



ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN



ĐỒ ÁN TỐT NGHIỆP
CAPSTONE PROJECT

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

ĐỀ TÀI:

**NGHIÊN CỨU VÀ THIẾT KẾ BÃI ĐỖ XE THÔNG
MINH ỨNG DỤNG TẠI DỰ ÁN DANANG TIMES
SQUARE CỦA CÔNG TY CỔ PHẦN VIỆT CHÀO,
HÀ NỘI**

Người hướng dẫn: **TS. NGUYỄN QUỐC ĐỊNH**

Sinh viên thực hiện:

- 1. ĐOÀN VĨNH QUANG – MSSV: 105200423 – LỚP: 20TDHCLC2**
- 2. HỒ NGUYỄN QUỐC HIẾU – MSSV: 105200405 – LỚP: 20TDHCLC2**

Đà Nẵng, tháng 6 năm 2025

TÓM TẮT

Tên đề tài: Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Chèo, Hà Nội.

Sinh viên thực hiện: Hồ Nguyên Quốc Hiếu (105200405)

Đoàn Vĩnh Quang (105200423)

Lớp: 20TDHCLC2

Đề tài tập trung nghiên cứu và thiết kế hệ thống bãi đỗ xe thông minh nhằm tối ưu hóa không gian đỗ xe, nâng cao hiệu quả khai thác tầng hầm của dự án Times Square Đà Nẵng. Hệ thống được ứng dụng công nghệ tự động hóa trong việc phân loại, sắp xếp và vận hành xe ra vào, kết hợp phần mềm quản lý, cảm biến, hệ điều khiển và cơ khí chính xác.

Giải pháp thiết kế dựa trên các tài liệu kỹ thuật của Công ty Việt Chèo và hệ thống đỗ xe cơ khí PSH của Công ty XIZI, đảm bảo phù hợp với diện tích, kết cấu công trình, đáp ứng các yêu cầu về an toàn, vận hành và thẩm mỹ.

Mục tiêu:

- Tối ưu diện tích đỗ xe tầng hầm.
- Giảm thiểu thời gian ra/vào xe.
- Phân loại xe ra vào
- Nhận diện biển số xe nhanh chóng
- Đảm bảo an toàn trong vận hành.
- Ứng dụng công nghệ hiện đại, phù hợp tiêu chuẩn kỹ thuật.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Ngành
1	Hồ Nguyên Quốc Hiếu	105200405	20TDHCLC2	Kỹ thuật điều khiển và tự động hóa
2	Đoàn Vĩnh Quang	105200423	20TDHCLC2	Kỹ thuật điều khiển và tự động hóa

1. Tên đề tài đồ án:

Nghiên cứu và thiết kế bãi đỗ xe thông minh ứng dụng tại dự án Times Square Đà Nẵng của Công ty Cổ phần Việt Chèo

2. Đề tài thuộc diện: Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

Dựa trên hệ thống đỗ xe cơ khí PSH của công ty XIZI (bao gồm thông số kỹ thuật, bản vẽ thiết kế, nguyên lý hoạt động) và các tài liệu kỹ thuật, tiêu chuẩn thiết kế, yêu cầu kỹ thuật do công ty Cổ phần Việt Chèo cung cấp phục vụ dự án Times Square Đà Nẵng.

- Bản vẽ kiến trúc: Tầng hầm CT3-CT7 Times Square Đà Nẵng

- Diện tích công trình: Tầng 1: 1100 m², Tầng 2: 1100 m²

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

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Hồ Nguyên Quốc Hiếu	- Phân tích tổng quan đề tài, giới thiệu hệ thống PSH, lựa chọn phương án thiết kế, tính toán kết cấu cơ khí.
2	Đoàn Vĩnh Quang	

		- Thiết kế sơ đồ nguyên lý điều khiển, tính toán điều khiển tự động hóa, đề xuất giải pháp phần mềm điều khiển, giao diện.
--	--	--

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Hồ Nguyên Quốc Hiếu	- Tính toán kết cấu cơ khí hệ thống đỗ xe (khung thép, sàn đỡ, truyền động) - Tính toán tải trọng, sức chứa và bố trí mặt bằng hệ thống. - Đề xuất giải pháp an toàn cơ khí và thiết bị nâng hạ
2	Đoàn Vĩnh Quang	- Thiết kế giao diện phần mềm quản lý (giao diện người - máy) - Thiết kế mô hình AI xử lý ảnh - Xây dựng phần mềm bãi đỗ xe thông minh

5. Các bản vẽ, đồ thị:

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Hồ Nguyên Quốc Hiếu	- Bản vẽ tổng thể hệ thống bãi đỗ xe thông minh.
2	Đoàn Vĩnh Quang	- Bản vẽ mặt bằng bố trí hệ thống trong không gian thực tế (tầng hầm) - Bản vẽ hệ thống chỉ dẫn và báo chỗ trống - Bản vẽ hệ thống cấp nguồn cho bãi đỗ xe

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Hồ Nguyên Quốc Hiếu	- Bản vẽ chi tiết kết cấu khung, sàn đỡ, cơ cấu truyền động.

		- Bản vẽ phân bố không gian xe - Bản vẽ hệ thống phân loại xe
2	Đoàn Vĩnh Quang	- Sơ đồ tổng thể phần mềm - Mô hình phân lớp

6. <i>Họ tên người hướng dẫn:</i>	<i>Phần/ Nội dung:</i>
TS. Nguyễn Quốc Định	Kiểm tra tiến độ của đồ án
Kĩ sư Nguyễn Đông Triều	Quy trình hệ thống đỗ xe cơ khí và các thiết bị
Kĩ sư Huỳnh Văn Huy	Quy trình hệ thống đỗ xe cơ khí và các thiết bị

7. Ngày giao nhiệm vụ đồ án: 6/3/2025

8. Ngày hoàn thành đồ án: 26/6/2025

Đà Nẵng, ngày tháng 6 năm 2025

Trưởng Bộ môn Tự động hóa

Người hướng dẫn

TS. Giáp Quang Huy

PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Hồ Nguyên Quốc Hiếu (105200405)

Đoàn Vĩnh Quang (105200423)

Tên đề tài ĐATN: Nghiên cứu và thiết kế bãi đỗ xe thông minh ứng dụng tại dự án Times Square Đà Nẵng của Công ty Cổ phần Việt Chèo

Họ tên người HD: TS. Nguyễn Quốc Định

Đơn vị: Khoa Điện

Tuần	Ngày	Khối lượng		GVHD ký tên
		đã thực hiện (%)	tiếp tục thực hiện (%)	
1	6/3	Nhận đề tài và nghiên cứu cơ bản về đề tài (10%)	Nghiên cứu tài liệu, phân tích yêu cầu, xây dựng hướng thiết kế	
2	10/3	Phân tích hệ thống, lập sơ đồ khối hệ thống cơ khí (20%)	Thiết kế sơ bộ cơ khí, điện	
3	18/3	Duyệt lần 1: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
4	24/3	Chi tiết thiết bị cơ khí và cách lắp đặt (30%)	Bổ sung bản vẽ, chức năng các thiết bị	
5	2/4	Duyệt lần 2: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		

Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Chèo, Hà Nội

6	7/4	Nghiên cứu hệ thống điện và tủ điều khiển của hệ thống cơ khí (40%)	Hoàn thiện bản vẽ điện, đấu nối	
7	14/4	Mô phỏng phân cứng phân loại xe (55%)	Hoàn thiện mô phỏng	
8	21/4	Duyệt lần 3: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
9	28/4	Thiết kế phần mềm nhận diện biển số xe AI (70%)	Hoàn thiện phần mềm, kiểm tra logic	
10	5/5	Thiết kế giao diện phần mềm bãi đỗ xe thông minh (80%)	Hoàn thiện giao diện, kiểm tra lỗi	
11	15/5	Duyệt lần 4: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
12	1/6	Viết báo cáo và kiểm tra lại toàn bộ phần mềm (95%)	Chuẩn bị bảo vệ ĐATN	
13	24/6	Bảo vệ ĐATN, hoàn thành nhiệm vụ		

LỜI NÓI ĐẦU VÀ CẢM ƠN

Đồ án tốt nghiệp chuyên ngành Kỹ thuật điều khiển và Tự động hóa với dự án “*Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Chèo, Hà Nội*” là kết quả của quá trình cố gắng không ngừng của chúng em và được sự giúp đỡ, động viên khích lệ của các thầy, bạn bè, đồng môn và người thân. Qua trang viết em xin gửi lời cảm ơn tới những người đã giúp đỡ chúng em trong thời gian học tập – nghiên cứu đồ án tốt nghiệp vừa qua.

Đầu tiên, chúng em xin gửi lời cảm ơn và biết ơn sâu sắc đối với thầy TS. Nguyễn Quốc Định đã trực tiếp tận tình hướng dẫn, giúp đỡ chúng em trong suốt quá trình thực hiện đồ án và tạo mọi điều kiện để chúng em có thể hoàn thành tốt đồ án này.

Đặc biệt, chúng em xin cảm ơn hai anh Nguyễn Đông Triều và Huỳnh Văn Huy, những người đã trực tiếp hỗ trợ kỹ thuật, không quản ngại khó khăn, dành thời gian hướng dẫn chi tiết về hệ thống cơ khí. Sự giúp đỡ tận tâm của hai anh đã giúp chúng em hiểu rõ hơn về các vấn đề kỹ thuật phát sinh, từ đó đưa ra những giải pháp tối ưu cho dự án.

Cuối cùng, chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Bách khoa Đà Nẵng, nơi đã trang bị cho chúng em những kiến thức nền tảng vững chắc và tạo môi trường học tập, nghiên cứu tốt nhất để chúng em có thể hoàn thành đồ án này.

Sự hỗ trợ quý báu từ tất cả mọi người là nguồn động lực to lớn, giúp chúng em vượt qua mọi thử thách và đạt được thành quả như ngày hôm nay. Chúng em xin chân thành cảm ơn!

Chúng em xin chân thành cảm ơn!

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Nhóm em gồm 2 sinh viên: Hồ Nguyên Quốc Hiếu và Đoàn Vĩnh Quang. Xin cam đoan rằng toàn bộ nội dung của đề án tốt nghiệp với đề tài “*Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Chèo, Hà Nội*” là kết quả nghiên cứu, tìm hiểu và thực hiện của riêng nhóm em dưới sự hướng dẫn của TS. Nguyễn Quốc Định và 2 anh kỹ sư Nguyễn Đông Triều và Huỳnh Văn Huy.

Nhóm em cam kết đã thực hiện đề án với tinh thần trung thực, liêm chính học thuật và không sao chép nội dung từ bất kỳ đề án, luận văn, tài liệu hoặc công trình nghiên cứu nào khác mà không trích dẫn rõ ràng nguồn gốc theo quy định.

Nếu phát hiện có sự gian lận, sao chép hoặc vi phạm liêm chính học thuật, nhóm em hoàn toàn chịu trách nhiệm trước Hội đồng chấm đề án và quy định của nhà trường.

Sinh viên thực hiện

MỤC LỤC

TÓM TẮT	I
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	II
PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP	V
LỜI NÓI ĐẦU VÀ CẢM ƠN	VII
MỤC LỤC	IX
DANH SÁCH HÌNH ẢNH VÀ BẢNG	XI
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT	XVI
CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ ĐƠN VỊ DOANH NGHIỆP	1
1.1. Giới thiệu về doanh nghiệp.....	1
1.2. Tổ chức bộ máy nhân sự của công ty:	3
1.3. Sản phẩm.....	4
1.4. Một số dự án tiêu biểu của Việt Chèo.....	9
CHƯƠNG 2: TỔNG QUAN DỰ ÁN	12
2.1. Sơ đồ bố trí mặt bằng tòa nhà Times Square Đà Nẵng.....	12
2.2. Bản vẽ mỗi block hệ thống đỗ xe cơ khí tại tầng hầm 1.....	14
2.2.2. Mặt bằng định vị cột mỗi block hệ thống đỗ xe cơ khí tại tầng hầm 2.....	14
2.2.2. Mô tả hệ thống đỗ xe cơ khí – 2 tầng – 3 xe.....	14
2.2.3. Mô tả hệ thống đỗ xe cơ khí – 2 tầng – 5 xe.....	18
2.3. Hệ thống chỉ dẫn, báo trống, phân loại và giám sát đỗ xe cơ khí.....	21
2.3.1. Hệ thống phân loại xe tại tầng hầm 1	23
2.3.2. Hệ thống chỉ, báo trống đỗ xe tại tầng hầm 2.....	24
2.4. Hệ thống giám sát bãi đỗ xe cơ khí	26
2.5. Hệ thống cấp nguồn bãi đỗ xe CT3-CT7	31
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG HƯỚNG DẪN ĐỖ XE	35
3.1. Hệ thống chỉ dẫn đỗ xe KPGS	35
3.2. Cấu hình hệ thống KPGS	36
CHƯƠNG 4: HỆ THỐNG ĐỖ XE CƠ KHÍ PUZZLE STACKER HORIZONTAL (PSH)	47
4.1. Cấu tạo chính của hệ thống đỗ xe cơ khí puzzle stacker horizontal (PSH).....	47
4.1.1 Kết cấu thép	47

4.1.2	Hệ thống truyền động.....	47
4.1.3	Pallet	48
4.1.4	Thiết bị bảo vệ.....	49
4.1.5	Tủ điều khiển	53
CHƯƠNG 5: THIẾT KẾ PHẦN MỀM PHÂN LOẠI XE VÀ NHẬN DIỆN BIỂN SỐ		64
5.1	Tổng quan kiến trúc phần mềm.....	64
5.1.1	Sơ đồ tổng thể hệ thống phần mềm	64
5.1.2	Mô hình phân lớp (Layered Architecture)	67
5.2	Mô hình phân loại xe và đọc thẻ RFID.....	70
5.2.1	Mô tả quy trình phân loại	70
5.2.2	Sử dụng cảm biến IR33 để phân loại.	70
5.2.3	Kết nối phần cứng IR33 và vi điều khiển	72
5.3	Giao tiếp và Xử lý tín hiệu RFID	74
5.3.1	RFID IC CARD / Tag Reader HS0430B	74
5.3.2	Kết nối phần cứng RFID và vi điều khiển	75
5.3.3	Xử lý UID từ RFID và gửi lên máy tính	76
5.4	Phần mềm nhận diện biển số.....	80
5.4.1	Giới thiệu về mô hình YOLOv10 và PaddleOCR	80
5.4.2	Quy trình chụp và xử lý ảnh biển số của YOLOv10 và PaddleOCR:.....	86
5.5	Thiết kế giao diện người dùng (GUI):	87
5.5.1	Công cụ và thư viện cần dùng:	87
5.5.2	Giao diện người dùng (GUI):.....	88
5.6	Chi tiết hoạt động của phần mềm:.....	89
5.6.1	Các chức năng chính của phần mềm:	89
5.6.2	Giải thích chi tiết từng hàm trong phần mềm:.....	90
5.7	Đánh giá hiệu quả phần mềm.....	100
5.7.1	Tỷ lệ nhận diện chính xác	100
5.7.2	Kết quả xử lý và phản hồi của phần mềm	101
5.7.3	Các hạn chế và đề xuất cải tiến.....	111
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		113
TÀI LIỆU THAM KHẢO		114
PHỤ LỤC.....		115

DANH SÁCH HÌNH ẢNH VÀ BẢNG

HÌNH 1.1 CÔNG TY VIỆT CHÈO.....	2
HÌNH 1.2 TỔ CHỨC BỘ MÁY NHÂN SỰ CỦA CÔNG TY	4
HÌNH 1.3 CÁC SẢN PHẨM CÔNG TY VIỆT CHÈO.....	5
HÌNH 1.4 CÁC SẢN PHẨM VỀ THANG MÁY CÔNG TY VIỆT CHÈO.....	5
HÌNH 1.5 HỆ THỐNG NÂNG THĂNG ĐỨNG (PCS)	6
HÌNH 1.6 HỆ THỐNG NÂNG TRƯỢT (PSH)	6
HÌNH 1.7 HỆ THỐNG TRƯỢT PHẪNG DÀNG CHO BÃI (PPY)	7
HÌNH 1.8 HỆ THỐNG NÂNG ĐƠN GIẢN (PJS)	8
HÌNH 1.9 THIẾT BỊ BẢO DƯỠNG TÒA NHÀ	9
HÌNH 1.10 DỰ ÁN THÁP BIDV.....	10
HÌNH 1.11 DỰ ÁN MASTERISE HOMES BASON	11
HÌNH 2.1 MẶT BẰNG TIMES SQUARE ĐÀ NẴNG.....	12
HÌNH 2.2 MẶT BẰNG TẦNG HẦM 1 (XE MÁY).....	13
HÌNH 2.3 MẶT BẰNG TẦNG HẦM 2 (XE Ô TÔ VÀ XE MÁY)	13
HÌNH 2.4 MẶT BẰNG ĐỊNH VỊ CỘT MỖI BLOCK HỆ THỐNG ĐỖ XE CƠ KHÍ.....	14
HÌNH 2.5 MẶT ĐỨNG VÀ MẶT CẮT HỆ THỐNG ĐỖ CƠ KHÍ – 2 TẦNG – 3 XE	14
HÌNH 2.6 BẢN VẼ TỔNG QUÁT HỆ THỐNG ĐỖ XE CƠ KHÍ – 2 TẦNG – 3 XE	16
HÌNH 2.7 MẶT ĐỨNG VÀ MẶT CẮT HỆ THỐNG ĐỖ CƠ KHÍ – 2 TẦNG – 5 XE	18
HÌNH 2.8 BẢN VẼ TỔNG QUAN HỆ THỐNG ĐỖ XE CƠ KHÍ – 2 TẦNG – 5 XE.....	20
HÌNH 2.9 BẢNG LED NGOÀI TRỜI DẠNG ĐỨNG HOẶC NGANG	22
HÌNH 2.10 TRẠM PHÂN LOẠI XE TẦNG 1	23
HÌNH 2.11 HÌNH ẢNH MINH HỌA TRẠM PHÂN LOẠI XE.....	24
HÌNH 2.12 MẶT BẰNG BỐ TRÍ BẢNG CHỈ DẪN VỊ TRÍ TRÔNG CỦA CÁC BLOCK	24
HÌNH 2.13 CHI TIẾT LẮP ĐẶT ĐIỂN HÌNH HỆ THỐNG.....	25
HÌNH 2.14 SƠ ĐỒ NGUYÊN LÝ GIÁM SÁT HỆ THỐNG ĐỖ XE CƠ KHÍ	26
HÌNH 2.15 SƠ ĐỒ NGUYÊN LÝ PHÂN LOẠI XE.....	27
HÌNH 2.16 BẢN VẼ BỐ TRÍ CẢM BIẾN VÀ BẢNG LED HIỆN THỊ PHÂN LOẠI XE.....	28
HÌNH 2.17 BỘ ĐIỀU KHIỂN KZ-VRC.NET	29
HÌNH 2.18 CẢM BIẾN HỒNG NGOẠI IR33.....	29
HÌNH 2.19 BẢNG LED HIỆN THỊ SUV/SEDAN.....	31
HÌNH 2.20 SƠ ĐỒ NGUYÊN LÝ CẤP NGUỒN TẠI HẦM 2 – TÒA CT3	31

HÌNH 2.21 SƠ ĐỒ NGUYÊN LÝ CẤP NGUỒN TẠI HẦM 2 – TÒA CT7	33
HÌNH 3.1 CẤU HÌNH HỆ THỐNG KPGS.....	36
HÌNH 3.2 BỘ XỬ LÝ TRUNG TÂM KZ-CCU.XXX.....	36
HÌNH 3.3 VỊ TRÍ LẮP BẮT VÍT VÀ CẤP NGUỒN	37
HÌNH 3.4 VỊ TRÍ KẾT NỐI TÍN HIỆU	38
HÌNH 3.5 NỐI DÂY TÍN HIỆU TRÊN BỘ CHUYỂN ĐỔI.....	38
HÌNH 3.6 GIAO DIỆN PHẦN MỀM BÁO CHỖ TRỐNG KZ-IPGS.....	39
HÌNH 3.7 BỘ ĐIỀU KHIỂN VÙNG K-ZCU	39
HÌNH 3.8 KHOANG LỖ BẮT VÍT	40
HÌNH 3.9 ĐẦU DÂY NGUỒN CHO ZCU	40
HÌNH 3.10 SƠ ĐỒ ĐI DÂY ZCU TỚI CẢM BIẾN.....	41
HÌNH 3.11 CẢM JACK MẠNG CHO CẢM BIẾN	41
HÌNH 3.12 SƠ ĐỒ KẾT NỐI CCU TỚI ZCU VÀ CÁC BẢNG LED	42
HÌNH 3.13 CẢM BIẾN BÁO TRỐNG TẦNG HẦM.....	43
HÌNH 3.14 BẢNG KED ĐƠN TRONG NHÀ KZ-LD.XX.NET	44
HÌNH 3.15 BẢNG LED NGOÀI TRỜI ĐỨNG KZ-LOV.XX	44
HÌNH 3.16 SWITCH IS170-26TF-AC.....	45
HÌNH 3.17 KZ-KPF KIOSK TÌM KIẾM XE VÀ THANH TOÁN.....	46
HÌNH 4.1 KẾT CẤU THÉP	47
HÌNH 4.2 MOTOR NÂNG LIFTING MOTOR PARKS	48
HÌNH 4.3 MOTOR TRƯỢT SLIDING MOTOR PARKS.....	48
HÌNH 4.4 BUFFET LÒ XÒ VÀ GIẢM CHẤN.....	49
HÌNH 4.5 CÔNG TẮT GIỚI HẠN HÀNH TRÌNH HẠ.....	49
HÌNH 4.6 MÓC CHỐNG RƠI	50
HÌNH 4.7 CÔNG TẮC GIỚI HẠN HÀNH TRÌNH NÂNG.....	50
HÌNH 4.8 CÔNG TẮC GIỚI HẠN HÀNH TRÌNH HẠ.....	51
HÌNH 4.9 CÔNG TẮT GIỚI HẠN HÀNH TRÌNH TRƯỢT	51
HÌNH 4.10 CẢM BIẾN XÂM NHẬP HỆ THỐNG.....	52
HÌNH 4.11 THANH CHẶN BÁNH SAU	52
HÌNH 4.12 TỔNG QUAN CỦA TỦ ĐIỀU KHIỂN	53
HÌNH 4.13 PLC OMRON CPM2AH-30CDR.....	53
HÌNH 4.15 MCB 2P BKN D6	56

HÌNH 4.16 CẦU CHỈ GIGA18N-32K	57
HÌNH 4.17 RELAY PHA RM17TE00.....	57
HÌNH 4.18 KHỞI ĐỘNG TỪ TỔNG GMC-12	59
HÌNH 4.19 KHỞI ĐỘNG TỪ LS GMC-6	60
HÌNH 4.20 KHỞI ĐỘNG TỪ LS GMC-9	61
HÌNH 4.21 RƠ LE NHIỆT GTH-22.....	62
HÌNH 4.22 NGUỒN TÔ ONG 24V	63
HÌNH 5.1 SƠ ĐỒ TỔNG THỂ HỆ THỐNG PHẦN MỀM	64
HÌNH 5.2 QUY TRÌNH CỦA HỆ THỐNG	66
HÌNH 5.3 MÔ HÌNH PHÂN LỚP.....	67
HÌNH 5.4 VỊ TRÍ ĐẶT CÁC CẢM BIẾN HỒNG NGOẠI PHÂN LOẠI XE	71
HÌNH 5.5 SƠ ĐỒ CHÂN TRONG ARDUINO MEGA 2560.....	73
HÌNH 5.6 SƠ ĐỒ KẾT NỐI TRÊN MÔ PHÒNG	74
HÌNH 5.7 ĐẦU ĐỌC THẺ RFID (HS0430B).....	75
HÌNH 5.8 SƠ ĐỒ KẾT NỐI TRÊN MÔ PHÒNG	76
HÌNH 5.9 PHẦN MỀM GIẢ LẬP CÔNG COM COM0COM.....	78
HÌNH 5.10 KẾT NỐI ARDUINO MEGA 2560 ĐẾN COM 5.....	79
HÌNH 5.11 BIÊN SỐ XE ĐƯỢC PHÁT HIỆN DÙNG YOLOV10	80
HÌNH 5.12 CÔNG CỤ ROBOFLOW UNIVERSE	81
HÌNH 5.13 GIAO DIỆN GOOGLE COLAB.....	82
HÌNH 5.14 KẾT NỐI GPU CỦA GOOGLE COLAB	82
HÌNH 5.15 CLONE YOLOV10 VỀ MÁY ẢO	83
HÌNH 5.16 CÀI ĐẶT CÁC GÓI THƯ VIỆN	83
HÌNH 5.17 TẢI DATASET TỪ ROBOFLOW.....	83
HÌNH 5.18 GIAO DIỆN GOOGLE COLAB.....	84
HÌNH 5.19 CHỈ SỐ KẾT QUẢ HUẤN LUYỆN	84
HÌNH 5.20 KẾT QUẢ BIÊN SỐ XE ĐỌC ĐƯỢC	86
HÌNH 5.21 GIAO DIỆN CỦA PHẦN MỀM QT DESIGNER	87
HÌNH 5.22 LỆNH CÀI ĐẶT PYQT5	87
HÌNH 5.23 GIAO DIỆN NGƯỜI DÙNG (GUI).....	88
HÌNH 5.24 HÀM KHỞI TẠO	90
HÌNH 5.25 HÀM ĐỌC SERIAL.....	91

HÌNH 5.26 HÀM CẬP NHẬT CAMERA VÀO	92
HÌNH 5.27 HÀM CẬP NHẬT CAMERA RA	93
HÌNH 5.28 HÀM PHÁT HIỆN BIỂN SỐ	94
HÌNH 5.29 HÀM PHÁT NHẬN DẠNG KÍ TỰ	95
HÌNH 5.30 HÀM LƯU TRỮ DỮ LIỆU.....	95
HÌNH 5.31 HÀM XEM DỮ LIỆU	96
HÌNH 5.32 HÀM XUẤT DỮ LIỆU	98
HÌNH 5.33 HÀM XỬ LÝ XE VÀO THỦ CÔNG	98
HÌNH 5.34 HÀM XỬ LÝ XE RA THỦ CÔNG	99
HÌNH 5.35 ĐỒ THỊ ĐỘ TỰ TIN VÀ CHÍNH XÁC	100
HÌNH 5.36 GIAO DIỆN PHẦN MỀM KHI VỪA KHỞI ĐỘNG	101
HÌNH 5.37 NHẬP RFID VÀ CẢM BIẾN SENSOR 1 VÀ SENSOR 2 KÍCH HOẠT	101
HÌNH 5.38 KẾT QUẢ NHẬN ĐƯỢC TRÊN GIAO DIỆN	102
HÌNH 5.39 BARRIER IN MỞ RA.....	103
HÌNH 5.40 NHẬP RFID LÚC XE RA.....	103
HÌNH 5.41 KẾT QUẢ NHẬN ĐƯỢC KHI XE RA	104
HÌNH 5.42 BARRIER OUT MỞ RA CHO XE RA.....	105
HÌNH 5.43 XEM DỮ LIỆU ĐÃ LƯU	105
HÌNH 5.44 XUẤT FILE EXCEL	106
HÌNH 5.45 GIAO DIỆN XE VÀO Ở CHẾ ĐỘ BẰNG TAY.....	107
HÌNH 5.46 GIAO DIỆN XE RA Ở CHẾ ĐỘ BẰNG TAY	108
HÌNH 5.47 LỖI NHẬP CHƯA ĐẦY ĐỦ THÔNG TIN CỦA XE	109
HÌNH 5.48 LỖI NHẬP SAI DỮ LIỆU XE VÀO.....	109
HÌNH 5.49 CẢNH BÁO KHÔNG CÓ DỮ LIỆU VÀO CỦA XE TRONG DỮ LIỆU.....	110
HÌNH 5.50 CẢNH BÁO BIỂN SỐ ĐÃ CÓ TRONG DỮ LIỆU	110
BẢNG 1.1 HỆ THỐNG NÂNG THẲNG ĐỨNG (PCS).....	6
BẢNG 1.2 HỆ THỐNG NÂNG TRƯỢT (PSH)	7
BẢNG 1.3 HỆ THỐNG TRƯỢT PHẪNG DÀNH CHO BÃI (PPY).....	7
BẢNG 1.4 HỆ THỐNG NÂNG ĐƠN GIẢN (PJS).....	8
BẢNG 2.1 THÔNG SỐ KỸ THUẬT HỆ THỐNG ĐỖ XE CƠ KHÍ – 2 TẦNG – 3 XE	16
BẢNG 2.2 THÔNG SỐ KỸ THUẬT TỔNG QUAN	17

BẢNG 2.3 THÔNG SỐ LỰC TÁC DỤNG	17
BẢNG 2.4 THÔNG SỐ KỸ THUẬT HỆ THỐNG ĐỖ XE CƠ KHÍ – 2 TẦNG – 5 XE	19
BẢNG 2.5 THÔNG SỐ KỸ THUẬT TỔNG QUAN	21
BẢNG 2.6 THÔNG SỐ LỰC TÁC DỤNG	21
BẢNG 2.7 THÔNG SỐ KỸ THUẬT ĐÈN LED NGOÀI TRỜI	22
BẢNG 2.8 THÔNG SỐ KỸ THUẬT BỘ ĐIỀU KHIỂN KZ-VRC.NET	29
BẢNG 2.9 THÔNG SỐ KỸ THUẬT CỦA CẢM BIẾN HỒNG NGOÀI IR33	30
BẢNG 3.1 THÔNG SỐ KỸ THUẬT CỦA KZ-CCU.01M.....	37
BẢNG 3.2 THÔNG SỐ KỸ THUẬT CỦA KZ-ZCU.CL24.NET.....	39
BẢNG 3.3 THÔNG SỐ KỸ THUẬT CẢM BIẾN BÁO CHỖ TRỐNG KZ-CL.01.....	43
BẢNG 3.4 THÔNG SỐ KỸ THUẬT LED ĐƠN TRONG NHÀ LZ-LD.04.NET	44
BẢNG 3.5 THÔNG SỐ KỸ THUẬT CỦA KZ.LOV.XX.NET.....	45
BẢNG 3.6 THÔNG SỐ KỸ THUẬT CỦA SWITCH IS170-26TF-AC.....	46
BẢNG 4.1 THÔNG SỐ KỸ THUẬT CPM2AH-30CDR.....	54
BẢNG 4.2 THÔNG SỐ KỸ THUẬT CỦA MCB 3P BKN-D20.....	55
BẢNG 4.3 THÔNG SỐ KỸ THUẬT CỦA MCB 2P BKN-D6.....	56
BẢNG 4.4 THÔNG SỐ KỸ THUẬT CỦA GIGA18N-32K	57
BẢNG 4.5 THÔNG SỐ KỸ THUẬT CỦA RELAY PHA RM17TE00.....	58
BẢNG 4.6 THÔNG SỐ KỸ THUẬT CỦA LS GMC-12	59
BẢNG 4.7 THÔNG SỐ KỸ THUẬT CỦA LS GMC-6	60
BẢNG 4.8 THÔNG SỐ KỸ THUẬT CỦA LS GMC-9	61
BẢNG 4.9 THÔNG SỐ KỸ THUẬT CỦA GTH-22.....	62
BẢNG 4.10 THÔNG SỐ KỸ THUẬT NGUỒN NUÔI PLC.....	63
BẢNG 5.1 BẢNG CÁC NGUỒN PHÂN LOẠI XE	71
BẢNG 5.2 BẢNG KẾT NỐI CHÂN TÍN HIỆU CỦA IR VỚI ARDUINO MEGA 2560.....	74
BẢNG 5.3 KẾT NỐI RFID VỚI ARDUINO MEGA 2560	76
BẢNG 5.4 CÁC HẠN CHẾ CỦA HỆ THỐNG.....	111
BẢNG 5.5 ĐỀ XUẤT CẢI TIẾN.....	112

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

KÍ HIỆU:

- KZ-VRC.NET: BỘ ĐIỀU KHIỂN PHÂN LOẠI XE
- KZ-CCU.xx: BỘ ĐIỀU KHIỂN TRUNG TÂM (CENTER CONTROL UNIT)
- KZ-ZCU.xx: BỘ ĐIỀU KHIỂN VÙNG (ZONE CONTROL UNIT)
- KPGS: HỆ THỐNG HƯỚNG DẪN ĐỖ XE (PARKING GUIDANCE SYSTEM)
- KZ-LD: BẢNG LED ĐƠN TRONG NHÀ
- KZ-LOV: BẢNG LED NGOÀI TRỜI
- KZ-KPF: KIOSK TÌM KIẾM VÀ THANH TOÁN
- KZ-iPGS: PHẦN MỀM BÁO SỐ CHỖ TRỐNG
- KZ-CL: CẢM BIẾN BÁO CHỖ TRỐNG

CHỮ VIẾT TẮT:

- SEDAN: XE 4 CHỖ
- SUV: XE 7 CHỖ
- OVERSIZE: XE QUÁ KHỔ
- GUI: GIAO DIỆN PHẦN MỀM (GRAPHICAL USER INTERFACE)
- UID: CHUỖI KÍ TỰ (UNIQUE IDENTIFIER)
- OCR: KÝ TỰ QUANG HỌC (OPTICAL CHARACTER RECOGNITION)
- GPU: BỘ XỬ LÝ ĐỒ HỌA (GRAPHIC PROCESSING UNIT)
- TPU: BỘ XỬ LÝ TENSOR (TENSOR PROCESSING UNIT)
- CNN: MẠNG NƠ-RON TÍCH CHẬP (CONVOLUTIONAL NEURAL NETWORK)

CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ ĐƠN VỊ DOANH NGHIỆP

1.1. Giới thiệu về doanh nghiệp

Công ty Cổ phần Việt Chèo được thành lập từ năm 2004, trải qua hơn **20 năm** phát triển và trưởng thành, với sự đoàn kết, nhiệt huyết của từng cán bộ công ty, dưới sự lãnh đạo tâm huyết của ban lãnh đạo, đến nay Công ty Cổ phần Việt Chèo đã phát triển thành một nhà thầu chuyên nghiệp trong lĩnh vực tư vấn, cung cấp, lắp đặt, bảo trì, bảo dưỡng các loại thiết bị cho các công trình.

Qua nhiều năm hoạt động, Việt Chèo đã xây dựng được mối quan hệ hợp tác chiến lược với các nhà cung cấp có thương hiệu lớn trên thế giới. Đến nay, để đáp ứng nhu cầu của thị trường, bên cạnh ngành nghề kinh doanh chính là cung cấp và lắp đặt thang máy – thang cuốn thương hiệu **Shanghai Mitsubishi** (thuộc Tập đoàn thang máy Mitsubishi Nhật Bản), thang máy gia đình LiftingItalia & Arealift (Ý), MP (Tây Ban Nha)..., Việt Chèo đã mở rộng sang các lĩnh vực thiết bị công trình khác như:

- Hệ thống đỗ xe tự động: XIZI (Trung Quốc), Dongyang Park (Hàn Quốc)
- Thiết bị bảo dưỡng tòa nhà (BMU): GEDA (Thổ Nhĩ Kỳ), CHI (Trung Quốc)
- Điều hòa không khí: Daikin, Mitsubishi (Nhật Bản), Carrier (Mỹ), Midea (Trung Quốc)
- Robot/xe cứu hỏa: ITM (Đức), BAI (Ý)
- Máy phát điện: Cummins (Mỹ), Mitsubishi (Nhật Bản), FG Wilson (Anh), Onis Visa (Ý)

Công ty hoạt động tại Việt Nam với trụ sở chính đặt ở tầng 1+2, số 61/66 - Triều Khúc - Tân Triều - Thanh Trì - Tp. Hà Nội. Ngoài ra, công ty còn có các cơ sở ở Số 85 - Đ. Nguyễn Xuân Ôn - P. Hòa Cường Bắc - Quận Hải Châu - Tp Đà Nẵng; Số 01 - Đường Lê Doãn Sửu - Phường Hà Huy Tập - Tp Vinh - Nghệ An và Số 42 Trung Mỹ Tây 1- Phường Trung Mỹ Tây - Quận 12 - Tp. HCM

Số điện thoại: 0898 552 888

Website: <https://vietchao.vn/>



Hình 1.1 Công ty Việt Chào

Chứng nhận của công ty

- Việt Chào tiếp tục được hãng Shanghai Mitsubishi gia hạn Hợp đồng đại lý ủy quyền tại Việt Nam.
- Việt Chào sẽ tiếp tục là đại lý DUY NHẤT được Mitsubishi Thương Hải ủy quyền tại Việt Nam.

Tập thể cán bộ nhân viên Công ty Cổ phần Việt Chào đồng lòng hướng tới TÂM NHÌN – SỨ MỆNH – GIÁ TRỊ CỐT LÕI – MỤC TIÊU sau:

TÂM NHÌN

Luôn đặt chữ **TÍN** lên hàng đầu trong mọi hoạt động kinh doanh với khách hàng và đối tác, liên tục cập nhật các công nghệ tân tiến nhất và tìm kiếm các đối tác uy tín trên thế giới, Công ty Cổ phần Việt Chào hướng tới trở thành Top 3 công ty cung cấp các thiết bị tòa nhà như thang máy, thang cuốn, bãi đỗ xe tự động, thiết bị vệ sinh tòa nhà ... hàng đầu Việt Nam vào năm 2030.

SỨ MỆNH

Nỗ lực không ngừng để mang đến cho người Việt Nam những sản phẩm có chất lượng xứng đáng với giá thành từ các nhà sản xuất có uy tín trên thế giới.

GIÁ TRỊ CỐT LÕI

Chất lượng – Hiệu quả: Yếu tố cốt lõi làm nên thương hiệu Việt Chào là chất lượng gắn liền với hiệu quả tối ưu trong mọi hoạt động và trên mọi phương diện. Việt Chào cam kết chỉ cung cấp sản phẩm và dịch vụ có chất lượng, chính hãng, kiên quyết nói không với hàng trôi nổi, kém chất lượng, không giảm giá bằng cách giảm chất lượng.

Trung thực – Trách nhiệm: Việt Chào luôn đề cao tính trung thực: trung thực với lời tư vấn khách hàng, trung thực với hành động của mình. Trong bối cảnh thị trường thang máy còn đang bị bùng nổ quản lý, Việt Chào quyết định xây uy tín theo cách riêng, không thỏa hiệp với việc nhập nhèm nguồn gốc xuất xứ sản phẩm và giả mạo logo.

Chia sẻ – Đoàn kết: Sự đoàn kết, sẵn lòng chia sẻ thông tin, kiến thức, kinh nghiệm, nhiệm vụ và phát huy trí tuệ tập thể là yếu tố quan trọng đem lại thành công của Việt Chào ngày hôm nay. Mỗi thành viên là một nhân tố quan trọng của Việt Chào, nhưng Việt Chào chỉ có thể lớn mạnh nếu mỗi thành viên đó hợp tác, kết dính với nhau để tạo thành một khối vững chắc.

