

TRƯỜNG ĐẠI HỌC BÀ RỊA - VŨNG TÀU
KHOA KỸ THUẬT - CÔNG NGHỆ

-----***-----



BARIA VUNGTAU
UNIVERSITY
CAP SAINT JACQUES

ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

“MÔ HÌNH BÃI GỬI XE TỰ ĐỘNG”

Giảng viên hướng dẫn: ThS. Lưu Hoàng

Sinh viên thực hiện: Phạm Trần Phi Hùng – Nguyễn Văn Hiếu

MSSV: 20035280 20035503

Lớp: DH20TD

Khóa: 2020 – 2024

Chuyên ngành: Điều khiển và tự động hóa

Bà Rịa - Vũng Tàu, tháng 5 năm 2024

Phiếu giao đề tài

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và tên sinh viên 1: **Phạm Trần Phi Hùng**

MSSV: **20035280**

Lớp: **DH20TD**

Chuyên ngành: **Điều khiển và tự động hóa**

Hệ đào tạo: **Đại học chính quy**

Khóa học: **2020 – 2024**

Họ và tên sinh viên 2: **Nguyễn Văn Hiếu**

MSSV: **20035503**

Lớp: **DH20TD**

Chuyên ngành: **Điều khiển và tự động hóa**

Hệ đào tạo: **Đại học chính quy**

Khóa học: **2020 – 2024**

I. Tên đề tài:

“MÔ HÌNH BÃI GỬI XE TỰ ĐỘNG”

II. Nhiệm vụ

1. Các số liệu ban đầu

- Kích thước mô hình:
- Động cơ bước 12V
- Động cơ servo
- Cảm biến tiệm cận phát hiện kim loại
- Webcam kết nối máy tính qua cổng USB
- Cơ cấu truyền động bằng băng răng và dây xích
- Mạch arduino và mạch RFID

2. Nội dung thực hiện

- Tìm hiểu nghiên cứu giao thức kết nối giữa máy tính – mạch arduino – mạch RFID
- Tìm hiểu dữ liệu và phương thức xử lý dữ liệu thu thập từ mạch đọc thẻ RFID và

dữ liệu hình ảnh từ webcam

- Tìm hiểu cách thức hoạt động của mô hình để lựa chọn linh kiện
- Thiết kế mô hình phân cứng
- Tìm hiểu và nghiên cứu cách thức vận hành động cơ bước
- Tổng hợp kiến thức và viết chương trình chạy
- Lắp ráp mô hình và chạy thử nghiệm, căn chỉnh sửa lỗi mô hình
- Rút ra nhận xét, đánh giá chung và tổng hợp kiến thức thu được
- Viết báo cáo luận án
- Báo cáo đề tài tốt nghiệp

III. Thông tin giao đề tài

Giảng viên hướng dẫn: **ThS. Lưu Hoàng**

- Ngày giao nhiệm vụ: 15/01/2024
- Ngày hoàn thành nhiệm vụ: 30/05/2024

GIÁO VIÊN HƯỚNG DẪN
(Ký và ghi rõ họ tên)

ThS. Lưu Hoàng

Bà Rịa - Vũng Tàu, ngày tháng năm 2024
SINH VIÊN THỰC HIỆN
(Ký và ghi rõ họ tên)

Phạm Trần Phi Hùng Nguyễn Văn Hiếu

TRƯỞNG KHOA
(Ký và ghi rõ họ tên)

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN

Đề tài: “MÔ HÌNH BÃI GỬI XE TỰ ĐỘNG”

Thời gian	Nội dung	Xác nhận của GVHD
15/01 – 16/01/2024	- Đăng kí và nộp kế hoạch thực hiện đồ án tốt nghiệp	
17/01 – 20/02/2024	- Tìm hiểu, nghiên cứu về thiết kế khung, mạch điện cần thực hiện	
21/02 – 10/02/2024	- Lên kế hoạch tính toán chi phí - Thực hiện đặt mua và gia công các linh kiện	
11/02 – 14/02/2024	- Thu thập, thử nghiệm linh kiện - Lắp ghép mô hình	
15/02 – 20/02/2024	- Xây dựng chương trình thu thập dữ liệu đọc thẻ từ mạch RFID	
21/02 – 10/03/2024	- Xây dựng trang trình xử lý ảnh	
11/03 – 20/03/2024	- Xây dựng trang trình C# lập trình cho mạch arduino	
21/03	- Xây dựng giao diện điều khiển	

31/03/2024	<ul style="list-style-type: none"> - Xây dựng chương trình thu thập dữ liệu - Chạy thử nghiệm, rà soát lỗi, chỉnh sửa chương trình và phân cứng 	
01/04 – 13/04/2024	<ul style="list-style-type: none"> - Hoàn chỉnh mô hình - Viết luận văn 	
14/04 – 30/05/2024	<ul style="list-style-type: none"> - Hoàn chỉnh luận văn - Nộp báo cáo 	

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Bà Rịa vũng tàu, tháng năm 2024

Giảng viên hướng dẫn

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Bà Rịa vũng tàu, tháng năm 2024

Giảng viên phản biện

LỜI CAM ĐOAN

Đồ án tốt nghiệp được thực hiện bởi sinh viên Phạm Trần Phi Hùng, Nguyễn Văn Hiếu lớp DH20TD, chuyên ngành Điều khiển và Tự động hóa, sinh viên trường Đại học Bà Rịa Vũng Tàu cam đoan rằng đồ án tốt nghiệp đề tài “MÔ HÌNH BÃI GỬI XE TỰ ĐỘNG” là công trình nghiên cứu của riêng của nhóm dưới sự hướng dẫn của thầy ThS. Lưu Hoàng khoa kỹ thuật – công nghệ, tất cả các thông tin, dữ liệu và tài liệu được sử dụng trong đồ án này đều được trích dẫn đầy đủ và chính xác từ các nguồn tham khảo được ghi chú trong phần tài liệu tham khảo. Cam đoan rằng đồ án này là sản phẩm của công sức và kiến thức cá nhân của nhóm, và không sao chép hoặc sử dụng tài liệu từ nguồn khác mà không được ghi chú.

Nhóm chịu trách nhiệm hoàn toàn về nội dung và ý kiến của đồ án này, và sẵn sàng chịu trách nhiệm trước các cơ quan chức năng nếu có bất kỳ vi phạm nào liên quan đến quy định của trường hoặc pháp luật và không vi phạm bất kỳ nguyên tắc gian lận nào. Trên tinh thần của sự trung thực và trách nhiệm, nhóm chúng em cam đoan rằng tất cả những gì được nêu trên là đúng và chân thành.

Bà Rịa – Vũng Tàu, ngày , tháng , năm 2024

Sinh viên thực hiện

Phạm Trần Phi Hùng

Nguyễn Văn Hiếu

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các Thầy Cô của trường Đại Học Bà Rịa Vũng Tàu, đặc biệt là các Thầy Cô khoa Kỹ Thuật - Công Nghệ của trường đã tạo điều kiện cho em học tập ở khoa và em cũng xin chân thành cảm ơn thầy ThS. Lưu Hoàng đã nhiệt tình hướng dẫn em để hoàn thành đề tài tốt nghiệp.

Trong quá trình học, cũng như là trong quá trình làm bài báo cáo, khó tránh khỏi sai sót, rất mong quý thầy, cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, nhóm em rất mong nhận được ý kiến đóng góp từ thầy, cô để em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn.

Cuối cùng em xin kính chúc quý thầy cô tại Trường Đại Học Bà Rịa Vũng Tàu, và quý thầy cô khoa Công Nghệ - Kỹ Thuật thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh nhà giáo cao đẹp của mình.

Trường Đại Học Bà Rịa Vũng tàu

Khoa: Kỹ thuật – Công nghệ

Chuyên ngành: Điều khiển và Tự động hóa

Đồ án tốt nghiệp

Bà Rịa vũng tàu, ngày 25 tháng 4 năm 2024

Nhóm xin chân thành cảm ơn!

Nội dung

Phiếu giao đề tài.....	1
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP.....	1
LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN	3
LỜI CAM ĐOAN	7
LỜI CẢM ƠN.....	8
PHẦN MỞ ĐẦU	12
CHƯƠNG 1 TỔNG QUAN.....	13
1.1 Đặt vấn đề.....	13
1.2 Mục tiêu.....	15
1.3 Nội dung nghiên cứu.....	15
1.4 Giới hạn	15
1.5 Bố cục	16
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	18
2.1 Phân tích các phương án bãi đỗ xe.....	18
2.1.1 Các dạng bãi đỗ xe ô tô hiện nay	18
2.1.2 Phương án lựa chọn	24
2.2 Các cơ sở lý thuyết liên quan.....	25
2.2.1 Giới thiệu về ngôn ngữ Python và một số công cụ liên quan	25
2.2.2 Giới thiệu về ngôn ngữ C# và một số công cụ liên quan	29
2.2.3 Công Nghệ RFID	33
2.3 Giới thiệu một số phần mềm lập trình	35
2.3.1 Phần mềm Python IDLE	36
2.3.2 Phần mềm Arduino IDE.....	37
2.4.3 Phần mềm qt Designer	38
2.5 Giới thiệu phần cứng.....	39
CHƯƠNG 3 THIẾT KẾ	41
3.1 Giới thiệu	41
3.2 Thiết kế hệ thống	41
3.2.1 Thiết kế sơ đồ khối hệ thống.....	41
3.2.2 Thiết kế phần cứng.....	43
3.2.3 Sơ đồ nguyên lý toàn mạch.....	53
Chương 4 THI CÔNG HỆ THỐNG	55
4.1 Giới thiệu	55
4.2 Thi công mô hình.....	56
4.2.1 Thi công phần khung khung.....	57

4.2.2 Thi công phần mạch điện	59
4.3 Lập trình hệ thống	63
4.3.1 Lập trình cho ARDUINO	63
4.3.2 Lập trình xử lý hình ảnh từ máy tính	77
4.3.3 Thiết kế và lập trình giao diện	93
4.4 Tài liệu hướng dẫn sử dụng, thao tác	95
4.4.1 Hướng dẫn các bước sử dụng	95
4.4.2 Quy trình thao tác	95
CHƯƠNG 5 KẾT QUẢ THU HOẠCH	96
5.1 Hệ thống mô hình	96
5.2 Giao diện phần mềm	98
CHƯƠNG 6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	99
6.1 Kết luận	99
6.1.1 Ưu điểm	99
6.1.2 Nhược điểm	99
6.1.3 Kết luận	100
6.2 Hướng phát triển đề tài	101
6.2.1 Mở rộng mô hình	101
6.2.2 Hoàn thiện mô hình	101
6.2.3 Tích hợp phương thức tính tiền tự động	102
6.2.4 Tích hợp điều khiển giám sát từ xa thông qua web cũng như app điện thoại	103
TÀI LIỆU THAM KHẢO	105

DANH SÁCH HÌNH ẢNH

Hình 1: Bãi đậu xe truyền thống.....	18
Hình 2: Bãi gửi xe tầng hầm.....	19
Hình 3: Bãi đậu xe đa tầng.....	19
Hình 4: dạng tầng di động.....	20
Hình 5: Bãi gửi xe tự động dạng tầng.....	22
Hình 6: Bãi gửi xe tự động vòng xoay.....	23
Hình 7: thiết kế mô hình Bãi gửi xe tự động vòng xoay của nhóm.....	24
Hình 8: Giải thuật xử lý ảnh nhận diện biển số xe.....	28
Hình 9: Sơ đồ giải thuật C# điều khiển động cơ.....	32
Hình 10: cấu tạo Thẻ RFID.....	33
Hình 11: Thiết bị đọc RFID RC522.....	34
Hình 12: Nguyên lý hoạt động RFID.....	34
Hình 13: giao diện Phần mềm Python IDLE.....	36
Hình 14: Phần mềm Arduino IDE.....	37
Hình 15: giao diện làm việc trên Qt Designer.....	38
Hình 16: sơ đồ khối hệ thống.....	41
Hình 17: sơ đồ chức năng chân của mạch Arduino Nano CH3.....	44
Hình 18: Module đọc thẻ RC522.....	45
Hình 19: nguyên lý thu nhận tín hiệu của mạch RFID.....	46
Hình 20: Cách kết nối modul RFID với arduino.....	47
Hình 21: Cảm biến tiệm cận LJ8A3-2-Z/BX NPN 2mm.....	48
Hình 22: Nút nhấn nhà R13-507 16mm.....	49
Hình 23: Webcam máy tính AUKEY PC-W3.....	50
Hình 24: thông số kỹ thuật webcam sử dụng.....	50
Hình 25: Động cơ bước SUMTOR 42HS3413A4 – 0.28Nm – 1.3a – 42*34mm.....	51
Hình 26: Driver động cơ bước a4988.....	52
Hình 27: động cơ Servo SG90S.....	52
Hình 28: Sơ đồ nguyên lý đi dây của mạch thực hiện.....	54
Hình 29: bản thiết kế mạch in thực hiện.....	54
Hình 30: thông số kỹ thuật thanh nhôm V-Slot 20mmx20mm.....	57
Hình 31: mô phỏng phần khung mô hình thực hiện.....	59
Hình 32: Sơ đồ nguyên lý đi dây của mạch.....	61
Hình 33: mạch điện hoàn thành.....	62
Hình 34: giao diện arduino IDE.....	63
Hình 35: thiết lập loại board và cổng Serial.....	63
Hình 36: thiết lập COM3.....	64
Hình 37: tạo file mới để lập trình.....	64
Hình 38: viết lệnh code.....	64
Hình 39: Lưu đồ giải thuật cho Arduino.....	65
Hình 40: Lưu đồ thuật toán xử lý ảnh bằng python IDLE.....	77
Hình 41: giao diện chương trình sau thiết kế.....	94
Hình 42: mô hình thu được.....	96
Hình 43: giao diện điều khiển thu được.....	98

PHẦN MỞ ĐẦU

Lý do chọn đề tài

Xã hội hiện đại phát triển với hệ thống giao thông dày đặc, nhu cầu sử dụng phương tiện cá nhân, đặc biệt là xe ô tô ngày càng tăng cao. Điều này dẫn đến tình trạng thiếu hụt trầm trọng chỗ đỗ xe, đặc biệt tại các thành phố lớn. Việc đỗ xe tràn lan trên vỉa hè, lòng đường gây mất mỹ quan đô thị, tiềm ẩn nguy cơ mất an toàn giao thông và cản trở giao thông chung. Nhận thức được vấn đề cấp thiết này, nhóm chúng tôi quyết định nghiên cứu đề tài “MÔ HÌNH BÃI GỬI XE TỰ ĐỘNG” với mong muốn góp phần giải quyết thực trạng thiếu chỗ đỗ xe, đồng thời mang đến giải pháp an toàn, tiện lợi cho người sử dụng.

Hệ thống bãi gửi xe ô tô tự động được xây dựng dựa trên nền tảng công nghệ tiên tiến, bao gồm:

- Bo mạch Arduino: Thiết bị xử lý trung tâm, điều khiển toàn bộ hệ thống.
- Ngôn ngữ Python: Giám sát và quản lý hệ thống.
- Công nghệ RFID: Nhận diện biển số xe chính xác, bảo mật cao.
- Xử lý ảnh bằng C#: Phát hiện vị trí trống, hỗ trợ đỗ/lấy xe tự động.

Với những ưu điểm vượt trội như hoạt động hoàn toàn tự động, tiện lợi cho người sử dụng; an ninh, bảo mật cao; tối ưu diện tích và tiết kiệm chi phí vận hành, hệ thống bãi gửi xe ô tô tự động được kỳ vọng sẽ trở thành giải pháp tối ưu cho bài toán đỗ xe trong đô thị.

Ngoài ra, đề tài nghiên cứu này còn có ý nghĩa khoa học và thực tiễn cao, giúp sinh viên ứng dụng kiến thức đã học vào thực tiễn, rèn luyện kỹ năng thiết kế, lập trình và điều khiển hệ thống tự động, góp phần nâng cao chất lượng đào tạo và đáp ứng nhu cầu thị trường lao động.

Hệ thống bãi gửi xe ô tô tự động - Giải pháp thiết thực cho tương lai!

CHƯƠNG 1

TỔNG QUAN

1.1 Đặt vấn đề

Thị trường ô tô Việt Nam đang chứng kiến sự tăng trưởng mạnh mẽ, đặc biệt là sau đại dịch Covid-19. Nhu cầu về chỗ đỗ xe ngày càng cao, đặc biệt tại các thành phố lớn. Tuy nhiên, các bãi đỗ xe truyền thống thường gặp nhiều hạn chế như:

- **Quá tải:** Do số lượng xe tăng cao, các bãi đỗ xe truyền thống thường xuyên rơi vào tình trạng quá tải, gây khó khăn cho việc đỗ xe và di chuyển.
- **Lãng phí diện tích:** Việc sử dụng các phương pháp đỗ xe truyền thống lãng phí diện tích, dẫn đến hiệu quả sử dụng đất thấp.
- **Thiếu an toàn:** Các bãi đỗ xe truyền thống thường thiếu hệ thống camera giám sát và bảo vệ, dẫn đến nguy cơ mất cắp và hư hỏng xe cao.
- **Gây ô nhiễm môi trường:** Việc tập trung nhiều xe trong một khu vực nhỏ có thể dẫn đến ô nhiễm tiếng ồn và khí thải, ảnh hưởng đến môi trường xung quanh.

Để giải quyết những vấn đề trên, cần có giải pháp hiện đại, tự động hóa cao để quản lý bãi đỗ xe hiệu quả. Bãi đỗ xe cần tiết kiệm diện tích, đảm bảo an toàn, thân thiện với môi trường và có tính kinh tế cao.

“MÔ HÌNH BÃI GỬI XE TỰ ĐỘNG” là đề tài có tính cấp thiết, khoa học, thực tiễn và sáng tạo. Mô hình đề xuất có thể góp phần giải quyết vấn đề thiếu chỗ đỗ xe hiện nay, đồng thời góp phần nâng cao chất lượng cuộc sống cho người dân.

Đề tài này sẽ tập trung vào nghiên cứu hiện trạng các hệ thống bãi gửi xe ô tô tự động trên thế giới, phân tích nhu cầu và yêu cầu về bãi đỗ xe tại Việt Nam, đề xuất mô hình hệ thống bãi gửi xe ô tô tự động phù hợp với điều kiện Việt Nam, thiết kế chi tiết các thành phần của hệ thống và lập dự toán kinh tế - kỹ thuật cho mô hình đề xuất.

Nghiên cứu này được kỳ vọng sẽ mang lại những lợi ích thiết thực sau:

- Giải quyết vấn đề thiếu chỗ đỗ xe hiện nay, đặc biệt tại các thành phố lớn.
- Nâng cao hiệu quả sử dụng đất và tiết kiệm diện tích.
- Đảm bảo an toàn cho xe cộ và tài sản của người dân.
- Góp phần bảo vệ môi trường và nâng cao chất lượng cuộc sống.

Với tính cấp thiết, tính khoa học, tính thực tiễn và tính sáng tạo, đề tài nghiên cứu này hứa hẹn sẽ mang lại những đóng góp thiết thực cho xã hội.

1.2 Mục tiêu

Ứng dụng những tính năng tiện ích và vượt trội của mạch arduino, các thiết bị tự động như cảm biến, động cơ bước, động cơ servo.... để thực hiện một mô hình bãi giữ xe ô tô mang tính tự động cao, có khả năng tự động cất xe và trả xe. Ngoài ra, với đề tài này nhóm còn ứng dụng công nghệ RFID và xử lý ảnh nhận diện biển số xe để tăng cường tính bảo mật cho hệ thống.

1.3 Nội dung nghiên cứu

Trong đề tài này, nhóm sẽ tiến hành nghiên cứu các nội dung như sau:

- Thu thập dữ liệu về nhu cầu sử dụng xe ô tô cá nhân và các bãi giữ hiện nay.
- Tìm hiểu ưu và nhược điểm của một số bãi giữ xe ô tô đang được áp dụng hiện nay.
- Nghiên cứu, xây dựng thuật toán điều khiển hệ thống bãi giữ xe ô tô tự động.
- Thiết kế và thi công mô hình.
- Đánh giá kết quả thực hiện.
- Nhận xét ưu nhược điểm của toàn bộ hệ thống và hướng phát triển để tải.

1.4 Giới hạn

Vì một số điều kiện khách quan cũng như chủ quan nên đề tài này sẽ có một số giới hạn như sau:

- Nhóm chỉ thực hiện sản phẩm ở mức độ mô hình có thể hoạt động trên quy mô nhỏ, phương thức hoạt động dựa trên sự sáng tạo của nhóm vì vậy sản phẩm mà nhóm làm ra không có tính ứng dụng thực tế cao cũng như không có tính đến thương mại hóa sản phẩm

- Kích thước mô hình nhỏ gọn: chiều cao 70cm, chiều rộng 50cm và chiều dài 50cm.
- Chỉ có 6 vị trí để cất xe ô tô.
- Sử dụng loại động cơ bước được vận hành thông qua tần số sau khi được khuếch đại từ mạch arduino.
- Hệ thống có thể tính chỉ phí giữ xe, còn việc thu phí phải cần nhân viên thực hiện.

1.5 Bộ cục

Đề tài được trình bày trong 6 chương:

- Chương 1: Tổng quan
- Chương 2: Cơ sở lý thuyết
- Chương 3: Thiết kế hệ thống
- Chương 4: Thi công hệ thống
- Chương 5: Kết quả, nhận xét và đánh giá
- Chương 6: Kết luận và hướng phát triển

Cụ thể như sau:

- Chương 1: Tổng quan

Đặt vấn đề dẫn tới lý do chọn đề tài, trình bày mục tiêu, nội dung nghiên cứu, xác định một số giới hạn và trình bày bố cục của đề tài.

- Chương 2: Cơ sở lý thuyết

Trình bày, phân tích ưu nhược điểm của các phương án bãi đỗ xe ô tô hiện có trên thị trường, từ đó chọn ra phương án phù hợp cho đề tài. Giới thiệu một số lý thuyết liên quan và những phần cứng cần sử dụng cho đề tài.

- Chương 3: Thiết kế hệ thống

Xây dựng sơ đồ khối, trình bày nguyên lý hoạt động của hệ thống. Thiết kế và lựa chọn linh kiện phù hợp cho từng khối. Xây dựng sơ đồ nguyên lý cho từng khối và toàn bộ hệ thống.

- Chương 4: Thi công hệ thống

Trình bày các bước thi công mô hình hoàn chỉnh từ phần cứng đến phần mềm. Xây dựng lưu đồ giải thuật, phương trình tính toán các thông số cụ thể, xử lý tín hiệu và lập trình cho C#, vi điều khiển.

- Chương 5: Kết quả, nhận xét và đánh giá

Trình bày những kết quả đạt được (cấu trúc mô hình, giao diện phần mềm). Sau đó đưa ra nhận xét tổng thể về đề tài cũng như những đánh giá dựa trên các mục tiêu đã đặt ra ban đầu.

- Chương 6: Kết luận và hướng phát triển

Trình bày những ưu nhược điểm của toàn bộ hệ thống. Từ đó đưa ra kết luận và hướng phát triển cho đề tài.

CHƯƠNG 2

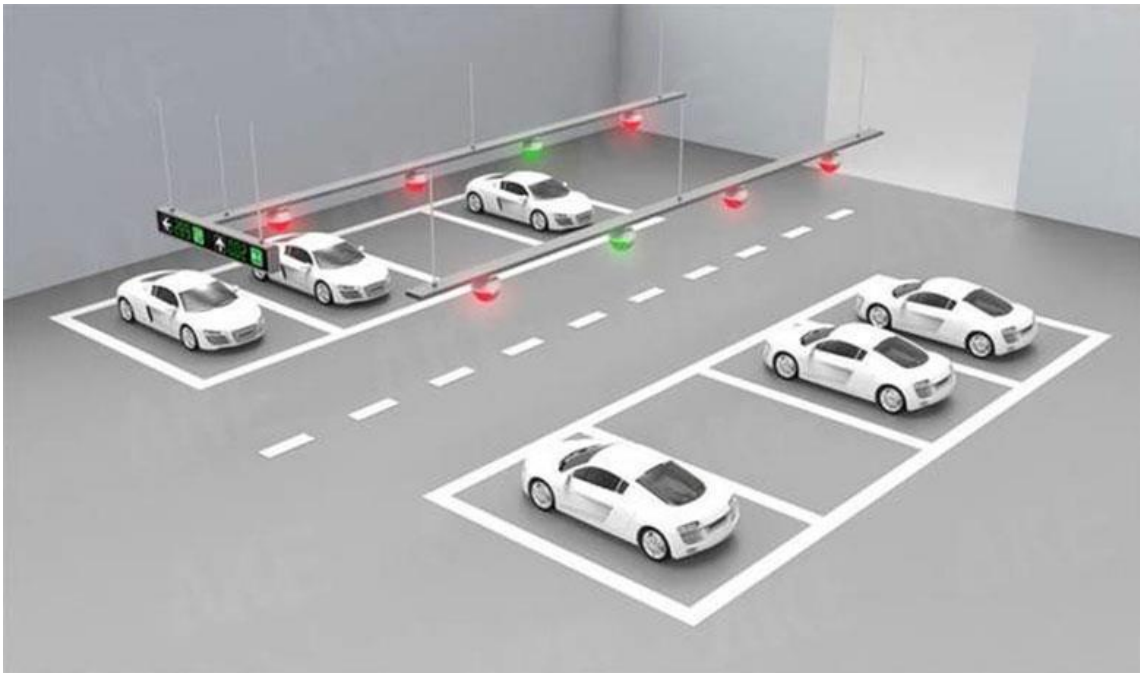
CƠ SỞ LÝ THUYẾT

2.1 Phân tích các phương án bãi đỗ xe

Do nhu cầu xã hội phát triển nên phương án bãi đỗ ô tô ngày càng đa dạng. Sau đây là phần trình bày đặc điểm, ưu, nhược của một số phương án được áp dụng phổ biến nhằm lựa chọn ra phương án phù hợp cho đề tài.

2.1.1 Các dạng bãi đỗ xe ô tô hiện nay

Bãi đậu xe truyền thống là hình thức lâu đời và thông dụng nhất hiện nay, thường được đặt ra ngoài trời hoặc trong các khu vực không gian mở.



Hình 1: Bãi đậu xe truyền thống

Bên cạnh ưu điểm bởi sự đơn giản về cách vận hành, bãi gửi xe truyền thống thường tốn khá nhiều diện tích mặt bằng hoạt động.

