    cv2.imshow("anh_b", final_b);# blue
    ser.write(b'xanh')
    ser.flush()
    time.sleep(3)
elif np.any(final_y):
    print("3");
    cv2.imshow("anh_y", final_y);# yellow
    ser.write(b'vang')
    ser.flush()
    time.sleep(3)
else: # khong phai red, yellow, blue
    print("ko co GRB");
    ser.write(b'ko co RGB')
    ser.flush()
    time.sleep(3)
cv2.imshow("anh_ori", img2); #anh goc

cv2.waitKey(500);
```