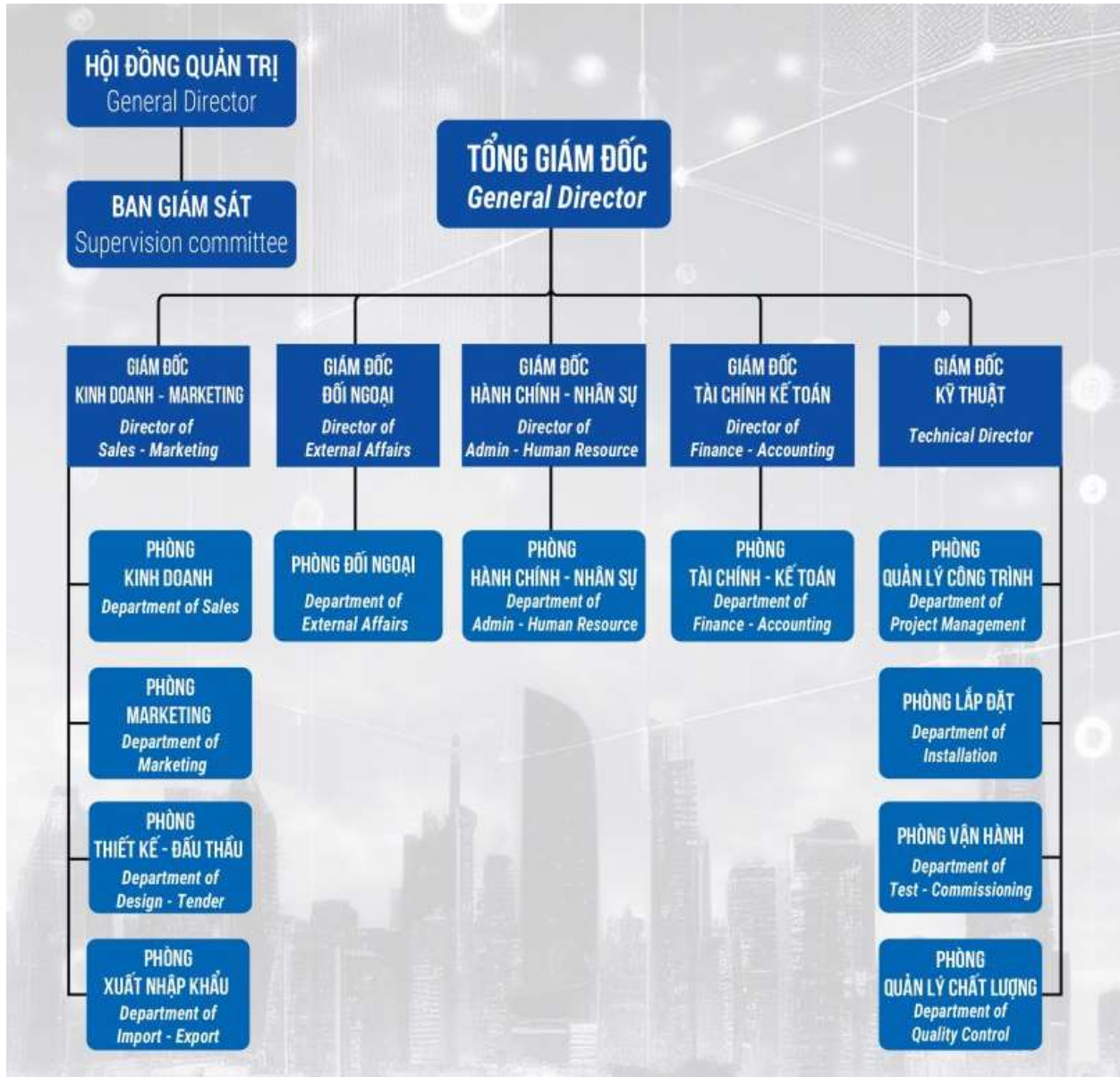
MỤC TIÊU

Phối hợp chặt chẽ với đối tác, nhà sản xuất có uy tín trên thế giới để liên tục cải tiến công nghệ, đáp ứng nhu cầu ngày càng cao của khách hàng, tối ưu hóa quy trình sản xuất nhằm giảm giá thành, hướng tới việc cung cấp sản phẩm chất lượng với giá thành hợp lý.

1.2. Tổ chức bộ máy nhân sự của công ty:

Việt Chào có đội ngũ nhân viên chuyên nghiệp với hơn 80 % là kỹ sư và nhân viên kỹ thuật, những người hằng ngày tham gia dự án, cung cấp dịch vụ kỹ thuật cho khách hàng hằng ngày.

Để trở thành nhà cung cấp giải pháp được chứng nhận của Shanghai Mitsubishi, các kỹ sư của Việt Chào đã tham gia các khóa đào tạo về chuyên môn tại Mitsubishi để lấy chứng chỉ. Không dừng lại ở đây, các kỹ sư của Việt Chào sau mỗi 2 năm sẽ phải ôn luyện và tham gia lại kỳ thi kiểm tra năng lực nhằm giữ danh hiệu nhà cung cấp giải pháp được chứng nhận.



Hình 1.2 Tổ chức bộ máy nhân sự của công ty

1.3. Sản phẩm

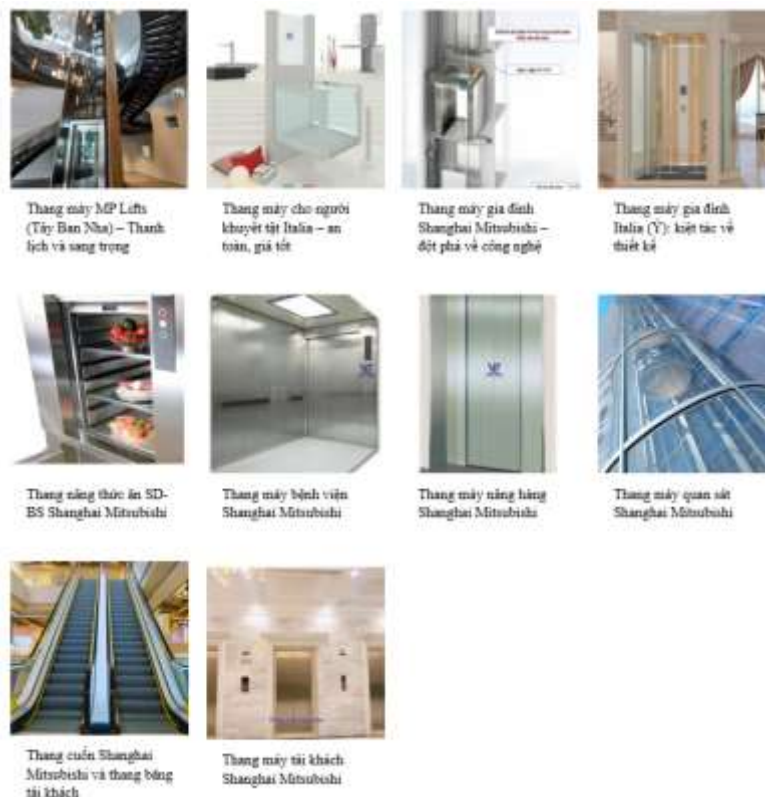




Hình 1.3 Các sản phẩm công ty Việt Chèo

➤ Về sản phẩm Thang máy – Thang cuốn

Với gần 20 năm kinh nghiệm trong ngành thang máy, thang cuốn, sản phẩm do Việt Chèo cung cấp đã được khách hàng cả nước đón nhận. Hai dòng sản phẩm chính của Việt Chèo được nhập từ các công ty thang máy có uy tín thế giới là Công ty thang máy Shanghai Mitsubishi (SMEC) và Công ty thang máy gia đình cao cấp Arealift/LiftingItalia chuyên về thang máy gia đình cao cấp và Arealift, chuyên thang máy sàn nâng và thang cho người khuyết tật. Sản phẩm của công ty đã có mặt trên 50 quốc gia.

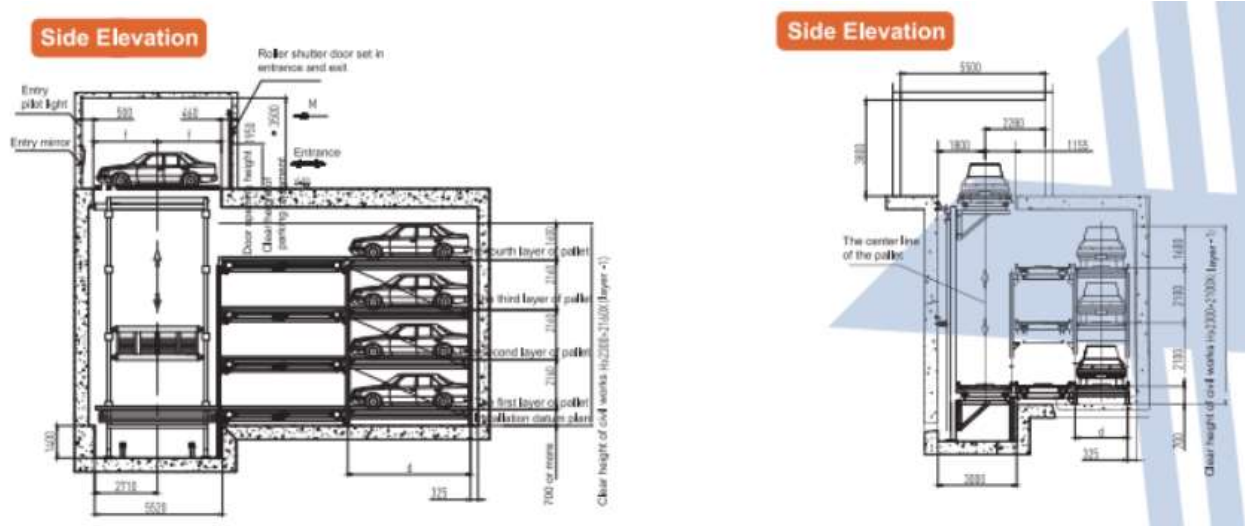


Hình 1.4 Các sản phẩm về thang máy công ty Việt Chèo

Hệ thống	Chiều dài xe (mm)	Chiều rộng xe (mm)	Tốc độ (m/p)	Khoảng cách giữa 2 lớp xe(mm)	Chiều cao(mm)	Tải trọng(kg)
PSH	4700~5300	1800~1950	4~8	1760~1900	1550	1500~2350

Bảng 1.2 Hệ thống nâng trượt (PSH)

- Hệ thống trượt phẳng dành cho bãi Parking Platform Y (PPY)

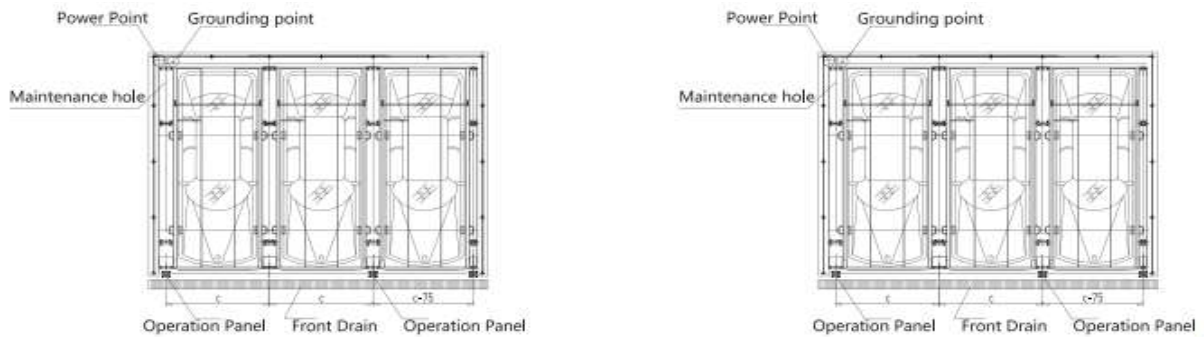


Hình 1.7 Hệ thống trượt phẳng dành cho bãi (PPY)

Hệ thống	Chiều dài xe (mm)	Chiều rộng xe (mm)	Tốc độ (m/s)	Số tầng	Nguồn	Tải trọng(kg)
PPY	5300	1900	50	5	18.5	1500~2350

Bảng 1.3 Hệ thống trượt phẳng dành cho bãi (PPY)

- Hệ thống nâng đơn giản Puzzle Sliding System (PJS)



Hình 1.8 Hệ thống nâng đơn giản (PJS)

Hệ thống	Chiều dài xe (mm)	Chiều rộng xe (mm)	Tốc độ (m/s)	Khoảng cách giữa 2 lớp xe(mm)	Chiều cao (mm)	Tải trọng(kg)
PJS	4700~5300	1800~1950	4~5	5	18.5	1500~2350

Bảng 1.4 Hệ thống nâng đơn giản (PJS)

➤ Điều hòa không khí

Công ty Cổ phần Việt Chà còn là đại lý phân phối cấp 1 sản phẩm của các hãng điều hòa danh tiếng trên thế giới như TOSHIBA (CARRIER), PANASONIC, YORK, SAMSUNG, HITACHI ...

Hệ thống điều hòa VRF với các dải công suất dàn nóng linh hoạt (tăng hoặc giảm 2HP) với công suất dàn nóng cao nhất lên tới 48HP, thuận tiện cho việc lựa chọn công suất hệ thống một cách linh hoạt. Giảm thiểu chi phí đầu tư ban đầu.

Với các chủng loại dàn lạnh đa dạng từ treo tường, cassette, âm trần nổi ống gió ... phù hợp với mọi mặt bằng kiến trúc cũng như đáp ứng mọi công năng sử dụng.

➤ Thiết bị bảo dưỡng tòa nhà

Công ty Cổ Phần Việt Chà tự hào là nhà phân phối thiết bị bảo dưỡng tòa nhà (BMU) của các hãng nổi tiếng thế giới như GEDA (Đức), Skypro (Mỹ). BMU đem lại cho công trình một giải pháp vệ sinh bề mặt kính, mặt ốp nhôm của tòa nhà một cách an toàn, hữu hiệu và BMU như một tác phẩm nghệ thuật xứng tầm với tòa nhà hiện đại mà nó chăm sóc.



Hình 1.9 Thiết bị bảo dưỡng tòa nhà

1.4. Một số dự án tiêu biểu của Việt Chèo

Dự án tháp BIDV 194 Trần Quang Khải, Hà Nội

Chủng loại: Bán tự động

Model: PSH2-XIZI

Số lượng: 95 xe

Được đầu tư xây dựng bởi Công Ty Liên Doanh Tháp BIDV, Tháp BIDV là công trình được xây dựng trên cơ sở liên doanh giữa Ngân hàng Thương mại Cổ phần (TMCP) Đầu tư và Phát triển Việt Nam (BIDV) và Công ty Bloomhill Holding (Singapore). Được thiết kế bởi Công ty Tư vấn thiết kế P&T (Singapore), Tháp BIDV đã được Hội đồng bình chọn Giải thưởng Xuất sắc về Thiết kế và An toàn kỹ thuật năm 2010 của Singapore trao tặng giải thưởng dành cho các công trình ở nước ngoài.



Hình 1.10 Dự án tháp BIDV

Dự án MASTERISE HOMES BASON, Q1, TP.HCM

Chủng loại: Bán tự động

Model: PSH

Số lượng: 90 xe

Dự án Masterise Homes là một trong những dự án được xem là biểu tượng của kiến trúc đẳng cấp tại khu vực. Dự án là khu căn hộ Bất Động Sản Hàng Hiệu mang thương hiệu Marriott và JW Marriott đầu tiên tại Việt Nam, góp phần tạo dựng quan trọng trong quá trình phát triển đất nước. Việt Chèo dù không phải là đơn vị chào thầu giá thấp nhất, nhưng vẫn là đơn vị được Chủ đầu tư lựa chọn bởi những ưu thế vượt trội mà khó có đối thủ nào có thể sánh bằng. Đây là một minh chứng sinh động cho thấy, trong các dự án lớn, không phải lựa chọn với chi phí đầu tư ban đầu thấp nhất mới là lựa chọn tối ưu, mà chất lượng của sản phẩm và độ tin cậy của nhà thầu mới là yếu tố quyết định.



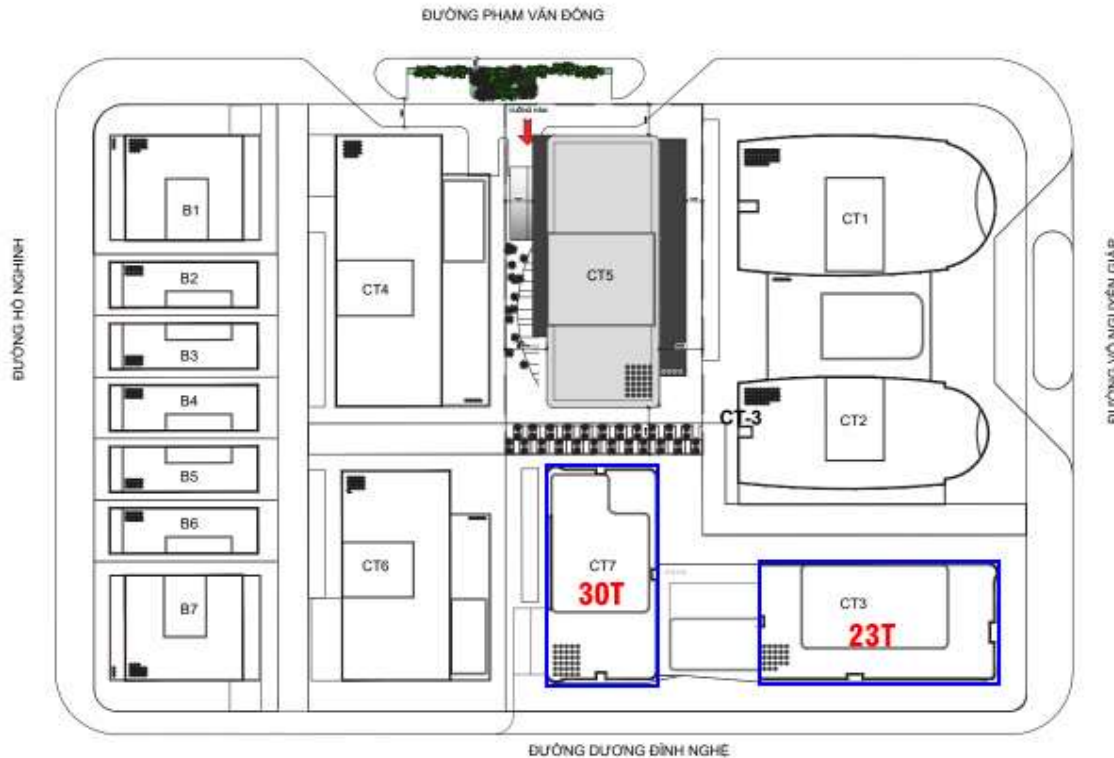
Hình 1.11 Dự án MASTERISE HOMES BASON

CHƯƠNG 2: TỔNG QUAN DỰ ÁN

Tên dự án: Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Chào, Hà Nội.

Vị trí thiết kế: Tầng hầm tòa CT3, CT7

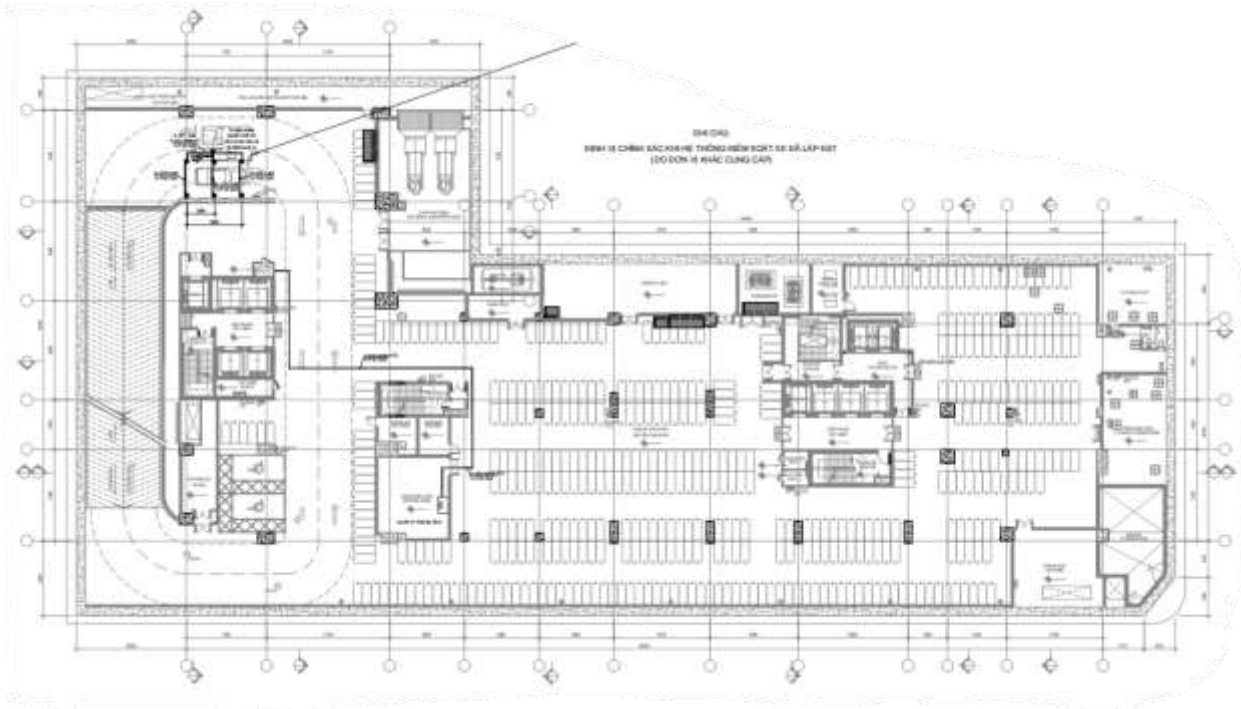
2.1. Sơ đồ bố trí mặt bằng tòa nhà Times Square Đà Nẵng



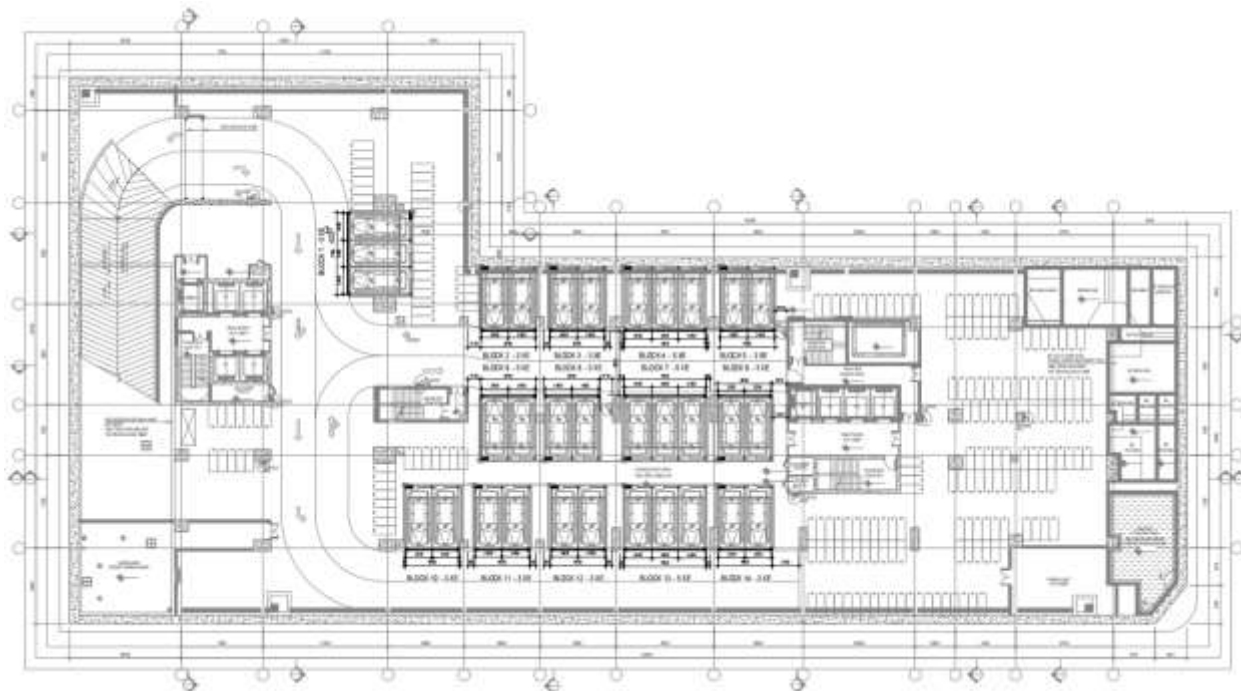
Hình 2.1 Mặt bằng Times Square Đà Nẵng

Sơ đồ mặt bằng trên có 2 khu bãi đỗ xe gồm CT3-CT7 mỗi khu đều có tầng hầm thông nhau, tầng hầm 1: 101 chỗ đỗ xe và tầng hầm 2: 198 chỗ đỗ xe trong đó có 50 chỗ đỗ xe ô tô ở tầng 2 còn lại là 249 chỗ đỗ xe cho xe máy nằm cả 2 tầng 1,2. 50 chỗ đỗ xe ô tô cho xe SUV và SEDAN được chia thành 14 block, trong đó có 4 block có 5 mâm và 9 block 3 mâm.

Sơ đồ mặt bằng tầng hầm CT3-CT7



Hình 2.2 Mặt bằng tầng hầm 1 (xe máy)



Hình 2.3 Mặt bằng tầng hầm 2 (xe ô tô và xe máy)

Cấu trúc tổng thể:

- Loại hệ thống: Hệ thống nâng trượt Puzzle Stacker Horizontal (PSH).
- Hệ thống có 2 tầng, với tầng trên sử dụng pallet di chuyển để nâng/hạ xe.
- Mỗi tầng có một hàng chứa xe (Single Row).
- Tổng sức chứa của hệ thống là 3 xe.

Các thành phần chính:

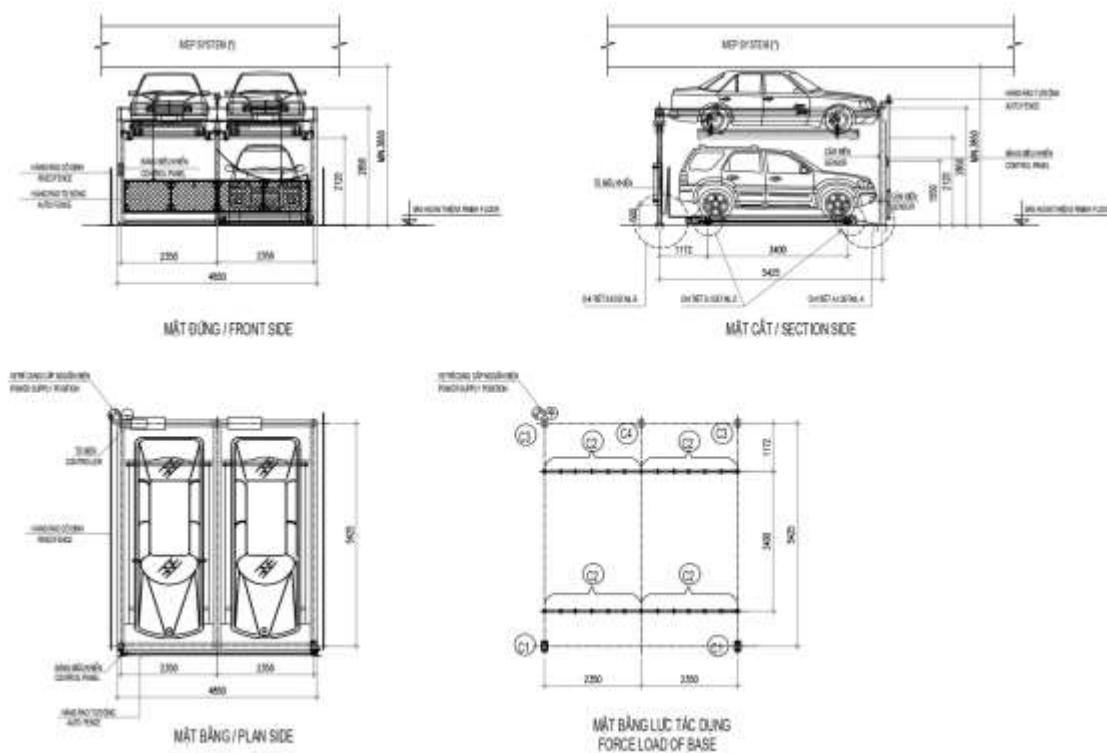
- Pallet: Dùng để đặt xe lên, có thể di chuyển nâng/hạ.
- Dầm chịu lực: Gồm dầm trước, dầm sau, dầm đỡ pallet giúp cố định hệ thống.
- Cột trước và cột sau: Hỗ trợ cấu trúc, đảm bảo an toàn.
- Cơ cấu nâng (Động cơ nâng, động cơ trượt):
 - Động cơ nâng giúp đưa pallet từ tầng dưới lên tầng trên và ngược lại.
 - Động cơ trượt hỗ trợ di chuyển pallet sang trái/phải để tối ưu hóa không gian.
- Hệ thống bảo vệ:
 - Hàng rào bảo vệ xung quanh giúp an toàn khi vận hành.
 - Sàn hoàn thiện đảm bảo nền vững chắc.

Thông số kỹ thuật

Material/vật liệu	Specification (thông số kỹ thuật của hệ thống)	
Lift motor/động cơ nâng	2.2 KW	
Slide motor/động cơ trượt	0.2 KW	
Front column/cột trước	150*150*4	
Rear column/cột sau	H100*100*6*8	
Front beam/dầm trước	H298x149*5.5*8	
Rear beam/dầm sau	H100*100*6*8	
Side beam/dầm bên	H248*124*5*8	
Platform/sàn nâng	Level 1 / tầng 1	Mạ kẽm, dày 2.5 mm
	Level 2 / tầng 2	Mạ kẽm, dày 2 mm

Platform beam/dầm sàn nâng	Level 1 / tầng 1	Mạ kẽm, dày 4 mm
	Level 2 / tầng 2	Mạ kẽm, dày 3 mm
Fixed slop plate/tấm dốc cố định	Thép gân, dày 3mm	
Finish surface/bề mặt hoàn thiện	Hot-dip galvanization / mạ kẽm nhúng nóng	

Bảng 2.1 Thông số kỹ thuật hệ thống đỗ xe cơ khí – 2 tầng – 3 xe



Hình 2.6 Bản vẽ tổng quát hệ thống đỗ xe cơ khí – 2 tầng – 3 xe

Bảng thông số kỹ thuật hệ thống / specification table		
Kích thước xe CAR SIZE	Loại / type	D
	Chiều dài / length	5000 mm
	Chiều rộng / width	1850 mm
	Chiều rộng / witch (bao gồm gương)	1950 mm
	Chiều cao / height	LEVEL 1 / tầng 1

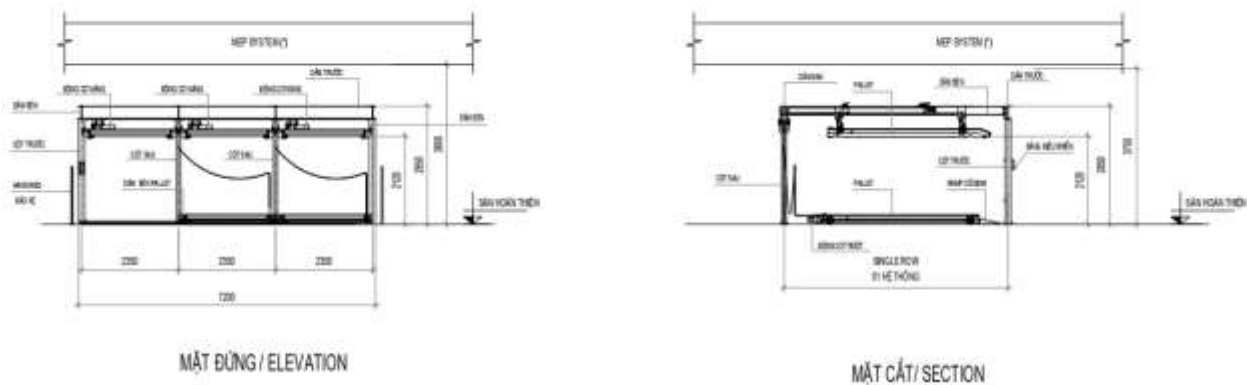
		LEVEL 2 / tầng 2	1550 – SEDAN mm
	Cân nặng / weight	LEVEL 1 / tầng 1	2600 kg
		LEVEL 2 / tầng 2	2000 kg
Tốc độ nâng / lift speed			4.0~5.0 m/min
Tốc độ trượt / slide speed			7.0~8.0 m/min
Chế độ nâng / lift drive mode			Motor - xích / motor - chain
Chế độ trượt / lift drive mode			Motor - xích / motor - chain
Motor nâng / lift motor			2.2 KW
Motor trượt / slide motor			0.2 KW
Công suất / power capacity			4 KVA
Nguồn điện / power			AC380V 50Hz 3 phase
Phương thức vận hành / operation method			Nút bấm + Thẻ từ / button + card reader
Cách thức đỗ xe / parking mode			Lùi xe vào, Đi xe ra / back in, forward out
Chiều rộng hệ thống / system width			C*N+150
Số xe đỗ / car lots			3

Bảng 2.2 Thông số kỹ thuật tổng quan

Bảng thông số lực tác dụng (LOAD PARAMETER TABLE)				
PILLAR	C1	C2	C3	C4
LOAD(N)	32360	13720	10460	20920

Bảng 2.3 Thông số lực tác dụng

2.2.3. Mô tả hệ thống đỗ xe cơ khí – 2 tầng – 5 xe



Hình 2.7 Mặt đứng và mặt cắt hệ thống đỗ xe cơ khí – 2 tầng – 5 xe

Cấu trúc tổng thể:

- Loại hệ thống: Hệ thống nâng trượt Puzzle Stacker Horizontal (PSH)
- Hệ thống có 2 tầng, với tầng trên sử dụng pallet di chuyển để nâng/hạ xe.
- Mỗi tầng có một hàng chứa xe (Single Row).
- Tổng sức chứa của hệ thống là 5 xe.

Các thành phần chính:

- Pallet: Dùng để đặt xe lên, có thể di chuyển nâng/hạ.
- Dầm chịu lực: Gồm dầm trước, dầm sau, dầm đỡ pallet giúp cố định hệ thống.
- Cột trước và cột sau: Hỗ trợ cấu trúc, đảm bảo an toàn.
- Cơ cấu nâng (Động cơ nâng, động cơ trượt):
 - Động cơ nâng giúp đưa pallet từ tầng dưới lên tầng trên và ngược lại.
 - Động cơ trượt hỗ trợ di chuyển pallet sang trái/phải để tối ưu hóa không gian.
- Hệ thống bảo vệ:
 - Hàng rào bảo vệ xung quanh giúp an toàn khi vận hành.
 - Sàn hoàn thiện đảm bảo nền vững chắc.

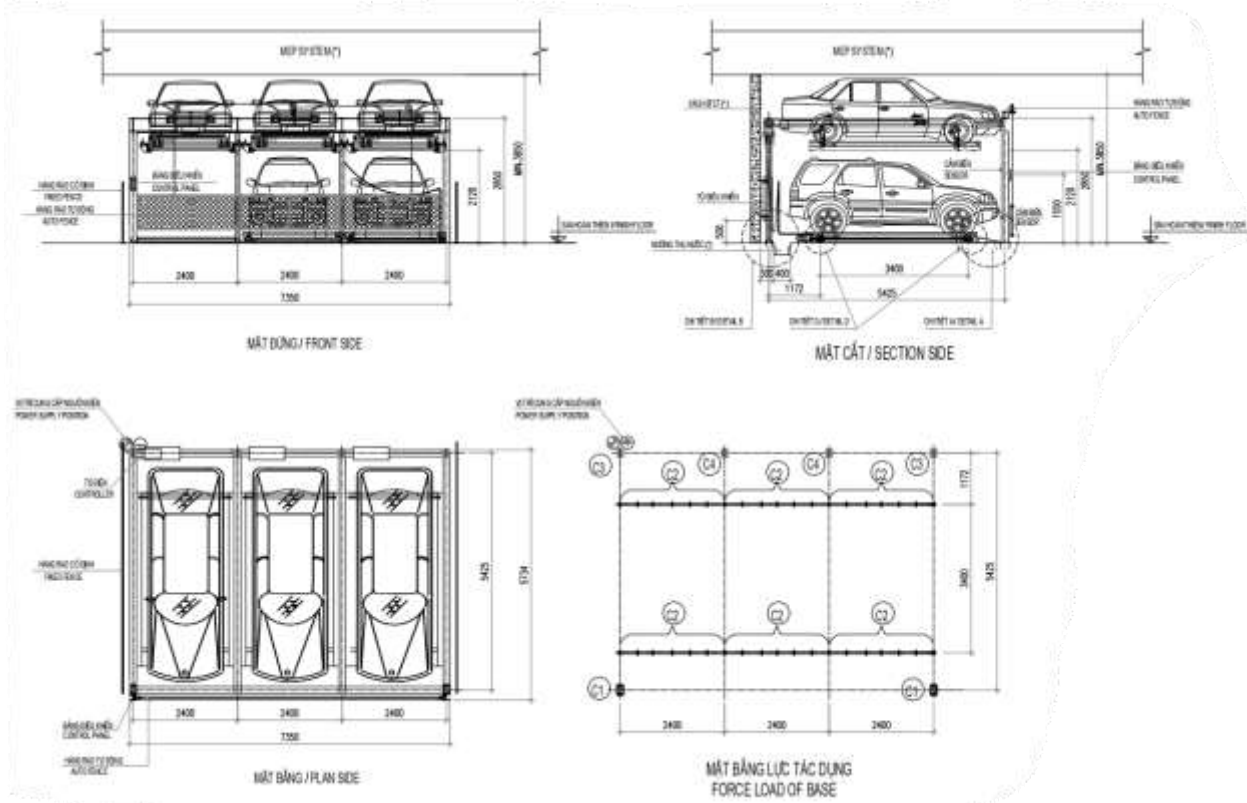
Thông số kỹ thuật

Material/vật liệu	Specification (thông số kỹ thuật của hệ thống)
Lift motor/động cơ nâng	2.2 KW
Slide motor/động cơ trượt	0.2 KW

Front column/cột trước		150*150*4
Rear column/cột sau		H100*100*6*8
Front beam/dầm trước		H298x149*5.5*8
Rear beam/dầm sau		H100*100*6*8
Side beam/dầm bên		H248*124*5*8
Platform/sàn nâng	Level 1 / tầng 1	Mạ kẽm, dày 2.5 mm
	Level 2 / tầng 2	Mạ kẽm, dày 2 mm
Platform beam/dầm sàn nâng	Level 1 / tầng 1	Mạ kẽm, dày 4 mm
	Level 2 / tầng 2	Mạ kẽm, dày 3 mm
Fixed slop plate/tấm dốc cố định		Thép gân, dày 3mm
Finish surface/bề mặt hoàn thiện		Hot-dip galvanization / mạ kẽm nhúng nóng

Bảng 2.4 Thông số kỹ thuật hệ thống đỗ xe cơ khí – 2 tầng – 5 xe

Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Trì, Hà Nội



Hình 2.8 Bản vẽ tổng quan hệ thống đỗ xe cơ khí – 2 tầng – 5 xe

Bảng thông số kỹ thuật hệ thống / specification table			
Kích thước xe CAR SIZE	Loại / type		D
	Chiều dài / length		5000 mm
	Chiều rộng / width		1850 mm
	Chiều rộng / witch (bao gồm gương)		1950 mm
	Chiều cao / height	LEVEL 1 / tầng 1	1900 – SUV mm
		LEVEL 2 / tầng 2	1550 – SEDAN mm
Cân nặng / weight	LEVEL 1 / tầng 1	2600 kg	
	LEVEL 2 / tầng 2	2000 kg	
Tốc độ nâng / lift speed		4.0~5.0 m/min	

Tốc độ trượt / slide speed	7.0~8.0 m/min
Chế độ nâng / lift drive mode	Motor - xích / motor - chain
Chế độ trượt / lift drive mode	Motor - xích / motor - chain
Motor nâng / lift motor	2.2 KW
Motor trượt / slide motor	0.2 KW
Công suất / power capacity	4 KVA
Nguồn điện / power	AC380V 50Hz 3 phase
Phương thức vận hành / operation method	Nút bấm + Thẻ từ / button + card reader
Cách thức đỗ xe / parking mode	Lùi xe vào, Đi xe ra / back in, forward out
Chiều rộng hệ thống / system width	C*N+150
Số xe đỗ / car lots	5

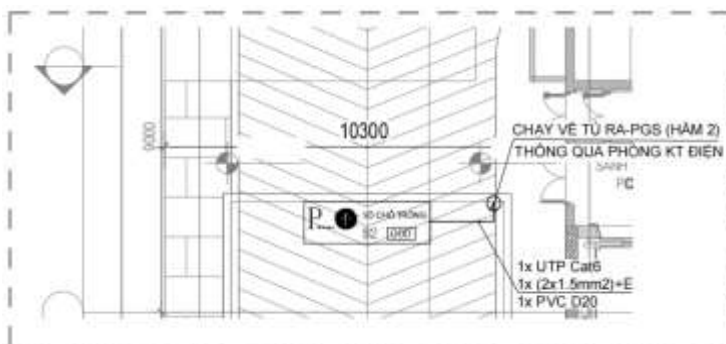
Bảng 2.5 Thông số kỹ thuật tổng quan

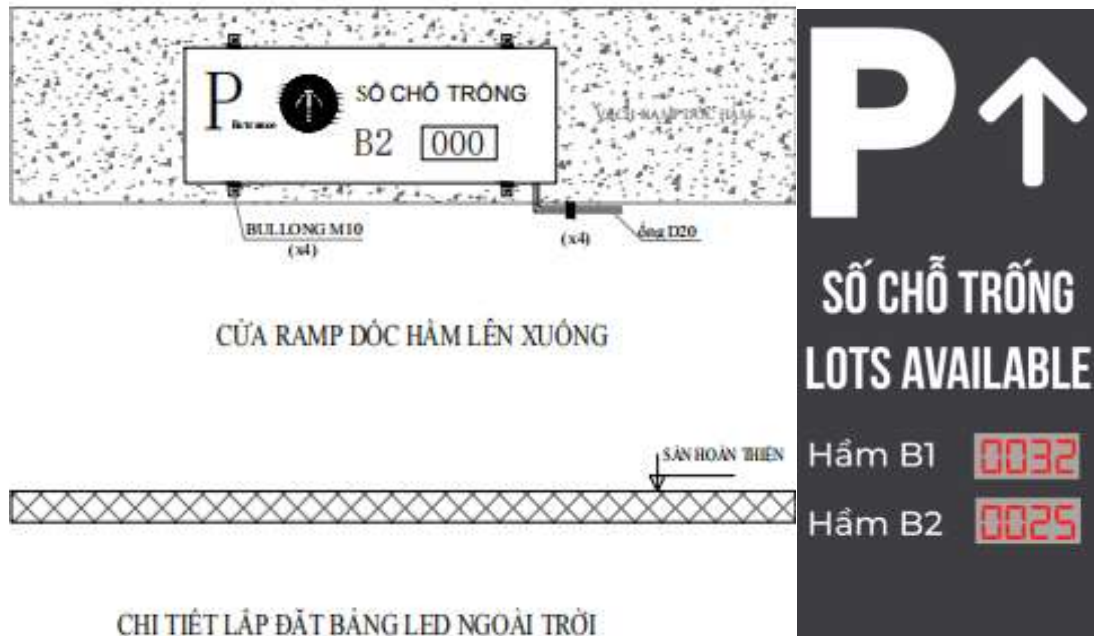
Bảng thông số lực tác dụng (LOAD PARAMETER TABLE)				
PILLAR	C1	C2	C3	C4
LOAD(N)	32360	13720	10460	20920

Bảng 2.6 Thông số lực tác dụng

2.3. Hệ thống chỉ dẫn, báo trống, phân loại và giám sát đỗ xe cơ khí

Bảng led ngoài trời





Hình 2.9 Bảng led ngoài trời dạng đứng hoặc ngang

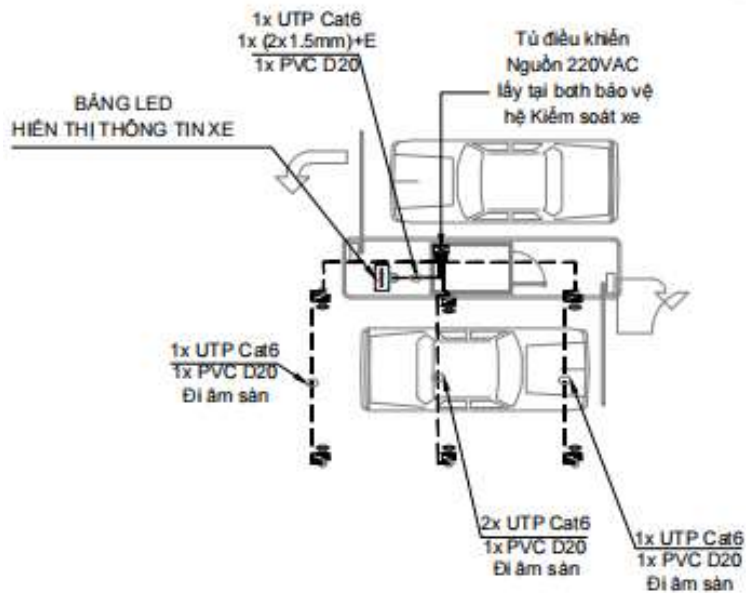
- Hiện thị số lượng chỗ đỗ trống tại mỗi vùng trong bãi đỗ xe.
- Vị trí lắp đặt: tầng 1 bãi đỗ xe lối vào.
- Thông báo cho khách hàng thông tin số chỗ trống trong bãi hết hay còn chỗ trống.