Ngoài ra dưới sự khan hiếm về mặt bằng cũng đã thôi thúc hình thức bãi gửi xe truyền thống phát triển thêm nhiều dạng mới như:

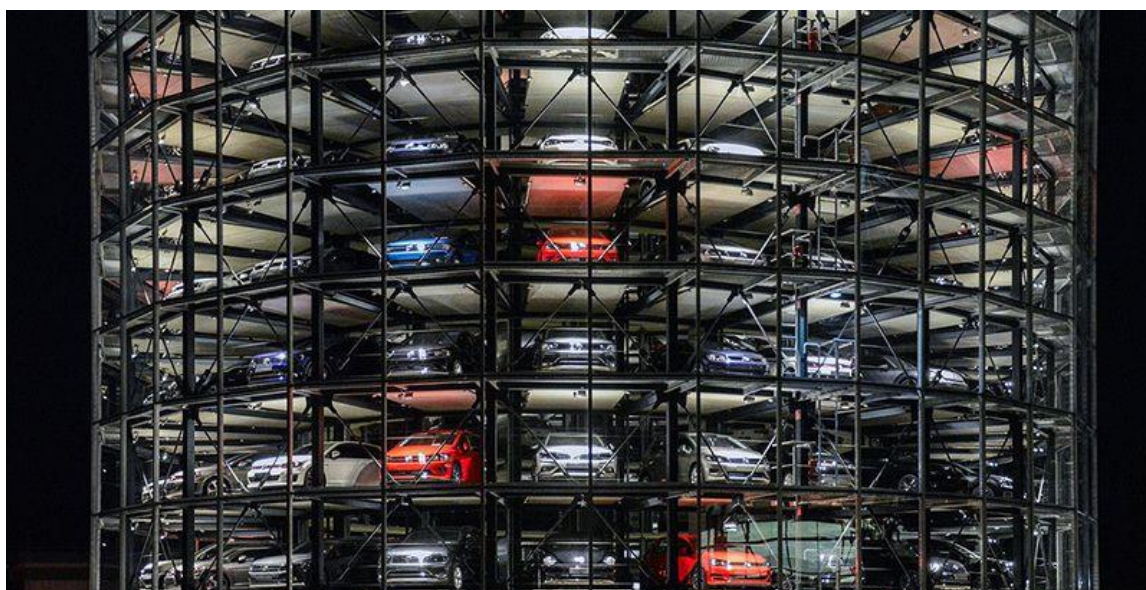
Bãi gửi xe tầng hầm



Hình 2: Bãi gửi xe tầng hầm

Đây là loại bãi đậu xe được xây dựng dưới lòng đất hoặc tầng hầm của các tòa nhà. Chúng thường có hình dạng xếp hình, với các dãy cột và hệ thống đường hầm để di chuyển xe vào và ra. Mặc dù chỉ là một dạng biến thể của bãi gửi xe truyền thống, nhưng việc tận dụng không gian tầng hầm của các tòa nhà đã giải quyết được phần nào đó cho nhược điểm của bãi gửi xe truyền thống về không gian.

Bãi đậu xe đa tầng



Hình 3: Bãi đậu xe đa tầng

Đây là hình thức bãi đậu xe có cấu trúc xếp chồng lên nhau thành nhiều tầng nhiều khu hoặc thậm chí là cả 1 toà nhà, tạo thành bãi gửi xe đồ sộ. Các tầng thường được kết nối bằng hệ thống đường đi để xe có thể di chuyển giữa các tầng. Mặc dù giải quyết được đa phần nhược điểm cũ nhưng lại đòi hỏi một lượng lớn chi phí đầu tư ban đầu, bên cạnh đó là khả năng luân chuyển xe trong tòa nhà và cần có lượng nhu cầu gửi xe lớn nhưng ổn định. Vì vậy với mô hình này vẫn hiếm khi được ứng dụng vào thực tế.

Bãi đỗ dạng tầng di động

Đây là phương án đỗ xe phù hợp với các thành phố lớn có lượng xe quá tải. Hệ thống sử dụng cấu trúc thiết kế một tháp và cao 2 hoặc 3 tầng, mỗi tầng giữ được từ 4 đến 5 xe.



Hình 4: dạng tầng di động

- Ưu điểm: chi phí thấp, phù hợp nhiều loại địa hình, tiết kiệm diện tích, bảo trì, bảo dưỡng, di chuyển đơn giản.
- Nhược điểm: chỉ phù hợp với quy mô nhỏ (5 đến 15 ô tô), cơ cấu điều khiển

chưa tự động.

Bãi gửi xe tự động dạng tầng



Hình 5: Bãi gửi xe tự động dạng tầng

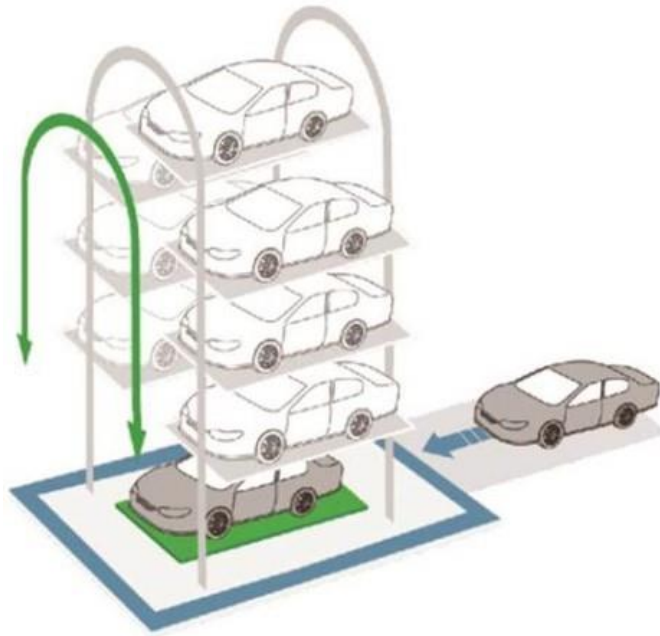
Bãi gửi xe tự động dạng tầng là một dạng tiên tiến của bãi đậu xe, thường được sử dụng trong các khu vực đô thị có diện tích đất hạn chế. Có cấu trúc xây dựng với nhiều tầng đậu xe được xếp lên nhau. Cấu trúc này giúp tận dụng không gian đất hiệu quả, cho phép đậu nhiều xe hơn trên diện tích nhỏ hơn. Xe được di chuyển giữa các tầng đậu xe thông qua hệ thống cơ khí hoặc hệ thống thang máy tự động. Hệ thống này thường được điều khiển bởi máy tính hoặc mạch điều khiển để đảm bảo sự chính xác và hiệu quả.

Người sử dụng thường sẽ gửi xe tại một điểm đặt xe tự động. Hệ thống sẽ tự động di chuyển xe vào các vị trí trống trên các tầng đậu xe và lấy lại xe khi cần thiết.

Bãi gửi xe tự động dạng tầng không chỉ giúp giải quyết vấn đề đỗ xe mà còn cải thiện sự tiện lợi và an toàn cho người sử dụng, đồng thời tối ưu hóa sử dụng không gian đất trong các khu vực đô thị.

Tuy nhiên để thực hiện được Việc xây dựng và triển khai bãi gửi xe tự động dạng tầng yêu cầu nhà đầu tư phải rót một số lượng lớn vốn đầu tư ban đầu. Các hệ thống máy móc, thiết bị và công nghệ phức tạp đều đóng góp vào chi phí này.

Bãi gửi xe tự động vòng xoay:



Hình 6: Bãi gửi xe tự động vòng xoay

Bãi gửi xe tự động dạng vòng xoay được thiết kế để tận dụng tối đa diện tích đất có sẵn. Cấu trúc xoay 2 quanh trục giúp nâng hạ các tầng xe một cách dễ dàng hơn, cho phép đậu được nhiều xe hơn trên diện tích nhỏ hơn so với các hệ thống đậu xe truyền thống, giảm thiểu diện tích cần thiết cho việc đậu xe.

Bãi gửi xe tự động dạng vòng xoay thường được trang bị các hệ thống an ninh như camera giám sát, cửa tự động và cơ chế kiểm soát truy cập để đảm bảo an toàn cho xe và người sử dụng. Việc tự động hóa quá trình đậu xe cũng giúp giảm thiểu nguy cơ va chạm hoặc hỏng hóc.

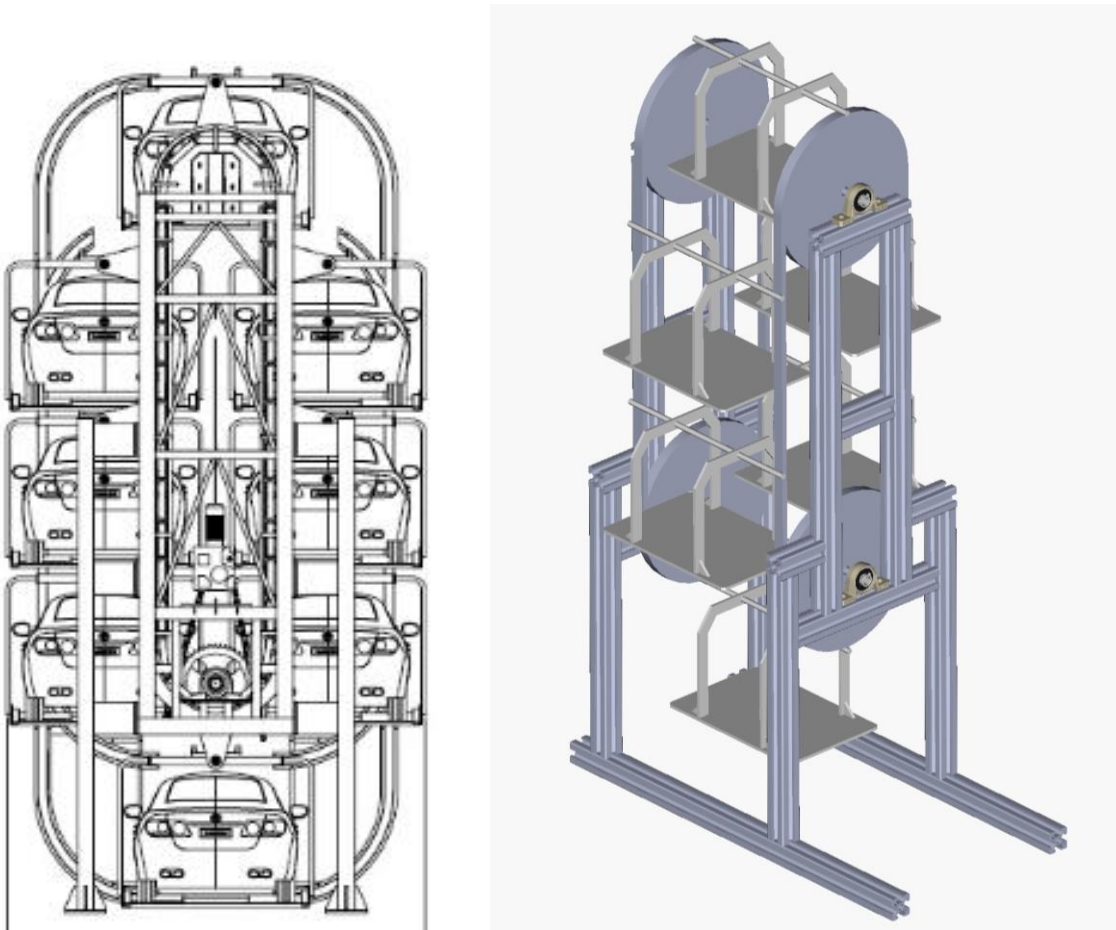
Người sử dụng có thể gửi xe tại một điểm đặt xe và hệ thống sẽ tự động đưa xe vào các vị trí đậu xe trống trên vòng xoay. Việc này giúp tiết kiệm thời gian và công sức cho người sử dụng, đồng thời tăng cường sự linh hoạt trong việc quản lý không gian đậu xe.

Hệ thống thanh toán thường được tích hợp trực tiếp vào quy trình đậu xe, cho phép người sử dụng thanh toán một cách dễ dàng thông qua các máy thu phí tự động hoặc ứng dụng di động.

Bãi gửi xe tự động dạng vòng xoay thường được thiết kế để hoạt động một cách hiệu quả về năng lượng, giảm thiểu lãng phí và tối ưu hóa hiệu suất hoạt động.

2.1.2 Phương án lựa chọn

Sau khi phân tích, phương án áp dụng vào đề tài được lựa chọn theo các chỉ tiêu đề ra như sau: tiết kiệm diện tích, hiệu quả kinh tế theo thời gian, tính tự động cao, dễ thực hiện về hiện thực hóa mô hình. Do đó phương án bãi gửi xe dạng vòng xoay được sử dụng vào đề tài của nhóm chúng em.



Hình 7: thiết kế mô hình Bãi gửi xe tự động vòng xoay của nhóm

2.2 Các cơ sở lý thuyết liên quan

2.2.1 Giới thiệu về ngôn ngữ Python và một số công cụ liên quan



Python là ngôn ngữ lập trình máy tính bậc cao thường được sử dụng để xây dựng trang web và phần mềm, tự động hóa các tác vụ và tiến hành phân tích dữ liệu. Python là ngôn ngữ có mục đích chung, nghĩa là nó có thể được sử dụng để tạo nhiều chương trình khác nhau và không chuyên biệt cho bất kỳ vấn đề cụ thể nào.

Tính linh hoạt này, cùng với sự thân thiện với người mới bắt đầu, đã khiến nó trở thành một trong những ngôn ngữ lập trình được sử dụng nhiều nhất hiện nay. Một cuộc khảo sát được thực hiện bởi công ty phân tích ngành RedMonk cho thấy rằng đây là ngôn ngữ lập trình phổ biến thứ hai đối với các nhà phát triển



Một số tiêu điểm về python:

- Python được phát triển vào cuối những năm 1980 bởi Guido van Rossum tại Viện Nghiên cứu Quốc gia về Toán học và Khoa học Máy tính ở Hà Lan với tư cách là người kế thừa ngôn ngữ ABC có khả năng xử lý và giao tiếp ngoài lề.
- Python có nguồn gốc từ các ngôn ngữ lập trình như ABC, Modula 3, small talk, Algol-68.
- Van Rossum đã chọn tên Python cho ngôn ngữ mới từ một chương trình truyền hình, Monty Python's Flying Circus.
- Trang Python là một tệp có phần mở rộng .py chứa có thể là sự kết hợp của Thẻ HTML và tập lệnh Python.
- Vào tháng 12 năm 1989, người sáng tạo đã phát triển trình thông dịch python đầu tiên như một sở thích, và sau đó vào ngày 16 tháng 10 năm 2000, Python 2.0 được phát hành với nhiều tính năng mới.
- Vào ngày 3 tháng 12 năm 2008, Python 3.0 được phát hành với nhiều thử nghiệm hơn và bao gồm các tính năng mới.
- Python là một ngôn ngữ kịch bản mã nguồn mở.
- Python là mã nguồn mở, có nghĩa là bất kỳ ai cũng có thể tải xuống miễn phí từ trang chủ và sử dụng nó để phát triển các chương trình. Mã nguồn của nó có thể được truy cập và sửa đổi theo yêu cầu trong dự án.
- Python là một trong những ngôn ngữ chính thức tại Google.

Ứng dụng của python

Python thường được sử dụng để phát triển trang web và phần mềm, tự động hóa tác vụ, phân tích dữ liệu và trực quan hóa dữ liệu. Vì tương đối dễ học, Python đã được nhiều người không phải là lập trình viên như kế toán và nhà khoa học áp dụng cho

nhiều công việc hàng ngày, chẳng hạn như tổ chức tài chính.

Ưu điểm:

- Dễ đọc và dễ hiểu: Python có cú pháp đơn giản và gần gũi với ngôn ngữ tự nhiên, giúp cho việc viết mã trở nên dễ dàng và dễ hiểu hơn. Điều này làm cho Python trở thành một lựa chọn tốt cho cả người mới bắt đầu và những lập trình viên kinh nghiệm.
- Cộng đồng lớn và sự hỗ trợ mạnh mẽ: Python có một cộng đồng lập trình viên rất lớn và năng động, với nhiều tài liệu, hướng dẫn và thư viện mã nguồn mở phong phú. Điều này giúp cho việc học và làm việc với Python trở nên dễ dàng hơn.
- Đa năng và linh hoạt: Python có thể được sử dụng cho nhiều mục đích khác nhau, từ việc viết script đơn giản cho đến phát triển ứng dụng phức tạp. Nó cũng có thể tích hợp được với nhiều ngôn ngữ và công nghệ khác.
- Tính mở rộng và phát triển: Python có một hệ thống thư viện và framework phong phú, cho phép lập trình viên mở rộng chức năng của ngôn ngữ theo nhu cầu cụ thể của dự án.
- Hỗ trợ cho nhiều nền tảng: Python có sẵn trên nhiều hệ điều hành khác nhau và có thể chạy trên nhiều loại thiết bị khác nhau, từ máy tính cá nhân đến thiết bị di động và thiết bị nhúng.

Với sự đa năng và dễ sử dụng của nó, Python là một lựa chọn lý tưởng cho việc lập trình các mô hình bãi gửi xe tự động. Bằng cách sử dụng Python, chúng ta có thể viết mã để điều khiển các hệ thống điều khiển, xử lý dữ liệu từ các cảm biến và thiết bị, và thậm chí xây dựng giao diện người dùng để tương tác với người dùng cuối. Do đó chúng em đã lựa chọn python là ngôn ngữ để lập trình xử lý ảnh và giao diện cho đề tài này.

Giải thuật xử lý ảnh nhận diện biển số xe

B1: Camera nhận lệnh chụp ảnh từ python và tiến hành chụp hình.

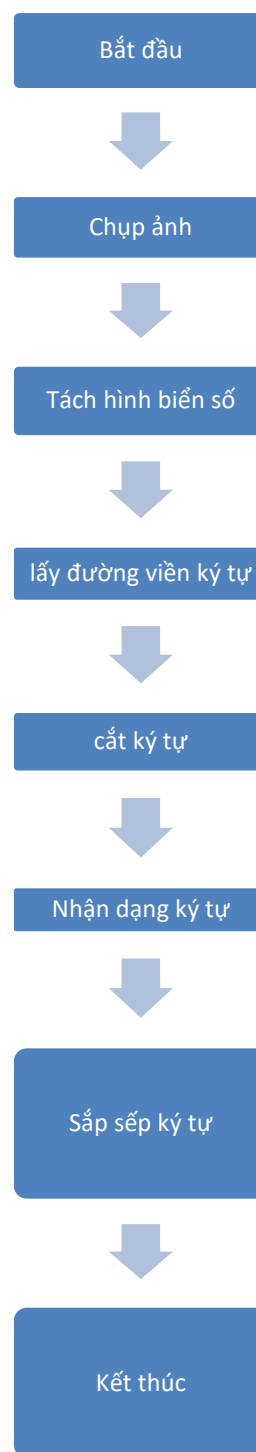
B2: Sau khi có được hình ảnh từ camera, chương trình tiến hành quá trình tách biển số xe ra khỏi hình.

B3: Từ biển số xe ta tiến hành lấy đường viền ảnh.

B4: Như bước ở trên ta đã có các đường viền xung quanh các ký tự, từ các đường viền này ta cắt riêng lẻ từng ký tự của biển số xe.

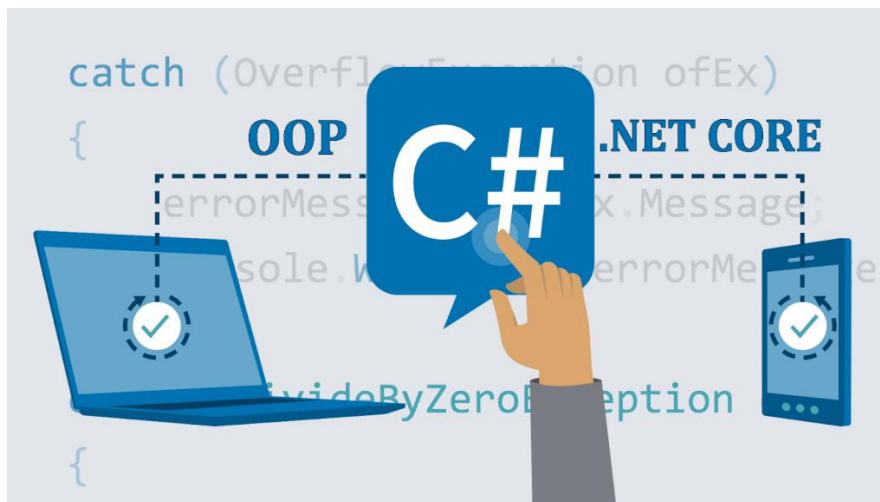
B5: Với những mẫu ký tự được cắt ra, ta đưa chúng vào thư viện nhận dạng ký tự có sẵn trong Emgu CV.

B6: Mặc dù đã nhận dạng được các ký tự, nhưng lúc này thứ tự của chúng vẫn rất lộn xộn. Do đó bước cuối cùng là sắp xếp chúng theo thứ tự dựa vào vị trí cắt của các ký tự ở bước



Hình 8: Giải thuật xử lý ảnh nhận diện biển số xe

2.2.2 Giới thiệu về ngôn ngữ C# và một số công cụ liên quan



C# (hay còn gọi là C Sharp, là ngôn ngữ lập trình hướng đối tượng, hiện đại do Microsoft phát triển như một phần của nền tảng .NET. Nó được giới thiệu lần đầu tiên vào năm 2000 và kể từ đó đã trở thành một trong những ngôn ngữ lập trình được sử dụng rộng rãi nhất trong ngành.

C# có thiết kế đơn giản, mạnh mẽ và linh hoạt. Sở hữu một cú pháp tương tự như các ngôn ngữ lập trình phổ biến, bao gồm Java và C++, giúp các nhà phát triển đã quen thuộc với các ngôn ngữ này dễ làm quen.

Một trong những tính năng chính của C# là an toàn kiểu của nó, giúp ngăn ngừa các lỗi lập trình có thể xảy ra với các ngôn ngữ khác. Không những vậy, nó còn hỗ trợ quản lý bộ nhớ tự động thông qua trình thu gom rác, giúp dễ dàng viết mã đáng tin cậy mà không làm rò rỉ bộ nhớ hay gây ra bất kỳ sự cố nào.

Nhờ những điều đó, nên C# được sử dụng rộng rãi trong việc phát triển các ứng dụng Windows dành cho máy tính để bàn, game, ứng dụng web và các ứng dụng dành cho thiết bị di động. Ngoài ra, nó còn được sử dụng trong phát triển phần mềm doanh nghiệp, với nhiều công ty lớn sử dụng C# để xây dựng các ứng dụng kinh doanh của họ.

Với tổng quan những điều trên, thì chúng ta dễ dàng nhận thấy C# là một ngôn ngữ mạnh mẽ và linh hoạt, phù hợp với nhiều nhiệm vụ lập trình. Cho dù bạn mới bắt đầu làm quen với lập trình hay là một nhà phát triển có kinh nghiệm, thì C# luôn là một ngôn ngữ tuyệt vời để học và sử dụng khi muốn xây dựng các ứng dụng mạnh mẽ, đáng tin cậy và có thể mở rộng.

Ứng dụng

Tương tự như các ngôn ngữ lập trình khác, C# được sử dụng để tạo các chương trình và ứng dụng khác nhau. Các ứng dụng đa dạng dành cho thiết bị di động, máy tính để bàn, dịch vụ dựa trên đám mây, trang web, phần mềm doanh nghiệp hay các ứng dụng trò chơi.

Do đó, C# được tin tưởng sử dụng nhiều trong các lĩnh vực chính:

- Phát triển ứng dụng máy tính: C# thường được sử dụng để phát triển các ứng dụng máy tính trên nền tảng Windows, bao gồm các ứng dụng desktop như ứng dụng quản lý, trò chơi, công cụ đồ họa, và phần mềm năng lượng.
- Phát triển ứng dụng di động: Sử dụng Xamarin, một công cụ phát triển ứng dụng di động được Microsoft sở hữu, bạn có thể sử dụng C# để phát triển ứng dụng di động cho cả iOS và Android.
- Phát triển ứng dụng web: C# thường được sử dụng trong việc phát triển ứng dụng web thông qua ASP.NET, một framework web được tích hợp trong .NET Framework. ASP.NET cung cấp các công cụ mạnh mẽ để phát triển ứng dụng web chuyên nghiệp và hiệu quả.

Ưu điểm

- C# giúp tiết kiệm thời gian: Sử dụng ngôn ngữ lập trình C# giúp tiết kiệm thời gian bởi nó được nhập tĩnh và rất dễ đọc. Mọi hành động tìm kiếm các tập lệnh để tìm ra các lỗi làm gián đoạn các chức năng của ứng dụng đều được tối ưu hoá thời gian ở mức độ tốt nhất. Ngoài ra, C# hoạt động theo

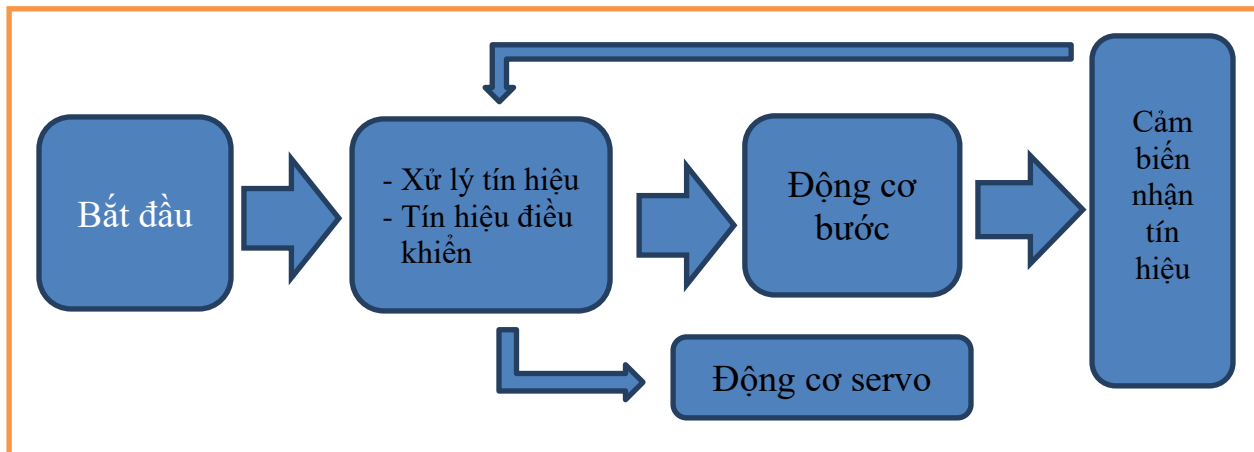
hướng đơn giản và hiệu quả cho phép các lập trình viên sử dụng ít thời gian để viết các đoạn mã phức tạp trong dự án của mình. Với một ngân hàng bộ nhớ đa dạng giúp tối ưu hoá thời gian lao động và cải thiện thời gian hoàn thành dự án đúng thời hạn.