Thông số bảng led ngoài trời (KZ-LOV.04.NET)

Mã hiệu	KZ-LOV.02.NET	
Nguồn điện	AC 220V/Max. 750mA	
Giao tiếp (tốc độ) về CCU	RS485 (9600 bps)	ETHERNET (10/100 Mbps)
Module hiển thị số chỗ đỗ trống	Ma trận LED, số điểm ảnh 16x32 Kích thước: 16cm x 32cm	
Số Module tiêu chuẩn	03 Module	
Hiện thị các ký tự cố định	Bảng mica chiếu sáng bên trong	
Số lượng phân vùng	Theo thiết kế bãi xe	
Vật liệu chính	Thép sơn tĩnh điện	
Số mặt hiển thị	01 mặt	
Tiêu chuẩn chống bụi/nước	IP65	
Kích thước tiêu chuẩn (RxCxS)	800x2000x300 (mm)	

Bảng 2.7 Thông số kỹ thuật đèn led ngoài trời

2.3.1. Hệ thống phân loại xe tại tầng hầm 1



Hình 2.10 Trạm phân loại xe tầng 1

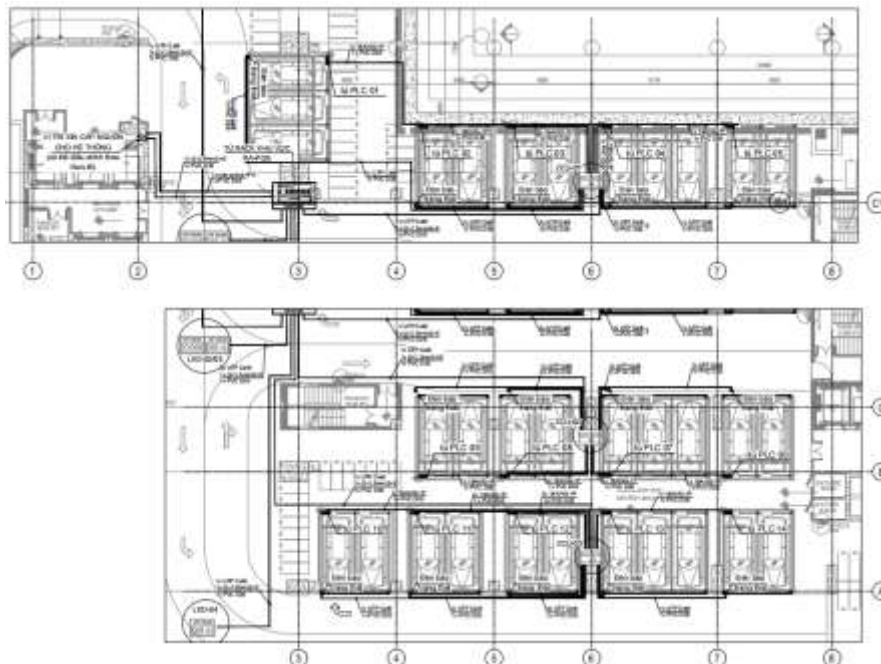
Trạm phân loại xe được đặt các trạm kiểm soát lối vào của làn ô tô để thực hiện phân loại xe SUV, SEDAN và OVERSIZE. Trạm phân loại xe sẽ kết hợp với phần mềm trạm trực vận hành và bảng Led hiển thị thông tin xe, giúp bảo vệ nắm bắt được loại xe, từ đó có thể phát thẻ xe lượt phù hợp với từng loại xe phục vụ việc ra vào hệ thống đỗ xe cơ khí được chính xác đảm bảo an toàn cho hệ thống cơ khí và phương tiện.

Hệ thống sẽ bao gồm các cảm biến quang với các đầu phát và nhận tín hiệu. Kết nối với bộ điều khiển để tổng hợp logic đưa ra tín hiệu được lập trình trước và máy trạm kiểm soát cho bảo vệ cũng như bảng LED hiển thị thông tin xe.



Hình 2.11 Hình ảnh minh họa trạm phân loại xe

2.3.2. Hệ thống chỉ, báo trống đỗ xe tại tầng hầm 2

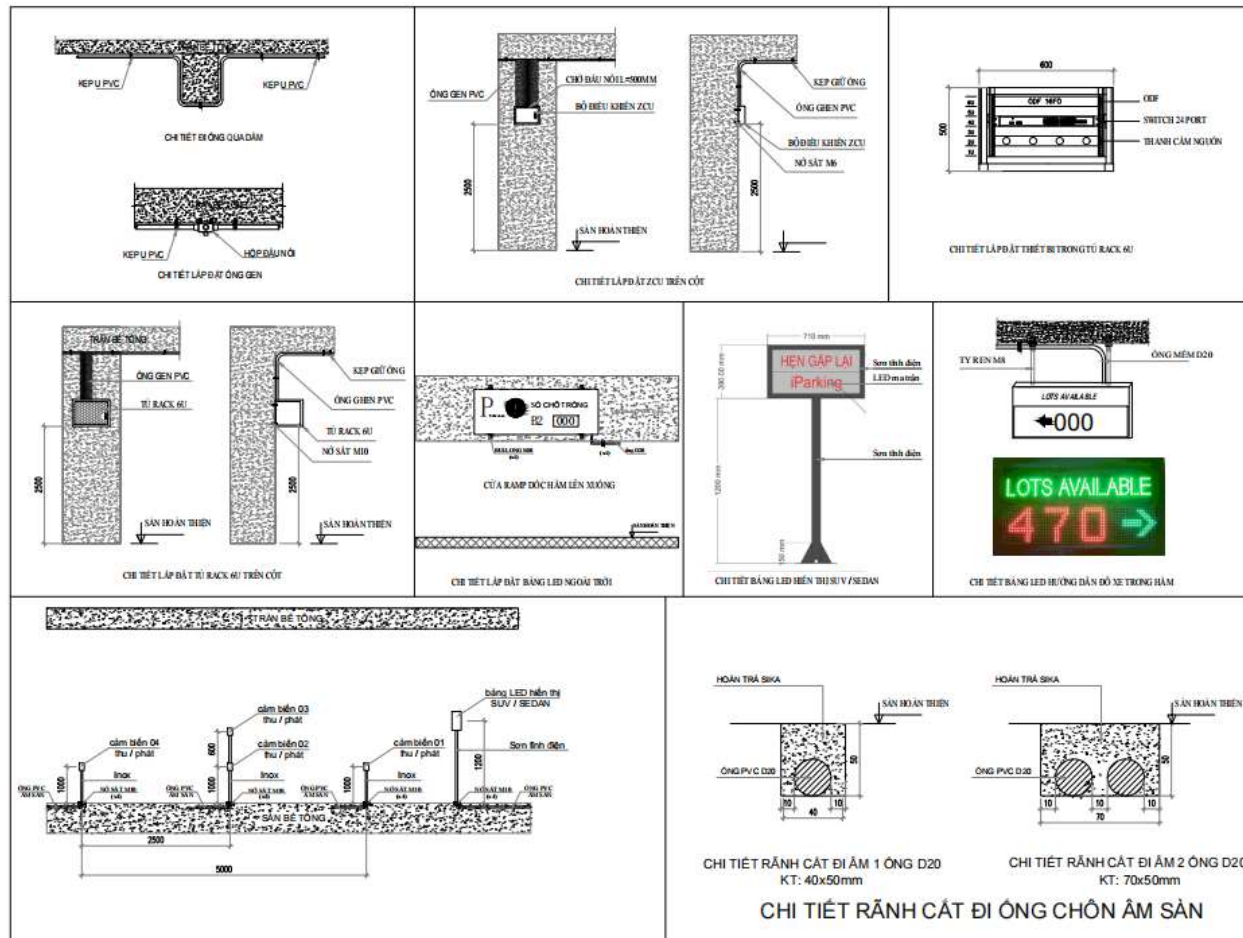


Hình 2.12 Mặt bằng bố trí bảng chỉ dẫn vị trí trống của các block

Hệ thống kiểm soát xe kết hợp sử dụng hệ thống báo chỗ trống trong tầng hầm có các ưu điểm sau:

- Hệ thống báo số chỗ trống và chỉ dẫn đỗ xe sử dụng công nghệ sóng siêu âm có thể hướng dẫn người điều khiển xe đến vị trí đậu xe còn trống rất dễ dàng và chính xác.

- Hệ thống giúp cho việc quản lý một cách dễ dàng và chính xác nhất. Giải pháp tạo ấn tượng tốt đối với khách hàng và nâng tầm thương hiệu của chủ sở hữu.
- Hệ thống cho thấy sự tiến bộ và hiện đại của tòa nhà hoặc khu đô thị.



CHI TIẾT LẮP ĐẶT ĐIỆN HÌNH HỆ THỐNG

Hình 2.13 Chi tiết lắp đặt điển hình hệ thống

Chi tiết lắp đặt ống gen và rãnh cắt đi ống chôn âm sàn:

- Hình vẽ mô tả cách lắp đặt ống gen PVC trong sàn bê tông.
- Kích thước rãnh cắt đi ống âm sàn (40x50mm và 70x50mm) cho ống PVC D20.

Chi tiết lắp đặt tủ rack và bộ điều khiển:

- Mô tả vị trí lắp đặt tủ rack trên cột và trong tường.
- Kèm theo bộ điều khiển ZCU và cách đi dây nối.

Chi tiết lắp đặt bảng LED chỉ dẫn:

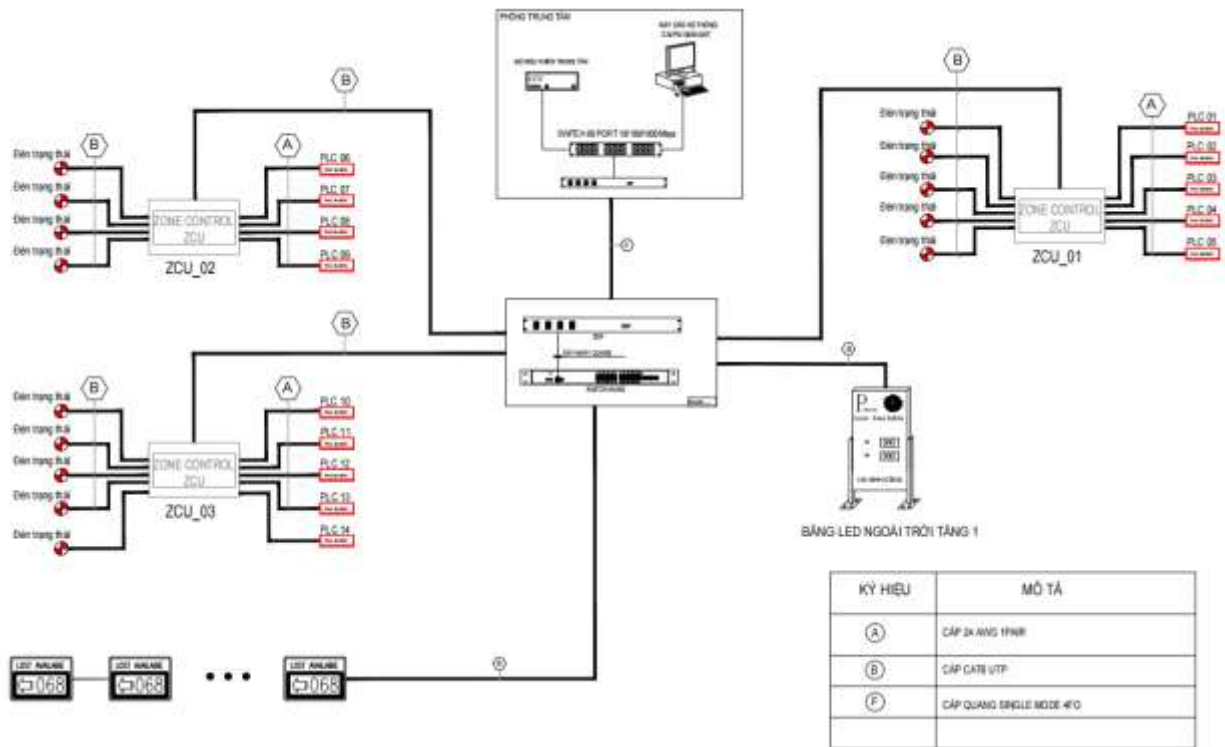
- Bảng LED hiển thị thông tin chỗ đỗ xe trong hầm.

- Bảng LED ngoài trời hướng dẫn xe SUV/SEDAN.

Chi tiết cửa rào và cảm biến lỗi vào:

- Mô tả hệ thống cửa rào kiểm soát vào bãi đỗ xe.
- Vị trí cảm biến phát hiện phương tiện.

2.4. Hệ thống giám sát bãi đỗ xe cơ khí



Hình 2.14 Sơ đồ nguyên lý giám sát hệ thống đỗ xe cơ khí

Giám sát trạng thái hoạt động của các thiết bị:

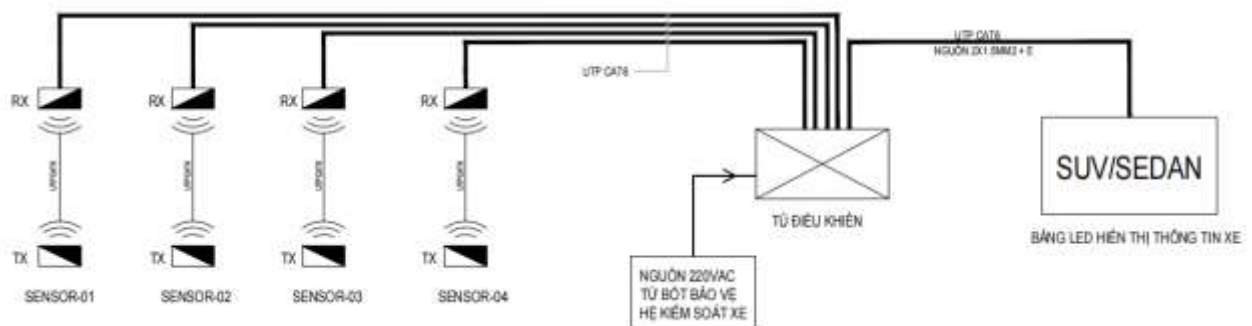
- **Đèn trạng thái (Đèn trạng thái):** Các đèn trạng thái được kết nối với PLC và ZCU cho phép giám sát trạng thái hoạt động của từng thiết bị. Điều này giúp người vận hành nhanh chóng phát hiện các lỗi hoặc sự cố.
- **PLC (Bộ điều khiển logic khả trình):** PLC được sử dụng để điều khiển các cơ cấu cơ khí của bãi đỗ xe. Thông qua PLC, người vận hành có thể giám sát vị trí, trạng thái của các cơ cấu này.
- **ZCU (Bộ điều khiển vùng):** ZCU quản lý các khu vực đỗ xe riêng biệt. Việc giám sát trạng thái của ZCU cho phép đảm bảo hoạt động ổn định của từng khu vực.

Giám sát kết nối mạng:

- **Switch 8 Port 10/100/1000 Mbps:** Switch này kết nối giữa máy tính trung tâm và CCU. Việc giám sát kết nối mạng đảm bảo rằng dữ liệu điều khiển và giám sát được truyền tải thông suốt.
- **Switch mạng:** Kết nối bộ điều khiển vùng ZCU, các bảng led hướng dẫn và led ngoài trời.
- **Cáp CAT6 UTP và cáp quang:** Các loại cáp này được sử dụng để truyền dữ liệu trong hệ thống. Việc giám sát tình trạng cáp đảm bảo kết nối ổn định.

Giám sát thông qua máy tính trung tâm:

- **Máy tính (Server):** Máy tính trung tâm có thể được sử dụng để hiển thị giao diện giám sát, cho phép người vận hành theo dõi trạng thái hoạt động của toàn bộ hệ thống.



Hình 2.15 Sơ đồ nguyên lý phân loại xe

Các thành phần chính:

- **Sensor-01, sensor-02, sensor-03, sensor-04:** Các cảm biến này được sử dụng để phát hiện sự hiện diện của xe và có thể đo kích thước hoặc đặc điểm khác của xe.
- **TX (Transmitter - Bộ phát):** Các bộ phát tín hiệu của cảm biến.
- **RX (Receiver - Bộ thu):** Các bộ thu tín hiệu của cảm biến.
- **Tủ điều khiển:** Đây là trung tâm xử lý dữ liệu từ các cảm biến và đưa ra quyết định phân loại xe.
- **Bảng led hiển thị thông tin xe:** Hiển thị kết quả phân loại xe (SUV/SEDAN).
- **Nguồn 220vac từ bộ bảo vệ hệ thống kiểm soát xe:** Nguồn điện cung cấp cho tủ điều khiển.
- **Nguồn 2x1.5mm2 + E:** Nguồn điện bổ sung hoặc nguồn dự phòng.
- **UTP CAT6:** Cáp mạng Ethernet, dùng để truyền dữ liệu giữa các cảm biến và tủ điều khiển.

Nguyên lý hoạt động:

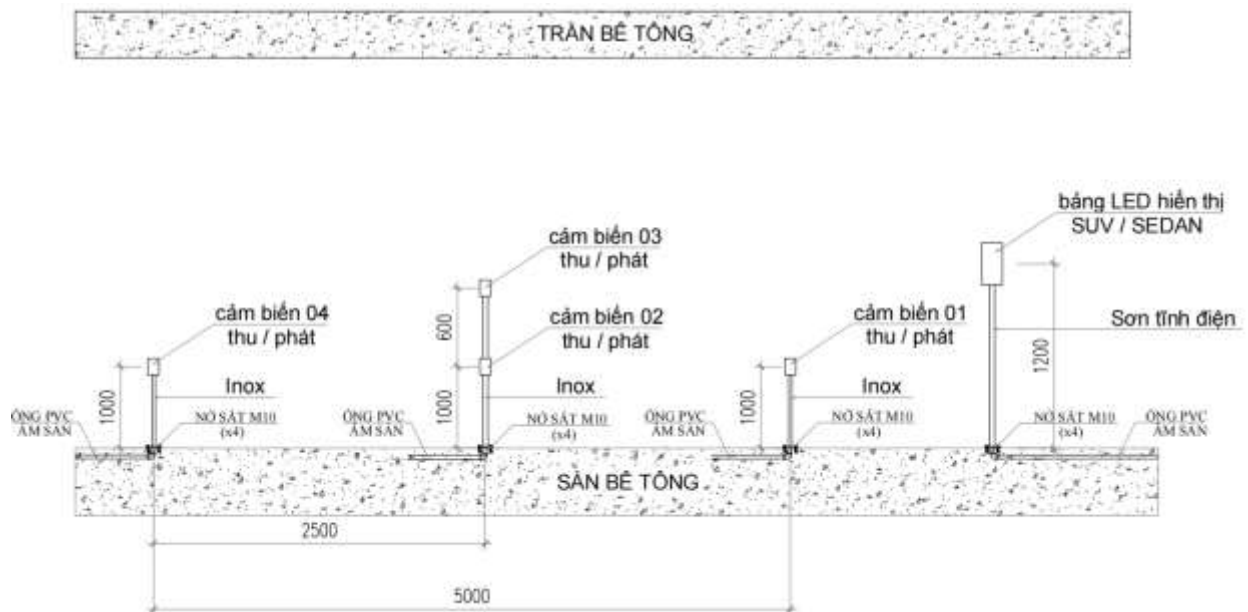
Phân loại xe: Khi xe đi qua khu vực cảm biến, các cảm biến (SENSOR-01 đến SENSOR-04) sẽ gửi tín hiệu về bộ điều khiển trung tâm để phân loại xe

Truyền dữ liệu: Dữ liệu từ các cảm biến được truyền đến tủ điều khiển thông qua cáp UTP CAT6.

Xử lý dữ liệu: Tủ điều khiển xử lý dữ liệu từ các cảm biến để xác định kích thước hoặc đặc điểm khác của xe.

Phân loại xe: Dựa trên dữ liệu đã xử lý, tủ điều khiển phân loại xe thành SUV hoặc SEDAN.

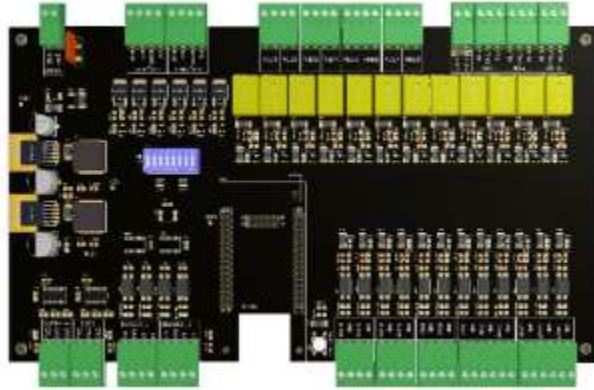
Hiển thị kết quả: Kết quả phân loại xe được hiển thị trên bảng LED.



Hình 2.16 Bản vẽ bố trí cảm biến và bảng LED hiển thị phân loại xe

Nguyên lý phân loại xe SUV và SEDAN trước khi vào bãi đỗ

- **Bộ điều khiển phân loại xe KZ-VRC.NET**
KZ-VRC.NET được sử dụng phân loại xe dựa trên tín hiệu đầu vào cảm biến giới hạn chiều cao và cảm biến giới hạn chiều dài để phân loại chủng loại xe SUV và SEDAN.



Hình 2.17 Bộ điều khiển KZ-VRC.NET

Thông số kỹ thuật

Mã hiệu	KZ-VRC.NET
Nguồn điện	24Vdc/500mA
Số đầu vào tiếp điểm khô	04
Số đầu ra điều khiển khóa (Relay)	02
Độ ẩm hoạt động	10% - 90%
Nhiệt độ hoạt động	0-600C
Kích thước(RxSxC)	160x250x20(mm)

Bảng 2.8 Thông số kỹ thuật bộ điều khiển KZ-VRC.NET

- **Cảm biến hồng ngoại IR33**



Hình 2.18 Cảm biến hồng ngoại IR33

Thông số kỹ thuật

Công nghệ	Hồng ngoại
Bộ thu	$\leq 15m$
Phạm vi	Phạm vi có thể giảm xuống 30% trong điều kiện thời tiết xấu: Đối với mưa, bụi, v.v.
Tần số hồng ngoại	1,92 Khz
Nguồn điện	12 đến 24VDC/AC
Chiều dài sóng	940nm
Đầu vào	R*15mA-T*30mA
Nhiệt độ hoạt động	-20C đến 50C
Đầu ra role	1A ma*30V
Kích thước	49,2*76*21,6mm
Số lượng	4

Bảng 2.9 Thông số kỹ thuật của cảm biến hồng ngoại IR33

• Nguyên lý hoạt động

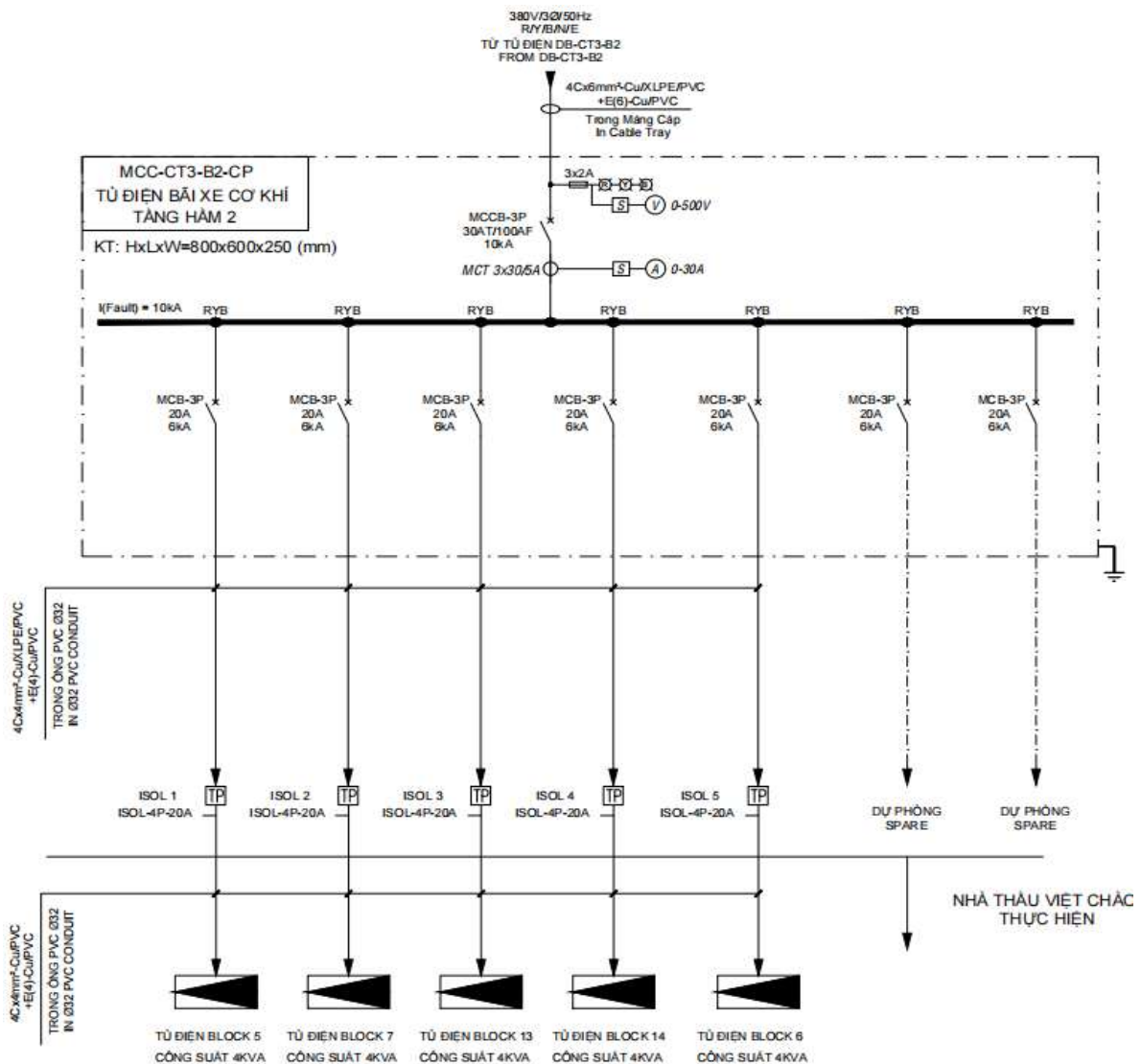
Có 4 cảm biến IR33 nằm ở các vị trí: Cảm biến 04 đặt ở vị trí xe vào và kích hoạt đầu tiên chiều cao 1000mm, cảm biến 02 đặt ở vị trí cách cảm biến 04 2500mm và cao 1000m, cảm biến 03 đặt ở vị trí trên vị trí cảm biến 02 cách 600mm và cảm biến 01 đặt ở vị trí cách cảm biến 04 5000mm cao 1000mm.

- Nếu cảm biến thấp nhất cảm biến 04 kích hoạt, tức là có xe đi vào.
- Dựa vào số cảm biến kích hoạt, xác định chiều cao xe:
- **SEDAN ($\leq 1475mm$):** Kích hoạt cảm biến 04, cảm biến 02 (có thể cảm biến 01 kích hoạt, nhưng cảm biến 04 không kích hoạt).
- **SUV ($\leq 1910mm$):** Kích hoạt. cảm biến 01, cảm biến 02, cảm biến 03 và cảm biến 04.
- Dữ liệu từ cảm biến gửi về vi điều khiển để xác định loại xe.
- Màn hình phân loại xe hiển thị loại xe để phát đúng loại thẻ.



Hình 2.19 Bảng LED hiển thị SUV/SEDAN

2.5. Hệ thống cấp nguồn bãi đỗ xe CT3-CT7



Hình 2.20 Sơ đồ nguyên lý cấp nguồn tại hầm 2 – tòa CT3

Nguồn cấp chính

- Hệ thống sử dụng nguồn 380V/3P/50Hz từ tủ điện tổng DB-CT7-B2.
- Cấp nguồn chính là 4x1Cx10mm² Cu/XLPE/PVC đi trong máng cáp và (E+10) Cu/PVC làm dây tiếp địa.

Bảo vệ và phân phối điện

- Sử dụng thiết bị đóng cắt MCCB-3P 50A/10kA làm thiết bị bảo vệ tổng.
- Sau MCCB tổng, dòng điện được phân nhánh đến các MCB-3P 20A/6kA (bảo vệ từng nhánh riêng lẻ).

Các nhánh cấp nguồn

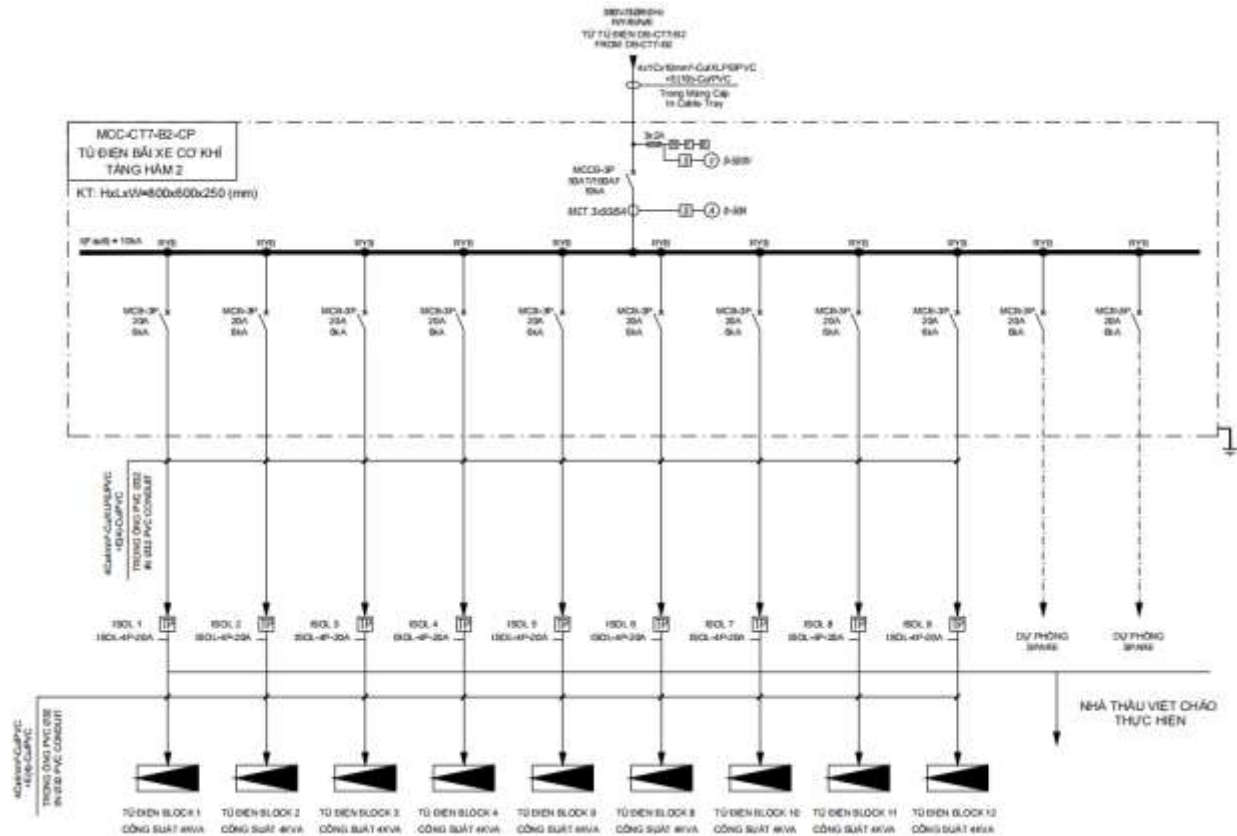
- Từ MCB-3P 20A, nguồn điện cấp xuống các bộ ISOL-4P-20A (công tắc cách ly 4 cực).
- Cấp nhánh sử dụng 4Cx6mm² Cu/XLPE/PVC đi trong ống nhựa PVC D32 hoặc D20 tùy vị trí lắp đặt.

Cấp nguồn đến từng tủ điện block

- Mỗi nhánh cấp đến một Tủ điện block (1, 3, 4, 8, 9, 10, 11, 12).
- Mỗi tủ có công suất 4kVA, đảm bảo cung cấp đủ điện cho thiết bị trong khu vực bãi đỗ xe cơ khí.

Dự phòng

- Có hai nhánh dự phòng (SPARE) để sử dụng khi cần mở rộng hoặc thay thế trong tương lai.



Hình 2.21 Sơ đồ nguyên lý cấp nguồn tại hầm 2 – tòa CT7

Tủ điện hệ thống đỗ xe cơ khí

- Đây là bộ điều khiển trung tâm, quản lý toàn bộ hoạt động của hệ thống đỗ xe cơ khí.
- Cung cấp nguồn điện cho các động cơ, cảm biến và thiết bị an toàn.

Các thành phần chính trong hệ thống

Hệ thống động cơ

Hệ thống này gồm ba loại động cơ chính:

- **Động cơ nâng:** Dùng để nâng hoặc hạ xe theo từng tầng của bãi đỗ xe cơ khí.
- **Động cơ trượt:** Di chuyển xe theo phương ngang để đưa xe vào đúng vị trí đỗ.
- **Động cơ hàng rào tự động:** Điều khiển cửa hoặc hàng rào bảo vệ, giúp kiểm soát vào/ra bãi đỗ xe.

Hệ thống cảm biến

Hệ thống cảm biến giúp giám sát và đảm bảo an toàn khi vận hành:

- **Cảm biến chiều cao/dài xe:** Xác định kích thước xe, tránh lỗi va chạm hoặc xe không phù hợp với vị trí đỗ.
- **Cảm biến xe ra/vào:** Nhận diện phương tiện khi vào hoặc ra khỏi bãi đỗ xe, giúp kích hoạt các thiết bị khác như cửa rào hoặc tín hiệu báo hiệu.

Hệ thống bảo vệ và điều khiển

- **Móc chống rơi:** Cơ chế an toàn để giữ xe cố định, ngăn ngừa rơi hoặc trượt ngoài ý muốn.
- **Công tắc giới hạn hành trình:** Giới hạn phạm vi di chuyển của động cơ, tránh quá tải hoặc vượt mức an toàn.
- **Bảng điều khiển:** Là trung tâm điều khiển hệ thống, nơi nhân viên hoặc hệ thống tự động vận hành các lệnh.
- **Còi báo hiệu:** Cảnh báo khi hệ thống đang hoạt động hoặc khi có sự cố.
- **Công tắc an toàn xích:** Đảm bảo rằng hệ thống xích vận hành đúng cách và không bị lỏng hoặc đứt.

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG HƯỚNG DẪN ĐỖ XE

3.1. Hệ thống chỉ dẫn đỗ xe KPGS

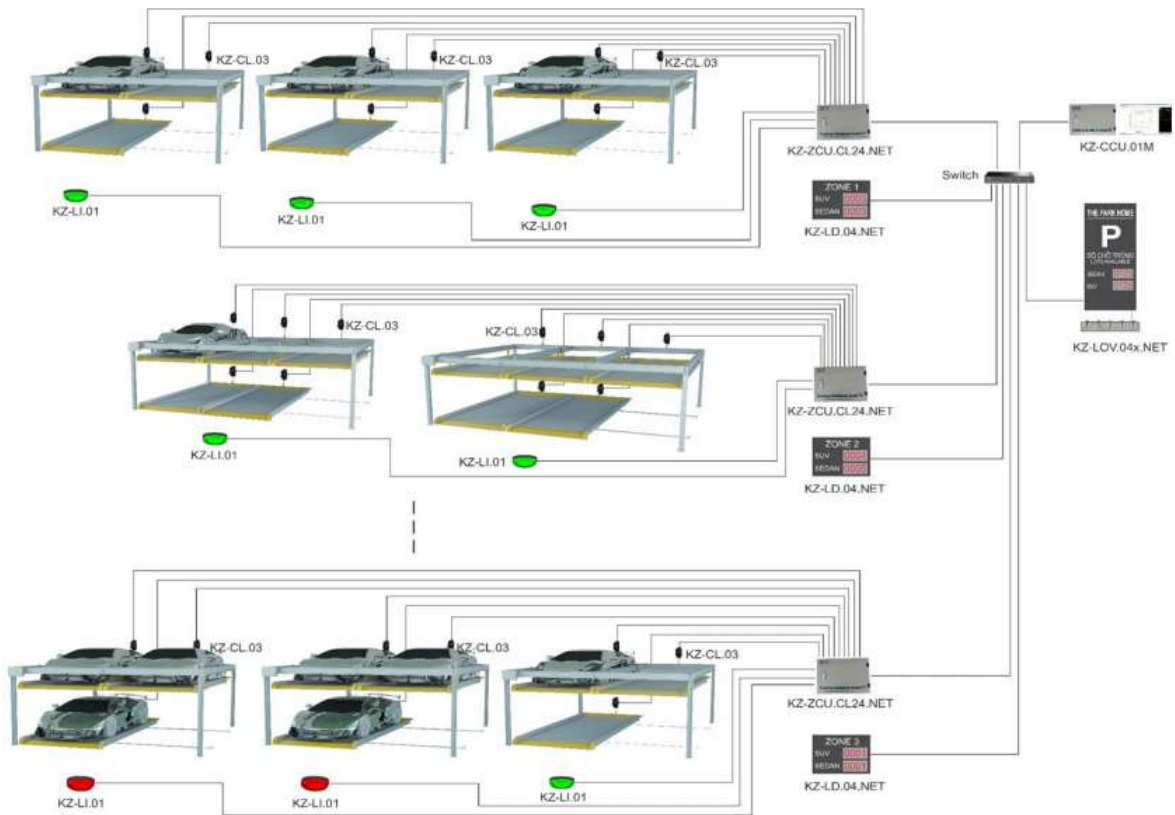
KPGS - Hệ thống hướng dẫn đỗ xe tiên tiến của KZTEK, dẫn đường cho người lái xe đến chỗ đỗ xe có sẵn, tiết kiệm thời gian và tạo ra trải nghiệm tích cực và hiệu quả cho người lái xe.

KPGS cung cấp cho các nhà khai thác bãi đỗ xe quyền kiểm soát lớn hơn đối với các cơ sở của họ, cung cấp tối ưu hóa lưu lượng giao thông và tăng công suất sử dụng. Người vận hành có thể đảm bảo rằng tất cả các chỗ đậu xe đều được sử dụng trước khi đóng bất kỳ khu vực đỗ xe nào.

KPGS có nghĩa là sự hài lòng của khách hàng nhiều hơn với ít rắc rối hơn khi họ được chỉ dẫn nhanh chóng và hiệu quả đến các vị trí đỗ xe có sẵn.

- Tính năng:
 - Mỗi vị trí đỗ xe trong bãi được trang bị cảm biến siêu âm với đèn báo tích hợp.
 - Cung cấp cho tài xế thông tin về số vị trí đỗ còn lại hiện tại của bãi đỗ xe ngay trước khi vào bãi đỗ.
 - Hướng dẫn phương tiện đến chỗ đỗ trống gần nhất.
 - Hệ thống có cấu trúc mô đun
 - Giao tiếp mạng LAN
 - Giám sát việc sử dụng chỗ đỗ xe
 - Cấu hình và giám sát hệ thống bằng phần mềm.
- Hoạt động của hệ thống
 - Đầu tiên: Chỉ dẫn ở tất cả các vị trí đỗ xe. Sử dụng một cảm biến siêu âm tích hợp đèn Led. Các cảm biến sẽ phát hiện xe vào vị trí đỗ. Đèn Led sẽ hiển thị kết quả, màu xanh lá cây khi chỗ đỗ trống và màu đỏ khi đã có xe.
 - Thứ hai: Chỉ dẫn ở tất cả những điểm mà phương tiện phải quyết định hướng đi. Sử dụng bảng led chỉ dẫn trong nhà đặt ngay trước các ngã rẽ, hiển thị những chỗ trống ở mỗi hướng. Giúp lái xe dễ dàng tìm thấy hướng đi còn chỗ đỗ xe.
 - Thứ ba: Quản lý và thu thập thông tin từ tất cả các cảm biến bằng phần mềm. Phần mềm sẽ gửi thông tin đến tất cả các bảng led chỉ dẫn, lưu trữ dữ liệu và tạo các báo cáo cần thiết.

3.2. Cấu hình hệ thống KPGS



Hình 3.1 Cấu hình hệ thống KPGS

- ❖ KZ-CCU.xxx – Bộ điều khiển trung tâm
 - Thu thập dữ liệu từ các cảm biến thông qua các bộ điều khiển vùng
 - Cung cấp các chỉ dẫn và số lượng vị trí trống đến các bảng led trong nhà và ngoài trời.



Hình 3.2 Bộ xử lý trung tâm KZ-CCU.xxx

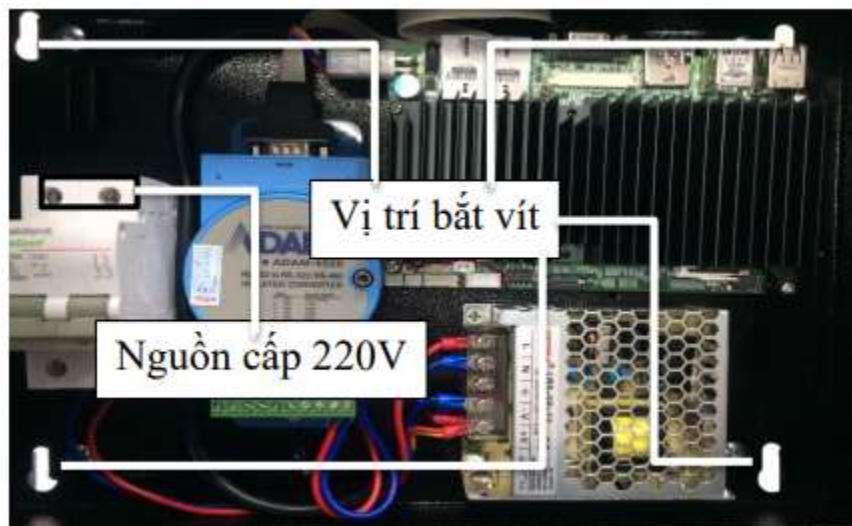
Thông số kỹ thuật

Mã hiệu	KZ-CCU.01M
Nguồn điện	AC 220V/Max. 200mA
Giao tiếp (tốc độ)	ETHERNET (10/100 Mbps)
Cổng hiển thị đầu ra	VGA, HDMI
Tính năng nâng cao	Remote desktop, Bản đồ bãi đỗ xe.
Số node(1) tối đa có thể quản lý	50
Kích thước (CxRxS)	28x38x12 cm

Bảng 3.1 Thông số kỹ thuật của KZ-CCU.01M

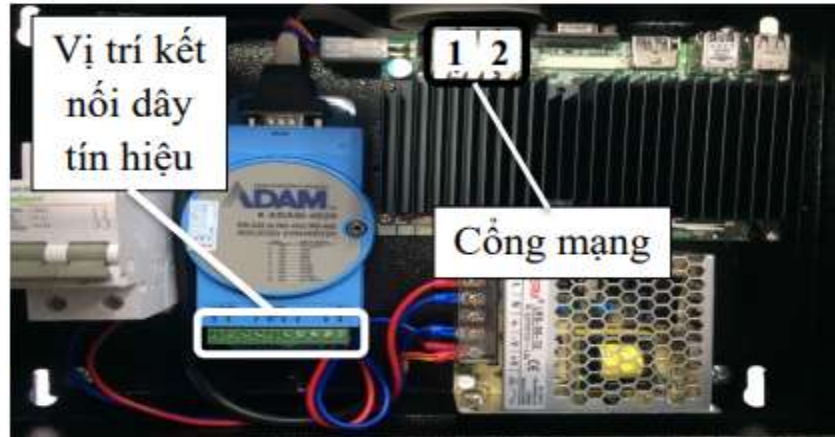
Hướng dẫn lắp đặt và đấu nối

- Tiến hành khoan lỗ bắt vít trên tường và cấp nguồn 220V AC cho bộ điều khiển trung tâm tại các vị trí như hình



Hình 3.3 Vị trí lắp bắt vít và cấp nguồn

- Vị trí kết nối dây tín hiệu và cổng mạng trên bộ điều khiển trung tâm:



Hình 3.4 Vị trí kết nối tín hiệu

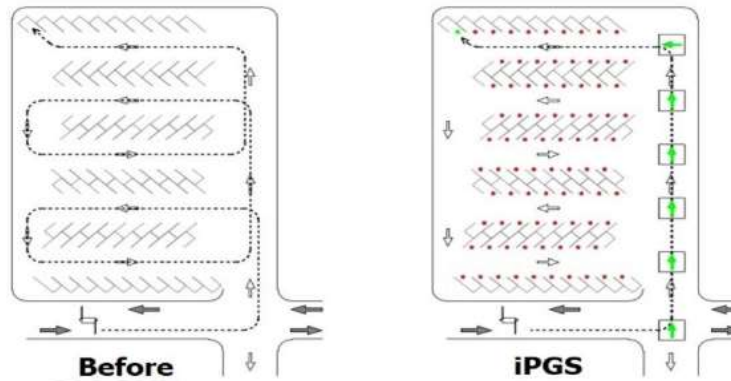
- Nối dây tín hiệu từ các bộ ZCU tới CCU (hệ thống sử dụng RS485 kết nối từ CCU tới ZCU)



Hình 3.5 Nối dây tín hiệu trên bộ chuyển đổi

- Dây tín hiệu tới ZCU và bảng Led là dây mạng CAT5e, tách 03 dây màu xanh lá cây (Data+), màu xọc xanh lá (Data-), màu xọc xanh da trời (GND) . Sau đó đấu vào các chân tương ứng trên bộ chuyển đổi ADAM-4520.

- ❖ KZ-iPGS – Phần mềm báo số chỗ trống
 - Quản lý Dữ liệu toàn bộ hệ thống.
 - Thu thập dữ liệu từ các bộ điều khiển khu vực, điều khiển hiển thị số chỗ trống và hướng cho bảng led trong nhà và ngoài trời.
 - Phần mềm được cài đặt trước trong bộ điều khiển trung tâm.
 - Cấu hình, giám sát và điều khiển hệ thống.
 - Hỗ trợ tính năng bản đồ: cho phép quan sát trực quan tất cả các vị trí trống trên bản đồ.



Hình 3.6 Giao diện phần mềm báo chỗ trống KZ-iPGS

- ❖ KZ-ZCU.CLxx – Bộ điều khiển vùng
 - Cấp nguồn và điều khiển cảm biến siêu âm.
 - Thu thập thông tin từ cảm biến, điều khiển đèn led chỉ thị
 - Gửi dữ liệu cho bộ điều khiển trung tâm.



Hình 3.7 Bộ điều khiển vùng K-ZCU

Thông số kỹ thuật

Mã hiệu	KZ-ZCU.CL24.NET
Nguồn	AC 220V/Max. 350mA
Số tín hiệu đầu vào(Số cảm biến tối đa được hỗ trợ)	24
Dạng tín hiệu đầu vào	Tín hiệu tiếp điểm khô
Tương thích với trung tâm	KZ-CCU.01M, KZ-CCU.01H, KZ-CCU.01U
Chất liệu vỏ	Thép sơn tĩnh điện
Kích thước (CxRxS)	28x38x12 cm

Bảng 3.2 Thông số kỹ thuật của KZ-ZCU.CL24.NET

Hướng dẫn lắp đặt và đấu nối

- Tiến hành khoan lỗ bắt vít trên tường, tại các vị trí trên hình vẽ



Hình 3.8 Khoang lỗ bắt vít

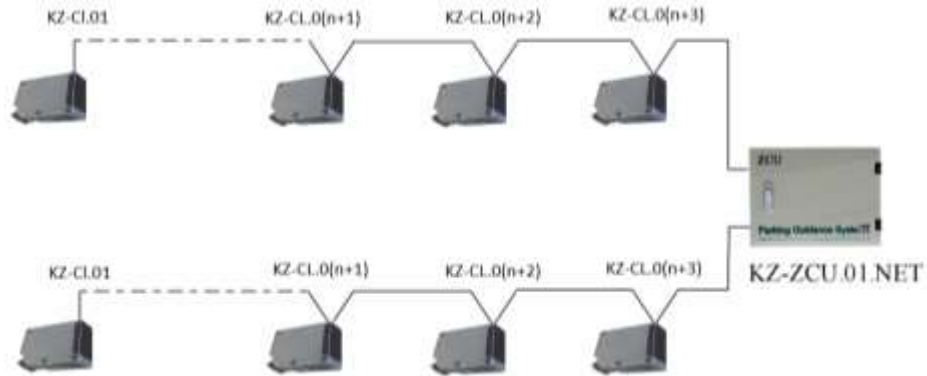
- Đấu nối dây nguồn 220V AC vào Aptomat:



Hình 3.9 Đấu dây nguồn cho ZCU

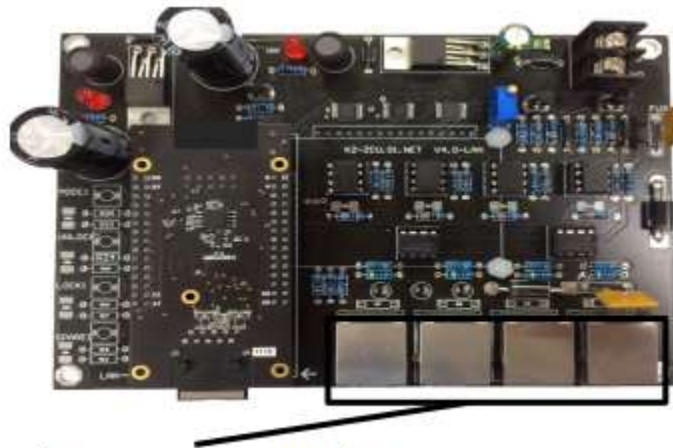
- Đấu dây tín hiệu và nguồn cho cảm biến báo trống

Cảm biến báo trống được cấp nguồn và truyền tín hiệu thông qua dây mạng (CAT5, CAT6). Các cảm biến được chia thành nhiều đường đến tủ ZCU để làm giảm nguy cơ sụt nguồn trên đường dây



Hình 3.10 Sơ đồ đi dây ZCU tới cảm biến

- Jack mạng từ các đường cảm biến được cắm vào vị trí SENSOR 1 – SENSOR 2 – SENSOR 3 – SENSOR 4

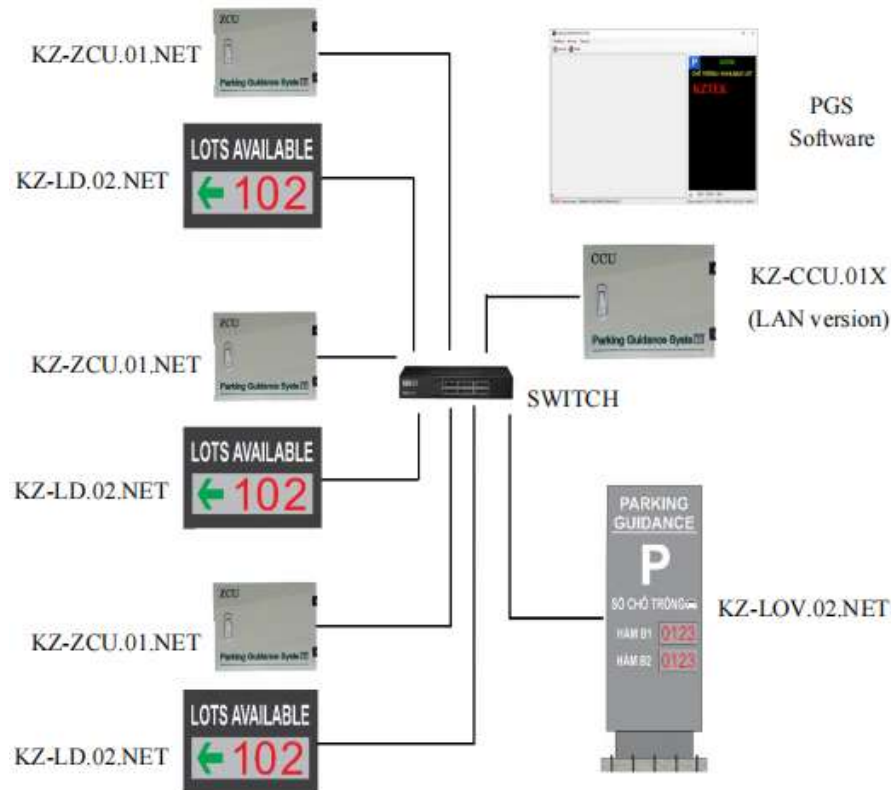


Vị trí cắm jack mạng từ cảm biến

Hình 3.11 Cắm jack mạng cho cảm biến

- Đầu dây kết nối về trung tâm

Trong cấu trúc mạng LAN mỗi thiết bị như CCU, ZCU, Bảng LED có một IP riêng, các thiết bị này được kết nối với nhau qua các switch mạng. Dưới đây là sơ đồ kết nối tổng quan hệ thống



Hình 3.12 Sơ đồ kết nối CCU tới ZCU và các bảng LED

❖ Cảm biến báo trống tầng hầm

- Cảm biến KZ-CL.01 được sử dụng để báo trống cho hệ thống bãi đỗ xe sử dụng kết cấu nâng hạ ô tô. Cảm biến sử dụng nguyên lý quang điện để phát hiện vật thể.
- Phát hiện vật thể lên tới 2 mét.
- Có độ nhạy cao.
- Đầu ra tiếp điểm khô.
- Công suất tiêu thụ thấp
- Tương thích với bộ điều khiển khu vực: KZ-ZCU.CL10, KZ-ZCU.CL10.NET



Hình 3.13 Cảm biến báo trống tầng hầm

Thông số kỹ thuật

Mã hiệu	KZ-CL.01
Nguồn	24VDC/45mA
Khoảng cách đo tối đa	2.0 mét
Tín hiệu đầu ra	Tiếp điểm khô
Led chỉ thị	02
Thời gian đáp ứng	20 ms
Vật liệu vỏ	ABS
Tiêu chuẩn chống bụi/nước.	IP67
Nhiệt độ hoạt động	0-600C
Độ ẩm hoạt động	35-85%
Trọng lượng	100g
Kích thước (RxSxC)	25x68x73 (mm)
Đèn Led chỉ thị	KZ-LI.01 - Nguồn: DC5V/Max. 30mA - Màu: Đỏ & Xanh lá cây - Kích thước: 8x8x2,5cm

Bảng 3.3 Thông số kỹ thuật cảm biến báo chỗ trống KZ-Cl.01

- ❖ KZ-LD.xx.NET – Bảng led đơn trong nhà
 - Hiện thị số lượng chỗ đỗ trống và mũi tên chỉ hướng đến khu vực có số chỗ trống đó.
 - Vị trí lắp đặt: Lôi vào một khu vực hoặc ở ngã rẽ trong bãi đỗ xe.



Hình 3.14 Bảng ked đơn trong nhà KZ-LD.xx.NET

Thông số kĩ thuật

Mã hiệu	KZ-LD.04.NET
Nguồn điện	AC 220V/Max. 80mA
Giao tiếp (tốc độ) về CCU	ETHERNET (10/100 Mbps)
Số điểm ảnh hiển thị	Matrix 16x64
Vật liệu	Thép sơn tĩnh điện
Chữ “LOTS AVAILABLE”	Bảng mica chiếu sáng bên trong
Kích thước (RxCxS)	545x262x90 mm

Bảng 3.4 Thông số kĩ thuật Led đơn trong nhà LZ-LD.04.NET

- ❖ KZ-LOV.xx – BẢNG LED NGOÀI TRỜI ĐỨNG.
 - Hiển thị số lượng chỗ đỗ trống tại mỗi vùng trong bãi đỗ xe.
 - Vị trí lắp đặt: phía trước cổng bãi đỗ xe



Hình 3.15 Bảng LED ngoài trời đứng KZ-LOV.xx

Thông số kỹ thuật

Mã hiệu	KZ-LOV.xx.NET
Nguồn điện	AC 220V/Max. 750mA
Giao tiếp (tốc độ) về CCU	ETHERNET (10/100 Mbps)
Module hiển thị số chỗ đỗ trống	Ma trận LED, số điểm ảnh 16x32 Kích thước: 16cm x 32cm
Số Module tiêu chuẩn	03 Module
Hiển thị các ký tự cố định	Bảng mica chiếu sáng bên trong
Số lượng phân vùng	Theo thiết kế bãi xe
Vật liệu chính	Thép sơn tĩnh điện
Số mặt hiển thị	01 mặt
Tiêu chuẩn chống bụi/nước	IP65
Kích thước tiêu chuẩn (RxCxS)	800x2000x300 (mm)

Bảng 3.5 Thông số kỹ thuật của KZ.LOV.xx.NET

❖ Switch IS170-26TF-AC

- Bộ chuyển mạch IS170 series là bộ chuyển mạch truy cập L2 không được quản lý hiệu suất cao do Maipu phát triển. Nó được áp dụng trong mạng LAN SMB và giải pháp chuyển mạch Layer2 để triển khai, cung cấp khả năng chuyển tiếp tốc độ đường truyền của tất cả các cổng.

- Nó áp dụng công nghệ lưu trữ và chuyển tiếp, kết hợp với phân bổ bộ nhớ động, để đảm bảo phân bổ hiệu quả cho từng cổng. Bộ chuyển mạch IS170 series đáp ứng tốt các yêu cầu toàn diện của khách hàng SMB như chi nhánh, khách sạn, nhà hàng, trường học, hậu cần, chuỗi cửa hàng, v.v.



Hình 3.16 Switch IS170-26TF-AC

Thông số kỹ thuật

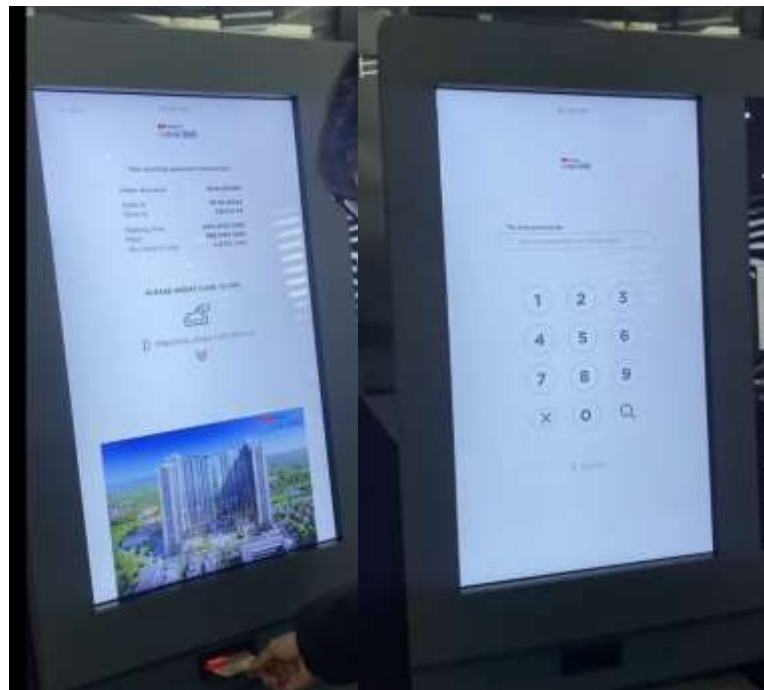
Mã hiệu	IS170-26TF-AC
Lưu lượng công vật lý	24*10/100/1000M Base-T giao diện 2*1G SFP

Khả năng chuyển đổi	52Gbps
Nguồn điện cố định	1
Quạt cố định	Có
Tiêu thụ điện năng	≤15W
Kích thước	442×220×44.2

Bảng 3.6 Thông số kỹ thuật của Switch IS170-26TF-AC

❖ KZ-KPF Kiosk tìm kiếm xe và thanh toán

- Tìm kiếm xe theo thẻ từ hoặc biển số
- Chỉ dẫn đường đi cho khách bằng bản đồ
- Hỗ trợ đầu đọc QR code để thanh toán
- Thanh toán qua ví điện tử hoặc Smart Banking
- Thanh toán tiền mặt
- Cảnh báo khi có sự cố
- Người dùng tương tác trực tiếp với màn hình cảm ứng 32inch



Hình 3.17 KZ-KPF Kiosk tìm kiếm xe và thanh toán

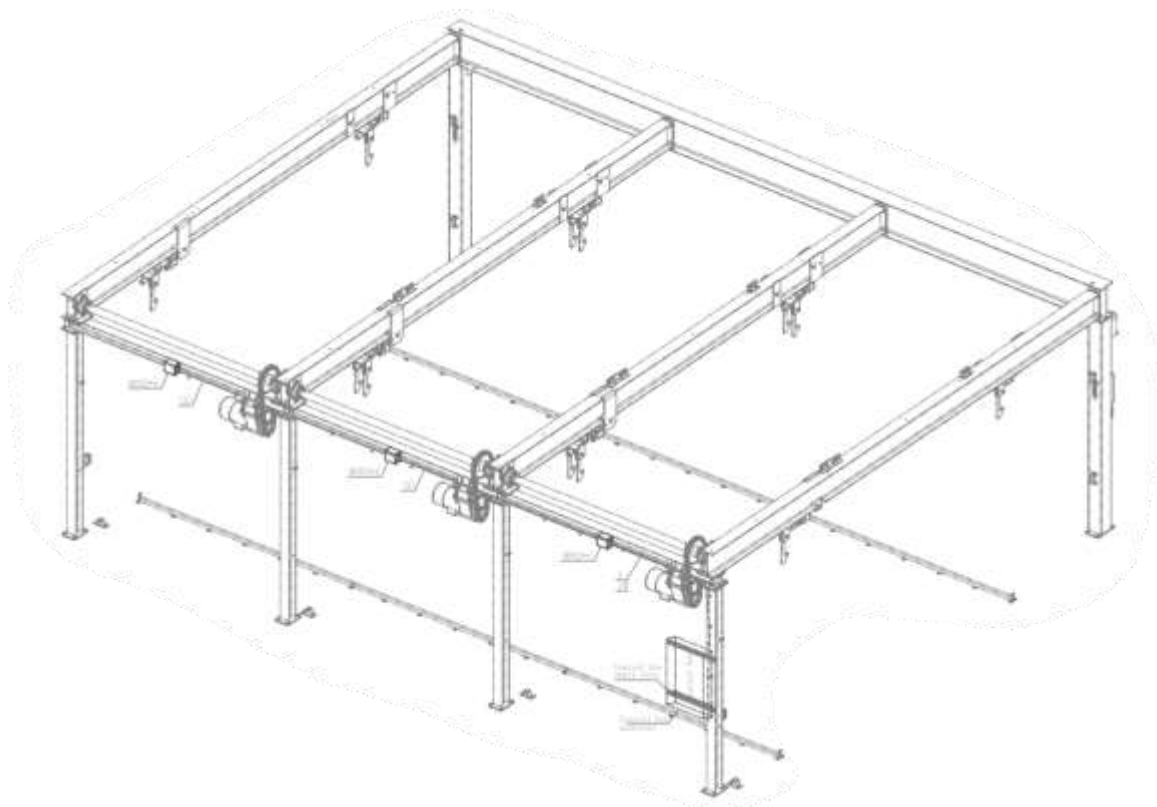
CHƯƠNG 4: HỆ THỐNG ĐỖ XE CƠ KHÍ PUZZLE STACKER HORIZONTAL (PSH)

4.1. Cấu tạo chính của hệ thống đỗ xe cơ khí puzzle stacker horizontal (PSH)

4.1.1 Kết cấu thép

- *Cột trước:* Sử dụng thép ống vuông 150×150 mm, thiết kế 2 cột giúp dễ dàng lắp đặt dây điện và bảng điều khiển, đồng thời tạo thuận lợi cho việc xe ra vào.
- *Cột sau:* Sử dụng thép H 100×100 mm, thiết kế 4 cột tăng khả năng chịu lực và giảm tác động lên sàn, đồng thời giảm rung lắc trong quá trình hoạt động.

Bề mặt kết cấu thép có thể được xử lý bằng sơn, sơn tĩnh điện hoặc mạ kẽm nhúng nóng để tăng độ bền và chống ăn mòn.



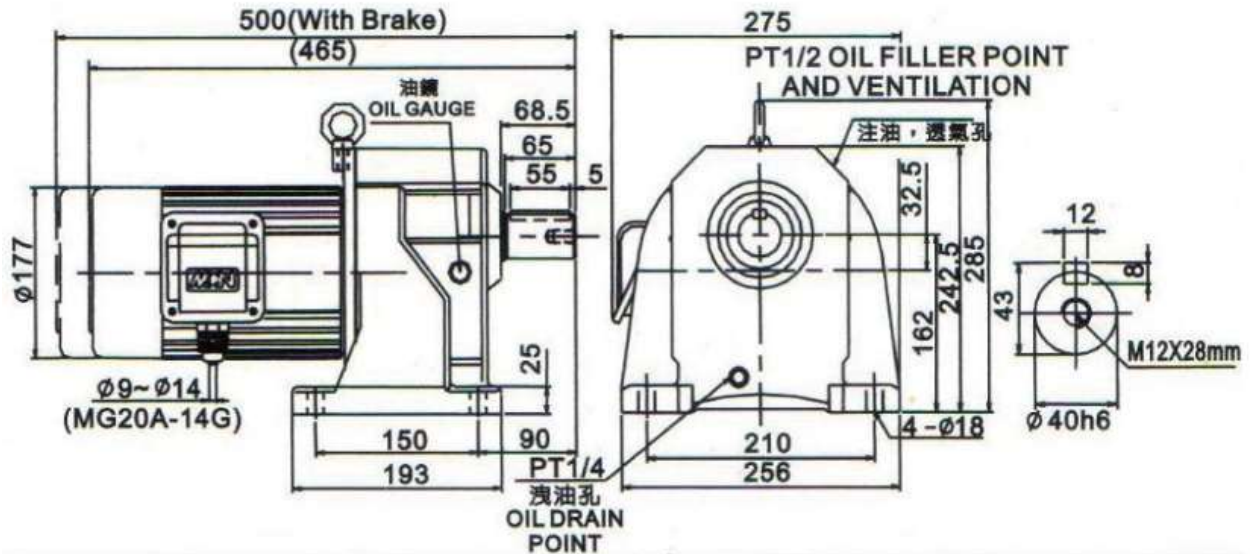
Hình 4.1 Kết cấu thép

4.1.2 Hệ thống truyền động

Sử dụng hệ xích 4 điểm độc lập và thiết kế xích dạng tròn để nâng hạ, đảm bảo hoạt động an toàn và ổn định.

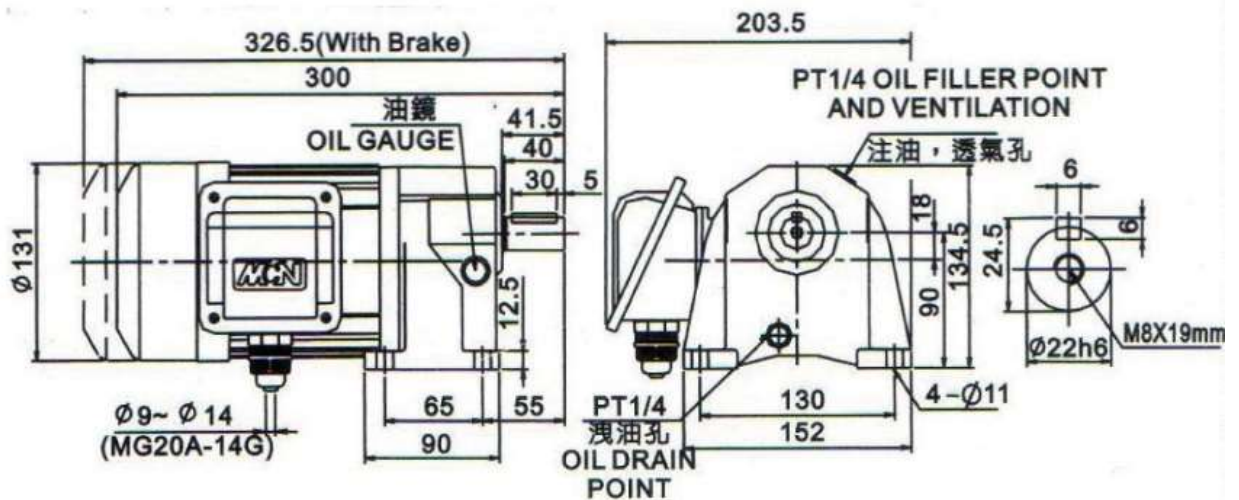
Gồm hai loại mô-tơ chính:

- *Mô-tơ nâng*: Công suất 2.2 kW, tỉ số truyền 1/20.



Hình 4.2 Motor nâng lifting motor parks

- *Mô-tơ trượt*: Công suất 0.2 kW, tỉ số truyền 1/10.



Hình 4.3 Motor trượt Sliding motor parks

Mô-tơ trượt ngang được kết nối trực tiếp với trục truyền lực ngang và sử dụng truyền động đồng cấp. Điều này giúp hệ thống trượt ổn định hơn, ít ồn hơn.

4.1.3 Pallet

- Thiết kế buffer lò xo và giảm chấn cao su ở đáy pallet giúp nâng hạ êm ái
- Thanh chặn bánh sau sử dụng thép tròn với các lỗ có thể điều chỉnh và bảo vệ hiệu quả bánh xe
- Với cáp điện + dây bảo vệ, khung và cáp không bị biến dạng và cáp được bảo vệ không bị đứt



Hình 4.4 Buffet lò xò và giảm chấn

4.1.4 Thiết bị bảo vệ

- **Công tắc giới hạn hành trình hạ:** Khi Pallet xuống đến vị trí có thể chạm vào công tắc giới hạn xuống, nguồn điện truyền động sẽ bị cắt khiến pallet ngừng chuyển động



Hình 4.5 Công tắc giới hạn hành trình hạ

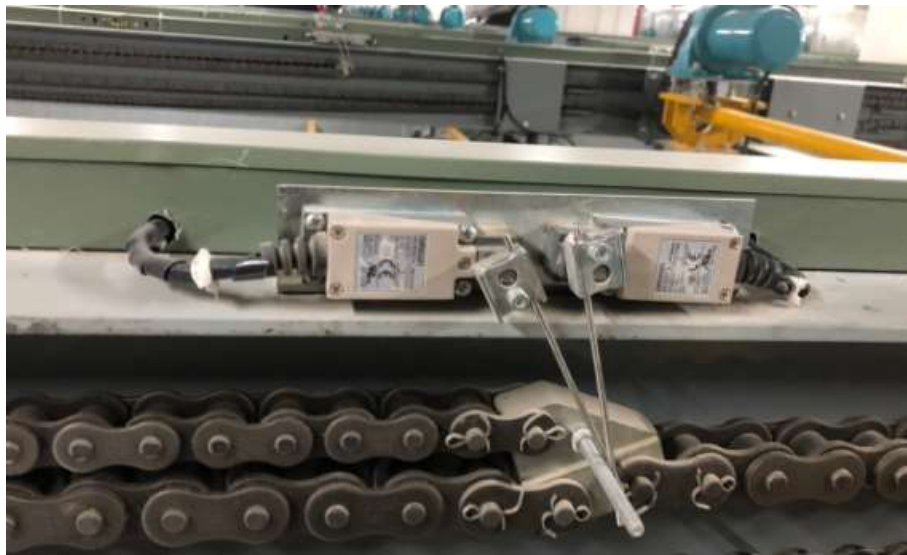
- **Móc chống rơi:** Hệ thống được trang bị móc chống rơi hoạt động bằng điện tại 4 đầu pallet. Khi không hoạt động, móc chống rơi luôn luôn ở trạng thái khóa. Tuyệt đối an toàn cho xe. Chỉ khi hệ thống hoạt động, móc chống rơi được tiếp điện, lúc này khóa mới mở ra thì xe mới có thể hạ xuống



Hình 4.6 Móc chống rơi

Khi nâng pallet, hệ thống có 3 lớp bảo vệ, bao gồm:

- **Công tắc giới hạn hành trình nâng:** Giúp dừng chuyển động khi pallet đạt đến độ cao tối đa cho phép
- **Công tắc quá giới hạn hành trình nâng (công tắc dự phòng):** Hoạt động như một lớp bảo vệ bổ sung, ngắt hệ thống nếu công tắc chính không hoạt động, ngăn ngừa việc nâng vượt quá giới hạn an toàn.



Hình 4.7 Công tắc giới hạn hành trình nâng và quá hành trình nâng

- **Công tắc chùng xích / cáp:** Khi xích bị chùng hoặc nhảy khỏi puli, thiết bị sẽ ngắt tín hiệu điều khiển/motor hoặc gửi cảnh báo để ngăn chặn tai nạn cơ khí nghiêm trọng.



Hình 4.8 Công tắc giới hạn hành trình hạ

- **Công tắc giới hạn hành trình trượt:** Khi bộ phận cơ khí trượt đến vị trí cuối hành trình, nó sẽ tác động vào công tắc, làm công tắc đóng hoặc ngắt mạch điện làm pallet dừng.



Hình 4.9 Công tắc giới hạn hành trình trượt

- **Cảm biến xâm nhập hệ thống:** Khi hệ thống đang hoạt động, nếu có người hoặc động vật xâm nhập hướng xe vào thì hệ thống dừng lại ngay lập tức và phát cảnh báo nguy hiểm.