- Hỗ trợ quá trình học tập hiệu quả: Không chỉ hỗ trợ thời gian triển khai dự án hiệu quả, C# chỉ yêu cầu lượng thời gian nhỏ để có thể học và thành thạo các thao tác triển khai công việc. C# cung cấp các tính năng đơn giản và dễ sử dụng cho phép người dùng mới có ít kỹ năng công nghệ có thể học tập và thao tác dễ dàng hơn.
- Ngôn ngữ lập trình mở rộng và dễ duy trì: Ngôn ngữ lập trình C# với khả năng mở rộng tốt và tính năng bảo trì hiệu quả bởi tính chất nghiêm ngặt trong quá trình viết mã tĩnh. Do đó mà các chương trình C# thường nhất quán và có mức độ đáng tin cậy cao. Cho phép người dùng điều chỉnh và bảo trì tốt hơn so với các chương trình được viết bằng ngôn ngữ khác. Đặc biệt, các ngăn xếp của ngôn ngữ lập trình C# không thay đổi sau nhiều lần triển khai và nâng cấp giúp cho các dự án được lưu trữ không bị thay đổi vị trí.
- Có một cộng đồng lớn: Mọi ngôn ngữ lập trình hoạt động không chỉ dựa vào một nền tảng hay dịch vụ được hỗ trợ bởi các chuyên gia CNTT. Các lập trình viên sẽ phải dựa vào sự hỗ trợ của cộng đồng trong cùng lĩnh vực nghiên cứu ngôn ngữ lập trình với những kinh nghiệm và kiến thức chuyên môn cao. Các cộng đồng chuyên gia lập trình này có thể tìm kiếm trên StackOverflow.

Hầu hết các trang web Hỏi và Đáp được triển khai dựa trên ngôn ngữ lập trình C# nên các nhà phát triển đã tạo nên một phần lớn cộng đồng cho phép người dùng hỏi, trả lời hoặc thực hiện các tương tác khác nhau. C# còn có cộng đồng tương tác cá nhân trên Meetuo.com cho phép mọi thành viên tham gia các cuộc thảo luận trực tuyến và IRL được lên lịch ngẫu nhiên hoặc nhất quán.

- C# là hướng đối tượng: Hướng đối tượng là tính năng đặc biệt nhất của ngôn ngữ lập trình C# với nhiều lợi thế khác nhau như tính hiệu quả và linh hoạt. Tuy nhiên, với một số nhà phát triển không quen với OOP có thể không muốn lựa chọn ngôn ngữ lập trình C#. Hãy yên tâm vì C# là một ngôn ngữ lập trình hướng đối tượng khá thân thiện với người dùng

Sơ đồ giải thuật C# điều khiển động cơ



Hình 9: Sơ đồ giải thuật C# điều khiển động cơ

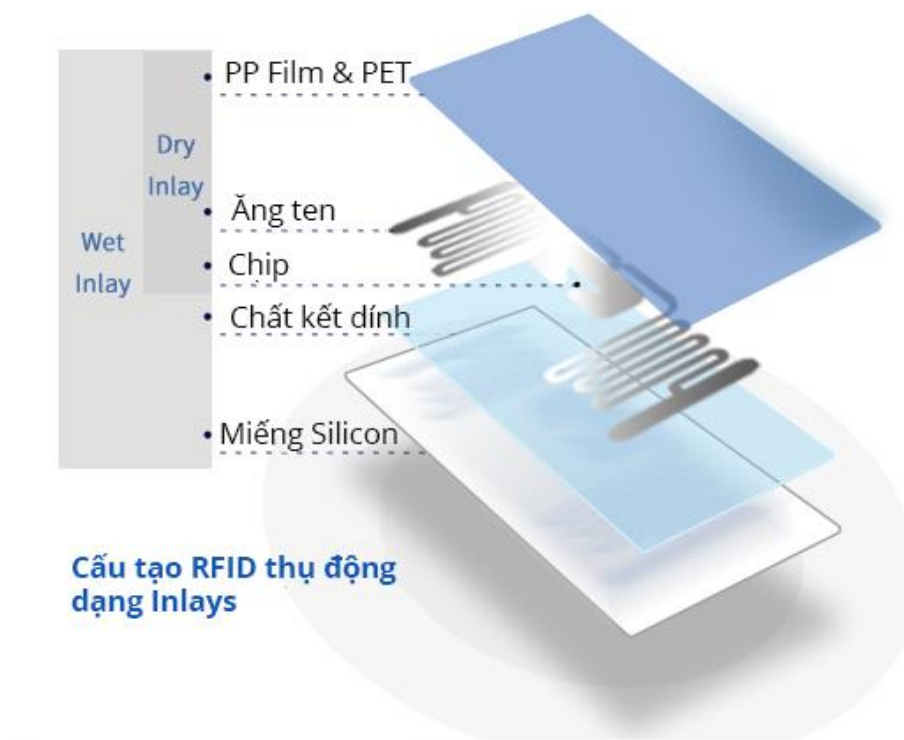
2.2.3 Công Nghệ RFID

Giới thiệu

RFID là công nghệ nhận dạng đối tượng bằng sóng vô tuyến. Công nghệ này cho phép nhận biết các đối tượng thông qua hệ thống thu phát sóng radio, cho phép ghi và đọc dữ liệu từ một thẻ hoặc tag thông qua sóng radio từ đó có thể giám sát, quản lý hoặc lưu vết từng đối tượng. RFID thường được sử dụng để nhận diện và theo dõi các đối tượng, hàng hóa, và người dùng một cách tự động và không cần tiếp xúc.

Cấu tạo

Hệ thống RFID gồm ba phần chính: tag (thẻ), thiết bị đọc (reader), và hệ thống phần mềm để xử lý dữ liệu.



Hình 10: cấu tạo Thẻ RFID

Thẻ RFID chứa một đoạn mã không trùng lặp giữa các thẻ được lưu trữ trong một chip và được truyền qua sóng radio.

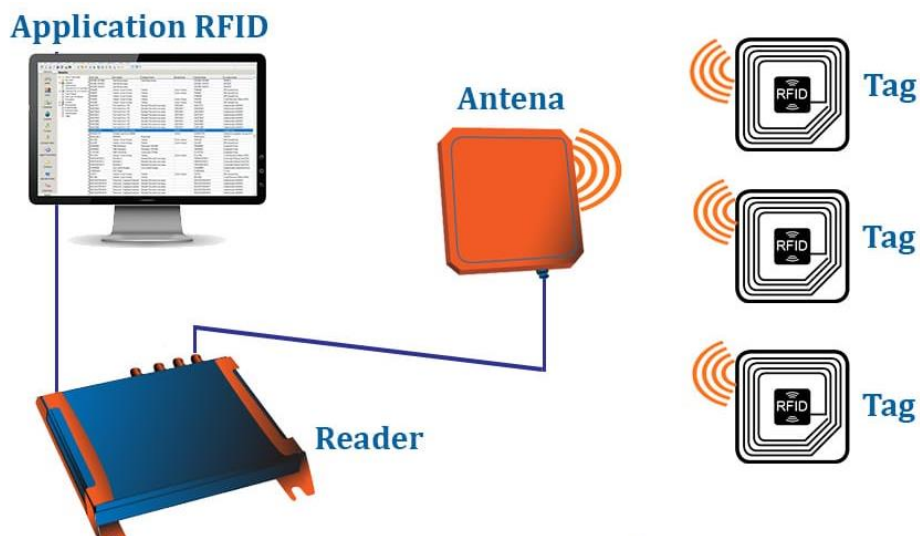


Hình 11: Thiết bị đọc RFID RC522

Thiết bị đọc RFID là một thiết bị điện tử có khả năng gửi và nhận sóng radio để tương tác với các thẻ RFID.

Nguyên lý hoạt động

Thiết bị RFID reader phát ra sóng điện tử ở một tần số nhất định, khi thiết bị thẻ RFID trong vùng hoạt động sẽ cảm nhận được sóng điện tử này và thu nhận năng lượng, từ đó phát lại cho thiết bị RFID reader biết mã số của mình. Từ đó thiết bị RFID reader nhận biết được tag nào đang trong vùng hoạt động.



Hình 12: Nguyên lý hoạt động RFID

Ứng dụng

Thẻ chip (tag) RFID chứa rất nhiều mã nhận dạng khác nhau, thông thường là 32 bit tương ứng với hơn 4 tỷ mã số khác nhau. Ngoài ra khi xuất xưởng mỗi thẻ chip RFID được gắn một mã số khác nhau. Do vậy, khi một vật được gắn chip RFID thì khả năng nhận dạng nhầm với 1 thẻ chip RFID khác là rất thấp, xác suất là 1 phần 4 tỷ. Với ưu điểm về mặt công nghệ như vậy nên sự bảo mật và độ an toàn của các thiết bị ứng dụng công nghệ RFID là rất cao.

- Theo dõi và quản lý hàng hóa trong chuỗi cung ứng và logistics.
- Quản lý hàng tồn kho trong các kho hàng và cửa hàng bán lẻ.
- Theo dõi vật nuôi trong nông nghiệp và công nghiệp thực phẩm.
- Điều khiển truy cập vào các khu vực an ninh và kiểm soát ra vào.
- Quản lý tài sản trong các tổ chức và doanh nghiệp.
- Thẻ thông minh cho giao thông công cộng và điều khiển ra vào trong các tòa nhà và công trình.

2.3 Giới thiệu một số phần mềm lập trình

Phần mềm lập trình, hay còn được gọi là trình biên dịch hoặc môi trường phát triển tích hợp (IDE), là các công cụ máy tính được thiết kế để hỗ trợ lập trình viên trong việc tạo, kiểm tra và gỡ lỗi mã nguồn.

Một số chức năng phổ biến của phần mềm lập trình:

- **Chỉnh sửa mã:** Hỗ trợ lập trình viên viết mã bằng cách cung cấp giao diện dễ dùng, kiểm tra cú pháp, và đôi khi đề xuất tự động một số câu lệnh.
- **Biên dịch và liên kết:** Chuyển đổi mã nguồn viết bằng một ngôn ngữ lập trình thành mã máy hoặc mã trung gian, sau đó liên kết nó với các thư viện hoặc module khác để tạo thành một chương trình hoàn chỉnh.
- **Gỡ lỗi:** Cung cấp các công cụ để giúp lập trình viên tìm ra và sửa lỗi trong mã

nguồn của họ.

- Phân tích mã: Cung cấp phân tích về chất lượng mã, hiệu suất và các vấn đề khác.
- Quản lý dự án: Giúp tổ chức và quản lý các tệp, thư mục, và tài nguyên khác của một dự án phần mềm.
- Hỗ trợ version control: Tích hợp với hệ thống quản lý phiên bản như Git để giúp lập trình viên quản lý các phiên bản khác nhau của mã nguồn.
- Emulator và simulator: Cho phép lập trình viên kiểm tra chương trình của họ trong một môi trường giả lập, thích hợp cho việc phát triển phần mềm trên nền tảng khác nhau.

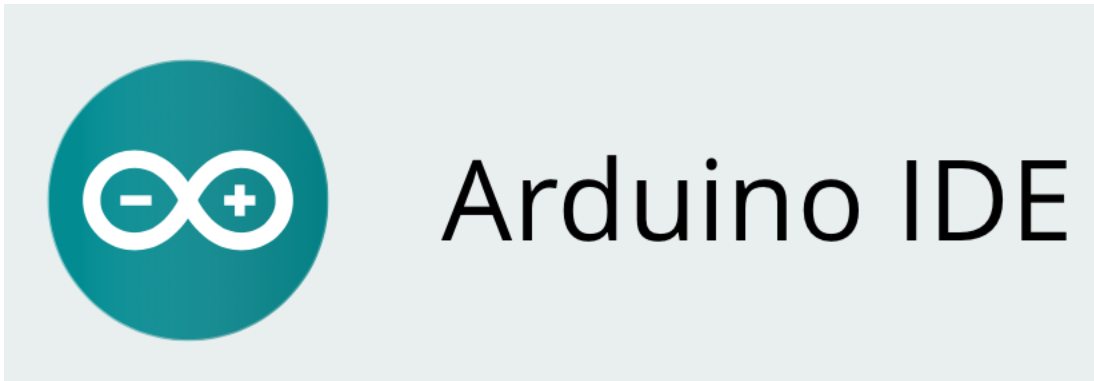
2.3.1 Phần mềm Python IDLE



Hình 13: giao diện Phần mềm Python IDLE

Là một IDE cơ bản được cung cấp sẵn với Python và hỗ trợ cho các tính năng như gỡ lỗi, trình biên tập mã nguồn và hỗ trợ cho shell Python. Python IDLE sẽ là một phần mềm lập trình thân thiện và dễ dùng cho bạn. Phần mềm này đi kèm khi cài đặt Python. IDLE cung cấp một trình biên tập mã nguồn với khả năng tô màu cú pháp, tự động hoàn thiện mã và các chức năng chỉnh sửa cơ bản khác. Ngoài ra, IDLE có các tính năng gỡ lỗi cơ bản, như đặt điểm dừng (breakpoints) và kiểm tra biến.

2.3.2 Phần mềm Arduino IDE



Hình 14: Phần mềm Arduino IDE

Giới thiệu

Arduino IDE là một phần mềm mã nguồn mở chủ yếu được sử dụng để viết và biên dịch mã vào module Arduino.

Đây là một phần mềm Arduino chính thức, giúp cho việc biên dịch mã trở nên dễ dàng mà ngay cả một người bình thường không có kiến thức kỹ thuật cũng có thể làm được.

Nó có các phiên bản cho các hệ điều hành như MAC, Windows, Linux và chạy trên nền tảng Java đi kèm với các chức năng và lệnh có sẵn đóng vai trò quan trọng để gỡ lỗi, chỉnh sửa và biên dịch mã trong môi trường.

Có rất nhiều các module Arduino như Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro và nhiều module khác.

Mỗi module chứa một bộ vi điều khiển trên bo mạch được lập trình và chấp nhận thông tin dưới dạng mã.

Mã chính, còn được gọi là sketch, được tạo trên nền tảng IDE sẽ tạo ra một file Hex, sau đó được chuyển và tải lên trong bộ điều khiển trên bo.

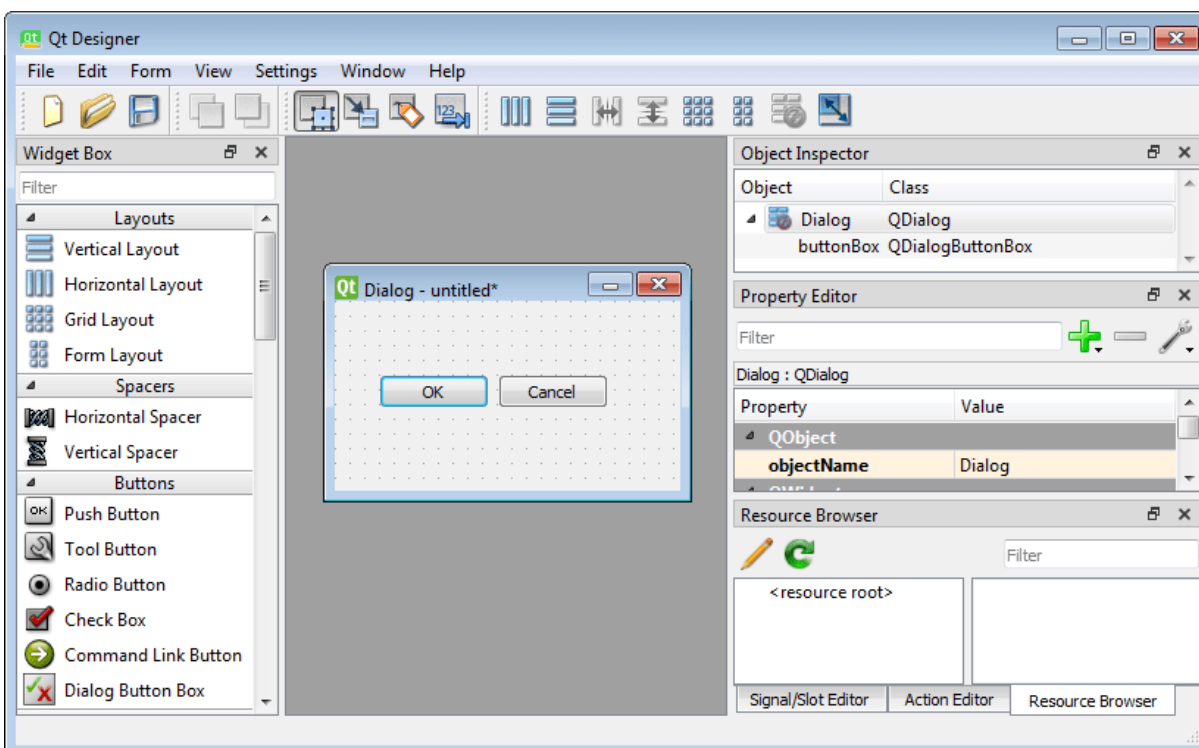
Môi trường IDE chủ yếu chứa hai phần cơ bản: Trình chỉnh sửa và Trình biên dịch, phần đầu sử dụng để viết mã được yêu cầu và phần sau được sử dụng để biên dịch và tải mã lên module Arduino.

Môi trường này hỗ trợ cả ngôn ngữ C và C++.

2.4.3 Phần mềm qt Designer

Giới thiệu

Qt Designer là một công cụ để nhanh chóng xây dựng giao diện người dùng đồ họa với các widget từ khung Qt GUI. Nó cung cấp cho bạn một giao diện kéo và thả đơn giản để bố trí các thành phần như nút, trường văn bản, hộp tổ hợp và hơn thế nữa. Đây là ảnh chụp màn hình của Qt Designer trên Windows:



Hình 15: giao diện làm việc trên Qt Designer

Qt Designer tạo ra “.ui” các tệp. Đây là một định dạng dựa trên XML đặc biệt để lưu trữ các widget dưới dạng cây. Ta có thể tải các tệp này trong thời gian chạy hoặc dịch chúng sang ngôn ngữ lập trình như C++ hoặc Python.

Chúng em sử dụng Qt Designer cùng với Python vì nó là một ngôn ngữ động có lợi cho việc tạo mẫu nhanh.

2.5 Giới thiệu phần cứng

Trong đề đề tài này chúng em sử dụng một số linh kiện phần cứng sau:

- Thiết bị xử lý trung tâm:
 - bo mạch arduino nano
 - máy tính
- Thiết bị đầu vào:
 - cảm biến tiệm cận phát hiện kim loại
 - webcam
 - nút bấm
 - mạch quét thẻ RFID
- thiết bị đầu ra:
 - động cơ bước
 - động cơ servo
 - đèn LED
 - loa
- các thiết bị khác:
 - nguồn adapter 12V 5A
 - jack nhận nguồn
 - modun relay
 - Driver động cơ bước
 - Mạch hạ áp
 - Ic ổn áp lm7805, transistor, tụ,...

- phần cứng cơ khí

- khung nhôm
- nhựa mica
- nhôm xích, xích
- trục quay inox, vòng bi (bạc đạn), đai cô định, đai ốc
- ốc vít, que nhôm, đai bánh răng, dây đai (curoa), giá cô định khung,...

CHƯƠNG 3 THIẾT KẾ

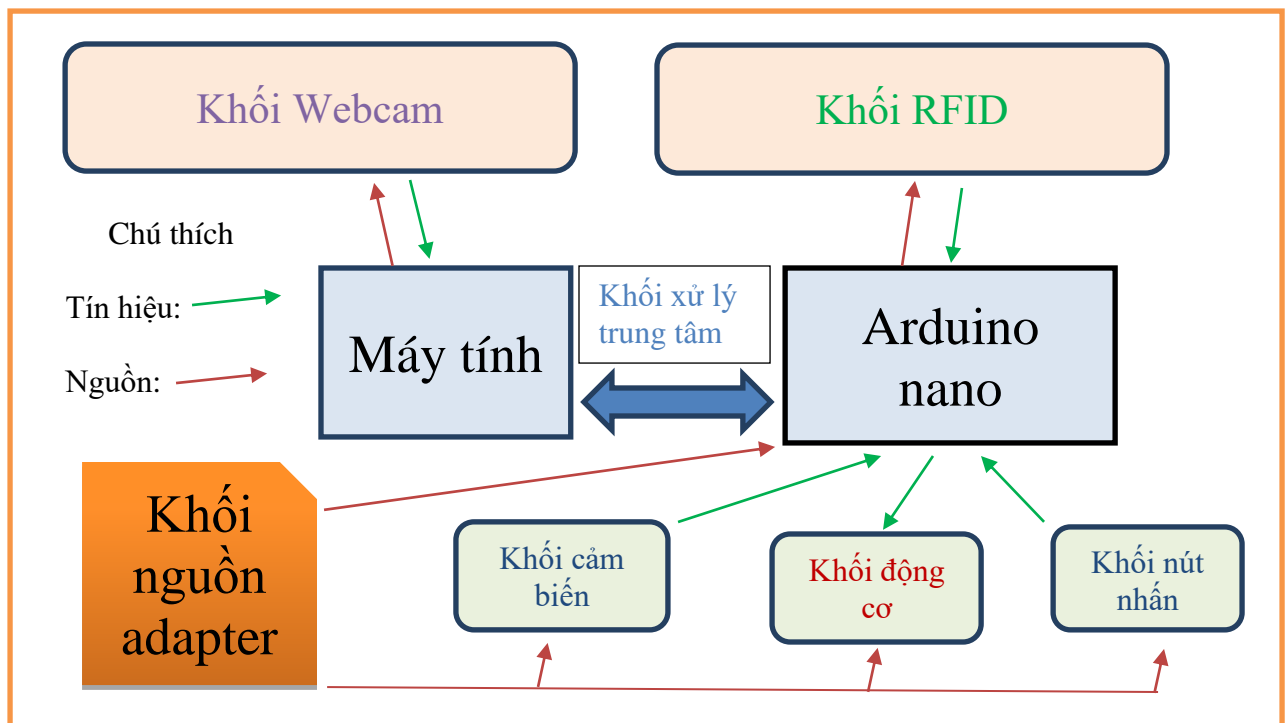
3.1 Giới thiệu

Với đề tài này thì yêu cầu phải làm sao cho cơ cấu có thể nâng, hạ và lấy xe ra vào khung để có thể dễ dàng thao tác trong việc lấy và cất xe, cùng với đó thì việc giao tiếp quét thẻ RFID và chụp ảnh từ Camera cũng như hoạt động của các motor. Vì vậy nhóm phải thiết kế các yêu cầu sau:

- Thiết kế mạch điều khiển các động cơ.
- Thiết kế mạch kết nối vào với cảm biến và công tắc.
- Thiết kế mạch driver kết nối ngõ ra của arduino để điều khiển động cơ bước.
- Thiết kế bộ đọc thẻ RFID từ mạch đọc thẻ RFID - RC522 và kết nối Arduino nano

3.2 Thiết kế hệ thống

3.2.1 Thiết kế sơ đồ khối hệ thống



Hình 16: sơ đồ khối hệ thống

Chức năng của từng khối

Khối nguồn adapter: cung cấp nguồn cho toàn bộ hệ thống hoạt động.

Khối đọc thẻ RFID: bộ đọc thẻ từ RFID có nhiệm vụ đọc mã thẻ.

Khối webcam: dùng camera để chụp ảnh.

Khối cảm biến: xác định vị trí tầng trung tâm khi khởi động.

Khối động cơ: động cơ bước 12VDC và động cơ servo để vận hành mô hình.

Khối xử lý trung tâm:

+ Sử dụng laptop tích hợp sẵn những công cụ lập trình có nhiệm vụ thu nhận, xử lý, lưu trữ mã thẻ từ đầu đọc thẻ RFID, nhận dạng biển số từ hình chụp được bằng Camera, giao tiếp với arduino.

+ Arduino nano nhận tín hiệu điều khiển từ máy tính và các tín hiệu ngõ vào để xử lý và điều khiển động cơ.

Hoạt động của hệ thống

Hệ thống gồm 6 thẻ ứng với 6 tầng, mỗi mã thẻ được lưu cố định với một tầng.

- Cất xe:

Khi có xe vào thì nhân viên bấm nút “vào” và quét thẻ, lúc này hệ thống bắt đầu việc so sánh xem mã thẻ xe vừa quét ứng với tầng nào, nếu mã thẻ nhận vào là đúng thì hệ thống điều khiển motor bước quay trục hệ thống hạ tầng ứng với mã thẻ xuống dưới cùng. Sau đó động cơ servo điều khiển thanh chắn barie trước hoạt động, thanh chắn barie được mở ra, ta cho xe vào và thực hiện nhận biển số. Khi xe vào tầng, ta quét thẻ thêm lần nữa, hệ thống bắt đầu chụp ảnh xử lý ảnh cho ra thông tin bằng kí tự văn bản ứng với biển số xe đó, sau đó lưu vào biến nhớ ứng với thẻ xe và điều khiển động cơ servo cho thanh chắn barie trước đóng lại, xe vào vị trí và hoàn tất việc cất xe.

- Trả xe

Cũng như việc cất xe thì việc lấy xe trả khách cũng tương tự. Lần lượt khách bấm nút “ra” và quét thẻ sau đó hệ thống sẽ xem xe đã có trong bãi chưa, nếu đã có thì bắt đầu di chuyển trực quay đến vị trí tầng trả xe cho khách, thanh chắn barie sau mở ra, xe rời khỏi bãi, thanh chắn barie đóng lại.