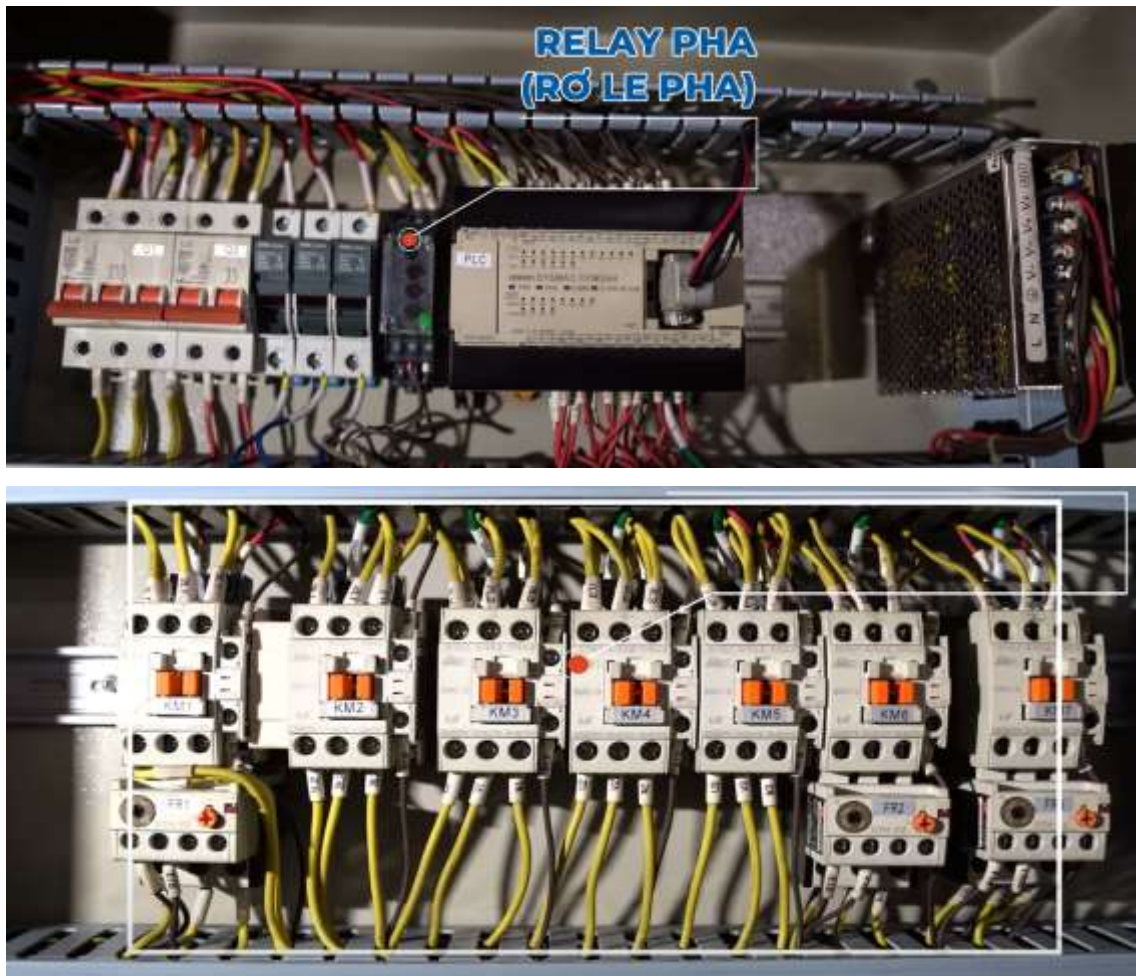
Hình 4.10 Cảm biến xâm nhập hệ thống

- **Hệ thống phanh điện từ:**
 - Chức năng chính: Phanh điện từ được tích hợp vào cơ cấu nâng hạ để giữ cố định khay đỗ xe khi motor ngừng hoạt động, ngăn chặn hiện tượng trượt hoặc rơi tự do.
 - Nguyên lý hoạt động: Khi cấp điện, phanh điện từ được giải phóng, cho phép motor hoạt động và di chuyển khay đỗ. Khi ngắt điện, phanh sẽ tự động kích hoạt, giữ chặt trục motor, đảm bảo khay đỗ đứng yên tại vị trí mong muốn.
 - Tích hợp an toàn: Phanh điện từ thường được kết hợp với các cảm biến hành trình và hệ thống điều khiển PLC để đảm bảo hoạt động chính xác và an toàn trong mọi tình huống.
- **Thanh chắn bánh sau:** Sử dụng thép tròn với các lỗ có thể điều chỉnh và bảo vệ hiệu quả



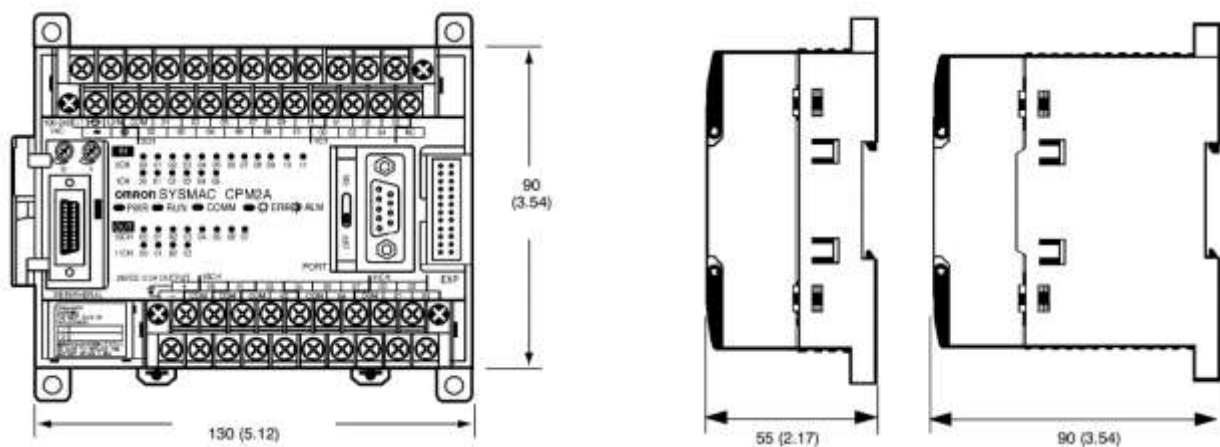
Hình 4.11 Thanh chắn bánh sau

4.1.5 Tủ điều khiển



Hình 4.12 Tổng quan của tủ điều khiển

- PLC Omron Sysmac CPM2AH-30CDR



Hình 4.13 PLC Omron CPM2AH-30CDR

Thông số kỹ thuật

Tên sản phẩm	PLC Omron CPM2AH-30CDR
Số ngõ vào, ra	30 điểm (18 input / 12 output)
Nguồn cấp	24VDC ($\pm 10\%$)
Kiểu ngõ ra	Relay output (loại R)
Tốc độ thực hiện lệnh	0.3 μ s cho lệnh cơ bản, 1.5 μ s cho lệnh phức hợp
Bộ nhớ chương trình	2K steps (khoảng 2048 bước lệnh)
Bộ đếm tốc độ cao	2 ngõ vào tốc độ cao, tối đa 2kHz
Ngõ ra xung tốc độ cao	2 kênh, tốc độ tối đa 1kHz (dạng transistor) – (Model relay không có ngõ ra xung)
Cổng truyền thông	1 cổng RS-232C (mini 8-pin), kết nối lập trình hoặc HM
Ngõ vào Analog	Không tích hợp sẵn (phải gắn thêm module)
Ngõ ra Analog	Không tích hợp sẵn (phải gắn thêm module)
Có thể gắn thêm	Module mở rộng (I/O, Analog, Communication, v.v.) qua cổng chuyên dụng
Chức năng	Lập trình ladder, chức năng logic, đếm, thời gian, truyền thông, mở rộng
Nhiệt độ làm việc	0°C ~ 55°C
Tiêu chuẩn	CE, UL, cUL, Lloyd, NK (tùy model)
Kích thước	Khoảng 90mm x 120mm x 60mm

Bảng 4.1 Thông số kỹ thuật CPM2AH-30CDR

- **MCB 3P BKN-D20**



Hình 4.14 MCB 3P BKN-D20

Thông số kỹ thuật

Hãng sản xuất	LS Electric – Hàn Quốc
Dòng sản phẩm	BKN series
Mã thiết bị	BKN-D20
Dòng định mức	20A
Số cực	3P (3 cực) – dùng cho 3 pha
Đặc tuyến bảo vệ	Loại D– dùng cho tải cảm (motor, máy bơm...)
Khả năng cắt (Icu)	6kA (theo tiêu chuẩn IEC 60898)
Điện áp định mức	400V AC / 50Hz
Tiêu chuẩn chế tạo	IEC 60898-1
Ứng dụng	Bảo vệ quá dòng, ngắn mạch cho hệ thống điện 3 pha
Xuất xứ	Hàn Quốc hoặc nhà máy LS tại Trung Quốc (tùy lô hàng)

Bảng 4.2 Thông số kỹ thuật của MCB 3P BKN-D20

- **MCB 2P BKN D6**



Hình 4.15 MCB 2P BKN D6

Thông số kỹ thuật

Hãng sản xuất	LS Electric – Hàn Quốc
Dòng sản phẩm	BKN series
Mã thiết bị	BKN-D6
Dòng định mức	6A
Số cực	2P (dùng cho mạch 1 pha 2 dây hoặc 2 pha)
Đặc tuyến bảo vệ	Loại D – chịu được dòng khởi động cao (10–20 lần I_n)
Khả năng cắt (I_{cu})	6kA (theo IEC/EN 60898-1)
Điện áp định mức	230/400VAC – 50/60Hz
Tiêu chuẩn chế tạo	IEC 60898-1
Cấp bảo vệ	IP20
Tiêu chuẩn chế tạo	IEC/EN 60898-1
Ứng dụng	Bảo vệ tải, motor nhỏ, mạch điều khiển có dòng khởi động cao
Xuất xứ	Hàn Quốc hoặc nhà máy LS tại Trung Quốc (tùy lô hàng)
Kích thước (2 module)	Rộng ~35mm x Cao ~80mm x Sâu ~68mm

Bảng 4.3 Thông số kỹ thuật của MCB 2P BKN-D6

- **Cầu chì GIGA18N-32K**



Hình 4.16 Cầu chì GiGa18N-32K

Thông số kỹ thuật

Tên sản phẩm	Cầu chì GIGA18N-32 Giga Electric
Dòng	GG18
Số cực	1P
Dòng nhiệt định mức	32A
Kích thước ruột cầu chì	22x58mm
Dòng ngắn mạch	100kA
Điện áp làm việc	500V
Lắp đặt	Din rail

Bảng 4.4 Thông số kỹ thuật của Giga18N-32K

- **Relay pha RM17TE00**



Hình 4.17 Relay pha RM17TE00

Thông số kỹ thuật

Thông số	Giá trị
Chức năng	Giám sát mất pha, đảo pha
Dòng sản phẩm	Zelio Control
Điện áp giám sát	3 pha 208 – 480 V AC
Tần số hoạt động	50/60 Hz
Điện áp cách ly định mức	4 kV giữa tiếp điểm và mạch điện áp giám sát
Chế độ bảo vệ	Mất pha, thứ tự pha sai
Ngưỡng điều chỉnh thời gian trễ	0.1 – 10 giây (biến trở mặt trước)
Tiếp điểm đầu ra	1 SPDT (1 thường mở + 1 thường đóng) – 8 A / 250 V AC
Loại tiếp điểm	Cơ khí (relay)
Nguồn cấp	Không cần cấp riêng – lấy trực tiếp từ pha giám sát
Chỉ thị LED	Có LED báo tình trạng hoạt động
Nhiệt độ làm việc	-20°C đến +50°C
Kích thước	17.5 mm (rộng) x 74 mm (cao) x 70 mm (sâu)
Lắp đặt	Gắn trên thanh DIN rail 35 mm
Tiêu chuẩn bảo vệ	IP20 (vỏ), IP30 (mặt trước)
Chứng nhận	CE, UL, CSA, EAC

Bảng 4.5 Thông số kỹ thuật của Relay pha RM17TE00

- Khởi động từ tổng LS GMC-12



Hình 4.18 Khởi động từ tổng GMC-12

Thông số kỹ thuật

Tên sản phẩm	Khởi động từ LS GMC-12
Dòng định mức (AC-3)	12A
Công suất chịu tải (3 pha 380V)	≈ 3.7 kW
Số cực chính	3 cực (L1, L2, L3 → T1, T2, T3)
Tiếp điểm phụ (thường có sẵn)	1 NO (1 tiếp điểm thường mở)
Điện áp cuộn hút (coil)	380V AC
Tần số hoạt động	50/60 Hz
Điện áp cách điện định mức (Ui)	690V AC
Điện áp chịu xung định mức (Uimp)	6 kV
Tiêu chuẩn sử dụng	IEC 60947-4-1
Cấp bảo vệ	IP20 (bên ngoài cần lắp trong tủ điện)
Gắn rơ le nhiệt	Có thể gắn rơ le nhiệt dòng tương ứng
Kích thước (W x H x D)	~45 x 77 x 87 mm (thay đổi nhẹ theo phiên bản)
Trọng lượng	Khoảng 0.4 – 0.5 kg

Bảng 4.6 Thông số kỹ thuật của LS GMC-12

- Khởi động từ cho động cơ (0.2kW) LS GMC-6



Hình 4.19 Khởi động từ LS GMC-6

Thông số kỹ thuật

Tên sản phẩm	Khởi động từ LS GMC-6
Dòng định mức (AC-3)	6 A
Công suất chịu tải (AC-3, 380V)	Tối đa ~1.5 kW
Điện áp cách điện định mức (Ui)	690V AC
Điện áp chịu xung (Uimp)	6 kV
Cuộn hút (coil voltage)	380V AC
Tần số hoạt động	50/60 Hz
Tiếp điểm chính	3 cực chính (L1, L2, L3 → T1, T2, T3)
Tiếp điểm phụ mặc định	1 NO (1 thường mở)
Nhiệt độ môi trường làm việc	-5°C đến +60°C
Gắn rơ le nhiệt	Có thể gắn MT-6 hoặc MT-32 tương ứng
Cấp bảo vệ	IP20
Kích thước (W x H x D)	Khoảng 45 x 77 x 80 mm (rất nhỏ gọn)
Trọng lượng	~0.3 – 0.4 kg
Tiêu chuẩn áp dụng	IEC 60947-4-1 / KS C IEC 60947-4-1

Bảng 4.7 Thông số kỹ thuật của LS GMC-6

- **Khởi động từ cho động cơ nâng hạ (2.2kW) LS GMC-9**



Hình 4.20 Khởi động từ LS GMC-9

Thông số kỹ thuật

Tên sản phẩm	Khởi động từ LS GMC-9
Hãng sản xuất	LS Electric (Hàn Quốc)
Dòng định mức (AC-3)	9 A
Công suất chịu tải (AC-3, 3 pha 380V)	Tối đa ~2.2 kW
Công suất chịu tải (1 pha 220V)	~0.75 – 1.1 kW
Điện áp cách điện định mức (Ui)	690 V AC
Điện áp chịu xung (Uimp)	6 kV
Tiếp điểm chính	3 cực chính (L1, L2, L3 → T1, T2, T3)
Tiếp điểm phụ mặc định	1 NO (1 thường mở)
Điện áp cuộn hút (coil)	380V AC
Tần số hoạt động	50/60 Hz
Khả năng gắn rơ le nhiệt	Dùng loại rơ le nhiệt MT-32 (hoặc MT-12)
Cấp bảo vệ	IP20
Nhiệt độ làm việc	-5°C đến +60°C
Kích thước (W x H x D)	~45 x 77 x 87 mm
Trọng lượng	~0.4 – 0.5 kg
Tiêu chuẩn áp dụng	IEC 60947-4-1

Bảng 4.8 Thông số kỹ thuật của LS GMC-9

- Rơ le nhiệt cho động cơ nâng hạ (2.2kW) LS GTH-22



Hình 4.21 Rơ le nhiệt GTH-22

Thông số kĩ thuật

Tên sản phẩm	Rơ le nhiệt LS GTH-22
Dải dòng chỉnh được	Từ 0.63 A đến 1 A
Điện áp cách điện định mức (U _i)	690V AC
Điện áp làm việc định mức (U _e)	380V – 415V AC
Số cực	3 cực chính
Tiếp điểm phụ	1 NO + 1 NC
Dòng cắt tức thời (I _{th})	~10 lần dòng định mức (tùy thời gian)
Loại reset	Tùy chọn: Manual (tay) hoặc Auto (tự động)
Chức năng bảo vệ	Quá tải, mất pha
Nhiệt độ làm việc	-5°C đến +60°C
Cấp bảo vệ	IP20
Tuổi thọ cơ khí / điện	~10 ⁶ lần đóng ngắt / ~10 ⁵ lần điện
Lắp kèm khởi động từ	GMC-6, GMC-9, GMC-12 (gắn trực tiếp được)
Tiêu chuẩn áp dụng	IEC 60947-4-1

Bảng 4.9 Thông số kĩ thuật của GTH-22

- Nguồn nuôi PLC nguồn tổ ong 24V



Hình 4.22 Nguồn tổ ong 24V

Thông số kỹ thuật

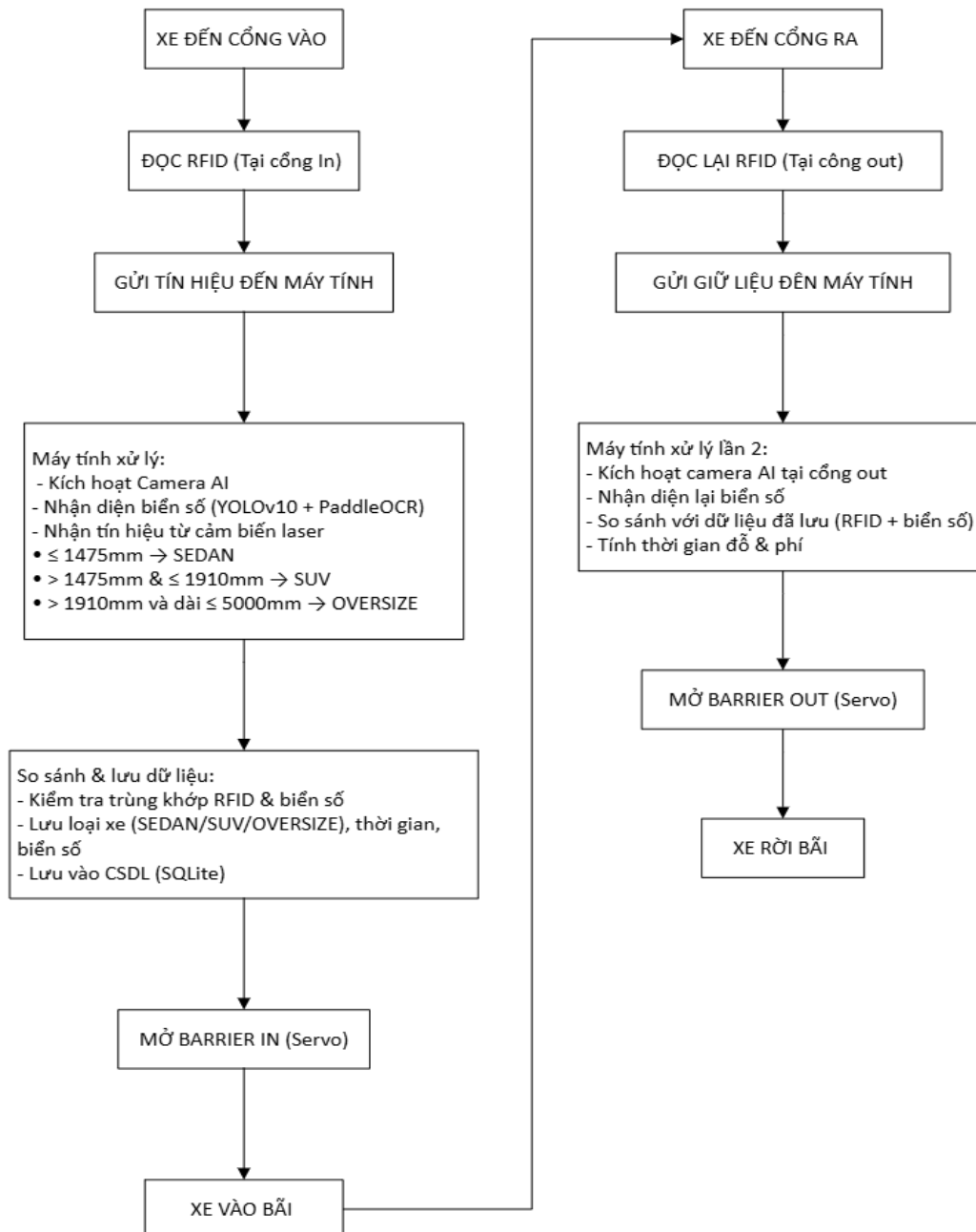
Hãng sản xuất	MZMW
Model	SV-250-24
Điện áp đầu vào	AC 380V (340-440V)
Điện áp đầu ra	DC 24V
Công suất	250W
Dòng điện đầu ra	10.4A
Cấp bảo vệ	- Bảo vệ quá dòng - Bảo vệ quá áp - Bảo vệ ngắn mạch - Lọc nhiễu EMI chống nhiễu tốt, độ ổn định cao

Bảng 4.10 Thông số kỹ thuật nguồn nuôi PLC

CHƯƠNG 5: THIẾT KẾ PHẦN MỀM PHÂN LOẠI XE VÀ NHẬN DIỆN BIỂN SỐ

5.1 Tổng quan kiến trúc phần mềm

5.1.1 Sơ đồ tổng thể hệ thống phần mềm



Hình 5.1 Sơ đồ tổng thể hệ thống phần mềm

Giai đoạn xe vào bãi (Cổng vào)

Khi xe đi đến cổng IN của bãi đỗ, hệ thống được kích hoạt để bắt đầu quá trình nhận diện và phân loại phương tiện. Lúc này, người điều khiển phương tiện phải dừng lại để hệ thống thực hiện các bước xử lý tiếp theo.

- ❖ Đọc thẻ RFID (tại cổng IN): Hệ thống sử dụng đầu đọc RFID HS0430B để nhận diện thẻ của phương tiện. Mỗi phương tiện được cấp một thẻ RFID riêng biệt có chứa mã UID duy nhất.
 - Tại cổng vào, đầu đọc RFID đọc dữ liệu UID từ thẻ.
 - Tín hiệu này được truyền đến vi điều khiển trung tâm của hệ thống.
- ❖ Gửi tín hiệu UID đến máy tính: Vi điều khiển phải xử lý tín hiệu từ đầu đọc và gửi chuỗi UID lên máy tính thông qua cổng Serial (UART).
 - Trong giai đoạn mô phỏng, phần mềm com0com được sử dụng để tạo cổng COM ảo, kết nối Arduino (Proteus) với phần mềm xử lý trên máy tính.
- ❖ Xử lý tại phần mềm máy tính: Sau khi nhận tín hiệu UID, phần mềm máy tính thực hiện các bước sau:
 - Kích hoạt Camera AI tại cổng IN: Camera được bật để chụp ảnh biển số của phương tiện.
 - Nhận diện biển số: Sử dụng mô hình AI YOLOv10 để phát hiện biển số, sau đó dùng PaddleOCR để nhận dạng ký tự trong biển số.
 - Tiếp nhận tín hiệu cảm biến chiều cao: Cảm biến laser hoặc IR (trong mô phỏng) cung cấp dữ liệu chiều cao xe. Dựa trên chiều cao này, hệ thống sẽ phân loại phương tiện:
 - Chiều cao $\leq 1475\text{mm}$ \rightarrow SEDAN
 - $1475\text{mm} < \text{chiều cao} \leq 1910\text{mm}$ \rightarrow SUV
 - Chiều cao $> 1910\text{mm}$ và chiều dài $\leq 5000\text{mm}$ \rightarrow OVERSIZE
- ❖ So sánh và lưu trữ dữ liệu
 - Hệ thống kiểm tra sự trùng khớp giữa UID RFID và biển số xe (để chống gian lận).
 - Ghi lại thông tin: Mã UID, Loại xe (SEDAN/SUV), Biển số xe và Thời gian vào bãi
 - Lưu toàn bộ dữ liệu vào Cơ sở dữ liệu SQLite để phục vụ cho xử lý tại cổng OUT.
- ❖ Điều khiển mở barrier IN: Sau khi xác thực thành công:
 - Phần mềm gửi tín hiệu điều khiển đến Arduino.

- Arduino kích hoạt servo motor mở barrier IN để phương tiện đi vào bãi.
- Barrier tự động đóng lại sau khoảng thời gian định trước.

Giai đoạn xe rời bãi (cổng OUT)

Khi phương tiện di chuyển đến cổng OUT, hệ thống bắt đầu quy trình xác thực lần 2.

❖ Đọc lại thẻ RFID (tại cổng OUT)

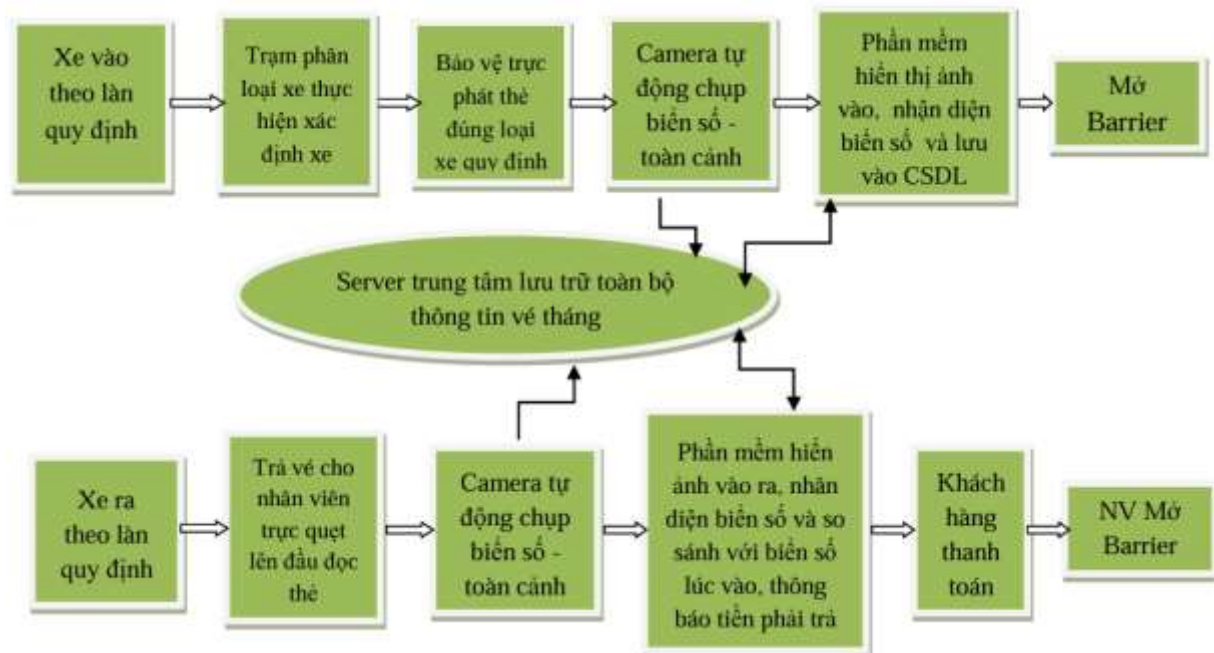
- Đầu đọc RFID tiếp tục quét UID từ thẻ của phương tiện.
- Dữ liệu UID được gửi về máy tính qua Arduino giống như ở bước vào bãi.

❖ Xử lý tại máy tính

- Kích hoạt lại Camera AI tại cổng OUT để chụp và nhận diện lại biển số xe.
- Phần mềm thực hiện:
 - So sánh UID và biển số với dữ liệu đã lưu từ khi xe vào bãi.
 - Nếu khớp, tính thời gian đỗ xe (thời gian vào – ra).
 - Tính phí đỗ xe nếu có quy định (trong phiên bản thực tế).

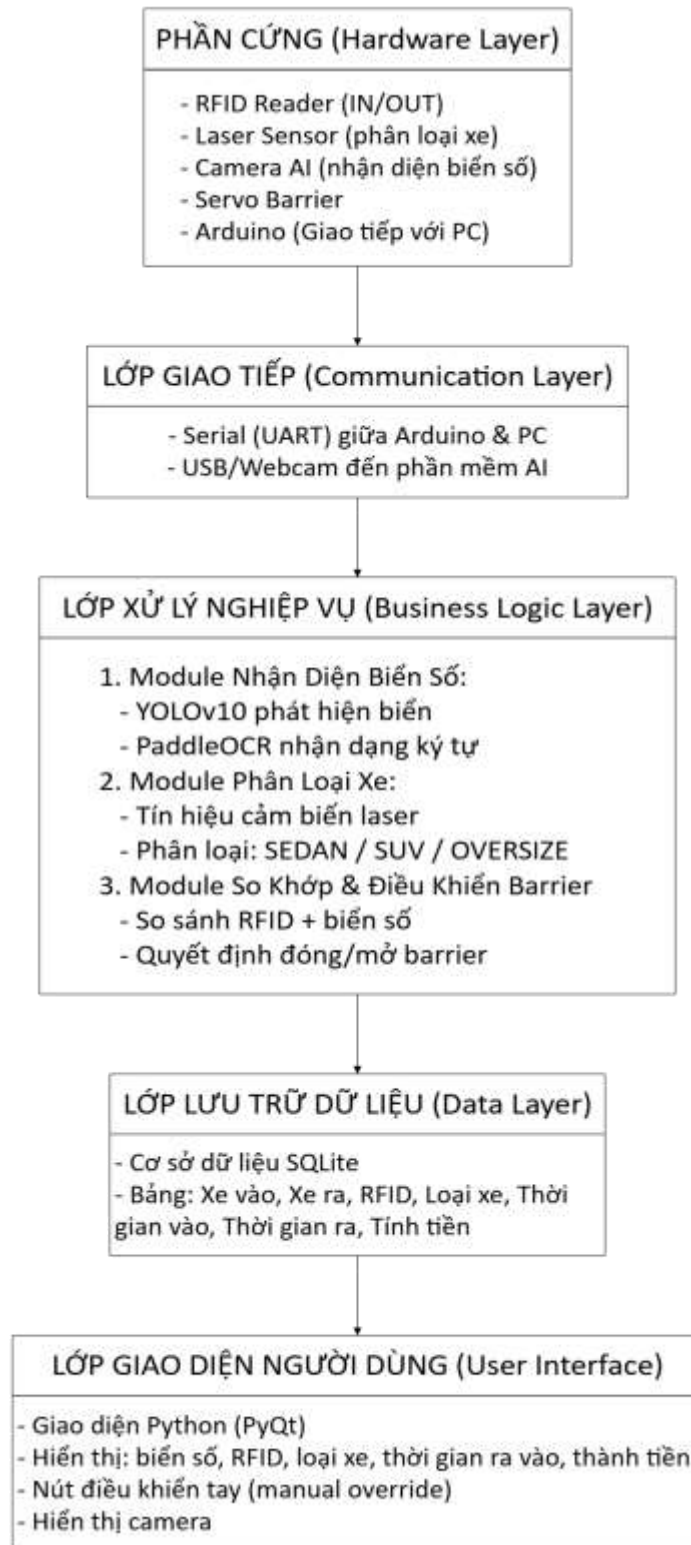
❖ Điều khiển mở barrier OUT

- Khi dữ liệu hợp lệ, phần mềm gửi tín hiệu điều khiển mở barrier OUT.
- Arduino kích hoạt servo để mở barrier cho xe rời bãi.



Hình 5.2 Quy trình của hệ thống

5.1.2 Mô hình phân lớp (Layered Architecture)



Hình 5.3 Mô hình phân lớp

Hệ thống được thiết kế theo mô hình phân lớp, bao gồm 5 lớp chính: Lớp Phần Cứng, Lớp Giao Tiếp, Lớp Xử Lý Nghiệp Vụ, Lớp Lưu Trữ Dữ Liệu và Lớp Giao Diện Người Dùng.

Lớp Phần Cứng (Hardware Layer)

Lớp này bao gồm các thiết bị vật lý chịu trách nhiệm thu thập dữ liệu và điều khiển các thành phần cơ điện:

- RFID Reader (IN/OUT): Thiết bị đọc thẻ RFID, được sử dụng để xác định các phương tiện ra/vào.
- Laser Sensor (phân loại xe): Cảm biến laser dùng để đo kích thước và phân loại xe (ví dụ: SEDAN, SUV, OVERSIZE).
- Camera AI (nhận diện biển số): Camera được tích hợp trí tuệ nhân tạo để chụp ảnh và nhận diện biển số xe.
- Servo Barrier: Rào chắn tự động, được điều khiển để đóng/mở lối đi.
- Vi điều khiển (Giao tiếp với PC): Vi điều khiển đóng vai trò cầu nối giao tiếp giữa các cảm biến, thiết bị ngoại vi và máy tính cá nhân (PC).

Lớp Giao Tiếp (Communication Layer)

Lớp này quản lý việc truyền thông giữa các thành phần phần cứng và phần mềm, cũng như giữa các module phần mềm:

- Serial (UART) giữa vi điều khiển & PC: Giao tiếp nối tiếp qua UART (Universal Asynchronous Receiver/Transmitter) để truyền dữ liệu và lệnh điều khiển giữa Arduino và PC.
- USB/Webcam đến phần mềm AI: Giao tiếp qua cổng USB hoặc sử dụng webcam để truyền hình ảnh từ camera đến phần mềm AI xử lý nhận diện biển số.

Lớp Xử Lý Nghiệp Vụ (Business Logic Layer)

Đây là "bộ não" của hệ thống, nơi thực hiện các thuật toán và quy tắc nghiệp vụ chính:

- Module Nhận Diện Biển Số:
 - YOLOv10 phát hiện biển: Sử dụng mô hình YOLOv10 (You Only Look Once phiên bản 10) để phát hiện vị trí biển số trong hình ảnh.
 - PaddleOCR nhận dạng ký tự: Áp dụng công nghệ PaddleOCR để nhận dạng các ký tự trên biển số đã được phát hiện.
- Module Phân Loại Xe:
 - Tín hiệu cảm biến laser: Sử dụng dữ liệu từ cảm biến laser để xác định kích thước xe.

- Phân loại: Thực hiện phân loại xe thành các loại như SEDAN / SUV / OVERSIZE dựa trên dữ liệu cảm biến.
- Module So Khớp & Điều Khiển Barrier:
 - So sánh RFID + biển số: So sánh thông tin RFID của xe với biển số đã nhận diện để xác minh.
 - Quyết định đóng/mở barrier: Dựa trên kết quả so sánh và các quy tắc nghiệp vụ (ví dụ: xe đã đăng ký, thanh toán hợp lệ), hệ thống sẽ quyết định điều khiển rào chắn đóng hoặc mở.

Lớp Lưu Trữ Dữ Liệu (Data Layer)

Lớp này chịu trách nhiệm lưu trữ và quản lý dữ liệu của hệ thống:

- Cơ sở dữ liệu SQLite: Sử dụng SQLite, một hệ quản trị cơ sở dữ liệu nhẹ, để lưu trữ thông tin.
- Bảng:
 - Xe vào: Lưu trữ thông tin chi tiết về xe khi vào.
 - Xe ra: Ghi lại thông tin khi xe rời đi.
 - Loại xe: Danh mục hoặc thông tin chi tiết về các loại xe.
 - Thời gian vào: Thời điểm xe đi vào.
 - Thời gian ra: Thời điểm xe rời đi.
 - Tính tiền: Thông tin liên quan đến việc tính toán phí.

Lớp Giao Diện Người Dùng (User Interface)

Lớp này cung cấp giao diện để người dùng tương tác với hệ thống:

- Giao diện Python (PyQt): Phát triển giao diện người dùng bằng thư viện PyQt của Python, cho phép tạo các ứng dụng đồ họa đẹp mắt và chức năng.
- Hiển thị:
 - Biển số: Hiển thị biển số xe đã được nhận diện.
 - RFID: Hiển thị thông tin RFID.
 - Loại xe: Hiển thị loại xe đã được phân loại.
 - Thời gian ra vào: Cung cấp thông tin thời gian xe vào và ra.
 - Thành tiền: Hiển thị số tiền cần thanh toán (nếu có).
- Nút điều khiển tay (manual override): Cung cấp khả năng can thiệp thủ công, cho phép người vận hành điều khiển rào chắn hoặc các chức năng khác trong trường hợp khẩn cấp hoặc cần thiết.

- **Hiện thị camera:** Trực tiếp hiển thị hình ảnh từ camera để người dùng có thể quan sát.

5.2 Mô hình phân loại xe và đọc thẻ RFID

5.2.1 Mô tả quy trình phân loại

- **Mục tiêu:** Phân loại các phương tiện thành các nhóm như SEDAN, SUV, hoặc OVERSIZE dựa trên chiều cao của xe khi đi qua trạm kiểm soát, phục vụ việc điều phối đỗ xe, kiểm soát ra vào và tính phí phù hợp.

- Quy trình phân loại gồm các bước sau:

- ❖ **Bước 1: Phát hiện phương tiện đến.**

Khi có xe di chuyển đến vị trí kiểm soát đầu vào, cảm biến hồng ngoại (IR) sẽ phát hiện vật cản và kích hoạt hệ thống.

- ❖ **Bước 2: Đo chiều cao xe bằng cảm biến laser.**

Cảm biến laser được gắn cố định phía trên đầu vào sẽ đo chiều cao xe.

Dữ liệu chiều cao được gửi về vi điều khiển.

- ❖ **Bước 3: Phân loại theo ngưỡng.**

Arduino sẽ so sánh chiều cao đo được với các ngưỡng xác định trước:

Nếu chiều cao ≤ 1475 mm \rightarrow Loại: SEDAN

Nếu chiều cao > 1475 mm và ≤ 1910 mm \rightarrow Loại: SUV

Nếu chiều cao > 1910 mm và chiều dài ≥ 5000 mm \rightarrow Loại: OVERSIZE

- ❖ **Bước 4: Gửi kết quả lên máy tính.**

Kết quả phân loại được gửi lên phần mềm giao diện trên máy tính qua cổng Serial .

Dữ liệu bao gồm: Loại xe (SEDAN/SUV).

Thời điểm đọc.

Giá trị chiều cao thực tế (mm).

- ❖ **Bước 5: Hiện thị và lưu trữ.**

Phần mềm trên máy tính nhận dữ liệu và hiển thị loại xe lên GUI.

Loại xe cùng các thông tin liên quan sẽ được lưu vào Cơ sở dữ liệu (SQLite) để kết hợp với thông tin nhận diện biển số và RFID sau đó.

5.2.2 Sử dụng cảm biến IR33 để phân loại.

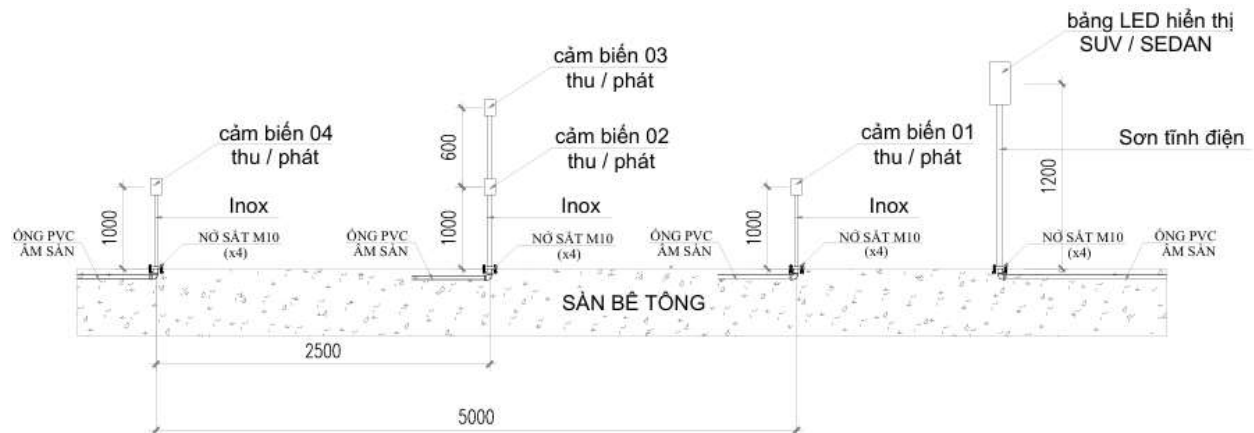
Trong hệ thống phân loại xe, cảm biến hồng ngoại IR33 đóng vai trò then chốt để xác định loại phương tiện dựa vào kích thước thực tế. Phương pháp sử dụng cảm biến hồng

ngoại giúp đảm bảo tính tự động, nhanh chóng và độ chính xác cao, đồng thời loại bỏ sự phụ thuộc vào yếu tố con người.

Nguyên lý hoạt động của cảm biến hồng ngoại IR33

Hệ thống sử dụng cảm biến hồng ngoại IR33 được lắp cố định ở phía trên làn xe, hướng xuống mặt đất. Khi một phương tiện tiến vào khu vực kiểm soát, cảm biến sẽ đo khoảng cách từ cảm biến đến điểm cao nhất của xe.

Thiết lập ngưỡng phân loại



Hình 5.4 Vị trí đặt các cảm biến hồng ngoại phân loại xe

Loại xe	Các cảm biến kích hoạt
SEDAN	Cảm biến 1 và cảm biến 2
SUV	Cảm biến 1, cảm biến 2 và Cảm biến 3
OVERSIZE	Tất cả các cảm biến hoặc cảm biến 1, 2, 4 (Pallet không quá 5m)

Bảng 5.1 Bảng các ngưỡng phân loại xe

Các ngưỡng được thiết lập dựa trên quy chuẩn kích thước thực tế của xe tại Việt Nam và kinh nghiệm từ thực tiễn hệ thống bãi đỗ.

Giao tiếp với hệ thống điều khiển

Khi cảm biến đo được chiều cao, giá trị này được truyền về bộ điều khiển phân loại xe KZ-VRC.NET, sau đó được xử lý và phân loại theo ngưỡng. Kết quả phân loại được gửi qua cổng Serial (UART) đến phần mềm máy tính.

Thông tin gửi bao gồm: Mã loại xe (SEDAN, SUV)

Khi hệ thống báo OVERSIZE thì cảnh báo cho khách hàng bãi đỗ xe không nhận xe vào bãi vì hệ thống đỗ xe cơ khí có PALLET chỉ dài 5m không chứa đủ kích thước xe.

5.2.3 Kết nối phần cứng IR33 và vi điều khiển

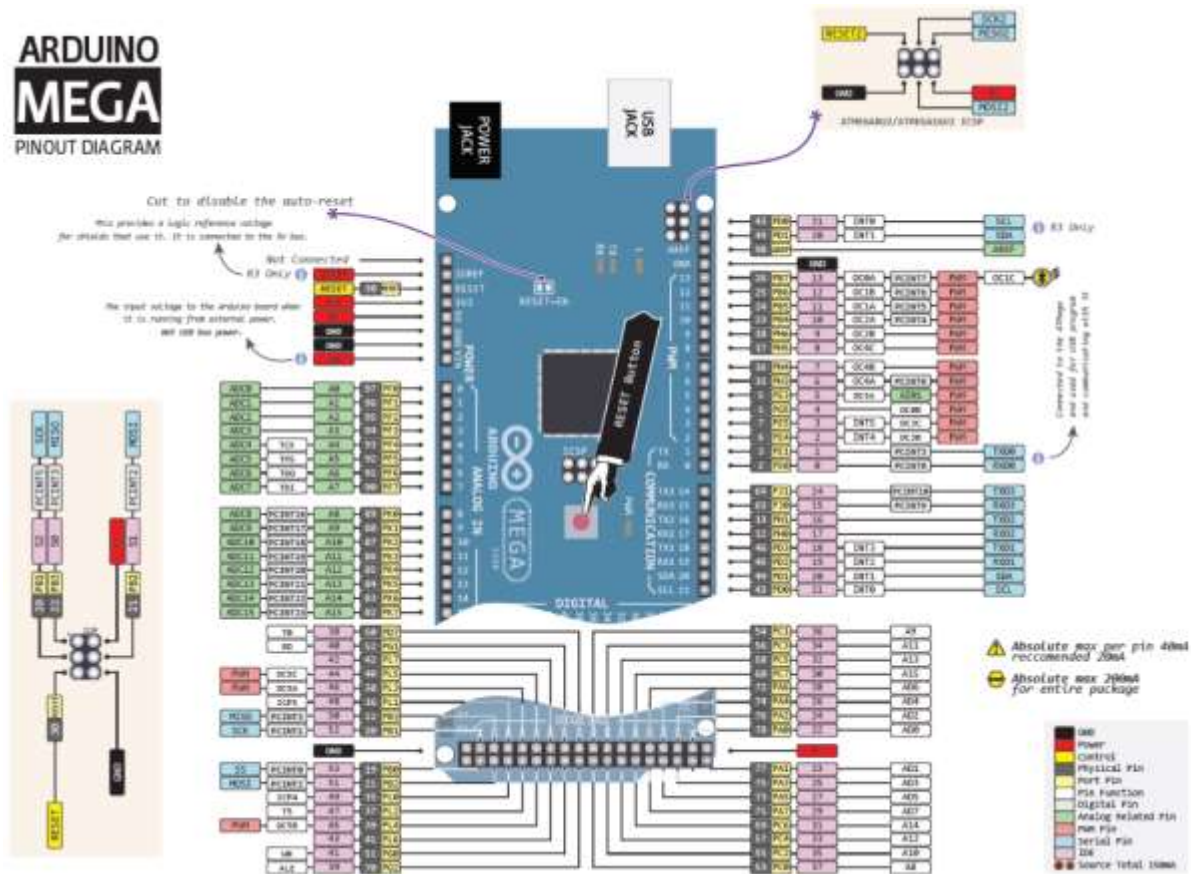
Trong giai đoạn thử nghiệm hệ thống phân loại xe, nhóm ban đầu sử dụng cảm biến thực tế IR33 kết hợp với bộ điều khiển KZ-VRC.NET. Đây là các thiết bị chuyên dụng do công ty KZtek thiết kế nhằm phục vụ cho việc phân loại phương tiện giao thông trong các hệ thống bãi đỗ xe thông minh. Tuy nhiên, do các thiết bị này là sản phẩm độc quyền, không được cung cấp công khai và không có mô hình mô phỏng tích hợp sẵn trong phần mềm Proteus, nhóm đã quyết định sử dụng Arduino MEGA 2560 để giả lập chức năng của bộ điều khiển KZ-VRC.NET trong quá trình mô phỏng và kiểm thử hệ thống.

Để mô phỏng tín hiệu đầu vào từ cảm biến chiều cao, nhóm sử dụng mô-đun IR Infrared Obstacle Avoidance Sensor, thiết bị này có sẵn trong thư viện Proteus và có khả năng phát hiện vật cản tương tự cảm biến chiều cao trong thực tế. Bằng cách bố trí các cảm biến ở các vị trí cao thấp khác nhau, Arduino MEGA sẽ đọc các tín hiệu từ cảm biến và xác định chiều cao xe để phân loại thành SEDAN, SUV hoặc OVERSIZE theo nguyên tắc định sẵn.

Song song với quá trình phân loại xe, hệ thống còn được tích hợp mô-đun RFID HS0430B nhằm quản lý thông tin phương tiện. Thẻ RFID gắn trên xe sẽ được đầu đọc thu nhận, sau đó dữ liệu UID của thẻ được vi điều khiển xử lý và gửi đến phần mềm máy tính để kiểm tra, đối chiếu và lưu trữ. Arduino MEGA chịu trách nhiệm giao tiếp qua giao thức UART (qua cổng Serial) để truyền dữ liệu UID về máy tính, nơi phần mềm quản lý tiếp nhận và thực hiện các bước xác thực cần thiết như đối chiếu biển số, kiểm tra thời gian vào/ra hoặc điều khiển rào chắn (barrier).

Giới thiệu về Arduino MEGA 2560

Arduino MEGA 2560 là một vi điều khiển mở rộng thuộc dòng Arduino, được phát triển dựa trên vi điều khiển ATmega2560 của hãng Atmel (nay là Microchip). Đây là phiên bản có nhiều chân I/O nhất trong họ Arduino phổ thông, phù hợp với các dự án cần xử lý nhiều tín hiệu vào/ra cùng lúc như: robot phức tạp, hệ thống điều khiển đa thiết bị, hay hệ thống phân loại xe như trong đề tài.

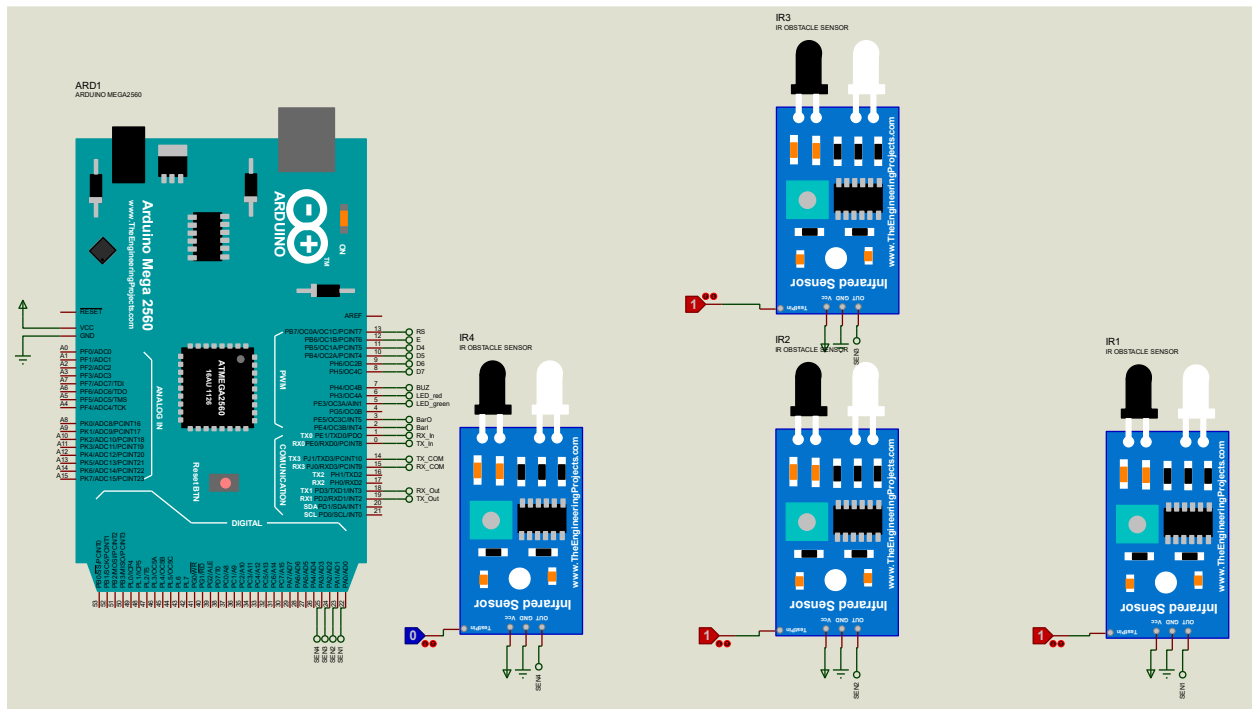


Hình 5.5 Sơ đồ chân trong Arduino MEGA 2560

Arduino Megas có sơ đồ linh kiện như các hình trên:

- 5 Chân GND
- 3 chân 5V
- 1 chân 3.3v
- 1 nút reset
- 16 chân analog
- 4 chân UART
- 54 Chân digital trong đó có 15 chân chúng ta có thể sử dụng như PWM
- 6 Chân lập trình ISP

Sơ đồ kết nối IR Infrared Obstacle Avoidance Sensor và Arduino MEGA 2560.



Hình 5.6 Sơ đồ kết nối trên mô phỏng

Bảng kết nối chân tín hiệu IR Infrared Obstacle Avoidance Sensor và Arduino MEGA 2560

IR Infrared Obstacle Avoidance Sensor	Arduino MEGA 2560
SENSOR 1	Chân 22
SENSOR 2	Chân 23
SENSOR 3	Chân 24
SENSOR 4	Chân 25

Bảng 5.2 Bảng kết nối chân tín hiệu của IR với Arduino MEGA 2560

5.3 Giao tiếp và Xử lý tín hiệu RFID

5.3.1 RFID IC CARD / Tag Reader HS0430B

Hệ thống sử dụng đầu đọc RFID HS0430B hoạt động ở tần số 125kHz, giao tiếp bằng UART (Serial) để nhận dạng phương tiện. Mỗi thẻ RFID có một mã số duy nhất (UID), được sử dụng để xác định xe khi vào và ra bãi đỗ xe.



Hình 5.7 Đầu đọc thẻ RFID (HS0430B)

Thông số kỹ thuật

- Điện áp sử dụng: 5VDC từ cổng USB
- Tương thích: Thẻ RFID IC Card / Tag
- Tần số hoạt động: 125 kHz
- Khoảng cách nhận thẻ: 8~15cm
- Chuẩn giao tiếp : UART
- Chuẩn chống nước : IP54
- Nhiệt độ hoạt động : -40 ~ 80 độ C
- Độ ẩm hoạt động : 5% - 95%
- Hệ điều hành hỗ trợ : Windows, Linux,...
- Kích thước : 96 x 60 x 11mm

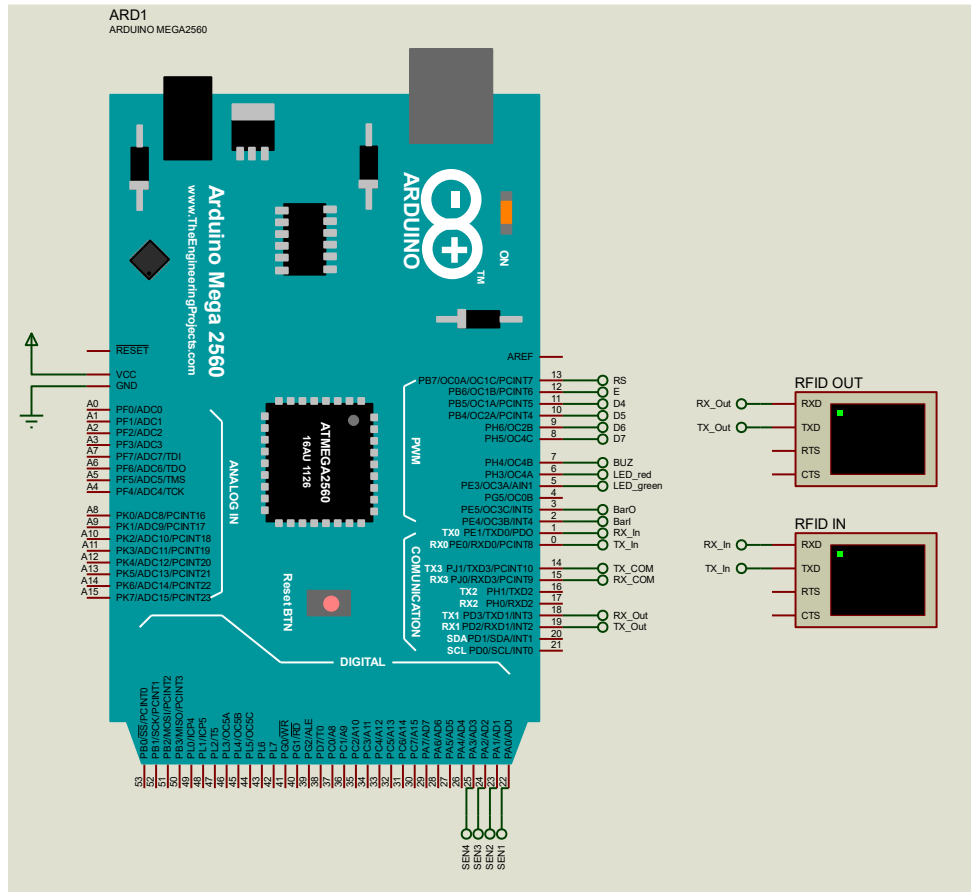
5.3.2 Kết nối phần cứng RFID và vi điều khiển

Sơ đồ chân nối giữa HS0430B và Arduino MEGA2560

- Hệ thống chỉ mô phỏng nên thay vì sử dụng HS0430B ta sử dụng VIRTUAL TERMINAL thay cho thiết bị RFID. Virtual Terminal trong Proteus đóng vai trò như một thiết bị UART gửi dữ liệu đến Arduino. Chúng ta có thể gõ tay vào Virtual Terminal chuỗi UID, ví dụ: 0A0045F23B, giống như HS0430B sẽ gửi thực tế.

RFID	Arduino MEGA 2560
HS0430B (IN) TX	RX0 (Chân 0)
HS0430B (OUT) TX	RX1 (Chân 19)

Bảng 5.3 Kết nối RFID với Arduino MEGA 2560



Hình 5.8 Sơ đồ kết nối trên mô phỏng

5.3.3 Xử lý UID từ RFID và gửi lên máy tính

Mục tiêu của quá trình xử lý UID

Mục tiêu chính của phần này là xử lý chuỗi UID (Unique Identifier) từ thẻ RFID đọc được bởi vi điều khiển (Arduino Mega 2560), sau đó gửi chuỗi dữ liệu này tới máy tính để thực hiện:

- Kiểm tra tính hợp lệ của thẻ
- Ghi nhận thông tin xe ra vào
- Điều khiển đóng/mở barrier thông qua phản hồi từ máy tính

Quy trình xử lý dữ liệu UID

Toàn bộ quá trình xử lý và truyền dữ liệu diễn ra như sau:

Bước 1: Nhận dữ liệu từ RFID

- Khi người dùng quét thẻ RFID, mô-đun HS0430B gửi chuỗi dữ liệu UID qua cổng Serial1 (UART1) của Arduino Mega 2560.
- Chuỗi này có độ dài cố định (thường 10 ký tự), dạng ký tự ASCII, ví dụ: "3A00B5C2F1".

Bước 2: Kiểm tra và chuẩn hóa chuỗi UID

- Arduino sẽ đọc toàn bộ chuỗi từ Serial (RFID IN) hoặc Serial1 (RFID OUT), loại bỏ ký tự không cần thiết như \r hoặc \n.
- Có thể áp dụng bước kiểm tra định dạng UID (ví dụ: chỉ chấp nhận ký tự số và chữ cái in hoa, độ dài = 10).

Bước 3: Gửi dữ liệu lên máy tính qua Serial3

- Arduino Mega sử dụng cổng Serial3 (UART3) kết nối với máy tính thông qua USB (chân TX3 – RX3).
- Sau khi đọc thành công UID, Arduino sử dụng lệnh Serial.print() để truyền chuỗi lên máy tính.
- Máy tính (phần mềm Python hoặc giao diện GUI) sẽ lắng nghe cổng COM để nhận dữ liệu.

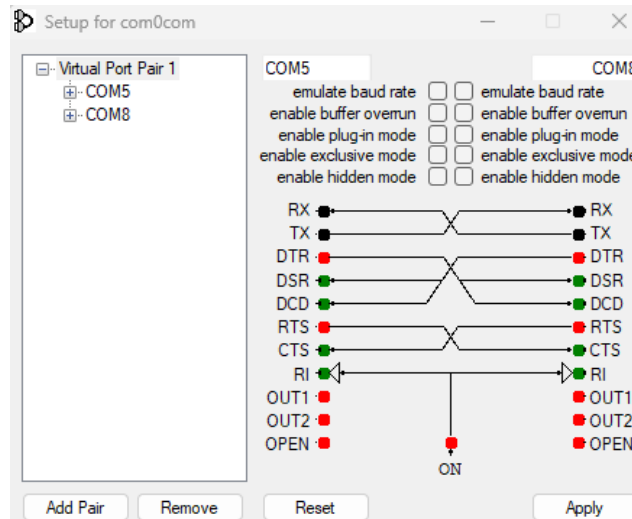
Kết nối Arduino MEGA 2560 đến phần mềm quản lý bãi đỗ xe thông minh

Giới thiệu com0com:

- Trong quá trình phát triển và mô phỏng hệ thống quản lý bãi đỗ xe thông minh, thay vì sử dụng thiết bị phần cứng thực tế, nhóm sử dụng công cụ com0com để giả lập cổng COM ảo. Đây là một trình điều khiển mã nguồn mở cho Windows, cho phép tạo hai cổng COM nối với nhau, từ đó mô phỏng quá trình giao tiếp giữa Arduino và phần mềm quản lý mà không cần kết nối vật lý.

Sử dụng com0com giúp nhóm thực hiện:

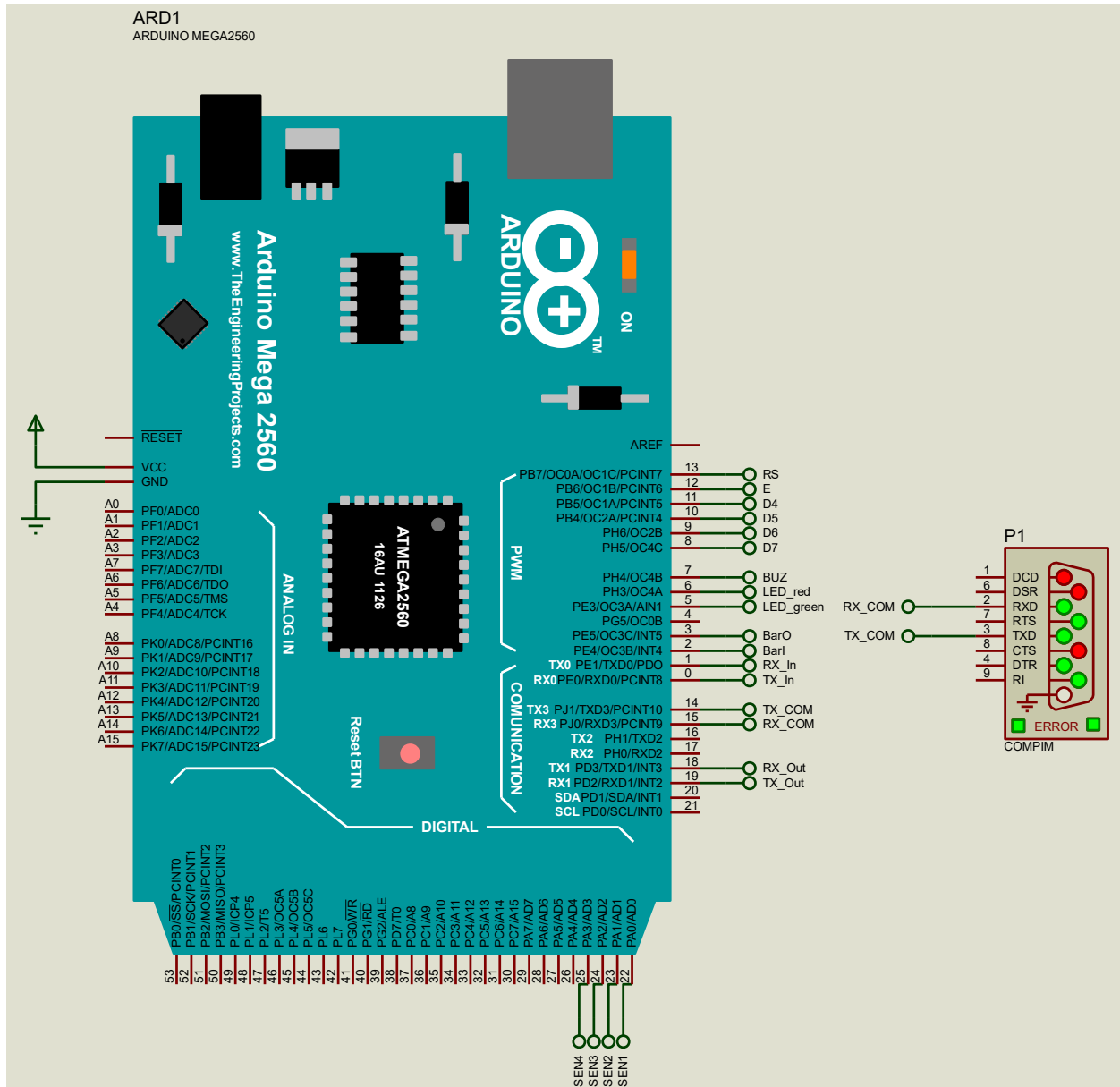
- Mô phỏng quá trình truyền dữ liệu từ Arduino đến phần mềm quản lý và ngược lại.
- Kiểm tra các chức năng truyền – nhận thông tin RFID, lệnh mở barrier mà không cần sử dụng phần cứng thực.
- Hỗ trợ quá trình lập trình và gỡ lỗi phần mềm trên máy tính một cách linh hoạt.



Hình 5.9 Phần mềm giả lập cổng Com com0com

Nguyên lý hoạt động của com0com

- com0com tạo một cặp cổng COM ảo, ví dụ: COM5 \Leftrightarrow COM8.
- Khi một ứng dụng (như Arduino IDE Serial Monitor, hoặc mô phỏng từ Proteus) gửi dữ liệu đến COM5, ứng dụng khác (ví dụ phần mềm quản lý viết bằng Python/C#) có thể đọc dữ liệu đó từ COM8 – và ngược lại.
- Điều này mô phỏng hoàn hảo quá trình giao tiếp UART giữa hai thiết bị thật.



Hình 5.10 Kết nối Arduino MEGA 2560 đến Com 5

Việc sử dụng com0com giúp hệ thống được kiểm thử hiệu quả, linh hoạt và tiết kiệm chi phí phần cứng. Trong giai đoạn phát triển, đây là công cụ không thể thiếu để đảm bảo tính đúng đắn và ổn định của phần mềm trước khi triển khai trên hệ thống thật.

5.4 Phần mềm nhận diện biển số

5.4.1 Giới thiệu về mô hình YOLOv10 và PaddleOCR

Mô hình YOLOv10:

Tổng quan:

Được phát triển bởi nhóm nghiên cứu Ultralytics, YOLOv10 là một thuật toán thị giác máy tính được sử dụng để phát hiện đối tượng dựa trên mạng thần kinh tích chập (convolutional neural network – CNN). YOLOv10 tiếp tục hướng tới việc cân bằng giữa tốc độ và độ chính xác, và đặc biệt tối ưu hơn cho các ứng dụng real-time object detection trên thiết bị edge hoặc cloud.

Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh.

Đặc điểm:

YOLOv10 là phiên bản nâng cấp của phiên bản trước đó là YOLOv8, nhanh hơn nhưng vẫn giữ được độ chính xác cao, hỗ trợ huấn luyện nhanh chóng, đơn giản hơn. Có thể dễ dàng triển khai trong các dự án thực tế như:

- Nhận diện biển số xe & kiểm soát bãi đỗ xe
- Giám sát và kiểm tra chất lượng trong công nghiệp
- Hệ thống giám sát an ninh & nhận diện khuôn mặt
- Robotics & Xe tự hành



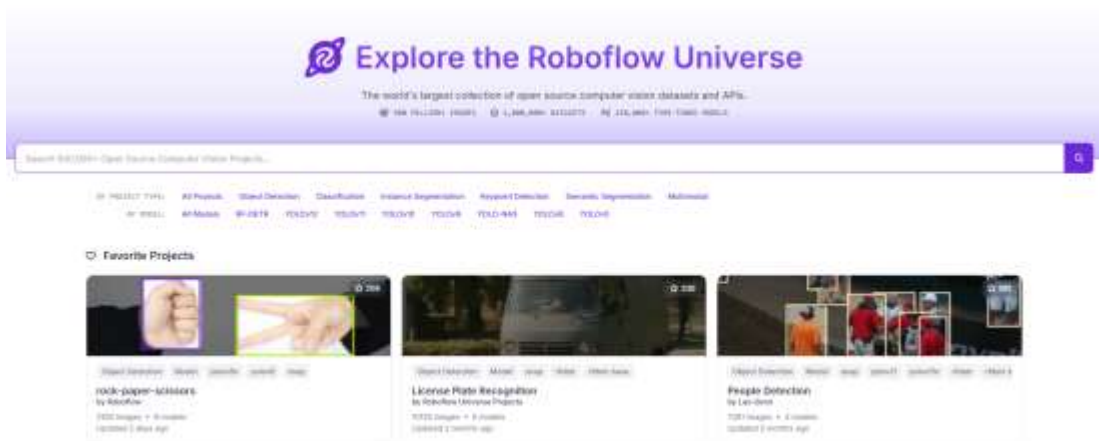
Hình 5.11 Biển số xe được phát hiện dùng YOLOv10

Các bước huấn luyện mô hình YOLOv10:

Bước 1: Chuẩn bị dữ liệu:

Để huấn luyện mô hình YOLOv10, ta cần có tập dữ liệu hình ảnh được gắn nhãn và chú thích đối tượng tương ứng của chúng. Các chú thích bao gồm nhãn lớp đối tượng (class) và tọa độ của hộp giới hạn (bounding box) của nó trong ảnh. Dữ liệu này được sử dụng để đào tạo mô hình phát hiện chính xác các đối tượng trong hình ảnh mới.

Việc tạo ra tập dữ liệu tùy chỉnh là một quá trình khó khăn. Có thể mất hàng chục hoặc thậm chí hàng trăm giờ để thu thập hình ảnh, gắn nhãn và chuyển đổi chúng ở định dạng thích hợp. Nhưng ngày nay đã có nhiều công cụ hỗ trợ việc xử lý dữ liệu như LabelImg, RectLabel hay Roboflow.



Hình 5.12 Công cụ Roboflow Universe

Bước 2: Huấn luyện mô hình:

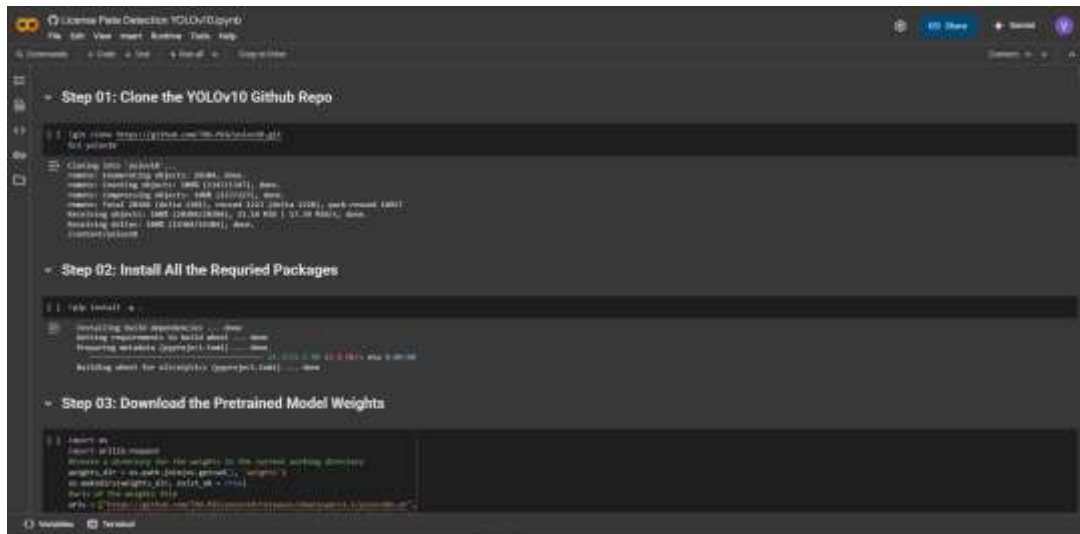
Vì việc huấn luyện yêu cầu rất nhiều tài nguyên phần cứng nên chúng ta sẽ sử dụng công cụ Google Colaboratory.

Google Colaboratory (hay Google Colab) là một nền tảng miễn phí do Google cung cấp, giúp người dùng viết và chạy mã Python trên nền tảng đám mây mà không cần cài đặt phần mềm trên máy tính. Colab chủ yếu được sử dụng trong máy học (Machine Learning), xử lý dữ liệu lớn và thị giác máy tính (Computer Vision).

Các tính năng:

- Chạy Python trên đám mây
- Hỗ trợ GPU & TPU miễn phí
- Kết nối với Google Drive
- Hỗ trợ Markdown và Jupyter Notebook
- Chạy mã Python tương tác

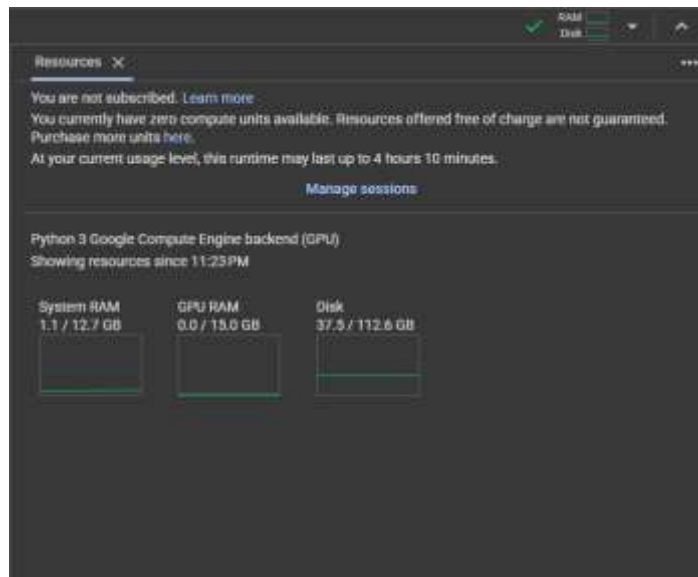
Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Chèo, Hà Nội



Hình 5.13 Giao diện Google Colab

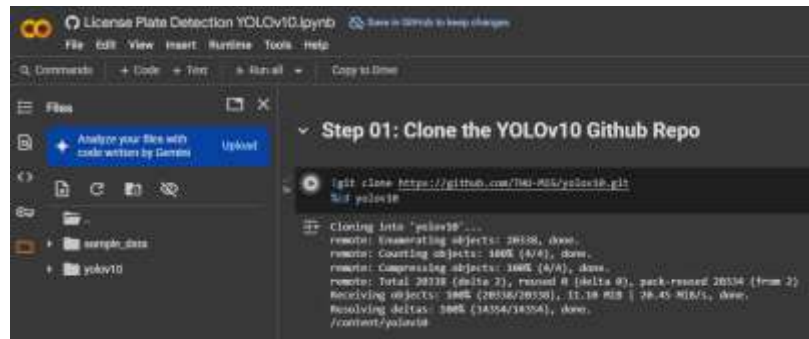
Theo tài liệu của Ultralytics, các bước huấn luyện như sau:

Kết nối Google Colab với GPU: Google Colab cung cấp GPU miễn phí như Tesla T4 hoặc P100 để tăng tốc huấn luyện.



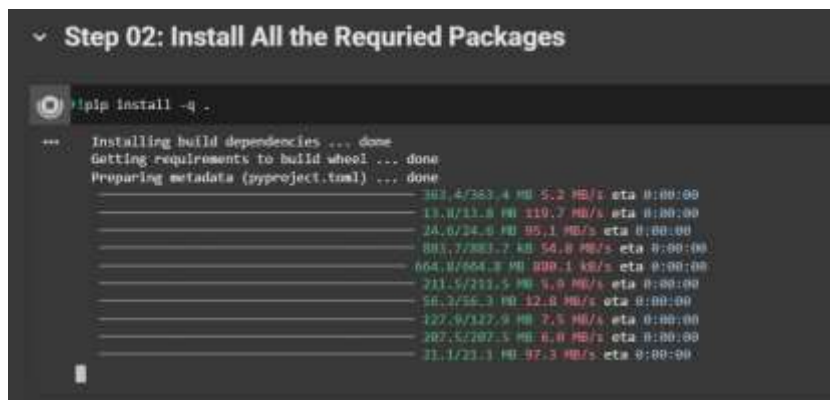
Hình 5.14 Kết nối GPU của Google Colab

Clone YOLOv10 về máy tính ảo Colab.



Hình 5.15 Clone YOLOv10 về máy ảo

Cài đặt các thư viện cần thiết: Một số thư viện quan trọng như: Ultralytics, Pytorch, numpy, OpenCV, ...



Hình 5.16 Cài đặt các gói thư viện

Chuẩn bị tập dữ liệu: YOLOv10 sử dụng dữ liệu theo định dạng chuẩn YOLO format, với mỗi ảnh đi kèm 1 file text chứa các tọa độ bounding box. Chúng ta sử dụng Roboflow để tìm các dataset có sẵn và tải về máy ảo để huấn luyện.



Hình 5.17 Tải Dataset từ Roboflow

Huấn luyện mô hình YOLOv10: Chạy lệnh huấn luyện.

```

Step 05: Train Custom Model

!python task-detect mode=train epochs=120 batch=16 plots=True model=weights/yolov10n.pt data=Car_license_Plates/data.yaml

New https://pypi.org/project/ultralytics/8.2.49 available. Update with 'pip install -U ultralytics'.
ultralytics YOLOv8.1.34 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15360MiB)
engine/trainer: task=detect, mode=train, model=weights/yolov10n.pt, data=Car_license_Plates/data.yaml, epochs=120, time=1000,
2024-07-02 08:47:02.710060: I external/local_cuda/stream_executor/cuda/cuda_device_context_impl.cc:387] Unable to register cudaPTX factory:
2024-07-02 08:47:02.718145: I external/local_cuda/stream_executor/cuda/cuda_device_context_impl.cc:353] Unable to register cudaBC factory:
Overriding model.yaml nc=88 with nc=1

      train  n  params  module  arguments
0         0   -1  1      464  ultralytics.nn.modules.conv.Conv  [3, 32, 3, 2]
1         1   -1  1     4672 ultralytics.nn.modules.conv.Conv  [16, 32, 3, 2]
2         2   -1  1     7360 ultralytics.nn.modules.block.C2F  [32, 32, 3, True]
3         3   -1  1    18560 ultralytics.nn.modules.conv.Conv  [32, 64, 3, 2]
4         4   -1  2    40608 ultralytics.nn.modules.block.C2F  [64, 64, 3, True]
5         5   -1  1    80320 ultralytics.nn.modules.block.SDown  [64, 128, 3, 2]
6         6   -1  2    192632 ultralytics.nn.modules.block.C2F  [128, 128, 3, True]
7         7   -1  1    385800 ultralytics.nn.modules.block.SDown  [128, 256, 3, 2]
8         8   -1  1    609280 ultralytics.nn.modules.block.C2F  [256, 256, 3, True]
9         9   -1  1    169680 ultralytics.nn.modules.block.SPPF  [256, 256, 5]
10        10  -1  1    249720 ultralytics.nn.modules.block.PSA  [256, 256]
11        11  -1  1         0 torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
12        12  [-1, 0] 1         0 ultralytics.nn.modules.conv.Concat  [1]
13        13  -1  1    148328 ultralytics.nn.modules.block.C2F  [384, 320, 1]
14        14  -1  1         0 torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
15        15  [-1, 0] 1         0 ultralytics.nn.modules.conv.Concat  [1]
16        16  -1  1    372480 ultralytics.nn.modules.block.C2F  [192, 64, 3]
17        17  -1  1    30092 ultralytics.nn.modules.conv.Conv  [64, 64, 3, 2]
18        18  [-1, 13] 1         0 ultralytics.nn.modules.conv.Concat  [1]
19        19  -1  1    123640 ultralytics.nn.modules.block.C2F  [192, 128, 3]
20        20  -1  1    188440 ultralytics.nn.modules.block.SDown  [128, 128, 3, 2]
21        21  [-1, 10] 1         0 ultralytics.nn.modules.conv.Concat  [1]
22        22  -1  1    702624 ultralytics.nn.modules.block.C2FCB  [384, 256, 1, True, True]
23        23  [10, 10, 22] 1    601710 ultralytics.nn.modules.head.v10Detect  [1, [64, 320, 256]]

YOLOv10n summary: 285 layers, 2694806 parameters, 270314 gradients, 8.4 GFLOPs
    
```

Hình 5.18 Giao diện Google Colab

Giải thích các tham số:

- **Batch size:** là một trong những tham số xác định số lượng dữ liệu huấn luyện được sử dụng trong một lần lặp lại thuật toán. Nó chỉ định số lượng mẫu sẽ được truyền qua mạng thần kinh (Neural Network) cùng một lúc. Việc lựa chọn batch size phụ thuộc vào kích thước của tập dữ liệu, độ phức tạp của mô hình và tài nguyên máy tính.
- **Epochs:** là một lần lặp hoàn chỉnh thông qua toàn bộ tập dữ liệu trong quá trình đào tạo. Trong mỗi Epochs, mô hình được trình bày với toàn bộ tập dữ liệu và nó cập nhật các trọng số của nó dựa trên các lỗi mà nó mắc phải trong khi dự đoán đầu ra chính xác. Mục tiêu của nhiều Epochs là cải thiện độ chính xác của mô hình bằng cách điều chỉnh weight của nó trên toàn bộ tập dữ liệu nhiều lần.
- **Model:** sử dụng trọng số đã được huấn luyện ban đầu của mô hình, ví dụ như yolov10n.pt, yolov10m.pt, yolov10x.pt.
- **Data:** sử dụng tập dữ liệu đã tải về trước đó để huấn luyện.

Sau khi huấn luyện, ta sẽ có một tập trọng số tốt nhất và các chỉ số huấn luyện:

```

YOLOv10n summary (fused): 285 layers, 2694806 parameters, 0 gradients, 8.2 GFLOPs
Class      Images  Instances  Box(P  R      mAP50  mAP50-95)
all         64       68         0.901  0.794  0.867   0.486
Speed: 0.2ms preprocess, 5.1ms inference, 0.0ms loss, 0.1ms postprocess per image
    
```

Hình 5.19 Chỉ số kết quả huấn luyện

- P (Precision): Độ chính xác (90,1%), mô hình ít bị nhầm lẫn, ít phát hiện sai.
- R (Recall): Tương đối tốt (79,4%), mô hình phát hiện khá đầy đủ các vật thể.
- mAP@0.5 (mean Average Precision): Khá cao (86,7%), rất tốt cho bài toán nhận diện
- mAP@0.5:0.95: Trung bình (48,6%), có thể cải thiện bằng cách thêm nhiều ảnh huấn luyện, đa dạng góc nhìn, ánh sáng, ...

PaddleOCR:

Tổng quan về OCR:

Hệ thống nhận dạng chữ viết (Optical Character Recognition – OCR) đã được sử dụng rộng rãi trong thực tế như đọc hoá đơn, các loại giấy tờ cá nhân. Tuy nhiên OCR vẫn là một vấn đề đầy thách thức về cả độ chính xác và tốc độ tính toán. Các công cụ và các thuật toán trích xuất ngày càng được phát triển và làm việc hiệu quả trong việc nhận dạng và phân tích cấu trúc.

Giới thiệu PaddleOCR:

PaddleOCR là một thư viện nhận dạng ký tự quang học (OCR) mã nguồn mở do Baidu phát triển, thuộc hệ sinh thái PaddlePaddle. Thư viện này hỗ trợ nhận dạng văn bản trong hình ảnh với độ chính xác cao và tối ưu hóa hiệu suất, đặc biệt phù hợp với ngôn ngữ đa dạng, bao gồm tiếng Việt.

- Các tính năng chính của PaddleOCR:
- Hỗ trợ nhận dạng nhiều ngôn ngữ
- Kiến trúc OCR mạnh mẽ
- Hỗ trợ nhiều mô hình OCR khác nhau
- Tích hợp GPU & CPU
- Dễ dàng tích hợp vào các hệ thống AI

5.4.2 Quy trình chụp và xử lý ảnh biển số của YOLOv10 và PaddleOCR:

- Nhận dữ liệu từ camera: dữ liệu đầu vào là ảnh/video từ webcam, camera IP, hoặc ảnh chụp sẵn.
- Phát hiện vùng chứa biển số bằng YOLOv10: dùng mô hình đã huấn luyện để xác định vị trí biển số xe. Kết quả sẽ được một bounding box (tọa độ).
- Cắt và xử lý ảnh biển số: dùng bounding box để cắt vùng chứa biển số.
- Nhận dạng kí tự biển số bằng PaddleOCR: Đưa vùng chứa biển số đã cắt vào PaddleOCR để nhận dạng kí tự, sau đó sẽ trả về chuỗi kí tự là biển số xe dựa vào giá trị độ tự tin có thể đặt trước.
- Xử lý hậu kỳ chuỗi biển số: Làm sạch chuỗi biển số (bỏ dấu chấm, kí tự lạ,...), kiểm tra đúng định dạng biển số xe.
- Lưu và hiển thị dữ liệu: Hiển thị biển số đọc được lên giao diện GUI, lưu các dữ liệu liên quan vào SQLite.