- Sự cố mất thẻ

Vì khách hàng mất thẻ từ nên việc lấy xe bằng quá trình quét thẻ không được thực hiện. Như vậy, hành khách muốn lấy được xe buộc phải chứng minh xe mình được gửi trong bãi bằng các giấy tờ tùy thân. Nhân viên kiểm tra giấy tờ của khách hàng, kiểm tra biển số xe có tồn tại trong cơ sở dữ liệu không. Nếu có thì truy xuất vị trí đỗ xe để tiến hành trả xe với chế độ điều khiển bằng tay và lưu lại thông tin khách hàng mất thẻ. Sau đó tiến hành xóa thông tin xe đã gửi trong cơ sở dữ liệu.

3.2.2 Thiết kế phần cứng

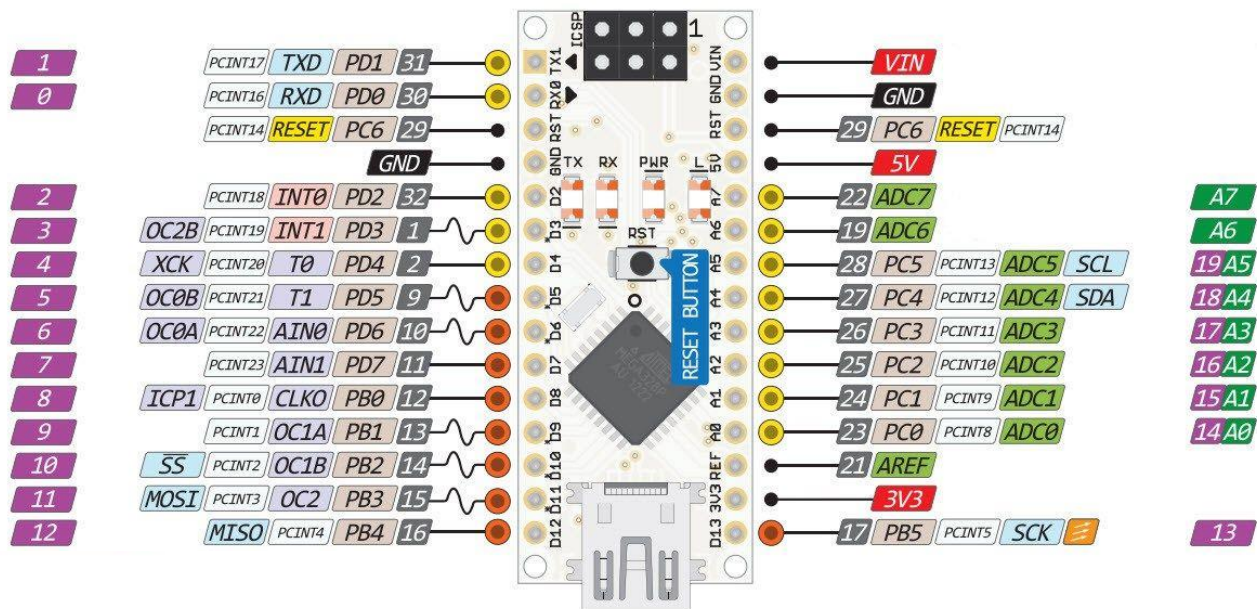
a) khối xử lý trung tâm:

Khối xử lý trung tâm bao gồm Arduino nano CH340 và máy tính giao tiếp với nhau qua giao thức UART hay hiểu một cách nôm na là Serial.

Arduino nano CH340 được lựa chọn trong đề tài này có các thông số cơ bản sau:

- Nguồn : 2.7V – 5.5V
- Input/Output : 23
- Số chân : 28
- Dòng hoạt động : 1.5mA
- Dòng chờ : 1uA
- Dòng output : 20mA
- Thạch anh ngoài : 16Mhz

- Thạch anh nội : 8Mhz
- Các chuẩn giao tiếp : I²C, SPI, UART / USART, USB
- Bộ nhớ ROM : 32KByte
- Bộ nhớ SRAM : 2KByte
- Bộ nhớ EEPROM: 1KByte
- Nhiệt độ hoạt động : -40°C ~ 85°C
- Cổng ADC : 10Bit – 8 Kênh
- Timer : 8Bit hoặc 16Bit

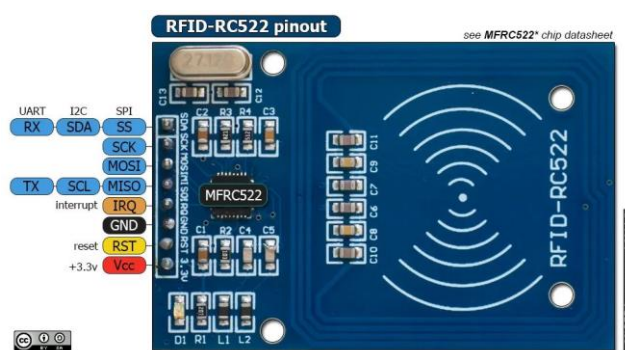


Hình 17: sơ đồ chức năng chân của mạch Arduino Nano CH3

b. Thiết kế mạch đọc thẻ RFID

RFID là chữ viết tắt của Radio Tần số Nhận dạng và đây là công nghệ tự động được sử dụng rộng rãi trong nhiều ngành cho các nhiệm vụ như theo dõi nhân sự, kiểm soát truy cập, quản lý chuỗi cung ứng, theo dõi sách trong thư viện, hệ thống trạm thu phí, v.v.

Trong bài này nhóm chúng em sử dụng mạch đọc thẻ RC522 cho mạch thực hiện



Hình 18: Module đọc thẻ RC522

Module đọc thẻ RC522 có thể đọc được các loại thẻ có kết nối không dây như NFC, thẻ từ (loại dùng làm thẻ giảm giá, thẻ xe bus, tàu điện ngầm...). Module có các thông số chính như:

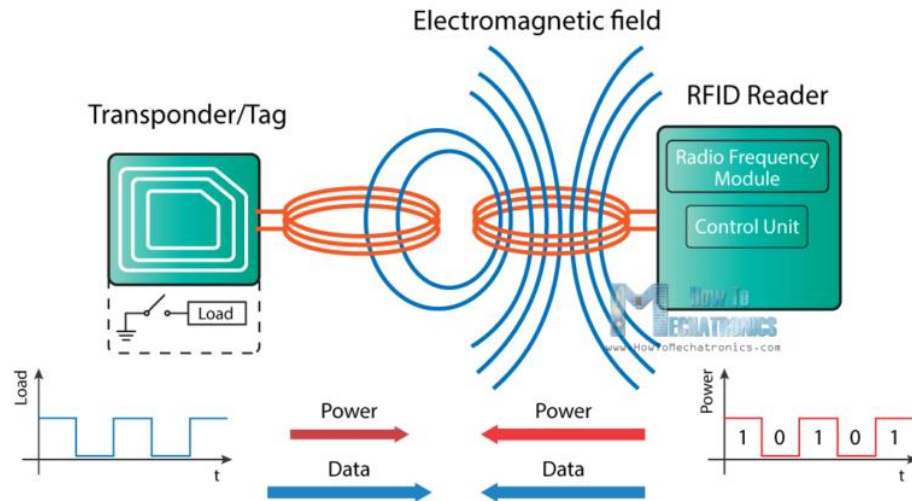
- Điện áp nuôi: 3.3V;
- Dòng điện nuôi :13-26mA
- Tần số hoạt động: 13.56MHz
- Khoảng cách hoạt động: 0 ~ 60 mm
- Cổng giao tiếp: SPI, tốc độ tối đa 10Mbps
- Kích thước: 40mm x 60mm
- Có khả năng đọc và ghi.

Phần cứng cần có:

- Mạch arduino
- Modul đọc thẻ RFID RC522
- Dây kết nối, bảng mạch
- Thẻ NFC

Đầu đọc RFID bao gồm một mô-đun tần số vô tuyến, bộ điều khiển và cuộn ăng ten

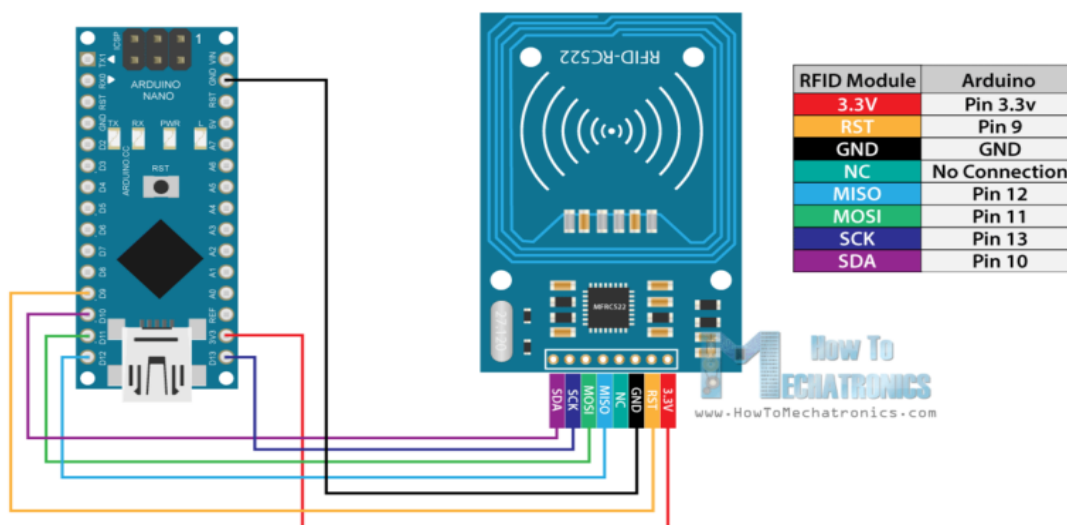
tạo ra trường điện từ tần số cao. Mặt khác, thẻ thường là một linh kiện thụ động, chỉ bao gồm ăng-ten và vi mạch điện tử, do đó, khi nó ở gần trường điện từ của bộ thu phát, do cảm ứng, một điện áp được tạo ra trong cuộn ăng ten của nó và điện áp phục vụ như là nguồn điện cho vi mạch.



Hình 19: nguyên lý thu nhận tín hiệu của mạch RFID

Khi thẻ được cấp nguồn, nó có thể trích xuất thông tin được truyền từ đầu đọc và gửi tin nhắn trở lại đầu đọc, nó sử dụng một kỹ thuật gọi là thao tác tải. Bật và tắt tải ở ăng-ten của thẻ sẽ ảnh hưởng đến mức tiêu thụ điện của ăng-ten của bộ đọc có thể được đo là sụt áp. Sự thay đổi điện áp này sẽ được ghi lại dưới dạng 0 và 1 và đó là cách dữ liệu được truyền từ thẻ đến đầu đọc.

Cách kết nối modul RFID với arduino:



Hình 20: Cách kết nối modul RFID với arduino

- Chân VCC của mô-đun RFID RC522 được kết nối với chân 3.3V của Arduino Uno.
- Chân RST của mô-đun RFID RC522 được kết nối với chân digital 9 trên Arduino Uno.
- Chân SS (Slave Select) của mô-đun RFID RC522 được kết nối với chân digital 10 trên Arduino Uno.
- Chân MOSI của mô-đun RFID RC522 được kết nối với chân digital 11 trên Arduino Uno.
- Chân MISO của mô-đun RFID RC522 được kết nối với chân digital 12 trên Arduino Uno.
- Chân SCK của mô-đun RFID RC522 được kết nối với chân digital 13 trên Arduino Uno.

Để đọc dữ liệu từ thẻ RFID, chúng ta cần sử dụng thư viện MFRC522 và viết mã Arduino tương ứng.

c. khối cảm biến, công tắc

Trong bài này nhóm chúng em sử dụng một cảm biến tiệm cận để dò tầng



Hình 21: Cảm biến tiệm cận LJ8A3-2-Z/BX NPN 2mm

- Điện áp: 5-24V
- Loại cảm biến: NPN (NO)
- Khoảng cách phát hiện: ≤ 2 mm
- Dòng chịu tải: < 300 mA
- Đối tượng phát hiện: Kim loại
- Kích thước lắp đặt: lỗ 8mm

Ngoài ra còn có các nút bấm cơ học để thao tác sử dụng mô hình



Hình 22: Nút nhấn nhà R13-507 16mm

Các chức năng sử dụng gồm:

- 2 nút “star”
- 2 nút “stop” và
- 1 nút “reset”

d. Khôi webcam:

Trong bài này nhóm chúng em sử dụng một webcam dùng làm thiết bị thu thập hình ảnh để xử lý ảnh đọc biển số



Hình 23: Webcam máy tính AUKEY PC-W3

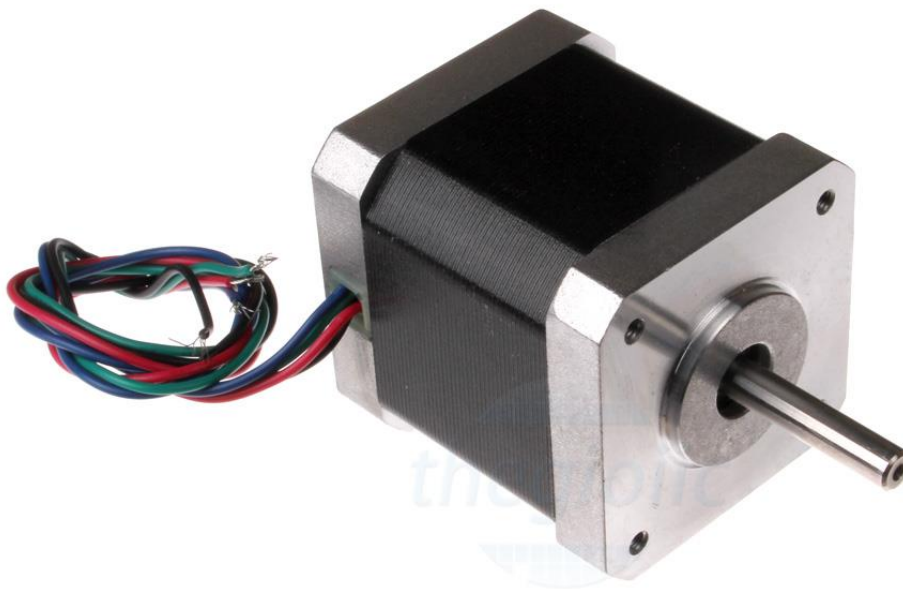
Với các thông số kỹ thuật như sau:

Model	PC-W3
Sensor	1/2.9" CMOS Image Sensor
Recording Resolution	1080p (30fps), 720p (30fps), 480p (30fps), 240p (30fps)
Video Format	MJPEG, YUV
Effective Pixels	2.0 Megapixels
Focus Distance	0.3 to 5m
Viewing Angle	85°
Minimum Illumination	≤5 lux
Microphone	Dual stereo microphones
Input	DC 5V 200mA (Max)
System Requirements	Windows XP / Mac OS 10.6 / Android 5.0 or above
Material	ABS
Operating Temperature	-10°C - +45°C (14°F - 113°F)
USB Cable Length	180cm / 5.9'
Size	81 x 26.5 x 25.5mm / 3.19" x 1.04" x 1.00"

Hình 24: thông số kỹ thuật webcam sử dụng

e. Khối động cơ:

Trong bài này nhóm chúng em sử dụng gồm 1 động cơ bước DC 12V và 2 động cơ servo:



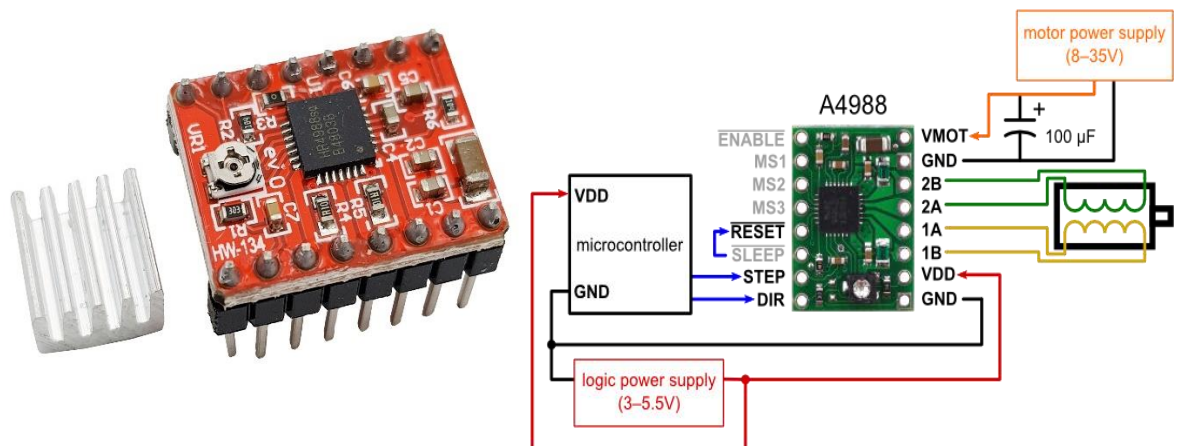
*Hình 25: Động cơ bước SUMTOR 42HS3413A4 – 0.28Nm – 1.3a – 42*34mm*

Động cơ bước SUMTOR 42HS3413A4 – 0.28Nm – 1.3a – 42*34mm

Được dùng làm động cơ chính của mô hình, để quay trục bánh răng để nâng hạ các tầng đồ xe

Thông số kỹ thuật chính:

- Kích thước mặt bích: 42×42 mm.
- Chiều dài thân: 34 mm.
- Dòng chịu tải: 1.3A.
- Moment xoắn: 0.28Nm
- Góc bước: 1.8°/step



Hình 26: Driver động cơ bước a4988

Để điều khiển được động cơ bước từ tín hiệu điều khiển arduino, ta cần thêm mạch driver giúp arduino giao tiếp được với động cơ, trong bài này nhóm chúng em sử dụng Driver động cơ bước a4988 để điều khiển động cơ bước.



Hình 27: động cơ Servo SG90S

Động cơ Servo SG90S được dùng để nâng hạ thanh chắn barie trước và sau mô hình bãi gửi xe với thông số chính:

- Trọng lượng: 9g
- Điện áp: 5V

- Màu sắc: Xanh
- Góc quay: 180^0

3.2.3 Sơ đồ nguyên lý toàn mạch

Nguyên lý hoạt động của hệ thống:

- **Cất xe:**

Khi có xe vào thì người dùng bấm nút “vào” và quét thẻ, lúc này hệ thống bắt đầu việc so sánh xem mã thẻ xe vừa quét ứng với tầng nào, nếu mã thẻ nhận vào là đúng thì hệ thống điều khiển motor bước quay trục hệ thống hạ tầng ứng với mã thẻ xuống dưới cùng. Sau đó động cơ servo điều khiển thanh chắn barie trước hoạt động, thanh chắn barie được mở ra, ta cho xe vào và thực hiện nhận biển số. khi xe vào tầng, ta quét thẻ thêm lần nữa, hệ thống bắt đầu chụp ảnh xử lý ảnh cho ra thông tin bằng kí tự văn bản ứng với biển số xe đó, sau đó lưu vào biến ứng với thẻ xe và điều khiển động cơ servo cho thanh chắn barie trước đóng lại, xe vào vị trí và hoàn tất việc cất xe.

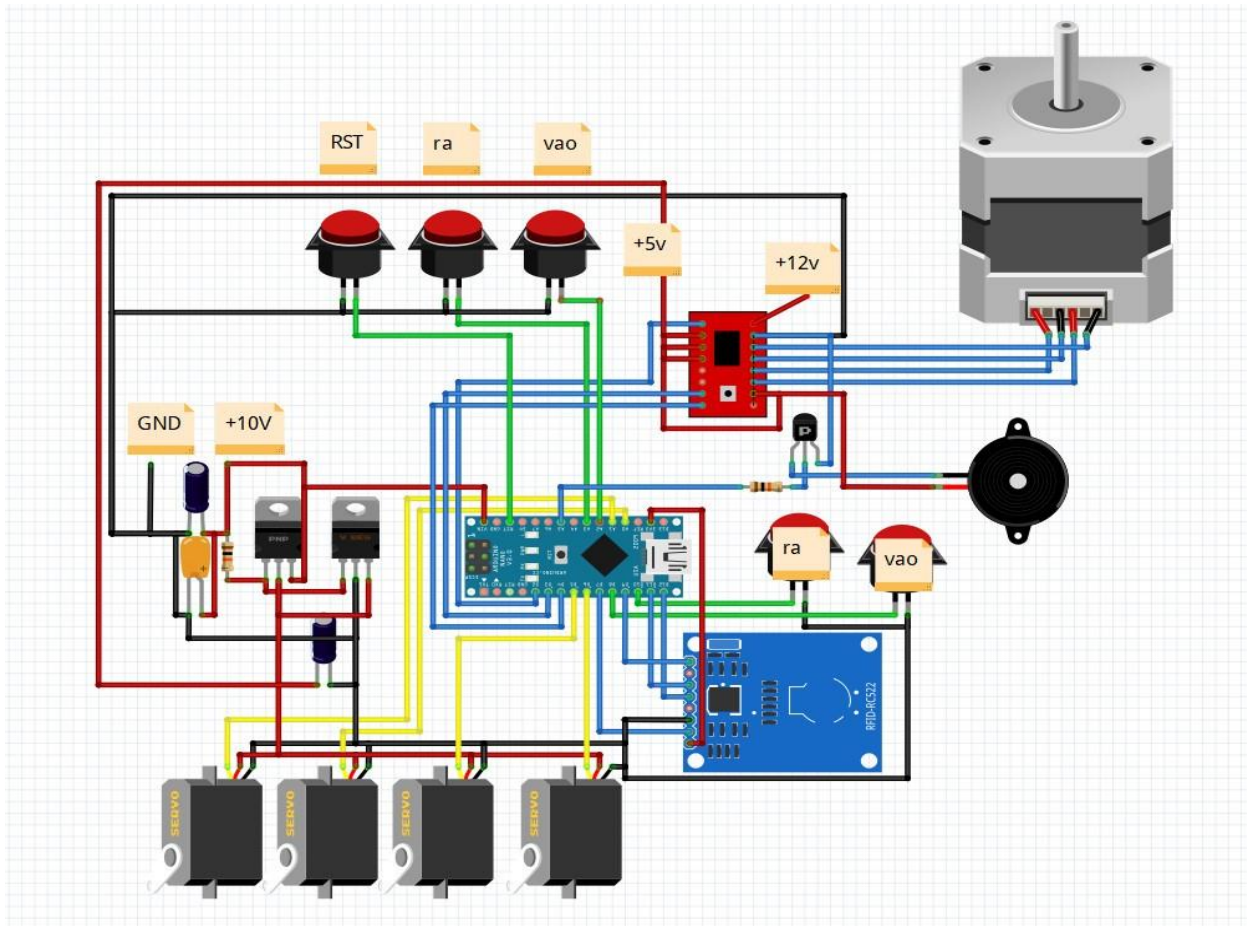
- **Trả xe**

Cũng như việc cất xe thì việc lấy xe trả khách cũng tương tự. Lần lượt khách bấm nút “ra” và quét thẻ sau đó hệ thống sẽ xem xe đã có trong bãi chưa, nếu đã có thì bắt đầu di chuyển trục quay đến vị trí tầng trả xe cho khách, thanh chắn barie sau mở ra, xe rời khỏi bãi, thanh chắn barie đóng lại.

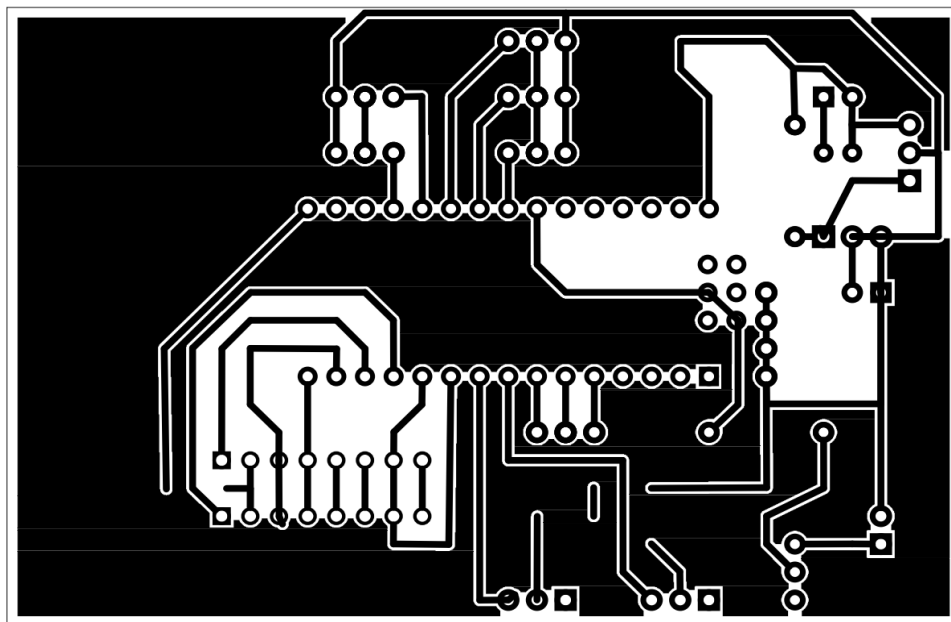
- **Sự cố mất thẻ**

Vì khách hàng mất thẻ từ nên việc lấy xe bằng quá trình quét thẻ không được thực hiện. Như vậy, hành khách muốn lấy được xe buộc phải chứng minh xe mình được gửi trong bãi bằng các giấy tờ tùy thân. Nhân viên kiểm tra giấy tờ của khách hàng, kiểm tra biển số xe có tồn tại trong cơ sở dữ liệu không. Nếu có thì truy xuất vị trí đỗ xe để tiến hành trả xe với chế độ điều khiển bằng tay và lưu lại thông tin khách hàng mất thẻ. Sau đó tiến hành xóa thông tin xe đã gửi trong cơ sở dữ liệu.

Sơ đồ kết nối toàn mạch



Hình 28: Sơ đồ nguyên lý đi dây của mạch thực hiện



Hình 29: bản thiết kế mạch in thực hiện

Chương 4

THI CÔNG HỆ THỐNG

4.1 Giới thiệu

Sau khi thiết kế hoàn chỉnh sơ đồ nguyên lý cho toàn hệ thống, nhóm tiến hành thi công mô hình. Hệ thống được thi công bao gồm hai phần chính là thi công phần cứng và thi công phần mềm. Cụ thể như sau:

- Về phần cứng: tiến hành lắp ráp các thiết bị vào mô hình đã gia công trước đó, kết nối các linh kiện điện tử, do nhóm sử dụng Arduino nano để kết nối các linh kiện nên nhóm có thiết kế một mạch in kết nối các mô đun với nhau.
- Về phần mềm: xây dựng giải thuật và viết chương trình cho hệ thống. Chương trình được lập trình dựa vào nguyên lý hoạt động của hệ thống từ khi cấp nguồn cho đến khi hệ thống ngừng hoạt động, áp dụng được giải thuật điều khiển vào mô hình một cách tối ưu nhất.