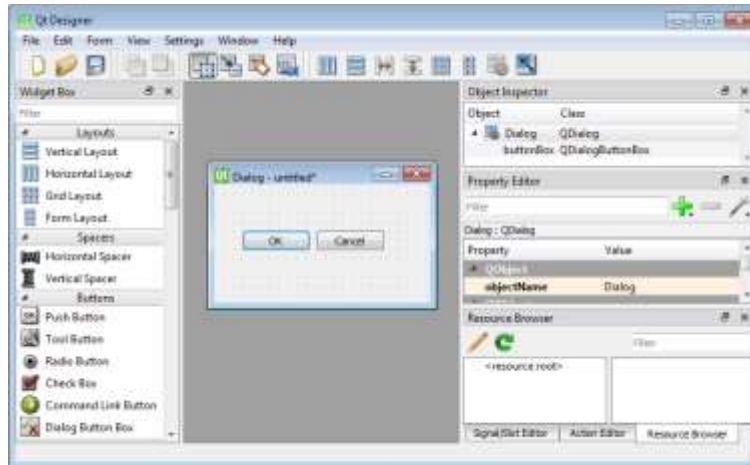
Hình 5.20 Kết quả biển số xe đọc được

5.5 Thiết kế giao diện người dùng (GUI):

5.5.1 Công cụ và thư viện cần dùng:

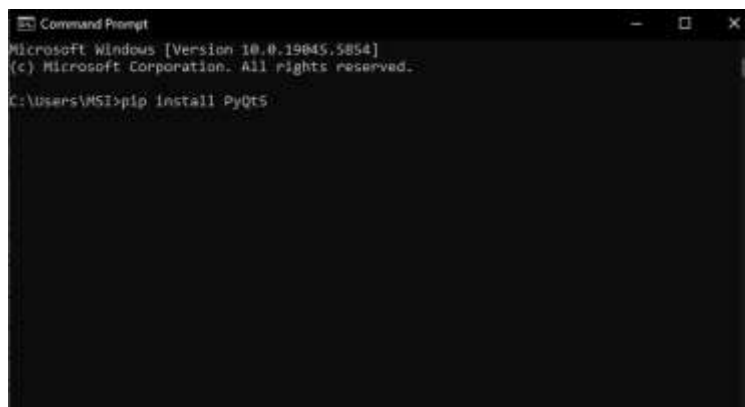
Giao diện người dùng (GUI) của phần mềm sử dụng thư viện PyQt5 và công cụ Qt Designer để thiết kế.

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 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 5.21 Giao diện của phần mềm Qt Designer

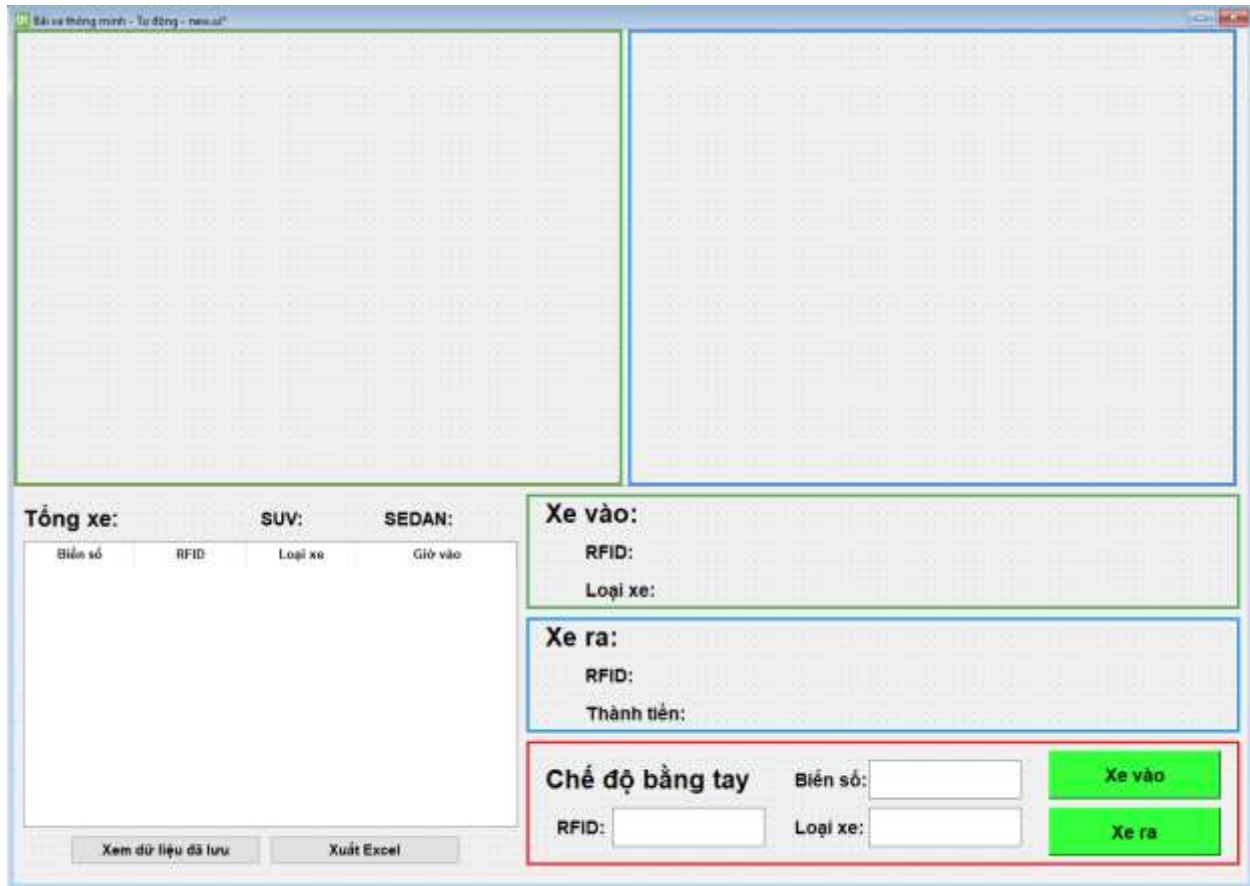
Qt là một Application framework đa nền tảng viết trên ngôn ngữ C++ , được dùng để phát triển các ứng dụng trên desktop, hệ thống nhúng và mobile. Hỗ trợ cho các platform bao gồm : Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS và một số platform khác. PyQt được phát triển bởi Riverbank Computing Limited, là Python interface của Qt, kết hợp của ngôn ngữ lập trình Python và thư viện Qt, một thư viện bao gồm các thành phần giao diện điều khiển (widgets , graphical control elements). Phiên bản mà phần mềm sử dụng là phiên bản thứ 5 – PyQt5.



Hình 5.22 Lệnh cài đặt PyQt5

5.5.2 Giao diện người dùng (GUI):

Sử dụng công cụ Qt Designer cùng với các Widgets có sẵn, giao diện người dùng (GUI) của phần mềm được thiết kế như sau:



Hình 5.23 Giao diện người dùng (GUI) của phần mềm

Trong đó:

- Hai khung camera: Vào (trái), ra (phải) dùng imageLabel và được đặt độ phân giải là 640x480.
- Khung hiển thị thông tin xe vào/ra: sử dụng các Label, hiển thị các thông tin của xe (biển số, RFID, loại xe, thành tiền).
- Danh sách xe đang gửi trong bãi: sử dụng tableWidget, hiển thị và lưu trữ thông tin của xe vào (biển số, RFID, loại xe, thời gian xe vào).
- Khung chế độ bằng tay: sử dụng các Label và lineEdit để nhập bằng tay thông tin của xe khi hệ thống nhận diện có sự cố, cùng với hai nút bấm vào/ra.
- Các nút bấm Xem dữ liệu và Xuất CSV: dùng để xem toàn bộ dữ liệu vào/ra của bãi xe, có thể xuất ra Excel.
- Hiển thị tổng số lượng xe đang gửi theo từng loại xe.

5.6 Chi tiết hoạt động của phần mềm:

5.6.1 Các chức năng chính của phần mềm:

Phần mềm quản lý bãi đỗ xe thông minh giúp tự động hóa quá trình nhận diện xe vào, ra dựa trên các chức năng chính như sau:

- Giao diện người dùng (UI): Trực quan, giúp người dùng dễ dàng giám sát hoạt động.
- Nhận diện biển số: Dùng YOLOv10 và PaddleOCR để nhận dạng biển số, chỉ chấp nhận các biển số đúng định dạng.
- Xử lý RFID qua cổng COM: Đọc dữ liệu nhận từ cổng COM, phân biệt xe vào/ra, và loại xe.
- Xử lý camera và nhận diện: Hai QTimer đọc khung hình liên tục từ camera, khi có RFID hợp lệ thì hệ thống mới cho phép đọc biển số và lưu vào bộ nhớ tạm. Khi xe ra, RFID và biển số phải trùng với lúc xe vào.
- Tính tiền gửi xe: Tính tiền bằng hiệu số giữa giờ ra và giờ vào.
- Lưu trữ dữ liệu: Dữ liệu đồng thời được lưu vào JSON file và SQLite database, có thể xuất ra Excel.
- Xử lý thủ công: Phòng trường hợp hệ thống nhận diện có sự cố, cho phép nhân viên trực có thể nhập thủ công biển số, RFID và loại xe.

5.6.2 Giải thích chi tiết từng hàm trong phần mềm: Hàm khởi tạo: `__init__(self)`

```
class ParkingApp(QMainWindow, Ui_MainWindow):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.serial_port = serial.Serial('COM6', 9600, timeout=1)
        self.timer = QTimer()
        self.timer.timeout.connect(self.read_serial)
        self.timer.start(100)
        (parameter) self: Self@ParkingApp
        self.cap_in = cv2.VideoCapture(0)
        self.cap_out = cv2.VideoCapture(1)

        self.timer_in = QTimer()
        self.timer_in.timeout.connect(self.update_frame_in)
        self.timer_in.start(30)

        self.timer_out = QTimer()
        self.timer_out.timeout.connect(self.update_frame_out)
        self.timer_out.start(30)

        self.entry_data = {}
        self.recent_detected_in = set()
        self.recent_detected_out = set()

        self.export_csv_button.clicked.connect(self.export_to_excel)
        self.db_button.clicked.connect(self.show_database)

        self.manual_in.clicked.connect(self.manual_entry)
        self.manual_out.clicked.connect(self.manual_exit)

        self.allow_plate_reading = False
        self.last_rfid = None
        self.last_type = None

        self.allow_plate_out = False
        self.last_rfid_out = None
```

Hình 5.24 Hàm khởi tạo

Chức năng:

- Khởi tạo giao diện người dùng.
- Kết nối camera và cổng Serial.
- Khởi tạo các biến tạm thời (cờ cho phép đọc biển số xe vào/ra), bộ nhớ tạm lưu xe đang đỗ (entry_data).
- Thiết lập các Timer để xử lý các khung hình từ camera, đọc Serial.
- Gán sự kiện cho các nút bấm trên giao diện (xem database, xuất excel, xe vào/ra thủ công).

Hàm đọc Serial: read_serial(self)

```
def read_serial(self):
    if self.serial_port.in_waiting:
        raw = self.serial_port.readline()
        try:
            line = raw.decode('utf-8').strip()
            if line.startswith("I;"):
                parts = line.split(';')
                if len(parts) == 3:
                    self.last_rfid = parts[1]
                    self.last_type = parts[2]
                    self.allow_plate_reading = True
                    self.label_rfid_in.setText(f"RFID: {self.last_rfid}")
                    self.label_type.setText(f"Loại xe: {self.last_type}")
                    QTimer.singleShot(10000, lambda: self.label_rfid_in.setText("RFID:"))
                    QTimer.singleShot(10000, lambda: self.label_type.setText("Loại xe:"))

            if line.startswith("O,"):
                parts = line.split(",")
                if len(parts) == 2:
                    self.last_rfid_out = parts[1].strip()
                    self.allow_plate_out = True
                    self.label_rfid_out.setText(f"RFID: {self.last_rfid_out}")
                    QTimer.singleShot(10000, lambda: self.label_rfid_out.setText("RFID:"))

        except Exception as e:
            print(f"Serial error: {e}")
```

Hình 5.25 Hàm đọc Serial

Chức năng:

- Đọc dữ liệu từ cổng COM, nếu dữ liệu dạng “I,<rfid>,<type>” thì xe vào, nếu dữ liệu dạng “O,<rfid>” thì xe ra. Hiển thị thông tin đọc được lên GUI.
- Phân tích và lưu thông tin RFID và loại xe, cho phép camera nhận diện biển số.

Hàm cập nhật camera vào: update_frame_in(self)

```
def update_frame_in(self):
    ret, frame = self.cap_in.read()
    if not ret:
        self.cap_in = cv2.VideoCapture(0)
        return

    self.show_frame(self.image_label_in, frame)

    if self.allow_plate_reading:
        plate = self.detect_plate(frame)
        if plate and plate not in self.entry_data and plate not in self.recent_detected_in:
            now = datetime.now()
            self.entry_data[plate] = {
                "entry_time": now,
                "rfid": self.last_rfid,
                "type": self.last_type
            }

            row_position = self.tableWidget.rowCount()
            self.tableWidget.insertRow(row_position)

            plate_item = QTableWidgetItem(plate)
            rfid_item = QTableWidgetItem(self.last_rfid)
            type_item = QTableWidgetItem(self.last_type)
            time_item = QTableWidgetItem(now.strftime("%Y-%m-%d %H:%M:%S"))

            for item in (plate_item, rfid_item, type_item, time_item):
                item.setTextAlignment(Qt.AlignCenter)

            self.tableWidget.setItem(row_position, 0, plate_item)
            self.tableWidget.setItem(row_position, 1, rfid_item)
            self.tableWidget.setItem(row_position, 2, type_item)
            self.tableWidget.setItem(row_position, 3, time_item)

            self.label_in.setText(f"Xe vào: {plate} lúc {now.strftime('%H:%M:%S')}")
            QTimer.singleShot(10000, lambda: self.label_in.setText("Xe vào:"))
            self.recent_detected_in.add(plate)
            QTimer.singleShot(10000, lambda: self.recent_detected_in.discard(plate))

            QTimer.singleShot(0, lambda: self.serial_port.write(b'IN\n'))

    self.allow_plate_reading = False
```

Hình 5.26 Hàm cập nhật camera vào

Chức năng:

- Đọc khung hình từ camera vào và hiển thị lên giao diện.
- Nếu đang ở chế độ chờ nhận diện (allow_plate_reading = True, biến này được gán giá trị True khi đã nhận được dữ liệu RFID, cho phép nhận diện biển số), gọi hàm detect_plate(frame) để tìm và nhận diện biển số.
- Nếu biển số hợp lệ, lưu biển số, RFID, loại xe và thời gian vào vào entry_data, hiển thị lên GUI ở phần danh sách xe đang gửi và phần thông báo xe vào.
- Gửi tín hiệu mở barrier để xe vào.

Hàm cập nhật camera ra: update_frame_out(self)

```
def update_frame_out(self):
    ret, frame = self.cap_out.read()
    if not ret:
        self.cap_out = cv2.VideoCapture(1)
        return

    self.show_frame(self.image_label_out, frame)

    if self.allow_plate_out:
        plate = self.detect_plate(frame)
        rfid_out = self.last_rfid_out

        if not plate:
            return

        if plate not in self.entry_data:
            self.label_out.setText(f"Biển số {plate} không có trong dữ liệu vào!")
            QTimer.singleShot(5000, lambda: self.label_out.setText("Xe ra:"))
            self.allow_plate_out = False
            return

        type = self.entry_data[plate]["type"]
        rfid_in = self.entry_data[plate]["rfid"]

        if rfid_out != rfid_in:
            self.label_out.setText(f"RFID không trùng khớp cho xe {plate}!")
            QTimer.singleShot(5000, lambda: self.label_out.setText("Xe ra:"))
            self.allow_plate_out = False
            return

        if plate not in self.recent_detected_out:
            now = datetime.now()
            entry_time = self.entry_data[plate]["entry_time"]
            exit_time = now
            duration = (exit_time - entry_time).total_seconds() / 3600
            fee = int(duration * 15000)

            self.save_record(plate, rfid_in, type, entry_time, exit_time, fee)
            self.label_out.setText(f"Xe ra: {plate} lúc {now.strftime('%d/%m/%Y %H:%M:%S')} | {duration:.2f}h")
            self.label_fee.setText(f"Thành tiền: {fee}d")
            QTimer.singleShot(10000, lambda: self.label_out.setText("Xe ra:"))
            QTimer.singleShot(10000, lambda: self.label_fee.setText("Thành tiền:"))

            for row in range(self.tableWidget.rowCount()):
                plate_item = self.tableWidget.item(row, 0)
                rfid_item = self.tableWidget.item(row, 1)
                if plate_item and rfid_item:
                    if plate_item.text() == plate and rfid_item.text() == rfid_out:
                        self.tableWidget.removeRow(row)
                        break

            del self.entry_data[plate]
            QTimer.singleShot(0, lambda: self.serial_port.write(b'OUT\n'))
            self.recent_detected_out.add(plate)
            QTimer.singleShot(10000, lambda: self.recent_detected_out.discard(plate))

        self.allow_plate_out = False
```

Hình 5.27 Hàm cập nhật camera ra

Chức năng:

- Tương tự như hàm cập nhật camera vào, nhưng dùng cho camera ra.
- Sau khi nhận diện được biển số xe, kiểm tra biển số có tồn tại trong entry_data không, so sánh RFID lúc ra và lúc vào có giống nhau không. Nếu hợp lệ, tính thời gian đỗ xe, phí đỗ xe và hiển thị lên giao diện ở phần xe ra, xóa dữ liệu trong bảng danh sách xe đang gửi.
- Ghi dữ liệu vào JSON, SQLite và gửi tín hiệu mở barrier.

Hàm phát hiện biển số: detect_plate(self, frame)

```
def detect_plate(self, frame):
    plate_text = ""
    results = model.predict(frame, conf=0.6)
    for result in results:
        for box in result.bboxes:
            x1, y1, x2, y2 = map(int, box.xyxy[0])
            label = paddle_ocr(frame, x1, y1, x2, y2)
            if plate_regex.fullmatch(label):
                plate_text = label
                cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
                cv2.putText(frame, label, (x1, y1 - 10),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
    return plate_text
```

Hình 5.28 Hàm phát hiện biển số

Chức năng:

- Gửi ảnh vào mô hình YOLOv10 để phát hiện vùng chứa biển số.
- Lấy vùng ảnh cắt ra từ YOLOv10, gọi hàm paddle_ocr() để nhận dạng kí tự từ vùng biển số.
- Trả về chuỗi biển số nhận được.

Hàm nhận dạng kí tự: paddle_ocr(frame, x1, y1, x2, y2)

```
def paddle_ocr(frame, x1, y1, x2, y2):
    roi = frame[y1:y2, x1:x2]
    result = ocr.ocr(roi, det=False, rec=True, cls=False)

    best_text = ""
    best_score = 0

    for r in result:
        text_candidate, score = r[0]
        if score > best_score and not np.isnan(score):
            best_score = score
            best_text = text_candidate

    if best_score < 0.85:
        return ""

    pattern = re.compile("[\W_]*")
    best_text = pattern.sub("", best_text)
    best_text = best_text.replace("???", "")
    best_text = best_text.replace("0", "0")

    return best_text
```

Hình 5.29 Hàm phát nhận dạng ký tự

Chức năng:

- Sử dụng PaddleOCR để nhận dạng chữ từ vùng biển số được cắt ra.
- Trả về chuỗi ký tự được phát hiện với số điểm đặt trước.
- Loại bỏ khoảng trắng và chữ thừa.

Hàm lưu trữ dữ liệu: save_record(self, plate_number, entry_time, exit_time, rfid, vehicle_type, fee)

```
def save_record(self, plate, rfid_in, type, entry_time, exit_time, fee):
    data = {
        "license_plate": plate,
        "rfid": rfid_in,
        "type": type,
        "entry_time": entry_time.isoformat(),
        "exit_time": exit_time.isoformat(),
        "fee": fee
    }

    try:
        with open("parking_data.json", "r") as f:
            all_data = json.load(f)
    except:
        all_data = []

    all_data.append(data)
    with open("parking_data.json", "w") as f:
        json.dump(all_data, f, indent=2)

    conn = sqlite3.connect("parking.db")
    cursor = conn.cursor()
    cursor.execute('''CREATE TABLE IF NOT EXISTS records (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        license_plate TEXT,
        rfid TEXT,
        type TEXT,
        entry_time TEXT,
        exit_time TEXT,
        fee INTEGER
    )''')
    cursor.execute('''INSERT INTO records (license_plate, rfid, type, entry_time, exit_time, fee)
    VALUES (?, ?, ?, ?, ?, ?)''',
        (plate, rfid_in, type, entry_time.isoformat(), exit_time.isoformat(), fee))
    conn.commit()
    conn.close()
```

Hình 5.30 Hàm lưu trữ dữ liệu

Chức năng:

- Ghi thông tin vào File JSON (parking_data.json) và cơ sở dữ liệu SQLite (parking.db).
- Thông tin bao gồm biển số xe, RFID, loại xe, giờ vào, giờ ra và phí.

Hàm xem dữ liệu: show_database(self)

```
def show_database(self):
    conn = sqlite3.connect("parking.db")
    cursor = conn.cursor()
    cursor.execute("SELECT license_plate, rfid, type, entry_time, exit_time, fee FROM records ORDER BY id DESC LIMIT 100")
    rows = cursor.fetchall()
    conn.close()

    dialog = QDialog(self)
    dialog.setWindowTitle("Dữ liệu bãi xe")
    dialog.setGeometry(150, 150, 1000, 500)
    layout = QVBoxLayout()

    text_edit = QTextEdit()
    text_edit.setReadOnly(True)
    text_edit.setFont(QFont("Consolas", 10))
    content = ""
    for row in rows:
        content += f"Biển số: {row[0]} | RFID: {row[1]} | Type: {row[2]} | Vào: {row[3]} | Ra: {row[4]} | Phí: {row[5]}đ\n"

    text_edit.setText(content)
    layout.addWidget(text_edit)
    dialog.setLayout(layout)
    dialog.exec_()
```

Hình 5.31 Hàm xem dữ liệu

Chức năng: Khi bấm nút “Xem dữ liệu” ở GUI thì chương trình sẽ gọi hàm này.

- Truy vấn 100 bản ghi gần nhất của cơ sở dữ liệu.
- Hiển thị ra cửa sổ popup, hiển thị thông tin biển số, RFID, loại xe, giờ vào, giờ ra và phí.

Hàm xuất dữ liệu: export_to_excel(self)

```
def export_to_excel(self):
    file_path, _ = QFileDialog.getSaveFileName(self, "Lưu file Excel", "", "Excel Files (*.xlsx)")
    if not file_path:
        return

    try:
        conn = sqlite3.connect("parking.db")
        cursor = conn.cursor()
        cursor.execute("SELECT license_plate, rfid, type, entry_time, exit_time, fee FROM records ORDER BY id DESC")
        rows = cursor.fetchall()
        conn.close()

        import pandas as pd
        from datetime import datetime

        data = []
        for row in rows:
            entry_dt = datetime.fromisoformat(row[3])
            exit_dt = datetime.fromisoformat(row[4])
            data.append({
                "Plate": row[0],
                "RFID": row[1],
                "Type": row[2],
                "Date_In": entry_dt.strftime('%Y-%m-%d'),
                "Time_In": entry_dt.strftime('%H:%M:%S'),
                "Date_Out": exit_dt.strftime('%Y-%m-%d'),
                "Time_Out": exit_dt.strftime('%H:%M:%S'),
                "Fee": row[5]
            })

        df = pd.DataFrame(data)

        df["Date_In_DT"] = pd.to_datetime(df["Date_In"])
        df["Month"] = df["Date_In_DT"].dt.strftime('%Y-%m')

        revenue_by_day = df.groupby(df["Date_In_DT"].dt.date)["Fee"].sum().reset_index()
        revenue_by_day.columns = ["Date", "Total Revenue (VND)"]

        revenue_by_month = df.groupby("Month")["Fee"].sum().reset_index()
        revenue_by_month.columns = ["Month", "Total Revenue (VND)"]

        with pd.ExcelWriter(file_path, engine='xlsxwriter') as writer:
            df.drop(columns=["Date_In_DT"]).to_excel(writer, index=False, sheet_name='Data')
            revenue_by_day.to_excel(writer, index=False, sheet_name='Revenue by Day')
            revenue_by_month.to_excel(writer, index=False, sheet_name='Revenue by Month')

        QMessageBox.information(self, "Thành công", "Xuất Excel thành công!")

    except Exception as e:
        QMessageBox.critical(self, "Lỗi", f"Lỗi khi xuất Excel: {str(e)}")
```

Hình 5.32 Hàm xuất dữ liệu

Chức năng: Khi bấm nút “Xuất Excel” ở GUI thì chương trình sẽ gọi hàm này.

- Truy vấn toàn bộ dữ liệu từ cơ sở dữ liệu SQLite
- Xuất ra file Excel, tạo các sheet chứa doanh thu theo ngày và theo tháng. Cảnh báo lỗi nếu có sự cố xảy ra.

Hàm xử lý xe vào thủ công: manual_entry(self)

```
def manual_entry(self):
    plate = self.plate_manual.text().strip()
    rfid = self.rfid_manual.text().strip()
    type = self.type_manual.text().strip()

    if not plate or not rfid or not type:
        QMessageBox.warning(self, "Thiếu thông tin", "Vui lòng nhập đầy đủ biển số, RFID và loại xe.")
        return
    if plate in self.entry_data:
        QMessageBox.warning(self, "Trùng biển số", "Biển số này đã có trong dữ liệu.")
        return

    now = datetime.now()
    self.entry_data[plate] = {
        "entry_time": now,
        "rfid": rfid,
        "type": type
    }

    row_position = self.tableWidget.rowCount()
    self.tableWidget.insertRow(row_position)

    plate_item = QTableWidgetItem(plate)
    rfid_item = QTableWidgetItem(rfid)
    type_item = QTableWidgetItem(type)
    time_item = QTableWidgetItem(now.strftime("%Y-%m-%d %H:%M:%S"))

    for item in (plate_item, rfid_item, type_item, time_item):
        item.setTextAlignment(Qt.AlignCenter)

    self.tableWidget.setItem(row_position, 0, plate_item)
    self.tableWidget.setItem(row_position, 1, rfid_item)
    self.tableWidget.setItem(row_position, 2, type_item)
    self.tableWidget.setItem(row_position, 3, time_item)

    self.label_in.setText(f"Thủ công - Xe vào: {plate}")
    self.label_rfid_in.setText(f"RFID: {rfid}")
    self.label_type.setText(f"Loại xe: {type}")
    QTimer.singleShot(5000, lambda: self.label_in.setText("Xe vào:"))
    QTimer.singleShot(5000, lambda: self.label_rfid_in.setText("RFID:"))
    QTimer.singleShot(5000, lambda: self.label_type.setText("Loại xe:"))

    QTimer.singleShot(0, lambda: self.serial_port.write(b'MANUAL_IN\n'))

    self.plate_manual.clear()
    self.rfid_manual.clear()
    self.type_manual.clear()
```

Hình 5.33 Hàm xử lý xe vào thủ công

Chức năng:

- Xử lý như một xe vào thông thường, nhưng thay vì quét RFID và nhận diện biển số thì cho phép nhân viên trực nhập bằng tay các thông tin biển số xe, RFID, loại xe.
- Các chức năng khác của chương trình hoạt động bình thường.
- Có cảnh báo khi nhập thiếu, trùng thông tin.
- Gửi tín hiệu mở barrier.
- Sử dụng khi hệ thống nhận diện biển số và RFID có sự cố.

Hàm xử lý xe ra thủ công: manual_exit(self)

```
def manual_exit(self):
    plate = self.plate_manual.text().strip()
    rfid = self.rfid_manual.text().strip()

    if not plate or not rfid:
        QMessageBox.warning(self, "Thiếu thông tin", "Vui lòng nhập đủ thông tin biển số, RFID để xe ra.")
        return

    if plate not in self.entry_data:
        QMessageBox.warning(self, "Không tìm thấy", "Không có dữ liệu vào của xe này.")
        return

    rfid_in = self.entry_data[plate][rfid]
    if rfid_in != rfid:
        QMessageBox.warning(self, "RFID không khớp", f"RFID không trùng khớp với dữ liệu xe vào.")
        return

    now = datetime.now()
    entry_time = self.entry_data[plate][rfid_in]["entry_time"]
    exit_time = now
    duration = (exit_time - entry_time).total_seconds() / 3600
    fee = int(duration * 15000)
    type = self.entry_data[plate][rfid_in]["type"]

    self.save_record(plate, rfid, type, entry_time, exit_time, fee)

    self.label_out.setText(f"Thủ công - Xe ra: {plate} | {duration:.2f}h")
    self.label_rfid_in.setText(f"RFID: {rfid}")
    self.label_fee.setText(f"Thành tiền: {fee:,}đ")
    QTimer.singleShot(10000, lambda: self.label_out.setText("Xe ra:"))
    QTimer.singleShot(10000, lambda: self.label_rfid_in.setText("RFID:"))
    QTimer.singleShot(10000, lambda: self.label_fee.setText("Thành tiền:"))

    for row in range(self.tableWidget.rowCount()):
        if self.tableWidget.item(row, 0).text() == plate:
            self.tableWidget.removeRow(row)
            break

    del self.entry_data[plate]

    QTimer.singleShot(0, lambda: self.serial_port.write(b'MANUAL_OUT\n'))

    self.plate_manual.clear()
    self.rfid_manual.clear()
    self.type_manual.clear()
```

Hình 5.34 Hàm xử lý xe ra thủ công

Chức năng:

- Xử lý như một xe ra thông thường, nhưng thay vì quét thẻ và nhận diện biển số thì cho phép nhân viên trực nhập bằng tay các thông tin biển số xe, RFID.
- Các chức năng khác của chương trình hoạt động bình thường.
- Có cảnh báo khi nhập thiếu thông tin, sai thông tin, không trùng khớp RFID.
- Gửi tín hiệu mở barrier.
- Sử dụng khi hệ thống nhận diện biển số và RFID có sự cố.

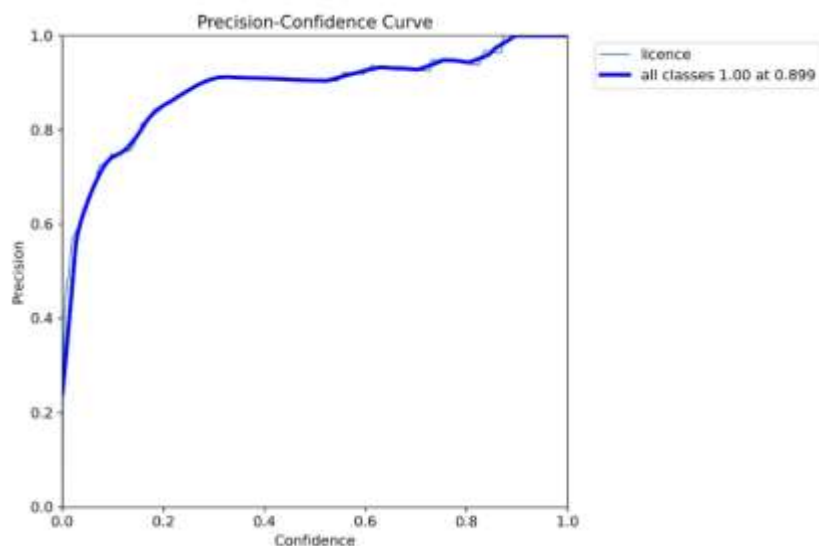
5.7 Đánh giá hiệu quả phần mềm

5.7.1 Tỷ lệ nhận diện chính xác

Sau khi thiết lập tham số thì việc đào tạo hoàn toàn được máy tính xử lý, tuy nhiên ta cần đánh giá mô hình sau khi hoàn thành là một bước quan trọng để kiểm tra hiệu suất của mô hình với những tệp dữ liệu mới. Một số chỉ số được sử dụng để đánh giá hiệu suất của mô hình như:

- Accuracy (độ chính xác): là số liệu đơn giản và phổ biến nhất đo lường số dự đoán chính xác do mô hình đưa ra chia cho tổng số dự đoán. Tuy nhiên, độ chính xác thôi có thể không đủ để đánh giá các mô hình cho dữ liệu mất ổn định.
- Precision (độ chuẩn xác): là phần nhỏ của các dự đoán tích cực thực sự trong số tất cả các dự đoán tích cực do mô hình đưa ra. Số liệu này rất quan trọng trong các tình huống mà kết quả dương tính giả trở thành vấn đề.
- Recall (khả năng thu hồi): là phần dự đoán tích cực thực sự trong số tất cả các điểm dữ liệu tích cực. Số liệu này rất quan trọng trong các tình huống mà âm tính giả trở thành vấn đề.
- F1 Score: là một chỉ số giữa độ chính xác và khả năng thu hồi. Nó cung cấp sự cân bằng giữa hai số liệu và hữu ích khi cả độ chính xác và khả năng thu hồi đều quan trọng.

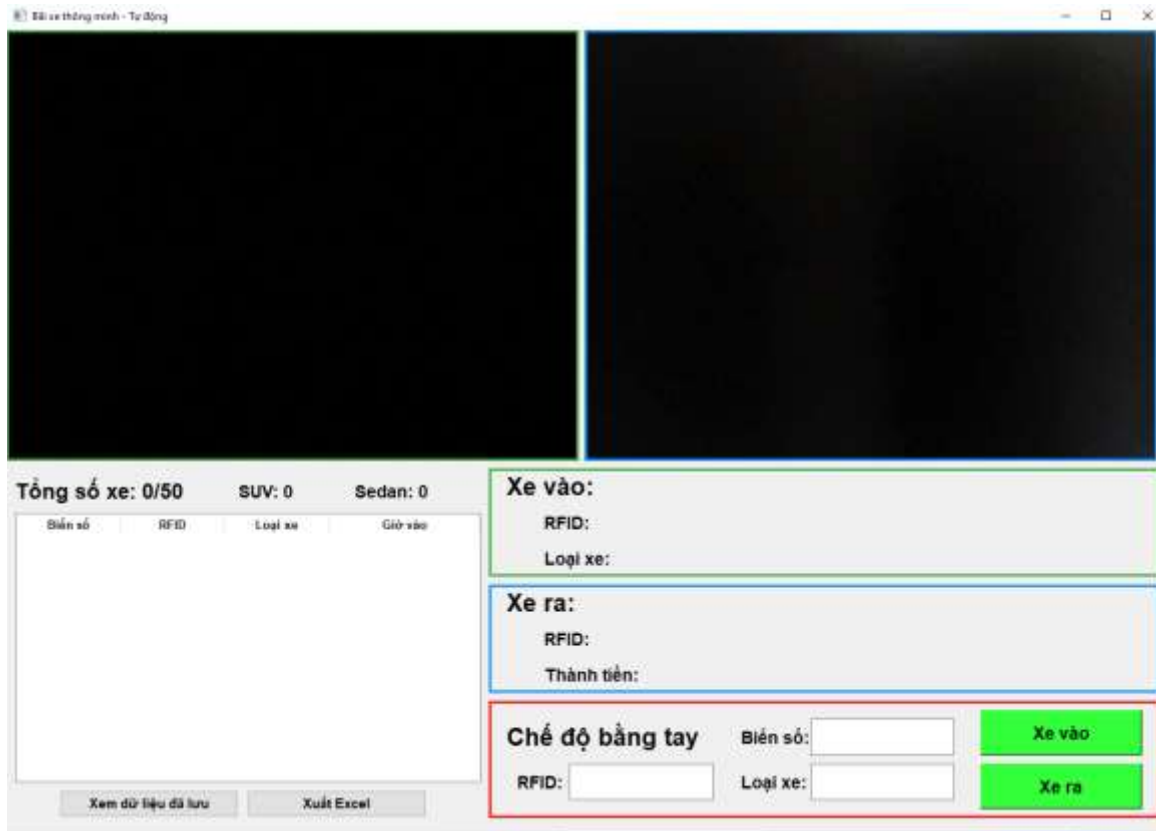
Nhìn chung, đánh giá mô hình là một bước quan trọng trong học máy giúp chúng ta hiểu hiệu suất của một mô hình và xác định các điểm cần cải thiện. Bằng cách hiểu được điểm mạnh và điểm yếu của một mô hình, chúng ta có thể cải thiện nó lặp đi lặp lại để đạt được kết quả tốt hơn.



Hình 5.35 Đồ thị độ tự tin và chính xác

5.7.2 Kết quả xử lý và phản hồi của phần mềm

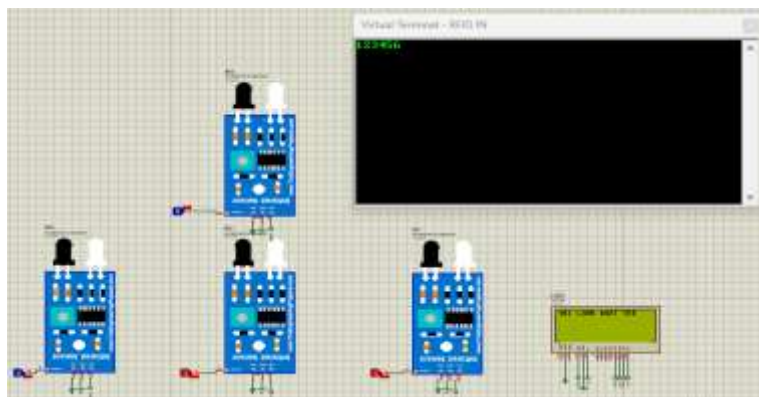
Khởi động phần mềm “Bãi đỗ xe thông minh – Tự động”



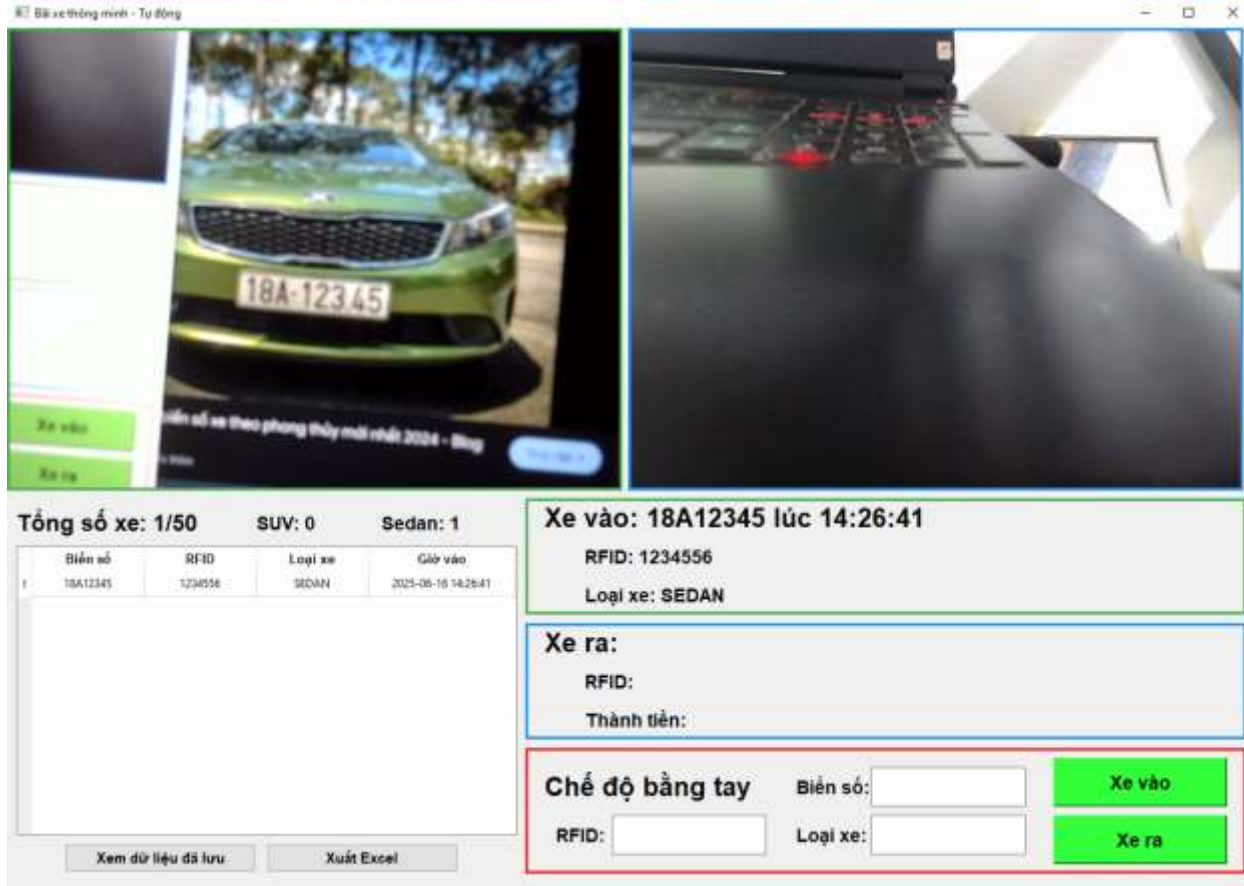
Hình 5.36 Giao diện phần mềm khi vừa khởi động

Xe vào bãi

Giả lập nhập RFID IN số thẻ 123456 ở Virtual terminal (phần mềm Proteus), kích hoạt 2 cảm biến SENSOR1 và SENSOR2, Camera AI quét biển số xe 18A-12345. Ta được kết quả:



Hình 5.37 Nhập RFID và cảm biến SENSOR 1 và SENSOR 2 kích hoạt



Hình 5.38 Kết quả nhận được trên giao diện

Nhận diện biển số bằng AI (YOLOv10 + OCR):

- Biển số nhận diện: 18A-123.45
- Thời gian nhận diện và xử lý: ~500–700 ms
- Ảnh chụp biển số được hiển thị ngay tại giao diện bên trái.