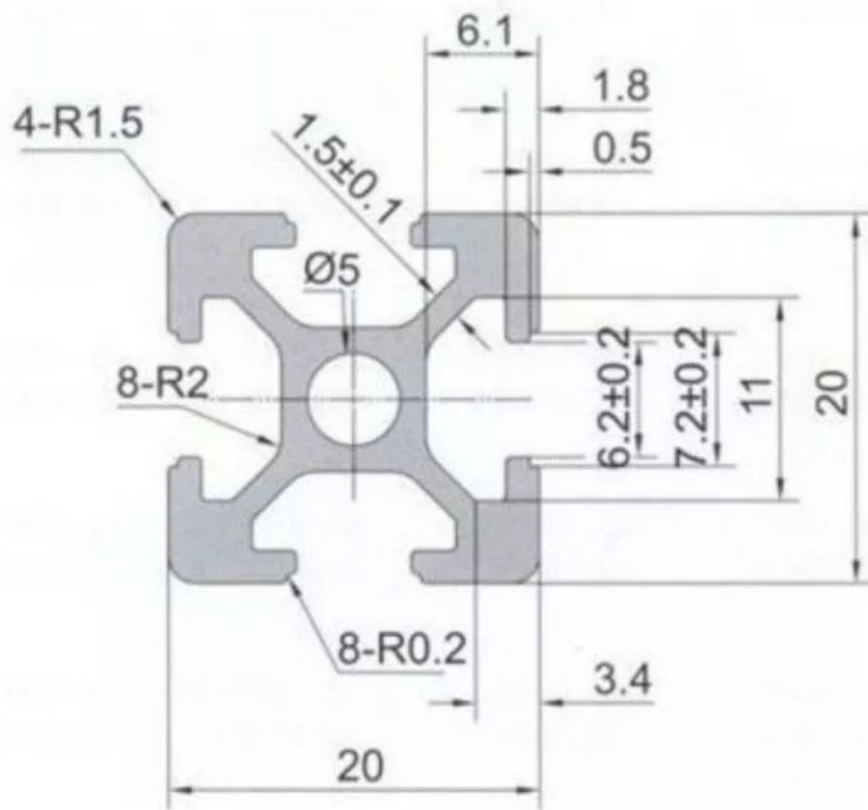
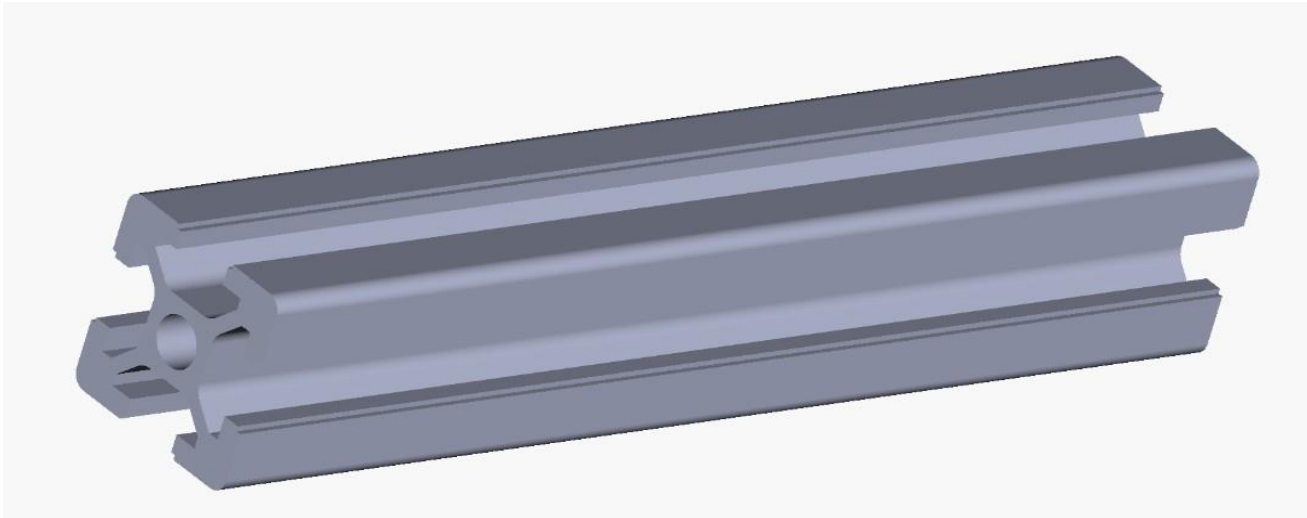
Toàn bộ quá trình thi công hệ thống phải đảm bảo tất cả những yêu cầu về thiết kế mà nhóm đã đặt ra ban đầu

4.2 Thi công mô hình

STT	Linh kiện chính	Số lượng
1	Mạch Arduino nano CH340	1
2	Module đọc thẻ RC522	1
3	Cảm biến tiệm cận LJ8A3-2-Z/BX NPN 2mm	1
4	Động cơ bước SUMTOR 42HS3413A4-0.28Nm-1.3a-42*34mm	1
5	Động cơ Servo SG90S	4
6	Webcam	3
7	Loa	1
8	Nguồn adaptor 12V 5A	1
Các linh kiện khác	<ul style="list-style-type: none"> - Nút nhất - Ic ổn áp 7805 - Transistor công suất TIP42C - Driver động cơ bước a4988 - Modul hạ áp DC LM2596 - Modul relay SRD-12VDC-SL-C - Đèn led - Tụ điện - Điện trở - Jack nguồn, dây điện,... 	

4.2.1 Thi công phần khung khung

Với phần khung mô hình, ta sử dụng vật liệu chính là các thanh nhôm V-Slot 20mmx20mm để lắp ghép định hình thành khung sườn

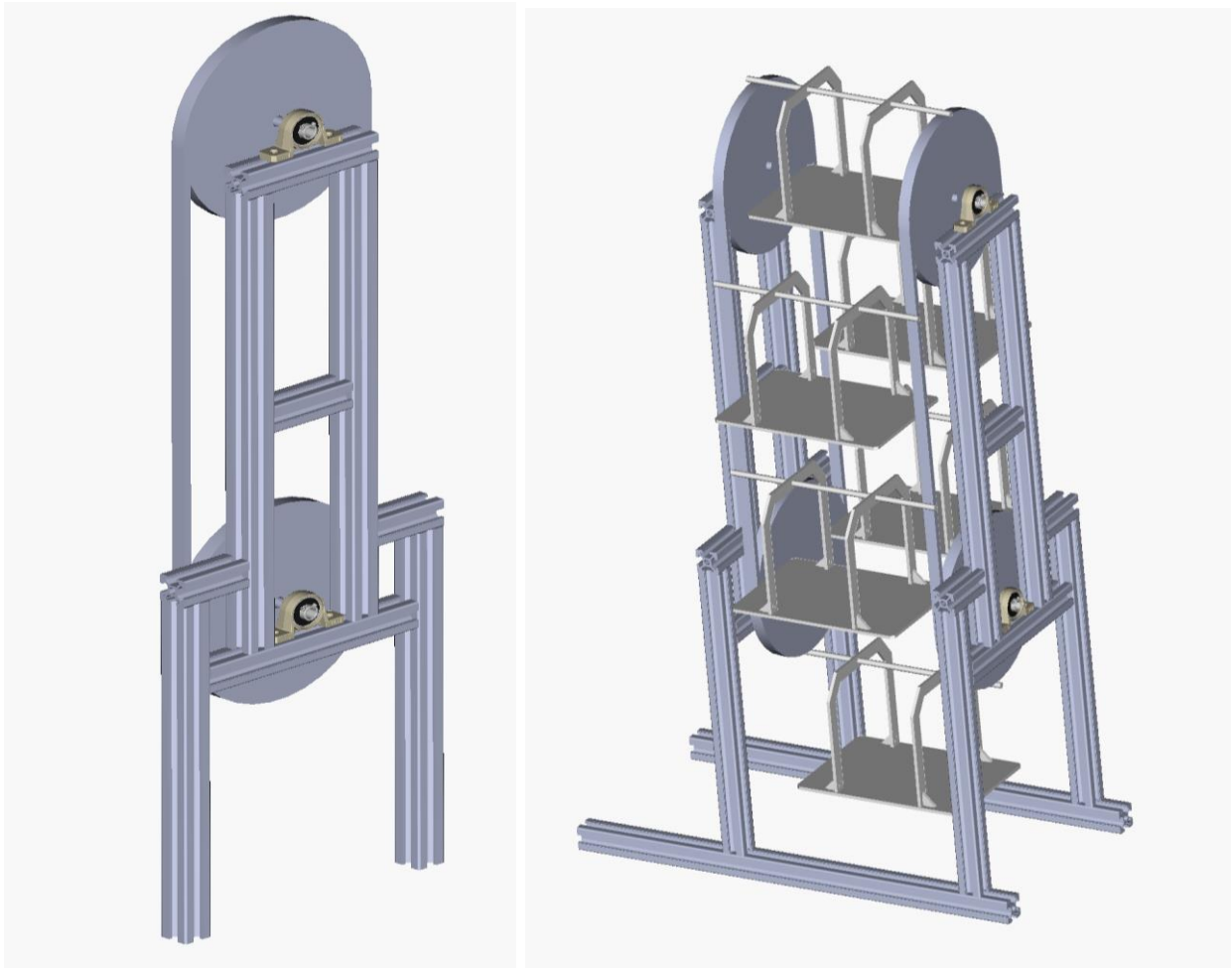


Hình 30: thông số kỹ thuật thanh nhôm V-Slot 20mmx20mm

Để có được mô hình như dự kiến chúng ta cần có 20 đoạn dài ngắn khác nhau:

- 400mm 2 đoạn
- 338mm 4 đoạn
- 250mm 4 đoạn
- 200mm 2 đoạn
- 160mm 2 đoạn
- 100mm 4 đoạn
- 60mm 2 đoạn
- 50mm 4 đoạn

Ngoài ra chúng em còn sử dụng thêm 2 trục inox hình trụ dài 190mm làm 2 trục quay trính cho 4 bánh xích



Hình 31: mô phỏng phân khung mô hình thực hiện

Trên đây là bản mô phỏng 3D của phần khung mô hình

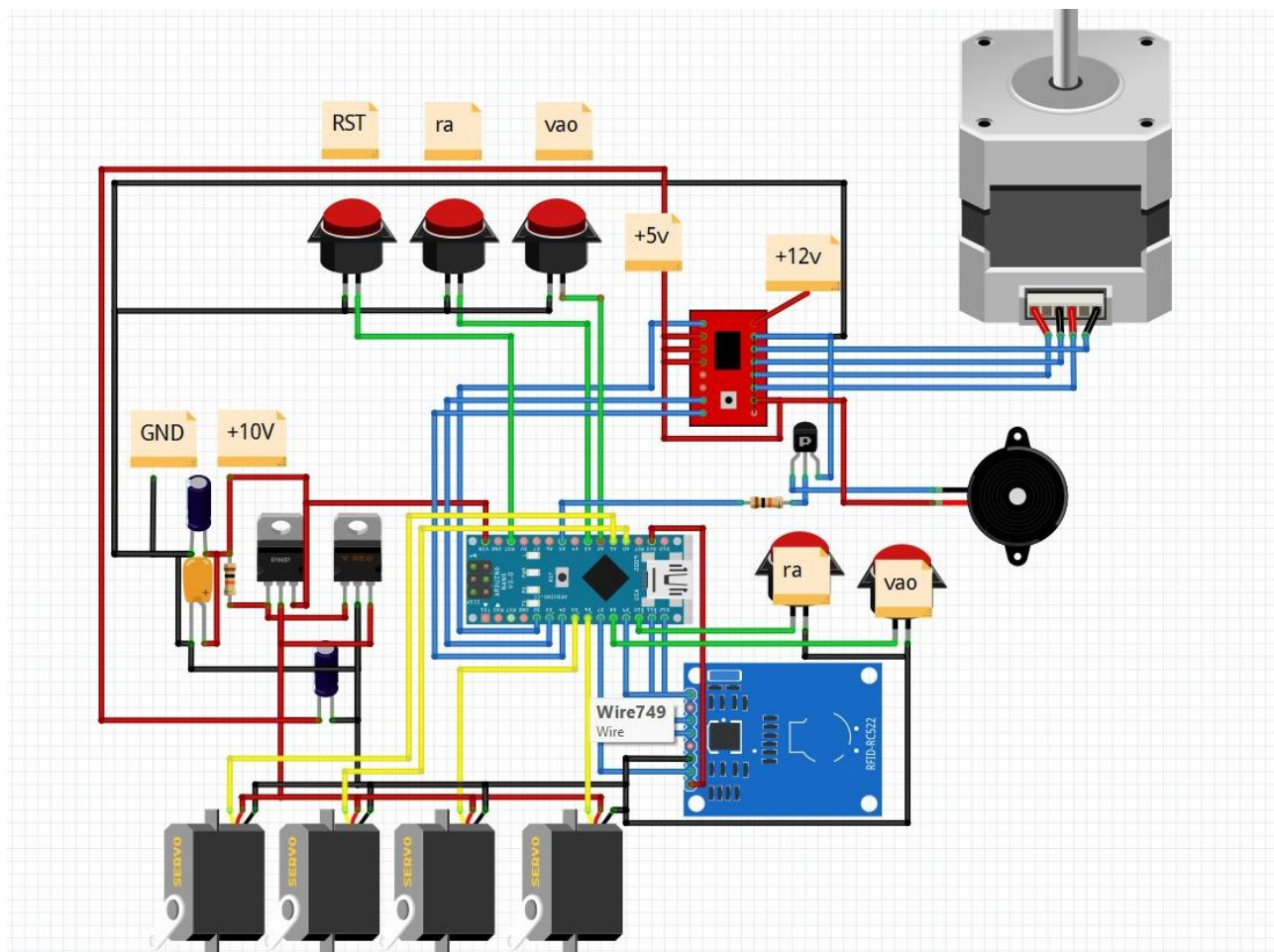
- Cách tầng gửi xe được ghép từ các tấm nhựa mica sau đó gắn vào trục xích chuyển động có thể di chuyển được
- Trên dây xích có gài các chốt để giữ các tầng để xe, khi trục xích di chuyển các tầng trên đó cũng luân phiên quay theo nó

4.2.2 Thi công phần mạch điện

Dựa vào kế hoạch thực hiện chúng em thực hiện mạch để nối các linh kiện sau:

- Mạch Arduino nano CH340
- Module đọc thẻ RC522

- Cảm biến tiệm cận
- Động cơ bước
- Động cơ Servo
- Nút nhất
- Ic ổn áp 7805
- Transistor công suất TIP42C
- Driver động cơ bước a4988
- Modul hạ áp DC LM2596
- Modul relay SRD-12VDC-SL-C
- Đèn led
- Tụ điện
- Điện trở
- Jack nguồn, dây điện,...



Hình 32: Sơ đồ nguyên lý đi dây của mạch

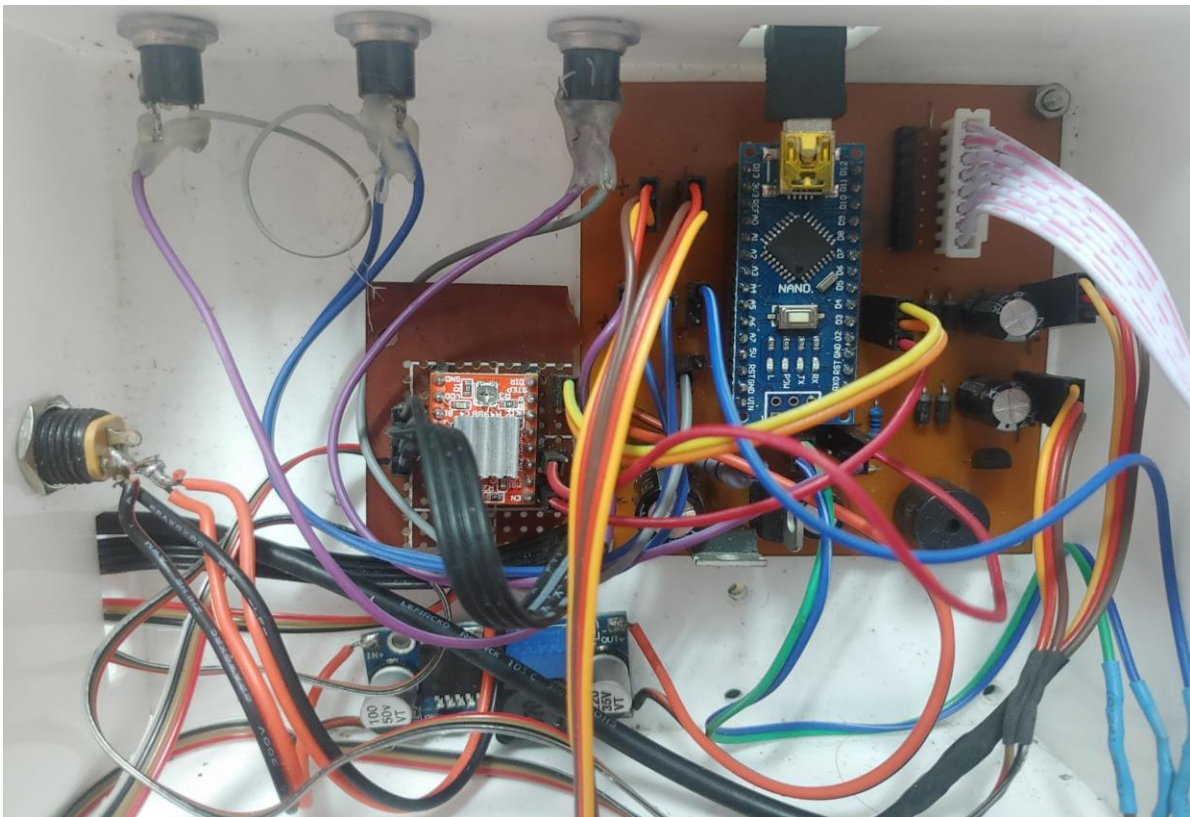
Như vậy chức năng của các chân arduino nano được sử dụng như sau:

- A2, D8 nhận tín hiệu nút “VAO”
- A3, D10 nhận tín hiệu nút “RA”
- Reset nhận tín hiệu nút “RESET”
- A5 suất tín hiệu âm thanh
- D2, D3, D4 suất tín hiệu điều khiển động cơ bước
- D5, D6, A0, A1 suất tín hiệu điều khiển động cơ servo
- D7 reset mạch RC522
- Chân VCC của mô-đun RFID RC522 được kết nối với chân 3.3V của

Arduino Uno.

- Chân RST của mô-đun RFID RC522 được kết nối với chân digital 9 trên Arduino Uno.
- Chân SS (Slave Select) của mô-đun RFID RC522 được kết nối với chân digital 10 trên Arduino Uno.
- Chân MOSI của mô-đun RFID RC522 được kết nối với chân digital 11 trên Arduino Uno.
- Chân MISO của mô-đun RFID RC522 được kết nối với chân digital 12 trên Arduino Uno.
- Chân SCK của mô-đun RFID RC522 được kết nối với chân digital 13 trên Arduino Uno.

Sau khi thực hiện thi công phân mạch điện chúng em thu được mạch điện như sau:



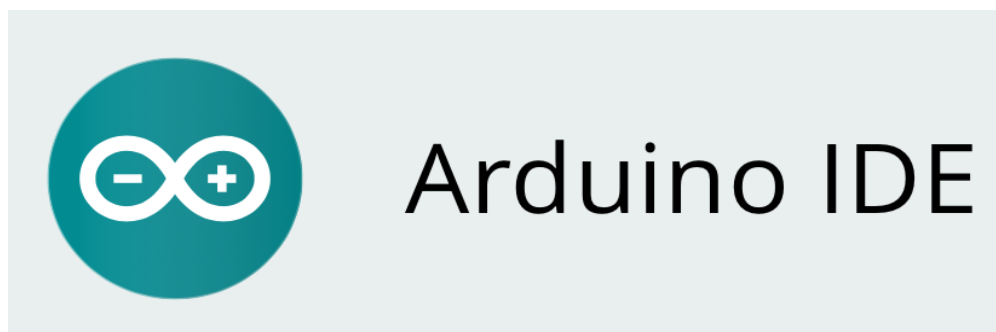
Hình 33: mạch điện hoàn thành

4.3 Lập trình hệ thống

4.3.1 Lập trình cho ARDUINO

Giới thiệu phần mềm:

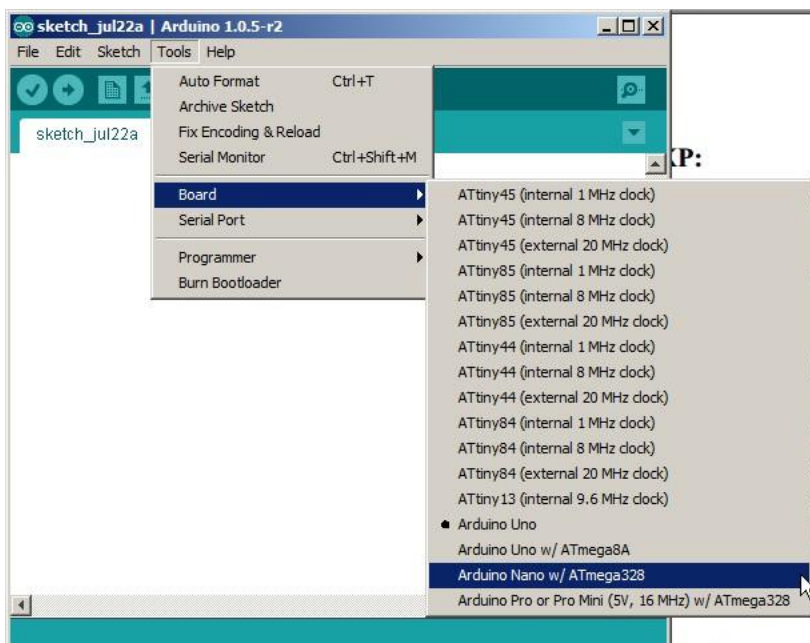
Để viết chương trình lập trình và điều khiển cho arduino nano chúng em sử dụng phần mềm arduino IDE, đây là phần mềm cơ bản để lập trình cho arduino:



Hình 34: giao diện arduino IDE

Chúng ta sẽ làm việc với Arduino IDE theo các bước sau:

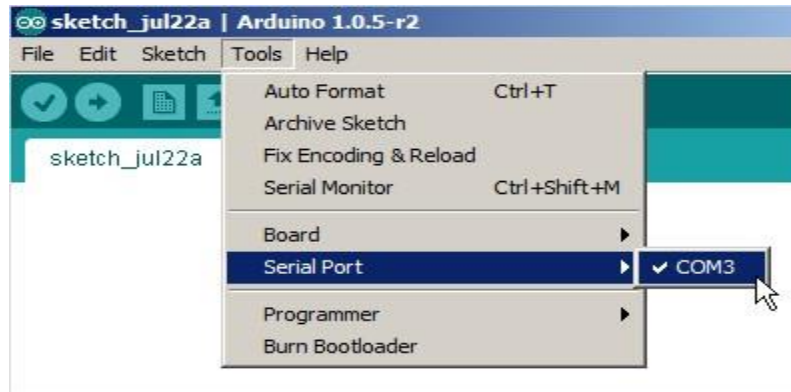
Bước 1: Sau khi vào giao diện chính của IDE, ta cần thiết lập loại board và cổng Serial mới cho phù hợp như hình bên dưới.



Hình 35: thiết lập loại board và cổng Serial

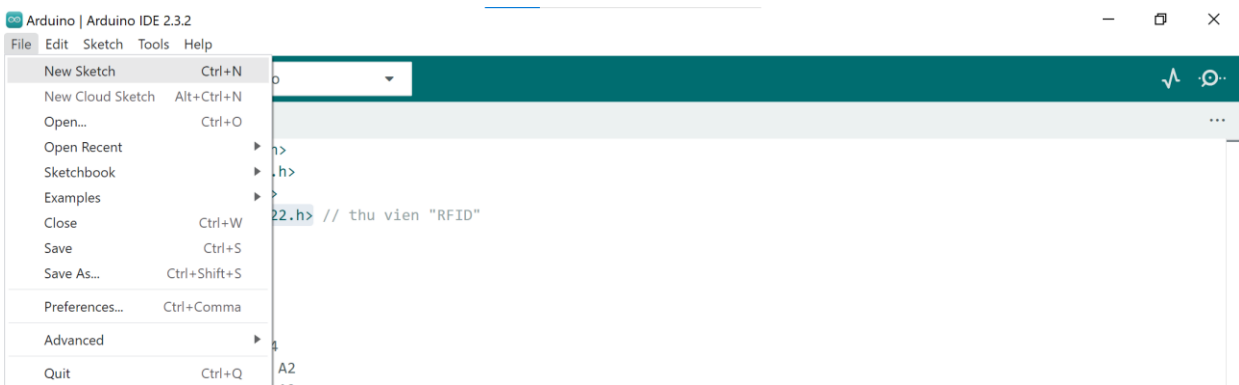
Lưu ý, cổng COM trong hình dưới đây là chỉ là hình minh họa trong máy tính của

chúng em, tùy thuộc vào mỗi máy và cổng kết nối khác nhau chung ta sẽ có các cổng COM cần thiết lập khác nhau:



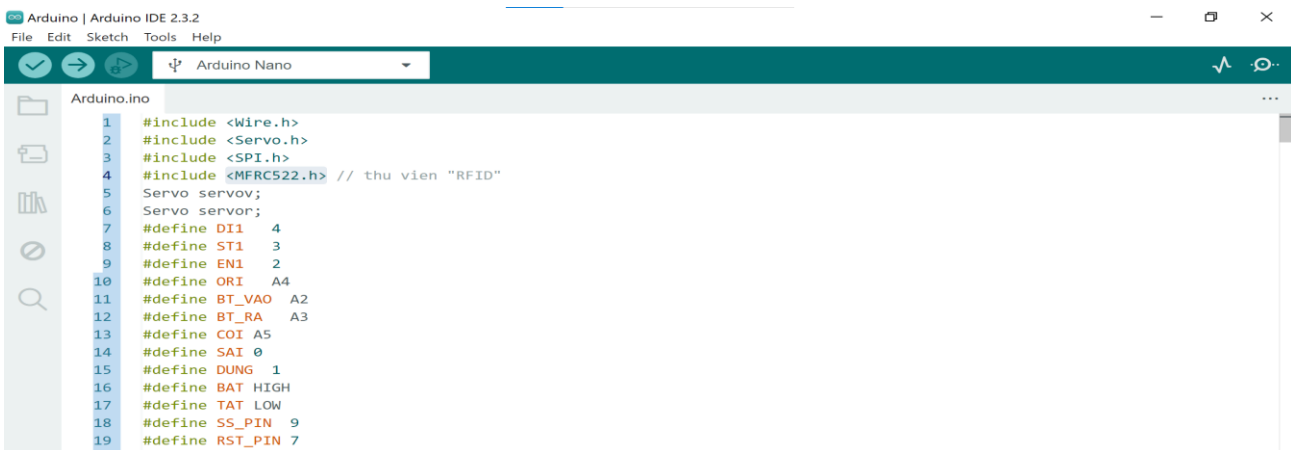
Hình 36: thiết lập COM3

Bước 2: vào mục file chọn new tạo file mới viết để lập trình code cho chương trình



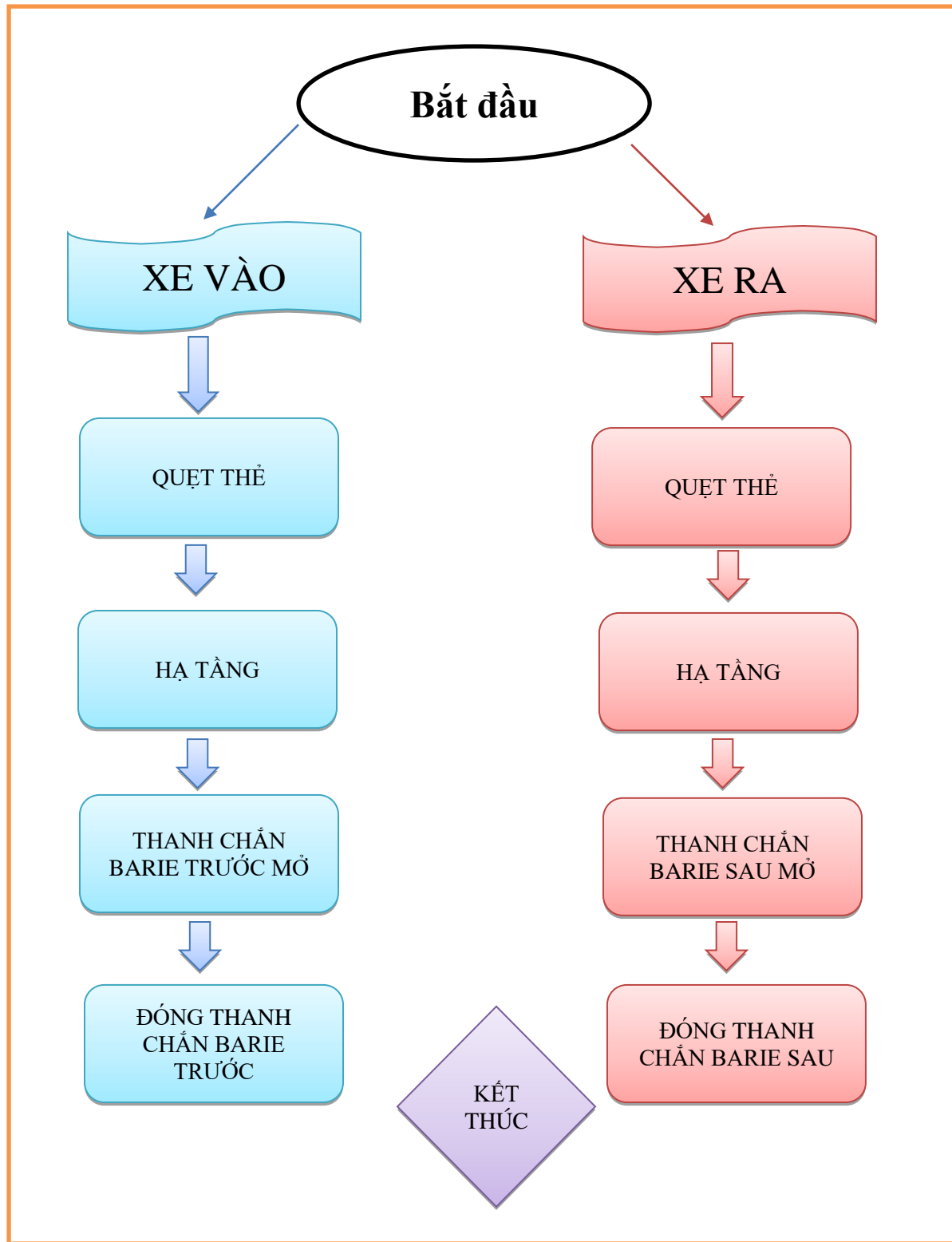
Hình 37: tạo file mới để lập trình

Bước 3: khai báo thư viện, khai báo biến và viết lệnh code cho chương trình



Hình 38: viết lệnh code

Lưu đồ giải thuật cho Arduino



Hình 39: Lưu đồ giải thuật cho Arduino

Giải thích lưu đồ:

Bắt đầu: nếu xe vào ta nhấn nút “vào” hệ thống sẽ chuyển sang chế độ xe vào, nếu xe ra ta nhấn nút “ra” hệ thống sẽ chuyển sang chế độ ra vào

- Xe vào: ở chế độ này hệ thống sẽ làm việc tuần tự theo các bước.
 - Hạ tầng: Sau khi quét thẻ xe hệ thống sẽ dựa trên mã số thẻ quét được sau đó lựa chọn tầng ứng với thẻ đó, điều khiển motor quay hạ tầng theo yêu cầu.
 - Thanh chắn barie trước mở: sau khi hạ tầng thành công, hệ thống điều khiển thanh chắn barie trước mở ra để cho xe vào tầng
 - Đóng thanh chắn barie trước: xe vô tầng thành công, ta quét thẻ lần nữa để xác nhận lấy mẫu ảnh xử lý mã biển số, lấy mẫu thành công thanh chắn barie trước đóng lại

- Xe ra: ở chế độ này hệ thống sẽ làm việc tuần tự theo các bước.
 - Hạ tầng: Sau khi quét thẻ xe hệ thống sẽ dựa trên mã số thẻ quét được sau đó lựa chọn tầng ứng với thẻ đó, điều khiển motor quay hạ tầng theo yêu cầu.
 - Thanh chắn barie trước mở: sau khi hạ tầng thành công, hệ thống điều khiển thanh chắn barie sau mở ra để cho xe vào tầng
 - Đóng thanh chắn barie sau: xe ra khỏi tầng thành công, thanh chắn barie sau đóng lại

Chương trình điều khiển trên arduino

B1: Khai báo thư viện

```
1  #include <Servo.h>
2  #include <MFRC522.h> // thu vien "RFID"
```

- Servo.h: Thư viện điều khiển servo.
- MFRC522.h: Thư viện giao tiếp với module RFID MFRC522

B2: đặt tên biến cho 4 động cơ server

```
4  Servo servov;
5  Servo servor;
6  Servo servovm;
7  Servo servorm;
```

- Servov: barie ô tô vào
- Servor: barie ô tô ra
- Servovm: barie xe máy vào
- Servorm: barie xe máy ra

B3: khai báo tên chân cho aduino

- DI1, ST1, EN1: Các chân điều khiển động cơ bước.
- ORI: Chân đọc cảm biến từ.
- BT_VAO, BT_RA: Các chân nút nhấn ô tô.
- BT_VAO_m, BT_RA_m: Các chân nút nhấn xe máy.
- COI: Chân điều khiển còi .
- SS_PIN, RST_PIN: Các chân giao tiếp với RFID.

```
9  #define DI1 4
10 #define ST1 3
11 #define EN1 2
12 //
13 #define ORI A4
14 //
15 #define COI A5
16 #define SS_PIN 9
17 #define RST_PIN 7
18 //
19 #define BT_VAO A2
20 #define BT_RA A3
21 //
22 #define BT_VAO_m 8
23 #define BT_RA_m 10
```

B4: khai báo trạng thái các biến

```
int find_rfid = 0;
int RFID = 0;
MFRC522 mfrc522(SS_PIN, RST_PIN);
unsigned long uidDec, uidDecTemp; // hiển thị số UID dạng thập phân
byte bCounter, readBit;
int mode = 0, dk_vao = 0, dk_ra = 0, vr = 0, cp = 0, so_xe = 0, tt_vao = 0, tt_ra = 0, dc_ra = 0;
int dk_vao_m = 0, dk_ra_m = 0, vr_m = 0, cp_m = 0, so_xe_m = 0, tt_vao_m = 0, tt_ra_m = 0, dc_ra_m = 0;
int i = 0, doc_bien = 0, tt = 0, tt2 = 0, tt3 = 0, tt4 = 0, lan = 0, read_id = 0, the = 0, the_m = 0;
String kqv = "", kqr = "";
String kqv1 = "", kqr1 = "";
String dataa = "";
char *token1, *token2;
char tama[40];
String mang_bien[10];
String mang_bien_m[10];
int mang_bai[10];
unsigned long int time_gui = 0;
int d = 1000;
int bai = 1, vong = 1, kcach = 0;
int dk1 = 0, dk_bai = 0;
```

- mode, dk_vao, dk_ra, vr, cp, so_xe, tt_vao, tt_ra, dc_ra, dk_vao_m, dk_ra_m, vr_m, cp_m, so_xe_m, tt_vao_m, tt_ra_m, dc_ra_m: Các biến trạng thái và điều khiển.
- i, doc_bien, tt, tt2, lan, read_id, the: Các biến tạm thời khác.
- time_gui: Biến đếm thời gian.
- d: Biến khoảng thời gian (1000 ms).
- bai, vong, kcach: Các biến điều khiển bãi đỗ xe.
- dk1, dk_bai: Các biến điều khiển khác.
- kqv, kqr: Chuỗi kết quả.
- dataa: Chuỗi dữ liệu.
- token1, token2: Các biến tách chuỗi.
- tama: Chuỗi tạm thời.
- mang_bien, mang_bien_m: Mảng chuỗi biển số xe ô tô và xe máy.
- mang_bao: Mảng cảnh báo.

B5: cấu hình các chân

```
void setup() {  
  Serial.begin(9600);  
  SPI.begin();  
  pinMode(COI, OUTPUT);  
  pinMode(DI1, OUTPUT);  
  pinMode(EN1, OUTPUT);  
  pinMode(ST1, OUTPUT);  
  pinMode(ORI, INPUT_PULLUP);  
  pinMode(BT_VAO, INPUT_PULLUP);  
  pinMode(BT_RA, INPUT_PULLUP);  
  pinMode(BT_VAO_m, INPUT_PULLUP);  
  pinMode(BT_RA_m, INPUT_PULLUP);  
  digitalWrite(ST1, LOW);  
  digitalWrite(DI1, LOW);  
  digitalWrite(COI, LOW);  
  digitalWrite(EN1, LOW);  
}
```

- `Serial.begin(9600)`: Khởi tạo giao tiếp Serial với tốc độ truyền 9600 bps để có thể in ra thông tin gỡ lỗi hoặc dữ liệu.
- `SPI.begin()`: Khởi tạo giao tiếp SPI, cần thiết để giao tiếp với module RFID.
- `pinMode(COI, OUTPUT)`: Cấu hình chân COI làm đầu ra để điều khiển loa.
- `pinMode(DI1, OUTPUT)`: Cấu hình chân DI1 làm đầu ra để điều khiển động cơ bước.
- `pinMode(EN1, OUTPUT)`: Cấu hình chân EN1 làm đầu ra để điều khiển động cơ bước.
- `pinMode(ST1, OUTPUT)`: Cấu hình chân ST1 làm đầu ra để điều khiển động cơ bước.
- `pinMode(ORI, INPUT_PULLUP)`: Cấu hình chân ORI làm đầu đọc tín hiệu từ cảm biến từ.
- `pinMode(BT_VAO, INPUT_PULLUP)`: Cấu hình chân BT_VAO để đọc tín hiệu từ nút nhấn vào.

- `pinMode(BT_RA, INPUT_PULLUP)`: Cấu hình chân BT_RA để đọc tín hiệu từ nút nhấn ra.
- `digitalWrite(ST1, LOW)`: Đặt chân ST1 ở mức thấp.
- `digitalWrite(DI1, LOW)`: Đặt chân DI1 ở mức thấp.
- `digitalWrite(COI, LOW)`: Đặt chân COI ở mức thấp để tắt loa.
- `digitalWrite(EN1, LOW)`: Đặt chân EN1 ở mức thấp.

B6: gán trạng thái điều khiển động cơ

```
while (digitalRead(ORI) == 1) {
    digitalWrite(ST1, HIGH);
    delayMicroseconds(d);
    digitalWrite(ST1, LOW);
    delayMicroseconds(d);
}
servov.attach(6);
servor.attach(5);
servov.write(0);
servor.write(90);
delay(3000);
//
servovm.attach(A0);
servorm.attach(A1);
servovm.write(0);
servorm.write(90);
delay(3000);
Serial.begin(9600);
```

- `while (digitalRead(ORI) == 1)`: Vòng lặp kiểm tra trạng thái của cảm biến từ. Nếu cảm biến từ chưa kích hoạt (đọc giá trị 1), động cơ bước sẽ được điều khiển để xoay một bước.
- `digitalWrite(ST1, HIGH)` và `digitalWrite(ST1, LOW)`: Điều khiển động cơ bước quay bằng cách thay đổi trạng thái của chân ST1.
- `delayMicroseconds(d)`: Tạo độ trễ giữa các bước để động cơ bước hoạt động.
- `servov.attach(6)`: Gán servo vào chân số 6.

- `servo.attach(5)`: Gán servo ra chân số 5.
- `servo.write(0)`: Đặt servo vào ở góc 0 độ (có thể là vị trí đóng công vào).
- `servo.write(90)`: Đặt servo ra ở góc 90 độ (có thể là vị trí mở công ra).
- `delay(3000)`: Đợi 3 giây để các servo di chuyển tới vị trí

B7: chương trình chế độ auto xe vào

```

95 void Auto() {
96     if(digitalRead(BT_VAO) == 0) vr = 1;
97     if(digitalRead(BT_RA) == 0) vr = 2;
98     if (vr == 1) {
99         if (so_xe < 6) {
100             while (read_id == 0) {
101                 doc_rfid();
102             }
103             if (mang_bai[the] == 0) {
104                 read_id = 0;
105                 quay();
106                 mov();
107                 delay(500);
108                 while (read_id == 0) {
109                     doc_rfid();
110                 }
111                 cp = 1;
112                 guidulieu2();
113                 cp = 0;
114                 while (doc_bien == 0) {
115                     serialEvent();
116                     guidulieu();
117                 }
118                 mang_bien[the] = kqv;
119                 doc_bien = 0;
120                 vr = 0;
121                 read_id = 0;
122                 so_xe++;
123                 mang_bai[the] = 1;
124                 guidulieu2();
125                 delay(200);
126                 dongv();
127             }
128         }
129     }
130     else{
131         read_id = 0;
132         vr = 0;
133         doc_bien = 0;

```


- `digitalRead(BT_VAO) == 0`: Nếu nút vào được nhấn, vr được đặt là 1.
- `digitalRead(BT_RA) == 0`: Nếu nút ra được nhấn, vr được đặt là 2.
- `if (so_xe < 6)`: Kiểm tra nếu số lượng xe trong bãi ít hơn 6.
- `while (read_id == 0)`: Vòng lặp đọc RFID để lấy ID xe.
- `if (mang_bao[the] == 0)`: Kiểm tra nếu RFID không tồn tại trong danh sách các xe đã vào bãi.

- `quay()`: Gọi hàm quay (có thể để điều chỉnh động cơ hoặc servo).
- `mov()`: Gọi hàm mở barie vào.
- `while (read_id == 0)`: Lặp lại quá trình đọc RFID.
- `cp = 1; guidulieu2(); cp = 0;`: Gửi dữ liệu trạng thái.
- `while (doc_bien == 0)`: Lấy biên số xe từ Serial.
- `mang_bien[the] = kqv;`: Lưu biên số xe vào mảng.
- `so_xe++;`: Tăng số lượng xe trong bãi.
- `mang_bao[the] = 1;`: Đánh dấu RFID đã vào bãi.
- `guidulieu2(); delay(200); dongv();`: Gửi dữ liệu và đóng cổng vào.

B8: chương trình chế độ auto xe ra

```
136     if (vr == 2) {
137         while (read_id == 0) {
138             doc_rfid();
139         }
140         quay();
141         delay(2000);
142         cp = 1;
143         guidulieu2();
144         cp = 0;
145         while (doc_bien == 0) {
146             serialEvent();
147             guidulieu();
148         }
149         doc_bien = 0;
150         vr = 0;
151         if (kqr == mang_bien[the]) {
152             mor();
153             if (so_xe > 0) so_xe--;
154             dc_ra = 1;
155             guidulieu2();
156             dc_ra = 0;
157             delay(7000);
158             dongr();
159             mang_bai[the] = 0;
160         }
161     }
162     else {
163         dc_ra = 2;
164         guidulieu2();
165         dc_ra = 0;
166     }
167     read_id = 0;
```

- `while (read_id == 0):` Vòng lặp đọc RFID để lấy ID xe.
- `quay(); và delay(2000);`: Gọi hàm quay và đợi 2 giây.
- `cp = 1; guidulieu2(); cp = 0;`: Gửi dữ liệu trạng thái.
- `while (doc_bien == 0) { serialEvent(); guidulieu();}`: lấy trạng thái chương trình đọc biển.
- `if (kqr == mang_bien[the]):` Kiểm tra biển số xe có khớp với dữ liệu lưu trữ không.
 - `mor();`: Gọi hàm mở cổng ra.
 - `if (so_xe > 0) so_xe--;`: Giảm số lượng xe trong bãi.
 - `dc_ra = 1; guidulieu2(); dc_ra = 0;`: Gửi dữ liệu trạng thái.
 - `delay(7000); dongr();`: đợi 7 giây cửa sau đóng lại

B9: thiết lập chương trình chọn và quay tầng cho động cơ bước

```
void quay() {  
    kcach = the - bai;  
  
    if (abs(kcach) <= 3) {  
        vong = abs(kcach);  
        if (kcach < 0 )  
            digitalWrite(DI1, LOW);  
        else  
            digitalWrite(DI1, HIGH);  
    }  
    else {  
        vong = 6 - abs(kcach);  
        if (kcach > 0 )  
            digitalWrite(DI1, LOW);  
        else  
            digitalWrite(DI1, HIGH);  
    }  
    digitalWrite(EN1, LOW);  
    for (int i = 0; i <= vong * 2100; i++) {  
        digitalWrite(ST1, HIGH);  
        delayMicroseconds(d);  
        digitalWrite(ST1, LOW);  
        delayMicroseconds(d);  
    }  
    bai = the;//
```

Cách tính tầng như sau

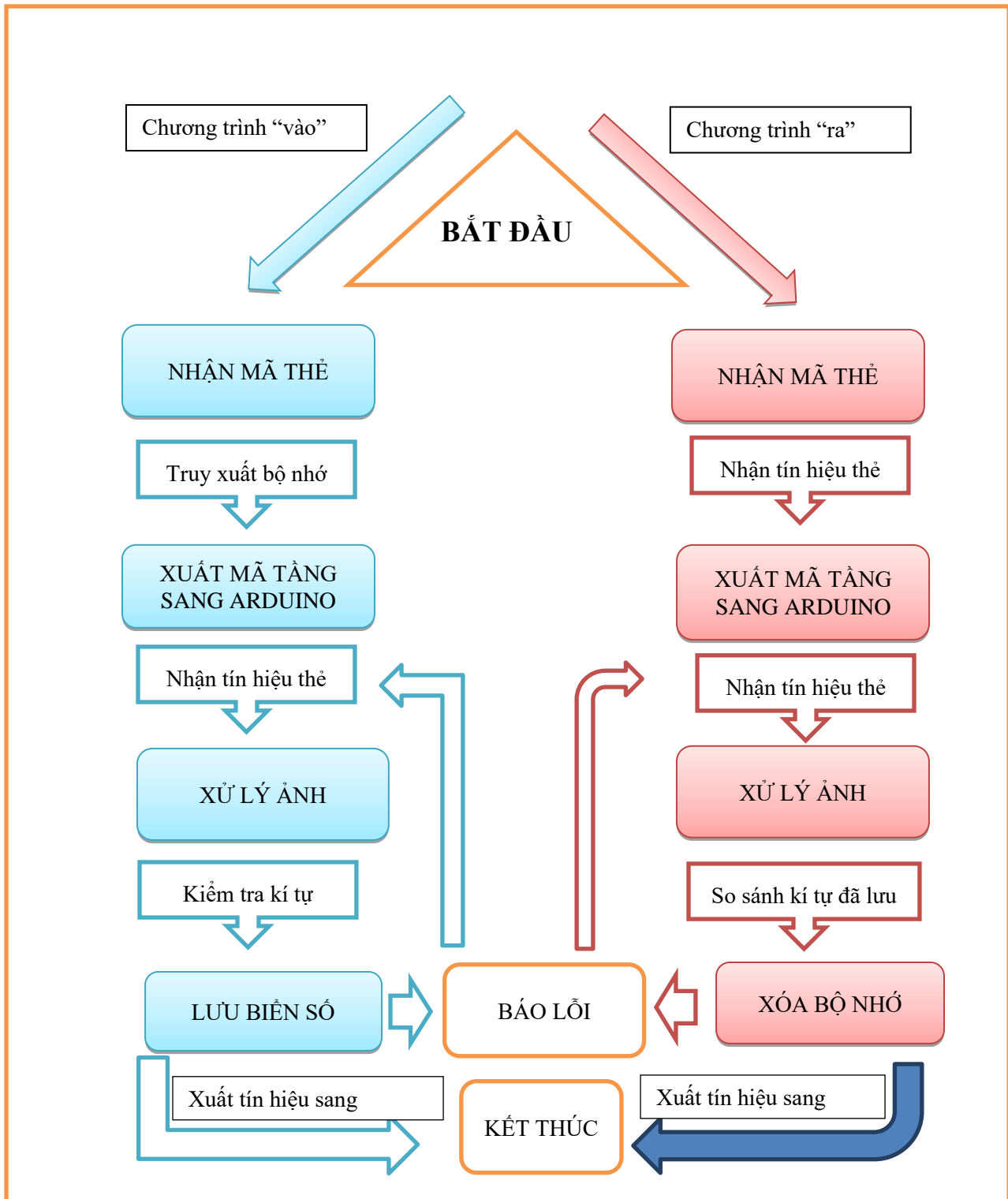
- $kcach = the - bai$: ta lấy số thẻ trừ đi số của bãi xe đang hạ
- $(abs(kcach) \leq 3)$: sau đó lấy trị tuyệt đối là ra khoảng cách

Cách lựa chọn hướng:

- Nếu giá trị tuyệt đối của khoảng cách nhỏ hơn hoặc bằng 3, động cơ sẽ quay một số bước bằng khoảng cách.
- Nếu khoảng cách là số âm, nó sẽ đặt chân điều hướng (DI1) xuống mức LOW; ngược lại, nếu khoảng cách là số dương, nó sẽ đặt chân điều hướng lên mức HIGH.

4.3.2 Lập trình xử lý hình ảnh từ máy tính

Lưu đồ thuật toán:



Hình 40: Lưu đồ thuật toán xử lý ảnh bằng python IDLE

Giải thích thuật toán:

Bước 1: khi bắt đầu tùy thuộc vào tín hiệu nhận về từ arduino được điều khiển bằng nút bấm mà máy tính sẽ chạy theo 1 trong 2 chương trình “VÀO” “RA”

Bước 2: Thẻ từ được quét vào thiết bị đọc thẻ, thiết bị sẽ có được chuỗi dữ liệu gồm 7 mã ACSII để đưa lên máy tính.

- Thiết bị gửi chuỗi mã ACSII và máy tính nhận dữ liệu.
- Từ chuỗi mã ACSII, rồi tiến hành truy cập cơ sở dữ liệu.

Bước 3: Trong cơ sở dữ liệu, máy tính tiến hành kiểm tra xem mã thẻ được lưu ở tầng nào? (có nghĩa là chiếc thẻ từ đã được lập trình lưu cố định ứng với 1 tầng, như vậy có 6 thẻ ứng với 6 tầng khác nhau của mô hình).

- Sau khi kiểm tra được tầng nào, máy tính sẽ gửi mã tầng sang arduino để di chuyển tới tầng đó

Bước 4.1: Sau khi chuyển tầng thành công, ta quét thẻ thêm lần nữa, quá trình chụp ảnh và nhận dạng biển số xe bắt đầu.