Đọc và xác thực RFID:

- Mã RFID: 1234556
- Thời gian xác thực: ~100 ms
- Hệ thống ghi nhận đúng RFID hợp lệ và đối chiếu biển số – hiển thị kết quả bên phải.

Phân loại xe:

- Loại xe: SEDAN (dựa trên chiều cao cảm biến hoặc dữ liệu đăng ký)
- Kết hợp dữ liệu RFID và AI, phần mềm phân loại và lưu vào danh sách xe đang gửi.

Thời gian vào được ghi nhận chính xác:

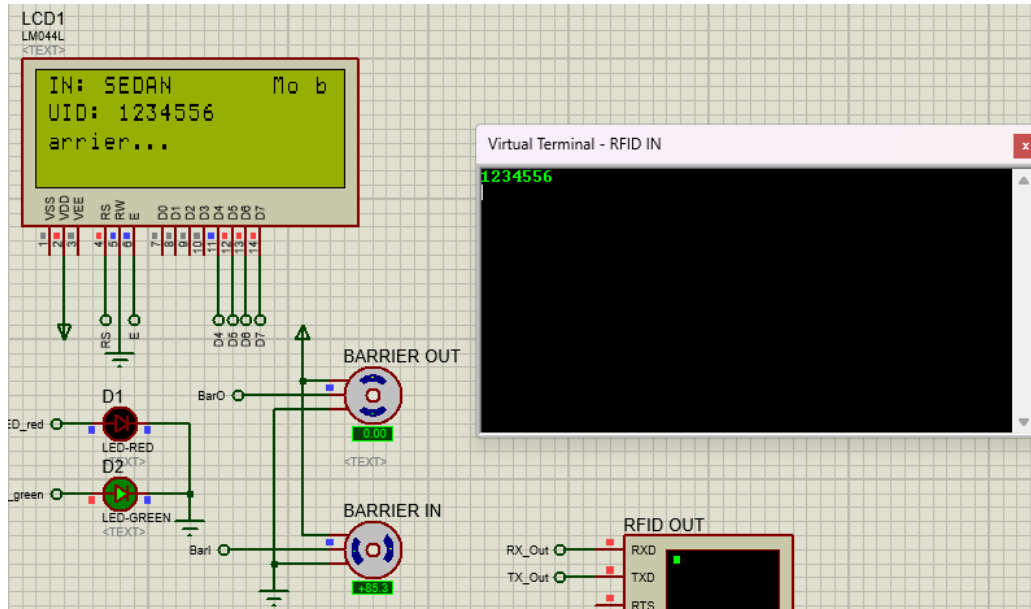
Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Chèo, Hà Nội

- 2025-06-16 14:26:41 hiển thị tại bảng thông tin xe vào.

Giao diện cập nhật thời gian thực:

- Danh sách xe vào được cập nhật tự động.

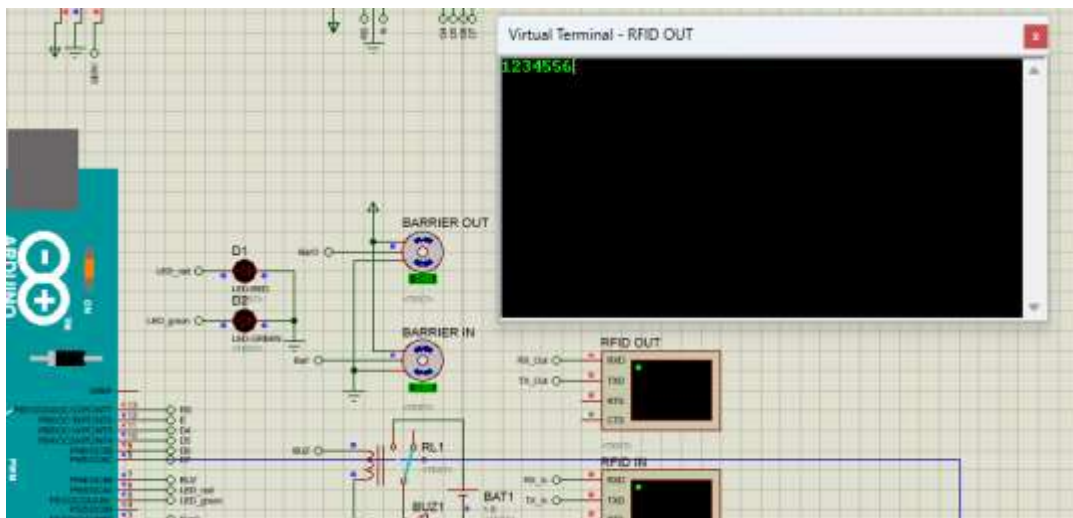
Đồng thời Phần mềm gửi tín hiệu xuống cho Vi điều khiển để mở Barrier IN



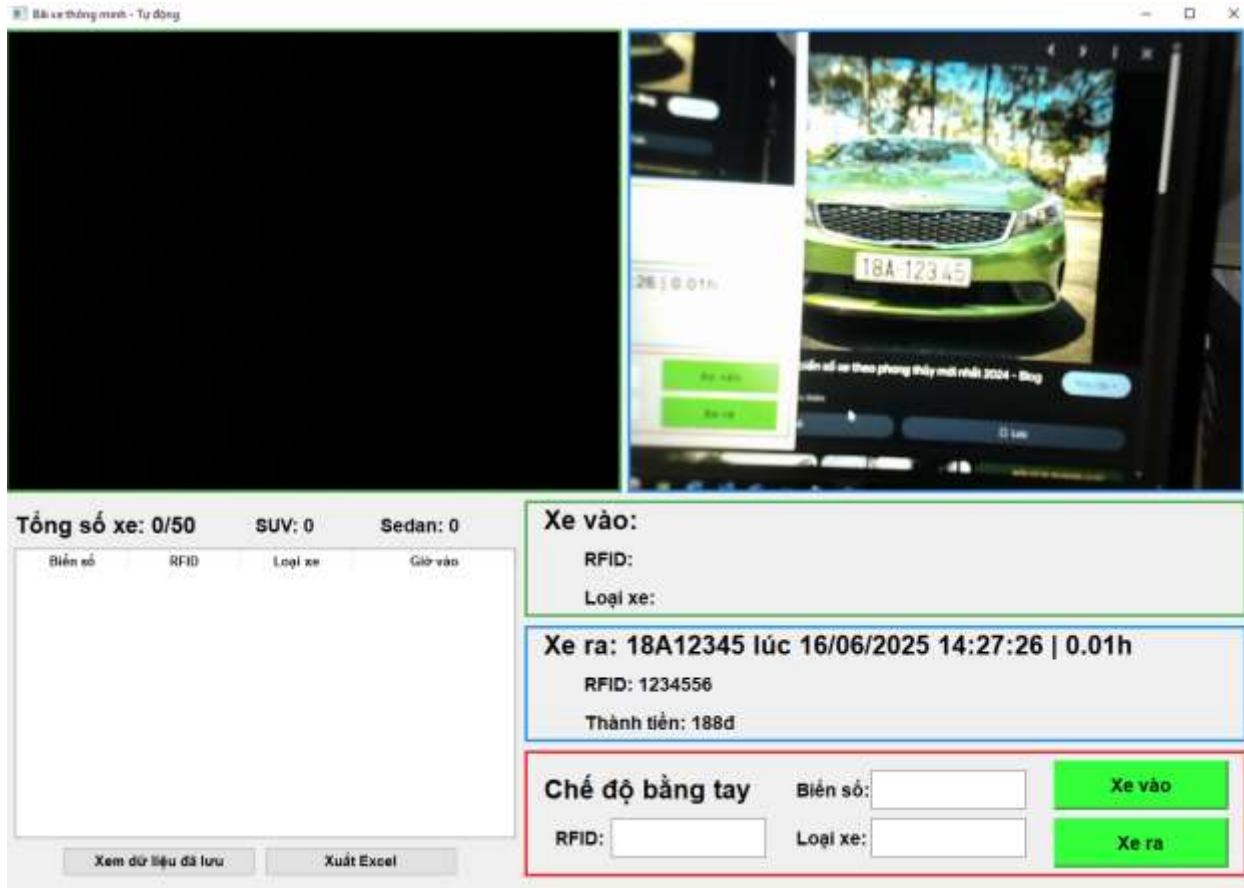
Hình 5.39 Barrier In mở ra

Xe ra khỏi bãi

Giả lập nhập RFID OUT 1234556, Camera Ai quét biển số lúc xe ra. Ta được kết quả:



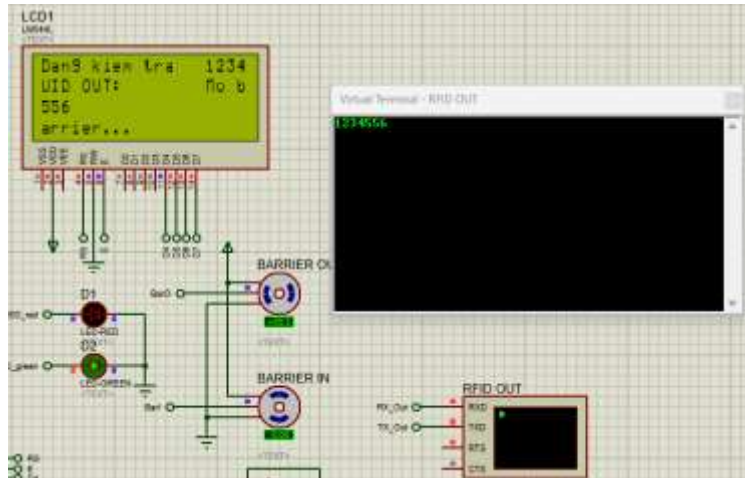
Hình 5.40 Nhập RFID lúc xe ra



Hình 5.41 Kết quả nhận được khi xe ra

- Biển số xe: 18A-123.45
- Thời gian ra: 16/06/2025 14:27:26
- Thời gian gửi xe: 0.01 giờ
- Thành tiền: 188 đồng
- Thông tin RFID: RFID hợp lệ, được truy xuất thành công từ cơ sở dữ liệu.

Sau khi đúng thông tin Biển số xe và RFID thì gửi tín hiệu xuống để mở Barrier cho xe ra khỏi bãi

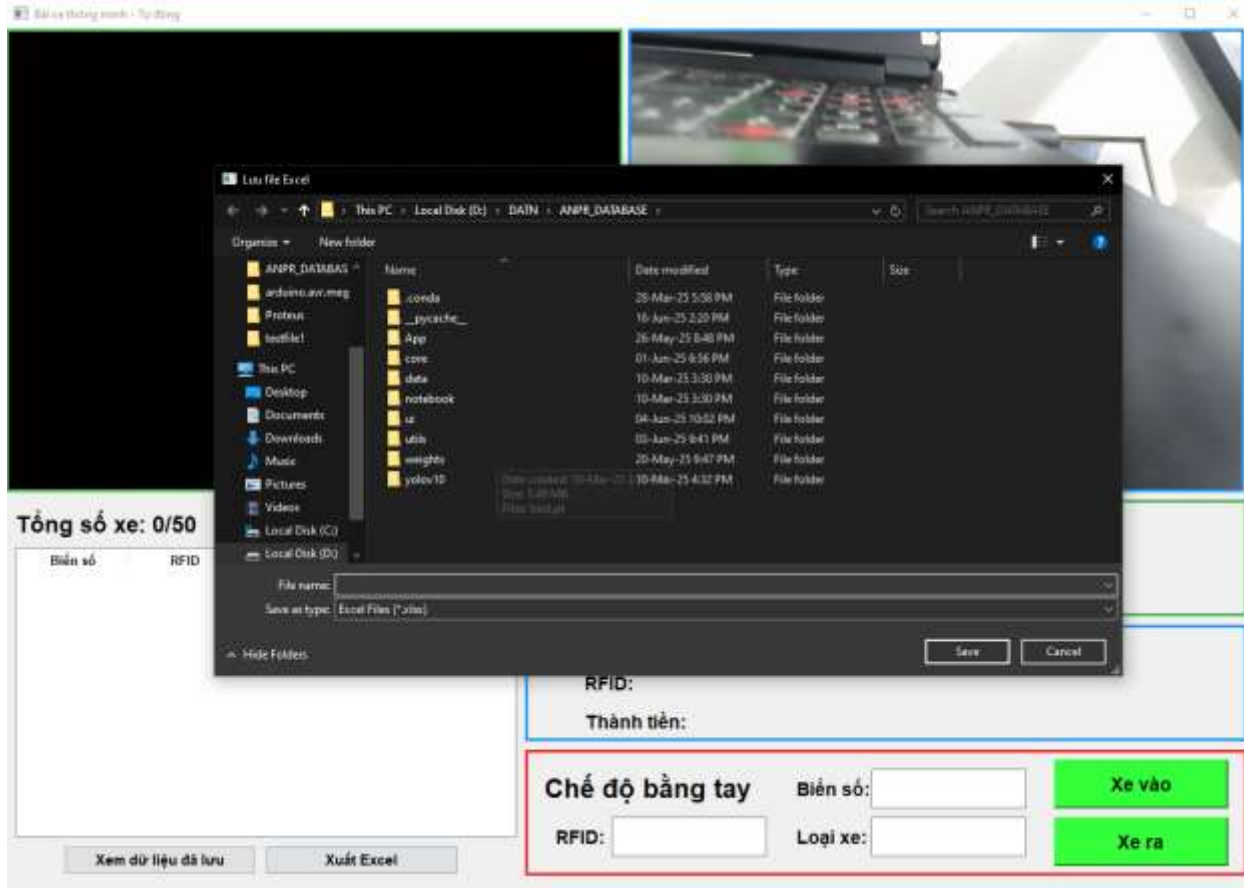


Hình 5.42 Barrier Out mở ra cho xe ra

Xem dữ liệu đã lưu và xuất file CSV

Biển số	RFID	Type	Vào	Ra	Phí
18A12345	1234556	SEDAN	2025-06-16T14:26:41.400641	2025-06-16T14:27:26.528207	180đ
18A12345	222222	SEDAN	2025-06-16T14:25:18.527523	2025-06-16T14:25:48.623268	120đ
18A12345	222222	SEDAN	2025-06-16T14:23:15.585726	2025-06-16T14:24:05.782485	280đ
12A12345	RFID: 222222	Type: SEDAN	Vào: 2025-06-14T16:12:19.196152	Ra: 2025-06-14T16:13:22.221024	Phí: 220đ
18A12345	RFID: 111111	Type: SUV	Vào: 2025-06-14T16:12:09.111901	Ra: 2025-06-14T16:13:15.870188	Phí: 274đ
18A12345	RFID: 12345	Type: SUV	Vào: 2025-06-14T14:50:53.247678	Ra: 2025-06-14T14:51:28.975982	Phí: 123đ
12A12345	RFID: 1234	Type: SEDAN	Vào: 2025-06-14T14:50:36.167982	Ra: 2025-06-14T14:51:32.156657	Phí: 174đ
12A12345	RFID: 123	Type: SUV	Vào: 2025-06-14T14:47:33.491764	Ra: 2025-06-14T14:48:54.481885	Phí: 337đ
12A12345	RFID: 868686	Type: SEDAN	Vào: 2025-06-14T14:47:05.862810	Ra: 2025-06-14T14:48:47.387043	Phí: 423đ
15B12345	RFID: 555555	Type: SUV	Vào: 2025-06-14T14:46:46.482887	Ra: 2025-06-14T14:48:34.701549	Phí: 451đ
00A999999	RFID: 444444	Type: SUV	Vào: 2025-06-14T14:46:22.483374	Ra: 2025-06-14T14:48:21.762151	Phí: 496đ
30F55775	RFID: 222222	Type: SEDAN	Vào: 2025-06-14T14:45:20.823768	Ra: 2025-06-14T14:46:06.562845	Phí: 690đ
18A12345	RFID: 111111	Type: SEDAN	Vào: 2025-06-14T14:44:41.551187	Ra: 2025-06-14T14:47:51.651820	Phí: 792đ
18A12345	RFID: 222222	Type: SEDAN	Vào: 2025-06-10T14:48:12.884541	Ra: 2025-06-10T14:47:43.693328	Phí: 381đ
12A12345	RFID: 123456	Type: SEDAN	Vào: 2023-06-10T14:42:00.959102	Ra: 2023-06-10T14:43:49.180612	Phí: 451đ
18A12345	RFID: 111111	Type: SEDAN	Vào: 2025-06-10T14:39:15.861457	Ra: 2025-06-10T14:39:15.948414	Phí: 84đ
18A12345	RFID: 1234556	Type: SEDAN	Vào: 2025-06-10T14:33:09.168808	Ra: 2025-06-10T14:35:20.133946	Phí: 583đ
18A12345	RFID: 123456	Type: SEDAN	Vào: 2025-06-10T14:30:20.322315	Ra: 2025-06-10T14:31:11.589274	Phí: 213đ
12A11111	RFID: 111	Type: SUV	Vào: 2025-06-05T00:12:32.213141	Ra: 2025-06-05T00:12:48.122116	Phí: 66đ
12A11111	RFID: 222	Type: SUV	Vào: 2025-06-05T00:11:26.094828	Ra: 2025-06-05T00:11:48.795593	Phí: 78đ
12A12345	RFID: 111	Type: SEDAN	Vào: 2025-06-05T00:10:12.906220	Ra: 2025-06-05T00:10:54.919487	Phí: 175đ
12A12345	RFID: 111	Type: sedan	Vào: 2025-06-04T23:54:27.897182	Ra: 2025-06-04T23:55:05.718111	Phí: 157đ
12A12345	RFID: 111	Type: SUV	Vào: 2025-06-04T23:49:51.751882	Ra: 2025-06-04T23:50:26.155855	Phí: 143đ
18A12345	RFID: 111311	Type: SUV	Vào: 2025-06-04T23:25:17.483729	Ra: 2025-06-04T23:25:38.257981	Phí: 53đ
12A12345	RFID: 123	Type: sedan	Vào: 2025-06-04T23:24:39.729361	Ra: 2025-06-04T23:24:47.080980	Phí: 38đ
12A12345	RFID: 123456	Type: SUV	Vào: 2025-06-04T23:14:02.498492	Ra: 2025-06-04T23:14:14.550881	Phí: 58đ
18A12345	RFID: 111511	Type: SEDAN	Vào: 2025-06-04T23:07:31.889405	Ra: 2025-06-04T23:08:48.258184	Phí: 285đ
12A12345	RFID: 123456	Type: SUV	Vào: 2025-06-04T23:07:17.844153	Ra: 2025-06-04T23:08:13.515753	Phí: 273đ
12A12345	RFID: 1234	Type: SEDAN	Vào: 2025-06-04T23:06:12.697521	Ra: 2025-06-04T23:06:32.689793	Phí: 82đ
12A12345	RFID: 123456	Type: SEDAN	Vào: 2025-06-04T22:39:39.848979	Ra: 2025-06-04T22:40:11.032417	Phí: 153đ
18A12345	RFID: 111511	Type: SEDAN	Vào: 2025-06-04T21:50:18.514830	Ra: 2025-06-04T21:50:40.823551	Phí: 122đ

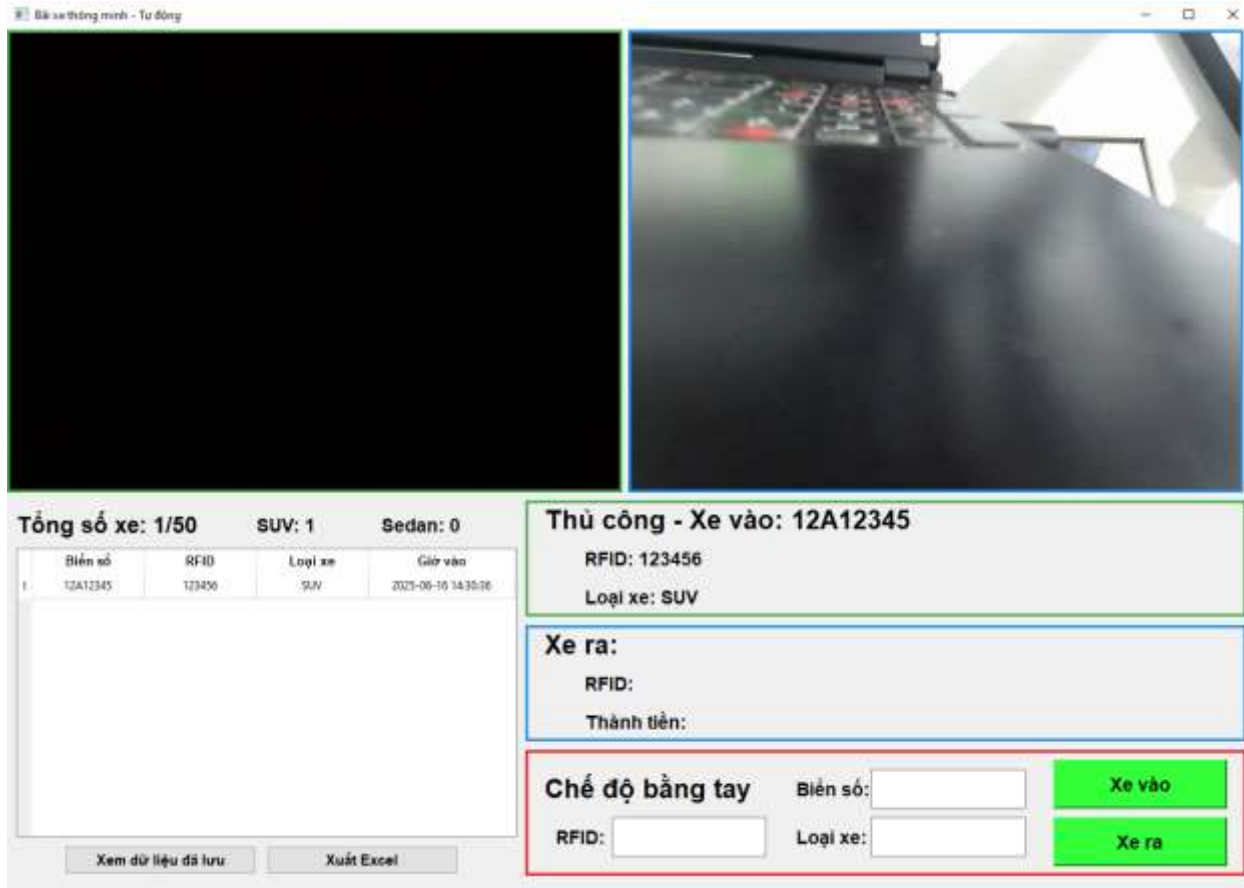
Hình 5.43 Xem dữ liệu đã lưu



Hình 5.44 Xuất file Excel

Chế độ bằng tay

Bảo vệ trực bãi xe khi gặp lỗi chuyển sang chế độ bằng tay nhập đủ thông tin Xe vào, Loại xe và RFID, khi nhập đủ thông tin giao diện sẽ “Thủ công – Xe vào:” và lưu nó vào danh sách xe đang gửi. Đồng thời mở barrier ở chế độ bằng tay.



Hình 5.45 Giao diện xe vào ở chế độ bằng tay

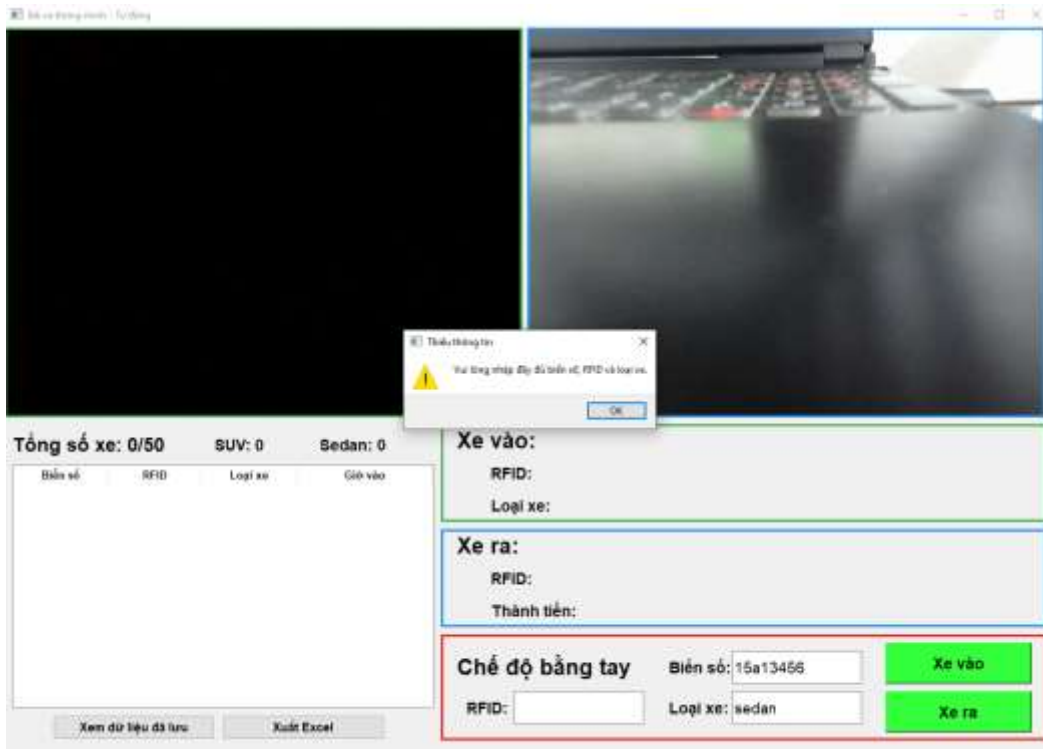
Nghiên cứu và Thiết kế bãi đỗ xe thông minh ứng dụng tại Dự án Danang Times Square của Công ty cổ phần Việt Chèo, Hà Nội

Chế độ bằng tay xe ra bảo vệ trực bãi nhập thông tin xe ra thông qua Camera và thẻ xe, kết quả ta nhận được:

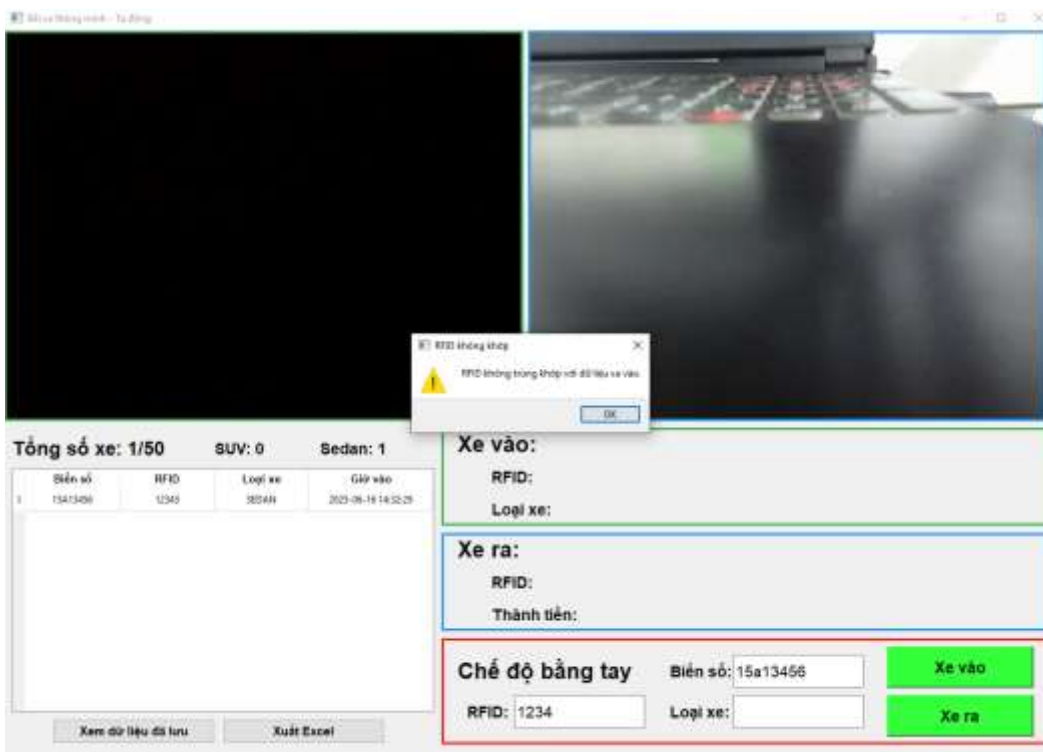
The screenshot displays a software window titled "Bãi xe thông minh - Tự động". The interface is divided into several sections:

- Summary:** "Tổng số xe: 0/50", "SUV: 0", "Sedan: 0".
- Table:** A table with columns: "Biển số", "RFID", "Loại xe", "Giờ vào".
- Manual Entry Section (highlighted with a red border):**
 - Section "Xe vào:" with fields for "RFID: 123456" and "Loại xe:".
 - Section "Thù công - Xe ra: 12A12345 | 0.00h" with fields for "RFID:" and "Thành tiền: 37đ".
 - Section "Chế độ bằng tay" with input fields for "Biển số:", "RFID:", and "Loại xe:", and two green buttons: "Xe vào" and "Xe ra".
- Footer:** "Xem dữ liệu đã lưu" and "Xuất Excel" buttons.

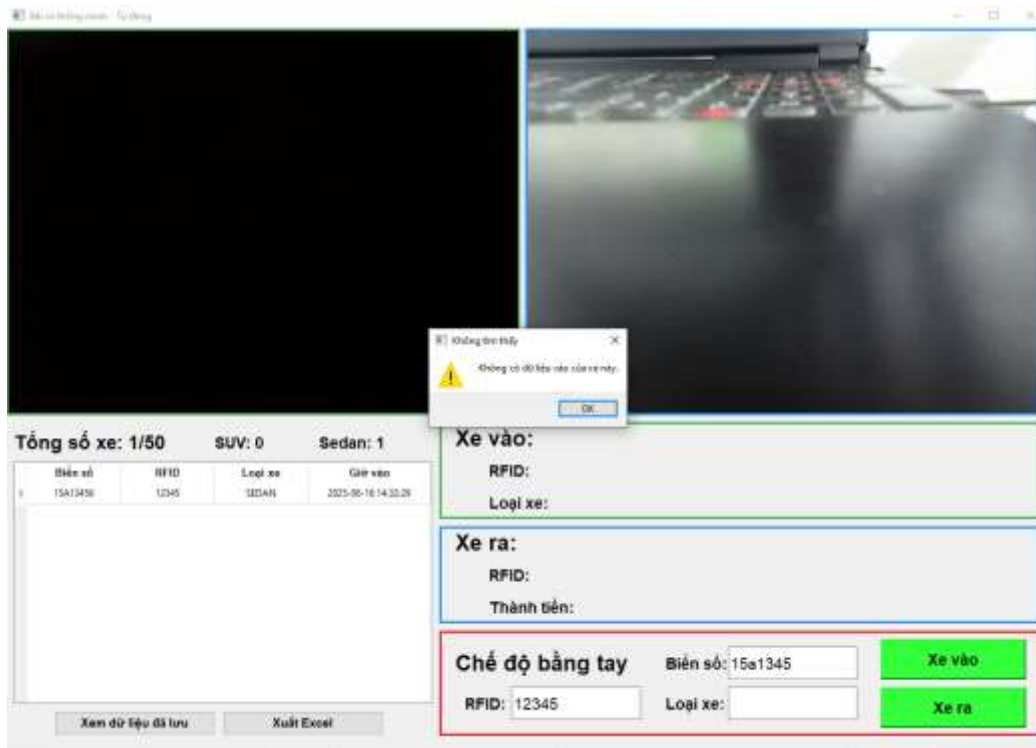
Hình 5.46 Giao diện xe ra ở chế độ bằng tay



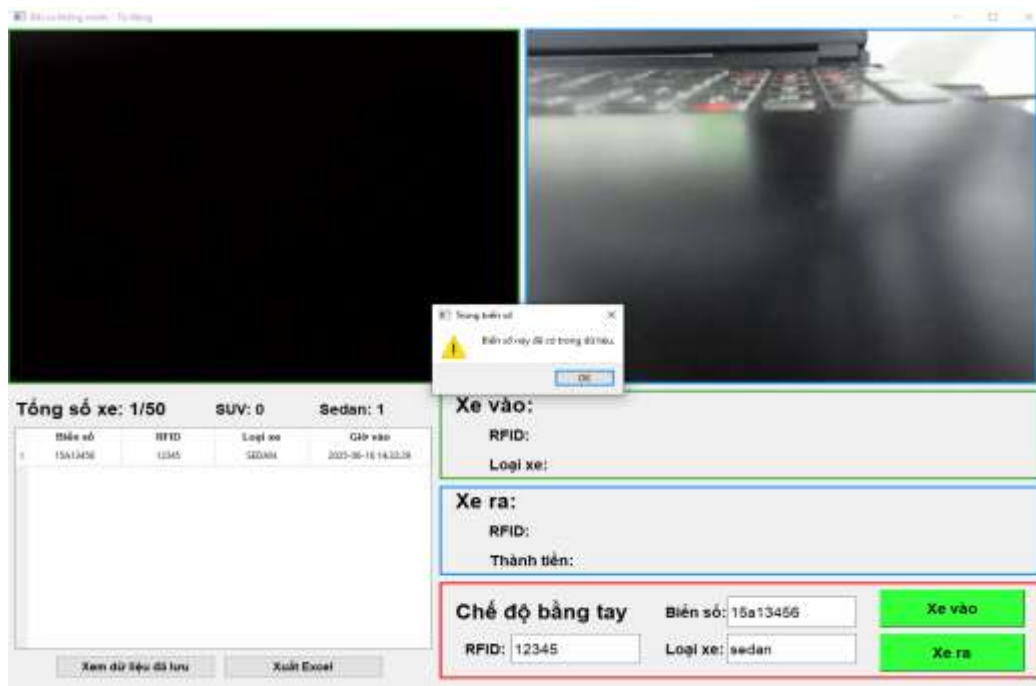
Hình 5.47 Lỗi nhập chưa đầy đủ thông tin của xe



Hình 5.48 Lỗi nhập sai dữ liệu xe vào



Hình 5.49 Cảnh báo không có dữ liệu vào của xe trong dữ liệu



Hình 5.50 Cảnh báo biển số đã có trong dữ liệu

5.7.3 Các hạn chế và đề xuất cải tiến

Các hạn chế

Hạn chế	Mô tả
Độ chính xác nhận diện biển số	Trong điều kiện ánh sáng yếu, biển số bị mờ, cong hoặc có vết xước, hệ thống AI (YOLOv10 + PaddleOCR) có thể nhận sai hoặc không nhận diện được.
Khả năng chống giả mạo	Hệ thống chỉ so sánh RFID và biển số, nên có thể bị qua mặt bằng thẻ giả hoặc dán biển số giả nếu không có thêm xác thực sinh trắc học hoặc camera toàn cảnh.
Phân loại xe đơn giản	Việc phân loại xe chỉ dựa vào chiều cao là tương đối, không đủ chính xác trong một số trường hợp đặc biệt (ví dụ SUV gầm thấp, SEDAN độ gầm cao).
Mất kết nối giữa Vi điều khiển và PC	Nếu kết nối Serial bị ngắt hoặc máy tính treo, hệ thống có thể không mở được barrier hoặc lưu dữ liệu sai.
Không có hệ thống cảnh báo lỗi	Khi một module gặp lỗi (ví dụ camera mất tín hiệu, cảm biến hỏng), người dùng không nhận được thông báo ngay để xử lý.
Khó mở rộng nhiều cổng IN/OUT	Thiết kế hiện tại phù hợp cho 1 cổng IN và 1 cổng OUT. Nếu mở rộng nhiều cổng, cần thiết kế lại hệ thống xử lý và giao tiếp.

Bảng 5.4 Các hạn chế của hệ thống

Đề xuất cải tiến

Đề xuất	Lợi ích
Thêm xử lý ảnh nâng cao (AI Enhancement)	Kết hợp thêm các kỹ thuật xử lý ảnh tiền xử lý (lọc nhiễu, tăng tương phản) để cải thiện độ chính xác nhận diện biển số.
Tích hợp camera toàn cảnh và cảnh báo giả mạo	So sánh hình ảnh toàn xe và biển số để phát hiện các tình huống biển giả, thẻ giả. Có thể tích hợp phát hiện “biển lạ” (chưa từng vào bãi).
Sử dụng nhiều cảm biến phân loại	Kết hợp chiều dài, chiều cao và chiều rộng để phân loại xe bằng mô hình AI thay vì chỉ dùng IR33.

Tích hợp hệ thống dự phòng/khôi phục (Fail-safe)	Tự động lưu dữ liệu tạm thời khi mất kết nối và đồng bộ lại khi phục hồi. Có chế độ "bypass" mở barrier thủ công kèm ghi log.
Giao diện cảnh báo lỗi thiết bị	Thêm chức năng kiểm tra tình trạng hoạt động của camera, cảm biến, RFID. Nếu có lỗi, thông báo và ghi log để xử lý kịp thời.
Thiết kế hướng mở rộng (Scalability)	Xây dựng kiến trúc hướng đối tượng và đa luồng để xử lý nhiều cổng IN/OUT cùng lúc. Lưu trữ tập trung để dùng chung dữ liệu.

Bảng 5.5 Đề xuất cải tiến

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận

Trong quá trình thực hiện đề tài, nhóm chúng em đã nghiên cứu và xây dựng mô hình mô phỏng một hệ thống bãi đỗ xe thông minh của dự án Time Square Đà Nẵng của công ty cổ phần Việt Chèo. Mô hình có khả năng phân loại phương tiện và nhận diện biển số xe. Hệ thống sử dụng RFID để kiểm soát vào/ra, cảm biến chiều cao để phân loại xe thành SEDAN, SUV và OVERSIZE, đồng thời kết hợp với công nghệ nhận dạng biển số bằng AI để tăng tính tự động hóa và chính xác.

Đặc biệt, nhóm đã tham khảo và nghiên cứu theo nguyên lý hoạt động của hệ thống đỗ xe cơ khí PSH – một mô hình đỗ xe hiện đại và phổ biến tại các đô thị. Từ đó, nhóm rút ra các tính năng phù hợp và áp dụng vào mô hình mô phỏng của mình. Tuy chỉ sử dụng phần cứng đơn giản như Arduino MEGA 2560 và mô phỏng bằng Proteus, nhưng hệ thống vẫn đảm bảo được chu trình cơ bản như: quét thẻ – phân loại – nhận diện – điều khiển barrier – lưu dữ liệu.

Đề tài không chỉ giúp nhóm nâng cao kỹ năng lập trình, giao tiếp giữa phần cứng và máy tính, mà còn rèn luyện tư duy tích hợp hệ thống, xử lý tín hiệu và thiết kế giao diện phần mềm.

Tuy rằng sản phẩm đã hoàn thành nhưng đối với bản thân nhóm em vẫn thấy sản phẩm còn nhiều thiếu sót, cần được chỉnh sửa và cải tiến nhiều hơn

Hướng phát triển

Trong tương lai, nhóm mong muốn được mở rộng hệ thống theo hướng thực tế hóa, tích hợp với mô hình cơ khí đỗ xe dạng nâng/hạ như PSH, giúp mô phỏng chính xác hơn quy trình vận hành thực tế.

Tích hợp thêm camera để xử lý ảnh thời gian thực và tăng độ chính xác cho hệ thống nhận dạng biển số.

Áp dụng trí tuệ nhân tạo để tối ưu hóa việc phân loại phương tiện và quản lý chỗ trống trong bãi xe.

Thiết kế ứng dụng di động để người dùng có thể theo dõi tình trạng xe, kiểm tra vị trí, thanh toán hoặc đặt chỗ trước.

Mở rộng hệ thống cơ sở dữ liệu để đáp ứng lượng xe lớn, hỗ trợ triển khai tại các tòa nhà cao tầng, trung tâm thương mại hoặc khu công nghiệp.

TÀI LIỆU THAM KHẢO

- [1] [GitHub - entbappy/License-Plate-Extraction-Save-Data-to-SQL-Database](#)
- [2] [Qt Widgets Designer Manual](#)
- [3] [YOLOv10: Phát hiện đối tượng đầu cuối theo thời gian thực - Ultralytics YOLO Tài liệu](#)
- [4] <https://kztek.net/giai-phap/giai-phap-quan-ly-xe-vao-ra/he-thong-kiem-soat-xe-vao-ra>
- [5] <https://vietchao.vn/product/he-thong-do-xe-ban-tu-dong-xizi-nhat-ban/>

PHỤ LỤC

- [1] Source Code phần mềm bãi đỗ xe thông minh
- [2] Source Code giao diện phần mềm
- [3] Source Code Arduino Mega 2560