- Nếu nhận dạng ảnh sai (không thể chụp được hình, nhận dạng không đủ ký tự). Máy tính sẽ nhận dạng lại một lần nữa khi ta quét thẻ
- Nếu nhận dạng đúng, hệ thống lưu dữ liệu biển số thu được và suất tín hiệu hoàn thành cho arduino để đóng thanh chắn barie trước, chương trình kết thúc

Bước 4.2: Nếu đang chạy chương trình “RA” thì hệ thống sẽ truy suất bộ nhớ và so sánh với kí tự biển số thu được thông qua quá trình xử lý ảnh

- Nếu nhận dạng sai (không thể chụp được hình, nhận dạng không đủ ký tự) hệ thống sẽ báo lỗi, máy tính sẽ nhận dạng lại một lần nữa khi ta quét thẻ
- Nếu nhận dạng đúng, hệ thống xóa dữ liệu biển số đã lưu trước đó và suất tín hiệu hoàn thành cho arduino để mở thanh chắn barie sau chương trình kết thúc

Chương trình chính trên PYTHON:

```
1 from PyQt5 import QtWidgets,uic
2 from PyQt5.QtWidgets import QApplication, QWidget,
3 QPushButton, QLabel, QVBoxLayout
4 from PyQt5 import QtCore
5 from PyQt5.QtGui import QIcon, QPixmap, QFont
6 from PyQt5.QtCore import QTimer, QTime, Qt
7 from PyQt5.QtGui import *
8 from PyQt5.QtWidgets import *
9 from PyQt5.QtCore import *
10 import time
11 import cv2
12 from time import sleep
13 import numpy as np
14 import array as arr
15 import math
16 import sys
17 import serial
18 import os
19 from imutils.video import VideoStream
20 from imutils import face_utils
21 from PIL import Image
22 import imutils
23 import pytesseract
24 import pygame
25 pygame.mixer.init()
26 import time
27 from datetime import datetime
28 #-----
29 ser = serial.Serial('COM8',baudrate = 9600, timeout = 1)
30 print(ser.name)
31 vs = VideoStream(src=1).start()
32 vs_m = VideoStream(src=0).start()
33 mode = 0; dk_vao = 0; dk_ra = 0; so_xe = 0; kq = 0; cp = 0
34 mode_ht = 0; tt_vao = 0; tt_ra = 0; vr = 0; dc_ra = 0
35 dk_vao_m = 0; dk_ra_m = 0; so_xe_m = 0; kq_m = 0; cp_m = 0
36 tt_vao_m = 0; tt_ra_m = 0; vr_m = 0; dc_ra_m = 0
37 time2 = 0; time_vao = 0; time_ra = 0; tien = 0; tgian_vao = ""
38 ; tgian_ra = ""; indextg = 0
39 img = None
40 img_m = None
```



```

41 mang_tg_vao = [""]*10; mang_time_vao = [0.0]*10; mang_bien = [""]*10;
42 data = ''; vr = 0; vr_m = 0
43 bao1 = 0; bao2 = 0; bao3 = 0; bao4 = 0; bao5 = 0; bao6 = 0
44 def Time():
45     global mode, dk_vao, dk_ra, so_xe, kq, cp, mode_ht, vr, tt_vao,
46     tt_ra, dc_ra
47     global vs, img
48     global time2, time_vao, time_ra, tien, indextg, mang_tg_vao,
49     mang_time_vao, mang_bien, data, vr
50     global bao1, bao2, bao3, bao4, bao5, bao6
51     global dk_vao_m, dk_ra_m, so_xe_m, kq_m, cp_m, vr_m, tt_vao_m,
52     tt_ra_m, dc_ra_m
53     global vs_m, img_m
54     #-----thời gian-----
55     today = datetime.today()
56     day = today.strftime("%d/%m/%Y")
57     now = datetime.now()
58     current_time = now.strftime("%H:%M:%S")
59     time_present = current_time + " - " + day
60     call.lb_tgian.setText(time_present)
61     #-----nhận dữ liệu-----
62     if (ser.in_waiting > 0):
63         s = ser.readline()           #Doc vao data
64         dataa = s.decode()           # decode s
65         dataa = dataa.rstrip()       # cut "\r\n" at last of string
66     try:
67         mang = dataa.split("|")
68         mode_ht = int(mang[0])
69         tt_vao = int(mang[1])
70         tt_ra = int(mang[2])
71         cp = int(mang[3])
72         vr = int(mang[4])
73         dc_ra = int(mang[5])
74         so_xe = int(mang[6])
75         tt_vao_m = int(mang[7])
76         tt_ra_m = int(mang[8])
77         cp_m = int(mang[9])
78         vr_m = int(mang[10])
79         dc_ra_m = int(mang[11])
80         so_xe_m = int(mang[12])

```

```

81         bao1 = int(mang[13])
82         bao2 = int(mang[14])
83         bao3 = int(mang[15])
84         bao4 = int(mang[16])
85         bao5 = int(mang[17])
86         bao6 = int(mang[18])
87     except:
88         a = 0
89     call.lb_so_xe.setText(str(so_xe))
90     call.lb_so_xe_m.setText(str(so_xe_m))
91     if mode_ht == 0:
92         call.lb_mode.setStyleSheet("background-color: rgb(0, 85, 255)")
93     if mode_ht == 1:
94         call.lb_mode.setStyleSheet("background-color: rgb(255, 170, 0)")
95     if tt_vao == 0:
96         call.BT_VAO.setStyleSheet("background-color: rgb(255, 0, 0)")
97     else:
98         call.BT_VAO.setStyleSheet("background-color: rgb(0, 255, 0)")
99     if tt_ra == 0:
100        call.BT_RA.setStyleSheet("background-color: rgb(255, 0, 0)")
101    else:
102        call.BT_RA.setStyleSheet("background-color: rgb(0, 255, 0)")
103    if so_xe < 6:
104        call.lb_so_xe.setStyleSheet("background-color: rgb(85, 255, 0)")
105    else:
106        call.lb_so_xe.setStyleSheet("background-color: rgb(255, 0, 0)")
107    if bao1 == 1:
108        call.lb1.setStyleSheet("background-color: rgb(255, 0, 0)")
109    else:
110        call.lb1.setStyleSheet("background-color: rgb(0, 255, 0)")
111    if bao2 == 1:
112        call.lb2.setStyleSheet("background-color: rgb(255, 0, 0)")
113    else:
114        call.lb2.setStyleSheet("background-color: rgb(0, 255, 0)")
115    if bao3 == 1:
116        call.lb3.setStyleSheet("background-color: rgb(255, 0, 0)")
117    else:
118        call.lb3.setStyleSheet("background-color: rgb(0, 255, 0)")
119    if bao4 == 1:
120        call.lb4.setStyleSheet("background-color: rgb(255, 0, 0)")

```

```

121     else:
122         call.lb4.setStyleSheet("background-color: rgb(0, 255, 0)")
123     if bao5 == 1:
124         call.lb5.setStyleSheet("background-color: rgb(255, 0, 0)")
125     else:
126         call.lb5.setStyleSheet("background-color: rgb(0, 255, 0)")
127     if bao6 == 1:
128         call.lb6.setStyleSheet("background-color: rgb(255, 0, 0)")
129     else:
130         call.lb6.setStyleSheet("background-color: rgb(0, 255, 0)")
131     #
132     if tt_vao_m == 0:
133         call.BT_VAO_m.setStyleSheet("background-color: rgb(255, 0, 0)")
134     else:
135         call.BT_VAO_m.setStyleSheet("background-color: rgb(0, 255, 0)")
136     if tt_ra_m == 0:
137         call.BT_RA_m.setStyleSheet("background-color: rgb(255, 0, 0)")
138     else:
139         call.BT_RA_m.setStyleSheet("background-color: rgb(0, 255, 0)")
140     if so_xe_m < 6:
141         call.lb_so_xe_m.setStyleSheet("background-color: rgb(85, 255, 0)")
142     else:
143         call.lb_so_xe_m.setStyleSheet("background-color: rgb(255, 0, 0)")
144     try:
145         # Đọc ảnh từ camera 1
146         frame1 = vs.read()
147         frame1 = imutils.resize(frame1, width=800)
148         imgg1 = QImage(frame1, frame1.shape[1], frame1.shape[0],
149                       QImage.Format_RGB888).rgbSwapped()
150         pix1 = QPixmap.fromImage(imgg1)
151         call.lb_vid.setPixmap(pix1)
152
153         # Đọc ảnh từ camera 2
154         frame2 = vs_m.read()
155         frame2 = imutils.resize(frame2, width=800)
156         imgg2 = QImage(frame2, frame2.shape[1], frame2.shape[0],
157                       QImage.Format_RGB888).rgbSwapped()
158         pix3 = QPixmap.fromImage(imgg2)
159         call.lb_vid_m.setPixmap(pix3)
160     except:

```

```

161     pass
162     if cp == 1:
163         if vr == 1:
164             if so_xe < 6:
165                 img = frame1
166                 xulyanh()
167                 pix1 = QPixmap('bien.jpg')
168                 call.lb_bien.setPixmap(pix1)
169                 time1 = time.time()
170                 if time1 - time2 >= 7:
171                     pygame.mixer.music.load("SOUND\WELCOME.MP3")
172                     pygame.mixer.music.play()
173                     time2 = time.time()
174                 mang_tg_vao[indextg] = time_present
175                 mang_time_vao[indextg] = time.time()
176                 mang_bien[indextg] = data
177                 call.lb_tgian_vao.setText(mang_tg_vao[indextg])
178                 indextg = indextg + 1
179                 if indextg >= 3:
180                     indextg = 0
181             else:
182                 print("day xe")
183                 time1 = time.time()
184                 if time1 - time2 >= 7:
185                     pygame.mixer.music.load("SOUND\FULL.MP3")
186                     pygame.mixer.music.play()
187                     time2 = time.time()
188         if vr == 2:
189             img = frame1
190             xulyanh()
191             pix1 = QPixmap('bien.jpg')
192             call.lb_bien.setPixmap(pix1)
193         cp = 0
194     if dc_ra == 1:
195         vitri = 0
196         for ii in range(0,3):
197             if mang_bien[ii] == data:
198                 vitri = ii
199                 print("ii = ",ii)
200     time_vao = mang_time_vao[vitri]

```

```

201 call.lb_tgian_ra.setText(time_present)
202 call.lb_tgian_vao.setText(mang_tg_vao[vitri])
203 time_ra = time.time()
204 tgian_gui = round(time_ra - time_vao)
205 tgian_gui = int(tgian_gui)
206 if tgian_gui <= 60:
207     tien = "5.000 VND"
208 elif tgian_gui <= 90:
209     tien = "10.000 VND"
210 else:
211     tien = "15.000 VND"
212 call.lb_tgian_gui.setText(str(tgian_gui) + " s")
213 call.lb_tien.setText(tien)
214 time1 = time.time()
215 if time1 - time2 >= 3:
216     pygame.mixer.music.load("SOUND\BYE.MP3")
217     pygame.mixer.music.play()
218     time2 = time.time()
219     dc_ra = 0
220 if dc_ra == 2:
221     time1 = time.time()
222     if time1 - time2 >= 7:
223         pygame.mixer.music.load("SOUND\ERROR.MP3")
224         pygame.mixer.music.play()
225         time2 = time.time()
226         dc_ra = 0
227 #
228 if cp_m == 1:
229     if vr_m == 1:
230         if so_xe_m < 6:
231             img_m = frame2
232             xulyanh2()
233             pix4 = QPixmap('bien.jpg')
234             call.lb_bien_m.setPixmap(pix4)
235             time1 = time.time()
236             if time1 - time2 >= 7:
237                 pygame.mixer.music.load("SOUND\WELCOME.MP3")
238                 pygame.mixer.music.play()
239                 time2 = time.time()
240 mang_tg_vao[indextg] = time_present

```

```

241         mang_time_vao[indextg] = time.time()
242         mang_bien[indextg] = data
243         call.lb_tgian_vao.setText(mang_tg_vao[indextg])
244         indextg = indextg + 1
245         if indextg >= 3:
246             indextg = 0
247         else:
248             print("day xe")
249             time1 = time.time()
250             if time1 - time2 >= 7:
251                 pygame.mixer.music.load("SOUND\FULL.MP3")
252                 pygame.mixer.music.play()
253                 time2 = time.time()
254         if vr_m == 2:
255             img_m = frame2
256             xulyanh2()
257             pix3 = QPixmap('bien.jpg')
258             call.lb_bien_m.setPixmap(pix3)
259         cp_m = 0
260     if dc_ra_m == 1:
261         vitri = 0
262         for ii in range(0,3):
263             if mang_bien[ii] == data:
264                 vitri = ii
265                 print("ii = ",ii)
266         time_vao = mang_time_vao[vitri]
267         call.lb_tgian_ra.setText(time_present)
268         call.lb_tgian_vao.setText(mang_tg_vao[vitri])
269         time_ra = time.time()
270         tgian_gui = round(time_ra - time_vao)
271         tgian_gui = int(tgian_gui)
272         if tgian_gui <= 60:
273             tien = "5.000 VND"
274         elif tgian_gui <= 90:
275             tien = "10.000 VND"
276         else:
277             tien = "15.000 VND"
278         call.lb_tgian_gui.setText(str(tgian_gui) + " s")
279         call.lb_tien.setText(tien)
280         time1 = time.time()

```

```

281         if time1 - time2 >= 3:
282             pygame.mixer.music.load("SOUND\BYE.MP3")
283             pygame.mixer.music.play()
284             time2 = time.time()
285             dc_ra_m = 0
286     if dc_ra_m == 2:
287         time1 = time.time()
288         if time1 - time2 >= 7:
289             pygame.mixer.music.load("SOUND\ERROR.MP3")
290             pygame.mixer.music.play()
291             time2 = time.time()
292             dc_ra_m = 0
293
294
295 #-----
296 def thuat():
297     ser.close()
298     call.close()
299     exit(app.exec())
300 #-----
301 def auto():
302     ser.write(b'a')
303     ser.write(b'|')
304     ser.write(b'0')
305     ser.write(b'\r\n')
306     ser.flush()
307 #-----
308 def manual():
309     ser.write(b'a')
310     ser.write(b'|')
311     ser.write(b'1')
312     ser.write(b'\r\n')
313     ser.flush()
314 #-----
315 def vao():
316     global dk_vao
317     dk_vao = dk_vao + 1
318     if dk_vao >= 2:
319         dk_vao = 0
320     ser.write(b'b')

```

```
321     ser.write(b'|')
322     ser.write(str(dk_vao).encode())
323     ser.write(b'\r\n')
324     ser.flush()
325
326 #-----
327 def ra():
328     global dk_ra
329     dk_ra = dk_ra + 1
330     if dk_ra >= 2:
331         dk_ra = 0
332     ser.write(b'c')
333     ser.write(b'|')
334     ser.write(str(dk_ra).encode())
335     ser.write(b'\r\n')
336     ser.flush()
337 #-----
338 def dk1():
339     ser.write(b'f')
340     ser.write(b'|')
341     ser.write(b'1')
342     ser.write(b'\r\n')
343     ser.flush()
344 #-----
345 def dk2():
346     ser.write(b'f')
347     ser.write(b'|')
348     ser.write(b'2')
349     ser.write(b'\r\n')
350     ser.flush()
351 #-----
352 def dk3():
353     ser.write(b'f')
354     ser.write(b'|')
355     ser.write(b'3')
356     ser.write(b'\r\n')
357     ser.flush()
358 #-----
359 def dk4():
360     ser.write(b'f')
```



```
361     ser.write(b'|')
362     ser.write(b'4')
363     ser.write(b'\r\n')
364     ser.flush()
365 #-----
366 def dk5():
367     ser.write(b'f')
368     ser.write(b'|')
369     ser.write(b'5')
370     ser.write(b'\r\n')
371     ser.flush()
372 #-----
373 def dk6():
374     ser.write(b'f')
375     ser.write(b'|')
376     ser.write(b'6')
377     ser.write(b'\r\n')
378     ser.flush()
379 #-----
380 def au_vao():
381     ser.write(b'e')
382     ser.write(b'|')
383     ser.write(b'1')
384     ser.write(b'\r\n')
385     ser.flush()
386 #-----
387 def au_ra():
388     ser.write(b'e')
389     ser.write(b'|')
390     ser.write(b'2')
391     ser.write(b'\r\n')
392     ser.flush()
393 #-----
394 def vao_m():
395     global dk_vao
396     dk_vao = dk_vao + 1
397     if dk_vao >= 2:
398         dk_vao = 0
399     ser.write(b'b1')
400     ser.write(b'|')
```

```

401     ser.write(str(dk_vao).encode())
402     ser.write(b'\r\n')
403     ser.flush()
404
405 #-----
406 def ra_m():
407     global dk_ra
408     dk_ra = dk_ra + 1
409     if dk_ra >= 2:
410         dk_ra = 0
411     ser.write(b'c1')
412     ser.write(b'|')
413     ser.write(str(dk_ra).encode())
414     ser.write(b'\r\n')
415     ser.flush()
416 #-----
417 def au_vao_m():
418     ser.write(b'e1')
419     ser.write(b'|')
420     ser.write(b'1')
421     ser.write(b'\r\n')
422     ser.flush()
423 #-----
424 def au_ra_m():
425     ser.write(b'e1')
426     ser.write(b'|')
427     ser.write(b'2')
428     ser.write(b'\r\n')
429     ser.flush()
430 #-----
431
432 def xulyanh():
433     global img, data, vr
434     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
435     thresh = cv2.adaptiveThreshold(gray, 255, cv2.
436     ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, 2)
437     contours, h = cv2.findContours(thresh, 1, 2)
438     largest_rectangle = [0, 0]
439     for cnt in contours:
440         lenght = 0.01 * cv2.arcLength(cnt, True)

```

```

441     approx = cv2.approxPolyDP(cnt, lenght, True)
442     if len(approx)==4:
443         area = cv2.contourArea(cnt)
444         if area > largest_rectangle[0]:
445             largest_rectangle = [cv2.contourArea(cnt),
446                                 cnt, approx]
447     x,y,w,h = cv2.boundingRect(largest_rectangle[1])
448
449     image=img[y:y+h, x:x+w]
450     cv2.drawContours(img,[largest_rectangle[1]],0,(0,255,0),8)
451
452     cropped = img[y:y+h, x:x+w]
453     #cv2.imshow('Dinh vi bien so xe', img)
454
455     cv2.drawContours(img,[largest_rectangle[1]],0,(255,255,255),18)
456     #DOC HINH ANH CHUYEN THANH FILE TEXT
457     pytesseract.pytesseract.tesseract_cmd =
458     'Tesseract-OCR/tesseract.exe'
459     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
460     blur = cv2.GaussianBlur(gray, (3,3), 0)
461     thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY_INV +
462                            cv2.THRESH_OTSU) [1]
463     #cv2.imshow('Crop', thresh)
464     cv2.imwrite("bien.jpg",thresh)
465     kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,3))
466     opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel,
467                               iterations=1)
468     invert = 255 - opening
469     data = pytesseract.image_to_string(invert, lang='eng',
470                                     config='--psm 6')
471
472     print("Bien so xe OTO la:")           # decode s
473     data = str(data)
474     data = data.rstrip()                 # cut "\r\n" at last of strin
475     data = data.replace('\n', '*')
476     print(data)
477
478     call.lb_bien_vao.setText(data)
479
480     ser.write(b'd')

```

```

481     ser.write(b'|')
482     ser.write(data.encode())
483     ser.write(b'\r\n')
484     ser.flush()
485 def xulyanh2():
486     global img_m, data_m, vr_m
487     gray = cv2.cvtColor(img_m, cv2.COLOR_BGR2GRAY)
488     thresh = cv2.adaptiveThreshold(gray,255,cv2.
489     ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY,11,2)
490     contours,h = cv2.findContours(thresh,1,2)
491     largest_rectangle = [0,0]
492     for cnt in contours:
493         lenght = 0.01 * cv2.arcLength(cnt, True)
494         approx = cv2.approxPolyDP(cnt, lenght, True)
495         if len(approx)==4:
496             area = cv2.contourArea(cnt)
497             if area > largest_rectangle[0]:
498                 largest_rectangle =
499                 [cv2.contourArea(cnt), cnt, approx]
500     x,y,w,h = cv2.boundingRect(largest_rectangle[1])
501
502     image=img_m[y:y+h, x:x+w]
503     cv2.drawContours(img_m,[largest_rectangle[1]],0,(0,255,0),8)
504
505     cropped = img_m[y:y+h, x:x+w]
506
507     cv2.drawContours(img_m,[largest_rectangle[1]],0,(255,255,255),18)
508     #DOC HINH ANH CHUYEN THANH FILE TEXT
509     pytesseract.pytesseract.tesseract_cmd =
510     'Tesseract-OCR/tesseract.exe'
511     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
512     blur = cv2.GaussianBlur(gray, (3,3), 0)
513     thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY_INV
514         + cv2.THRESH_OTSU)[1]
515     #cv2.imshow('Crop', thresh)
516     cv2.imwrite("bien.jpg",thresh)
517     kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,3))
518     opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel,
519         iterations=1)
520     invert = 255 - opening

```

```

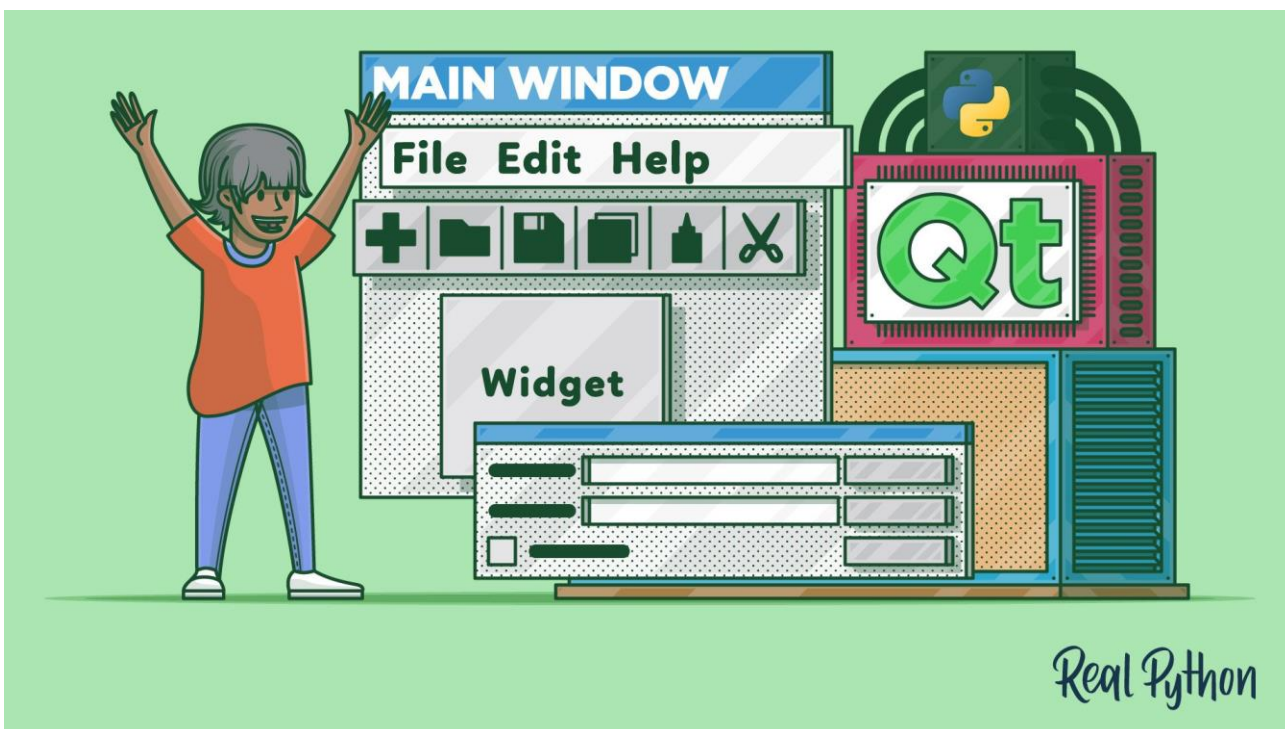
521 data = pytesseract.image_to_string(invert, lang='eng',
522                                   config='--psm 6')
523
524 print("Bien so xe may la:")           # decode s
525 data = str(data)
526 data = data.rstrip()                 # cut "\r\n" at last of strin
527 data = data.replace('\n', '*')
528 print(data)
529
530 call.lb_bien_vao_m.setText(data)
531
532 ser.write(b'd')
533 ser.write(b'|')
534 ser.write(data.encode())
535 ser.write(b'\r\n')
536 ser.flush()
537 print("ok")
538 #-----
539
540 #-----
541 while True:
542     app=QtWidgets.QApplication([])
543     call=uic.loadUi("GUID.ui")
544     call.QUIT.clicked.connect(thoat)
545     call.BT_AUTO.clicked.connect(auto)
546     call.BT_MAN.clicked.connect(manual)
547     call.BT_VAO.clicked.connect(vao)
548     call.BT_RA.clicked.connect(ra)
549     call.BT_K1.clicked.connect(dk1)
550     call.BT_K2.clicked.connect(dk2)
551     call.BT_K3.clicked.connect(dk3)
552     call.BT_K4.clicked.connect(dk4)
553     call.BT_K5.clicked.connect(dk5)
554     call.BT_K6.clicked.connect(dk6)
555     call.BT_XE_VAO.clicked.connect(au_vao)
556     call.BT_XE_RA.clicked.connect(au_ra)
557     #
558     call.BT_VAO_m.clicked.connect(vao_m)
559     call.BT_RA_m.clicked.connect(ra_m)
560     call.BT_XE_VAO_m.clicked.connect(au_vao_m)
561     call.BT_XE_RA_m.clicked.connect(au_ra_m)
562     call.timer = QTimer()
563     call.timer.timeout.connect(Time)
564     call.timer.start(50)
565     index1 = 1
566     call.show()
567     app.exec()

```

4.3.3 Thiết kế và lập trình giao diện

Giới thiệu phần mềm qt Designer:

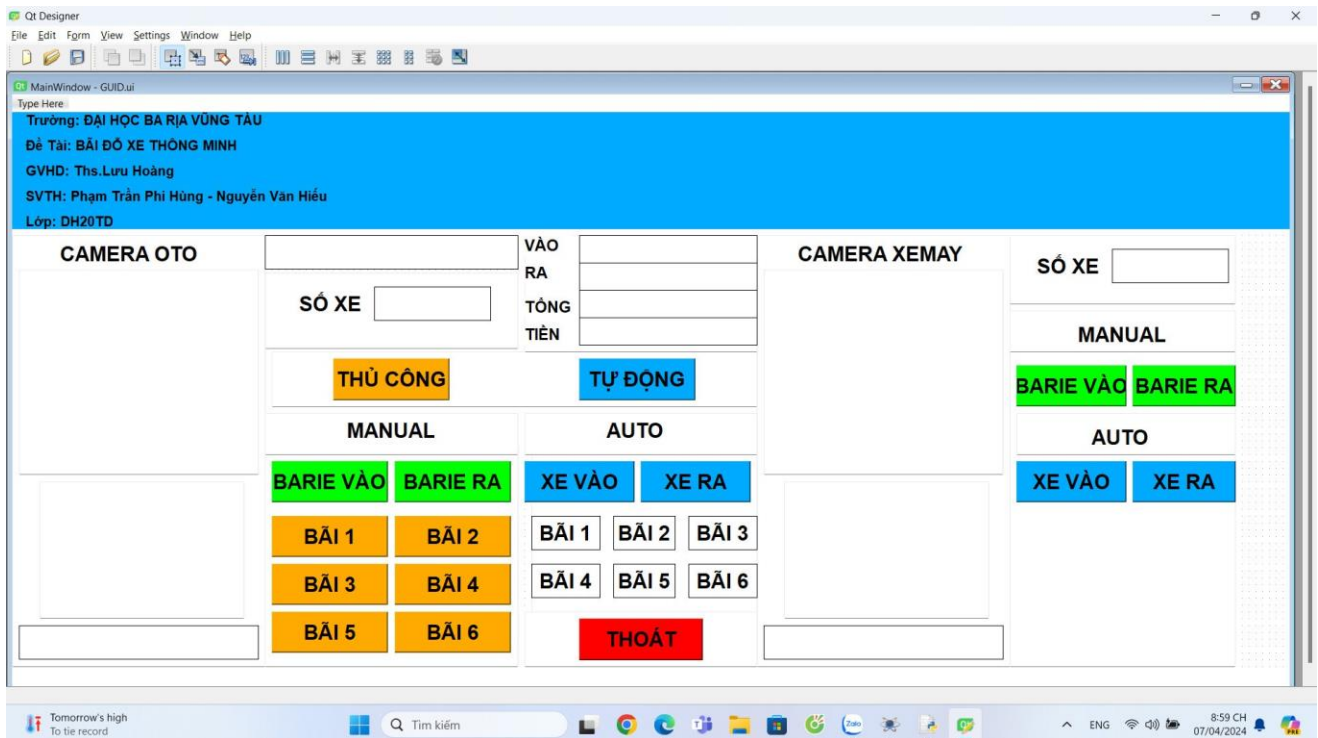
Qt Designer là một công cụ để nhanh chóng xây dựng giao diện người dùng đồ họa với các widget từ khung Qt GUI . Nó cung cấp cho chúng ta một giao diện kéo và thả đơn giản để bố trí các thành phần như nút, trường văn bản, hộp tổ hợp và hơn thế nữa.



Qt Designer tạo ra “.ui” các tệp. Đây là một định dạng dựa trên XML đặc biệt để lưu trữ các widget dưới dạng cây. Ta có thể tải các tệp này trong thời gian chạy hoặc dịch chúng sang ngôn ngữ lập trình như C ++ hoặc Python.

Với sự tương thích của phần mềm này đối với 2 loại code mà nhóm chúng em thực hiện là C# và Python, cùng với sự đơn giản dễ tiếp cận cho người mới làm quen, nhóm chúng em đã quyết định sử dụng phần mềm này để thiết kế giao diện điều khiển cho chương trình.

Giao diện chương trình thực hiện:



Hình 41: giao diện chương trình sau thiết kế

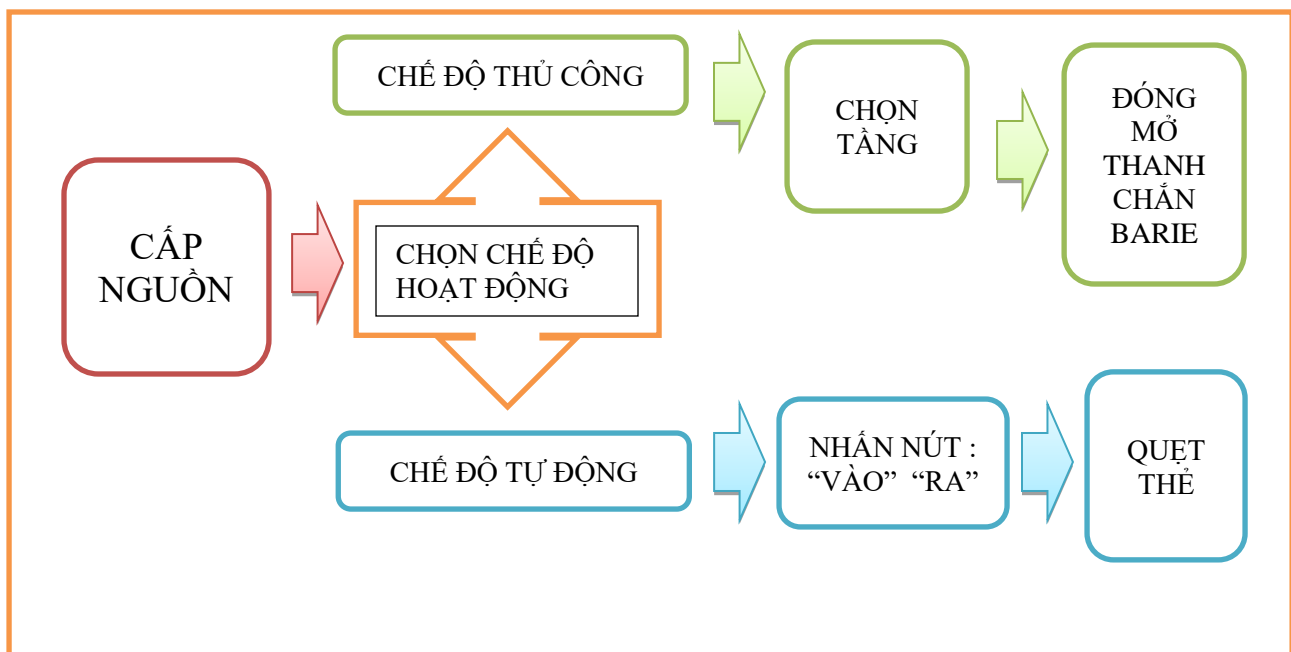
4.4 Tài liệu hướng dẫn sử dụng, thao tác

4.4.1 Hướng dẫn các bước sử dụng

Để có thể vận hành hệ thống cần thực hiện theo trình tự các bước sau:

- Bước 1: cấp nguồn cho hệ thống, hệ thống sử dụng nguồn chính là 220V AC, khi cấp nguồn thì đèn hệ thống tự động sáng nên, khi đẩy hệ thống bắt đầu hoạt động.
- Bước 2: Download chương trình chính trên phần mềm Arduino IDE xuống arduino nano
- Bước 3: Mở giao diện hệ thống trên qt Designer
- Bước 4: Mở chương trình trên Python IDLE để chương trình xử lý ảnh hoạt động
- Bước 5: Chọn chế độ hoạt động trên giao diện qt Designer sau đó sử dụng
- Bước 6: nếu có sự cố ta nhấn nút reset để hệ thống dừng hoạt động hay hoạt động lại từ đầu

4.4.2 Quy trình thao tác

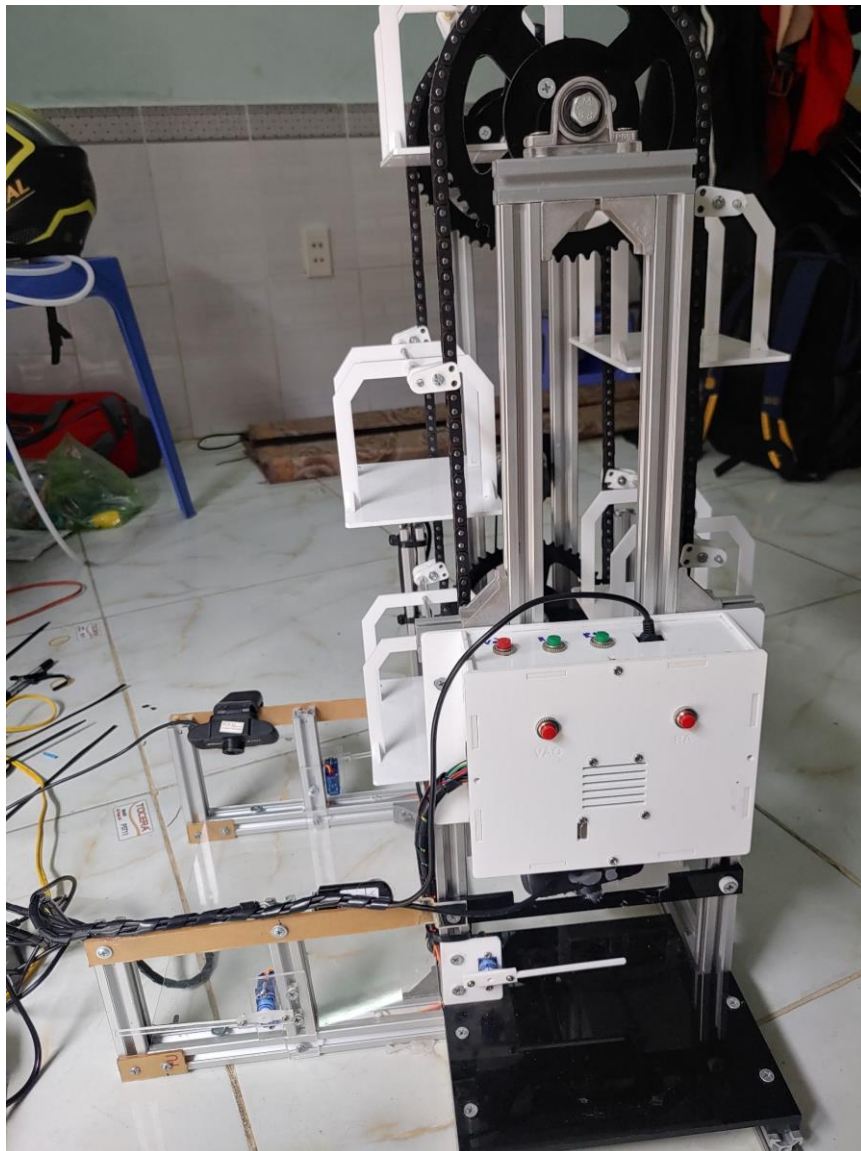


CHƯƠNG 5

KẾT QUẢ THU HOẠCH

5.1 Hệ thống mô hình

Qua những cố gắng và nỗ lực của bản thân cùng với sự giúp đỡ của giảng viên hướng dẫn cũng như thầy cô và các bạn bè đã giúp đỡ em trong thời gian qua. Nhờ đó mà em đã hoàn thành được đề tài “MÔ HÌNH BÃI GỬI XE TỰ ĐỘNG”. Mô hình thu được có kích thước như dự kiến gồm tổ hợp các linh kiện cơ khí được lắp ghép hoàn chỉnh như sau:



Hình 42: mô hình thu được

Cụ thể đề tài nghiên cứu đã đạt được những kết quả như sau:

- Hệ thống có thể di chuyển tầng ổn định
- Camera chụp ảnh hoạt động ổn định
- Thẻ RFID giao tiếp được với arduino và hoạt động ổn định
- Giao diện kết nối được với hệ thống và hoạt động ổn định
- Có thể tính phí
- Có thể xử lí ảnh
- Chế độ tay hoạt động tốt

Bên cạnh những kết quả thu được, hệ thống còn xảy ra một số lỗi như sau:

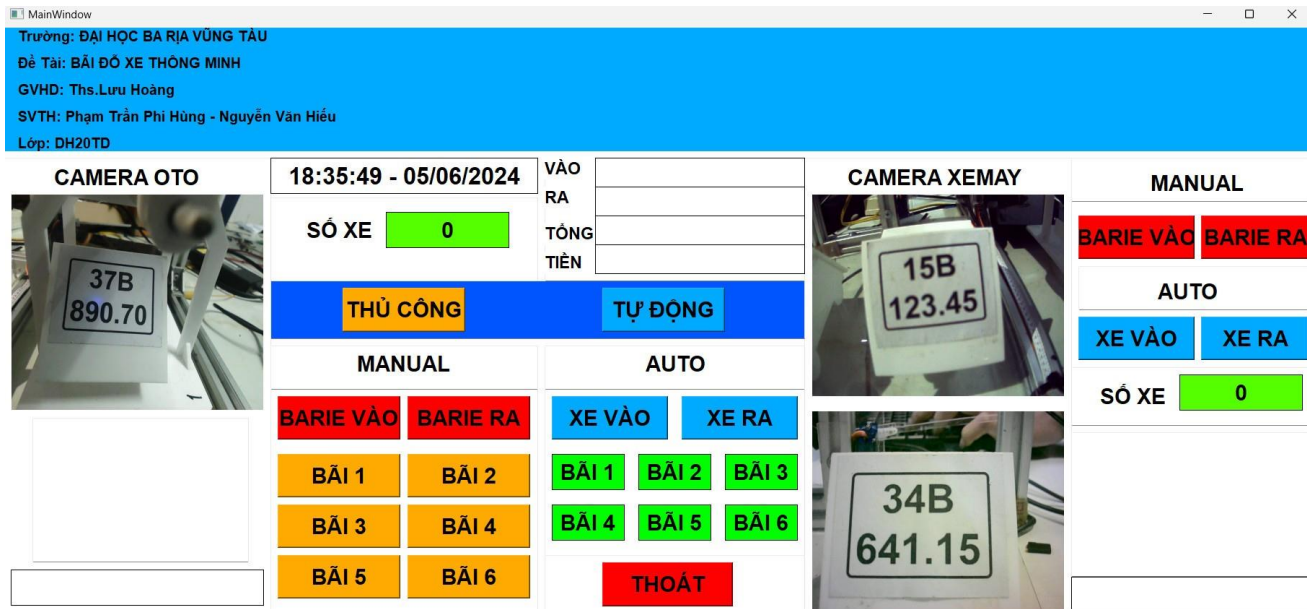
- Xử lí ảnh đôi khi còn bị lỗi, nhận dạng biển số chưa đúng
- Hệ thống chạy còn chậm trễ
- Đôi khi hệ trục quay còn bị kẹt xích

Lý do xảy ra lỗi trên:

- Có thể do thiếu hiểu biết về chiều sâu của code nên chương trình vẫn chưa thực sự hoạt động ổn định
- Hình ảnh thu được đôi khi vẫn có tạp ảnh mà chương trình không có chức năng khử tạp ảnh
- Khung tải và trục quay còn bị yếu do được cố định bằng ốc vít nên vẫn còn bị cong vênh

5.2 Giao diện phần mềm

Sau khi hoàn thành sắp xếp bố cục cho giao diện trên QT Designer, chúng em thực hiện liên kết giao diện với các tag trong python và chạy chương trình. Giao diện thu được như sau:



Hình 43: giao diện điều khiển thu được

Kết quả giao diện thu được gồm những thành phần chính như sau:

- Thông tin đề tài
- Có 3 ô lớn để hiển thị hình ảnh thu được qua 3 camera
- 1 ô hiển thị hình ảnh thu được sau khi xử lý tương phản đen trắng
- 1 ô hiển thị kí tự biển số thu được sau khi xử lý từ ảnh đen trắng
- 2 nút giúp chuyển chế độ qua lại giữa chế độ thủ công và chế độ tự động
- Ở chế độ thủ công gồm 6 nút chọn tầng và 4 nút điều khiển 4 thanh chắn barie
- Ở chế độ tự động gồm 10 đèn hiển thị trạng thái của 6 tầng và 4 thanh chắn barie
- Các ô hiển thị thời gian ra vào cổng, phí dịch vụ sau khi lấy xe

CHƯƠNG 6

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

6.1.1 Ưu điểm

- Hiệu quả xử dụng diện tích của mô hình đáp ứng tốt cho yêu cầu về diện tích mặt sàn sử dụng.
- Khả năng tự động hóa cao gần như đáp ứng được hết tất cả công đoạn.
- Việc tích hợp camera và tầng để xe cá nhân giúp nâng cao mức độ an ninh cho phương tiện xử dụng hệ thống.
- Cách bố trí phù hợp cho người xử dụng ra vào an toàn.
- Mô hình được thiết kế theo hình hộp vì vậy khá dễ dàng để có thể nâng cấp, lắp mở rộng thêm.

6.1.2 Nhược điểm

- Vì áp dụng nhiều trang thiết bị nên chi phí đầu tư ban đầu nhiều hơn so với bãi đỗ xe truyền thống
- Yêu cầu kĩ thuật cao trong việc lắp đặt và vận hành
- Việc sửa chữa khi có lỗi hay sự cố xảy ra cũng trở nên khó khăn
- Khả năng an toàn cháy nổ thấp nếu không trang bị hệ thống chữa cháy riêng biệt
- Không phù hợp với khách hàng có ác cảm với độ cao, không tin tưởng vào kết cấu hệ thống và muốn một phương án truyền thống an toàn hơn

6.1.3 Kết luận

Thiết kế và chế tạo thành công mô hình bãi đỗ xe tự động dạng vòng xoay đã đạt được những kết quả sau:

- Hệ thống bao gồm các bộ phận chính như: vòng xoay bánh xích, khay chứa xe, hệ thống điều khiển, hệ thống giao diện người dùng.
- Phát triển phần mềm điều khiển hệ thống. Phần mềm có giao diện trực quan, dễ sử dụng, cho phép người dùng thao tác di chuyển khay chứa xe, quản lý thông tin xe ra vào, thiết lập các tham số hệ thống.
- Thử nghiệm và đánh giá hiệu quả hoạt động của hệ thống. Hệ thống hoạt động ổn định, an toàn, đáp ứng được các yêu cầu đề ra.

Hệ thống có tính mới, tính sáng tạo ở những điểm sau:

- Sử dụng cơ cấu vòng xoay bánh xích để di chuyển khay chứa xe, giúp tiết kiệm diện tích và tăng hiệu quả sử dụng không gian.
- Tích hợp hệ thống điều khiển thông minh, tự động hóa cao các thao tác vận hành.
- Dễ dàng mở rộng và nâng cấp để đáp ứng nhu cầu sử dụng ngày càng cao.

Hệ thống có giá trị khoa học và ứng dụng thực tiễn cao:

- Có giá trị khoa học trong lĩnh vực tự động hóa và robot.
- Có thể ứng dụng trong các khu vực có diện tích hẹp, nhu cầu đỗ xe cao như: trung tâm thương mại, khu chung cư, bệnh viện, sân bay,...

6.2 Hướng phát triển đề tài

6.2.1 Mở rộng mô hình

Hiện tại, mô hình chỉ có 6 tầng, con số này còn khá khiêm tốn so với nhu cầu thực tế. Để đáp ứng tốt hơn nhu cầu đỗ xe ngày càng cao, đặc biệt là tại các khu vực đô thị, cần thiết phải mở rộng thêm nhiều tầng hơn trên mỗi cơ cấu xoay.

Việc mở rộng mô hình mang lại nhiều lợi ích thiết thực:

- Tăng số lượng chỗ đỗ xe: Mỗi tầng bổ sung thêm sẽ gia tăng đáng kể số lượng xe có thể lưu trữ, giúp giải quyết vấn đề thiếu hụt chỗ đỗ xe hiệu quả.
- Tối ưu hóa diện tích sử dụng: Mô hình nhiều tầng giúp tận dụng tối đa không gian chiều cao, tiết kiệm diện tích mặt bằng so với các bãi đỗ xe truyền thống.
- Nâng cao hiệu quả vận hành: Hệ thống điều khiển thông minh có thể quản lý hiệu quả việc di chuyển khay chứa xe giữa các tầng, đảm bảo an toàn và tối ưu hóa thời gian.
- Phù hợp với các khu vực có diện tích hẹp: Tại những nơi có diện tích hạn chế, mô hình nhiều tầng là giải pháp tối ưu để đáp ứng nhu cầu đỗ xe cao.

6.2.2 Hoàn thiện mô hình

Hiện tại, mô hình còn một số hạn chế về mặt thẩm mỹ và độ bền do thiếu kinh nghiệm thực hành thi công. Để khắc phục những hạn chế này và đưa mô hình đến gần hơn với thực tế, cần thực hiện những việc sau:

- Sử dụng vật liệu cao cấp hơn: Thay thế các vật liệu hiện tại bằng các vật liệu có độ bền cao, thẩm mỹ tốt như kim loại, nhựa ABS,...
- Gia công chi tiết: Chú ý đến độ chính xác và tỉ mỉ trong quá trình gia công các chi tiết, đảm bảo mô hình có hình thức đẹp mắt, tinh xảo.
- Hoàn thiện sơn phủ: Lựa chọn màu sơn phù hợp, chất lượng tốt để tạo vẻ ngoài sang trọng và thu hút cho mô hình.

- Tăng cường kết cấu: Sử dụng các thanh đỡ, khung gia cố để tăng độ cứng vững cho mô hình, đảm bảo khả năng chịu tải tốt.
- Lựa chọn linh kiện chất lượng cao: Sử dụng các linh kiện như bánh răng, xích, động cơ có nguồn gốc xuất xứ rõ ràng, đảm bảo độ bền bỉ trong quá trình vận hành.
- Kiểm tra và bảo dưỡng định kỳ: Thực hiện kiểm tra định kỳ các bộ phận của mô hình, tra dầu mỡ, siết ốc vít để đảm bảo hoạt động trơn tru và bền bỉ.

6.2.3 Tích hợp phương thức tính tiền tự động

Hệ thống tự động hóa hoàn chỉnh không thể thiếu tính năng thanh toán tự động. Việc thu phí thủ công không chỉ lãng phí nhân lực mà còn hạn chế hiệu quả vận hành, đi ngược lại xu hướng tự động hóa cao. Do vậy, kết hợp hệ thống tính tiền tự động thông qua phần mềm ngân hàng hoặc ví điện tử là hướng phát triển thiết yếu cho mô hình bãi đỗ xe tự động dạng vòng xoay bánh xích.

Lợi ích của hệ thống tính tiền tự động:

- Tiết kiệm nhân lực: Loại bỏ nhu cầu thu phí thủ công, giải phóng nhân viên cho các công việc khác, tối ưu hóa chi phí vận hành.
- Nâng cao hiệu quả: Tăng tốc độ thanh toán, giảm thời gian chờ đợi cho khách hàng, mang đến trải nghiệm thông minh và tiện lợi.
- Giảm thiểu sai sót: Loại bỏ sai sót do tính toán thủ công, đảm bảo tính chính xác và minh bạch trong thanh toán.
- Tăng tính tiện lợi: Hỗ trợ đa dạng phương thức thanh toán như thẻ ngân hàng, ví điện tử, mã QR,... đáp ứng nhu cầu đa dạng của khách hàng.

Tuy nhiên, việc triển khai hệ thống tính tiền tự động cũng gặp một số khó khăn:

- Yêu cầu kỹ thuật cao: Cần có kiến thức chuyên môn về lập trình, kết nối API với các ngân hàng và ví điện tử, đảm bảo an toàn bảo mật cho giao dịch thanh

toán.

- Chi phí đầu tư: Chi phí đầu tư cho hệ thống phần mềm, thiết bị thanh toán có thể cao hơn so với phương thức thanh toán thủ công.
- Sự phối hợp: Cần sự phối hợp chặt chẽ giữa các bên liên quan như nhà phát triển phần mềm, ngân hàng, ví điện tử để đảm bảo hệ thống hoạt động trơn tru.

Đề xuất hướng phát triển:

- Sử dụng dịch vụ thanh toán của bên thứ ba: Hợp tác với các công ty cung cấp dịch vụ thanh toán trực tuyến như VNPay, MoMo,... để tích hợp hệ thống thanh toán vào mô hình.
- Tự phát triển hệ thống thanh toán: Nếu có đủ nguồn lực và chuyên môn, ta có thể tự phát triển hệ thống thanh toán riêng, tuy nhiên sẽ tốn nhiều thời gian và chi phí hơn.

6.2.4 Tích hợp điều khiển giám sát từ xa thông qua web cũng như app điện thoại

Hệ thống tự động hóa hoàn chỉnh không chỉ dừng lại ở hoạt động tự động mà còn cần có khả năng giám sát và điều khiển từ xa hiệu quả. Do vậy, mặc dù rất muốn thực hiện nhưng chúng em vẫn chưa thực hiện được, đó là khả năng tích hợp hệ thống giám sát và điều khiển từ xa thông qua web và ứng dụng di động cho mô hình. Và đây cũng là hướng phát triển thiết yếu cho mô hình bãi đỗ xe tự động dạng vòng xoay bánh xích.

Lợi ích của hệ thống giám sát và điều khiển từ xa:

- Giám sát toàn diện: Theo dõi trạng thái hoạt động của hệ thống theo thời gian thực, phát hiện sớm các sự cố tiềm ẩn, đảm bảo vận hành trơn tru.
- Can thiệp kịp thời: Xử lý nhanh chóng các vấn đề phát sinh, hạn chế tối đa thiệt hại và gián đoạn hoạt động.
- Quản lý thông minh: Thu thập và phân tích dữ liệu vận hành, đưa ra các quyết

định điều chỉnh tối ưu hóa hiệu quả hệ thống.

- Tiện lợi và linh hoạt: Truy cập và điều khiển hệ thống từ mọi nơi, mọi lúc thông qua thiết bị di động, nâng cao trải nghiệm quản lý.

Tuy nhiên, việc triển khai hệ thống giám sát và điều khiển từ xa cũng gặp một số khó khăn:

- Yêu cầu kỹ thuật cao: Cần có kiến thức về lập trình web, ứng dụng di động, kết nối mạng, bảo mật dữ liệu để xây dựng hệ thống an toàn và hiệu quả.
- Chi phí đầu tư: Chi phí cho phần mềm, thiết bị và dịch vụ lưu trữ đám mây có thể cao hơn so với hệ thống giám sát truyền thống.
- Sự an toàn bảo mật: Cần phải lựa chọn và xác định phương tiện lưu trữ thông tin đảm bảo an toàn cho dữ liệu hệ thống, tránh rò rỉ thông tin tọa điều kiện cho kẻ xấu lợi dụng.

Mặc dù gặp khó khăn, tuy nhiên em vẫn đề xuất hướng phát triển cần thiết để nâng cao giá trị ứng dụng của mô hình. Một số giải pháp có thể tham khảo:

Sử dụng dịch vụ lưu trữ đám mây: Cung cấp nền tảng lưu trữ dữ liệu an toàn và có thể mở rộng dễ dàng như Google Cloud Platform, Amazon Web Services,...

Phát triển ứng dụng di động: Tự thiết kế ứng dụng di động với giao diện thân thiện, dễ sử dụng, tối ưu hóa cho các thiết bị di động.

TÀI LIỆU THAM KHẢO

1. Phạm Tấn Lộc. “Lập trình C# 9.0 từ cơ bản đến nâng cao”. Nhà xuất bản Đại học Bách khoa Hà Nội, 2023.
2. Phạm Đình Tuấn. “Lập trình Python 3.10 từ cơ bản đến nâng cao”. Nhà xuất bản Đại học Quốc gia Hà Nội, 2023.
3. Phan Minh Đức. “Giao diện người dùng với Qt 5”. Nhà xuất bản Công nghệ TP.HCM, 2020.
4. Đặng Văn Long. “Hệ thống tự động hóa trong kỹ thuật xây dựng”. Nhà xuất bản Xây dựng, 2022.