

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA



BÀI TẬP LỚN

MÔN THIẾT KẾ HỆ THỐNG NHÚNG

ĐỀ TÀI: BÃI XE THÔNG MINH

LỚP L01--- NHÓM 8 --- HK 251

Giảng viên hướng dẫn: TRẦN HOÀNG QUÂN

Sinh viên thực hiện	Mã số sinh viên	Điểm số
Nguyễn Xuân Đức	2310792	
Lê Minh Quân	2312824	
Trương Tấn Đạt	2310713	
Bạch Lê Phi Trường	2313697	
Nguyễn Huỳnh Đức	2310784	

SĐT nhóm trưởng: 0902458373

Thành phố Hồ Chí Minh – 2025

LỜI NÓI ĐẦU	3
1. Giới thiệu về hệ thống	4
1.1 Đặc tả hệ thống	4
1.2 Đặc tả về kỹ thuật.....	5
1.3 Đặc tả phần cứng.....	8
1.4 Đặc tả phần mềm	8
1.5 Thao tác kiểm thử	9
1.6 Thời gian thực hiện sản phẩm	11
2. Issue khi thiết kế hệ thống	13
2.1 Constraint issues	13
2.2 Functional issues	13
2.3 Real-time issues	13
2.4 Concurrent issues	13
2.5 Reactive issues	13
3. Phần cứng của hệ thống.....	14
4. Phần mềm của hệ thống.....	16
4.1 Lưu đồ giải thuật.....	16
4.2 Code chương trình	18
5. Sản phẩm cuối cùng	26
5.1 Mô hình bãi xe thông minh	26
5.2 Kiểm tra và đánh giá hệ thống vận hành	26
KẾT LUẬN.....	28

Bảng kế hoạch BTL	
Tên nhóm	Nhóm 8
Tên sản phẩm	Bãi xe thông minh
Chức năng chính	<ul style="list-style-type: none"> -Nạp ID thẻ mới vào dữ liệu lưu trữ -Tự cập nhật số lượng xe trong bãi. -Tự mở, đóng cửa khi có xe ra vào. -Nhận biết thẻ có hợp lệ hay không.
Thời gian ước tính	<ul style="list-style-type: none"> -1 tháng -Bắt đầu vào: tuần 43 -Kết thúc vào: tuần 48
Chi phí ước tính	-Khoản 500k
Thành viên	<ul style="list-style-type: none"> -Nguyễn huỳnh Đức: Trưởng nhóm, phần cứng, debug -Nguyễn Xuân Đức: làm mô hình, mua đồ -Lê Minh Quân: làm mô hình, mua đồ -Bạch Lê Phi Trường: Phần mềm -Trương Tấn Đạt: phần mềm, phần cứng, debug

LỜI NÓI ĐẦU

Trong bối cảnh các hệ thống tự động ngày càng được ứng dụng rộng rãi, những giải pháp quản lý bằng công nghệ nhúng đang trở thành xu hướng phổ biến trong đời sống hiện đại. Bãi xe thông minh là một ví dụ điển hình, nơi việc kiểm soát ra vào, nhận diện phương tiện và tối ưu quy trình quản lý được thực hiện tự động thay cho các phương pháp thủ công truyền thống. Việc nghiên cứu và xây dựng mô hình hệ thống bãi xe thông minh giúp sinh viên hiểu rõ hơn về cách kết hợp giữa phần cứng, phần mềm và công nghệ cảm biến trong một ứng dụng thực tế.

Trong quá trình thực hiện đề tài, nhóm chúng em tiến hành khảo sát yêu cầu hệ thống, đề xuất phương án điều khiển và lựa chọn các linh kiện phù hợp như vi điều khiển, cảm biến, mô-đun RFID, servo, màn hình hiển thị, cảm biến vật cản... Hệ thống được thiết kế nhằm mô phỏng quy trình hoạt động của một bãi xe thông minh, bao gồm nhận diện thẻ, điều khiển barie, hiển thị thông tin và xác định trạng thái chỗ trống. Các thành phần được kết nối và lập trình để hoạt động đồng bộ, đảm bảo mô hình có thể vận hành liên tục và phản ứng chính xác theo từng tình huống.

Báo cáo này trình bày toàn bộ quá trình làm việc, từ phân tích yêu cầu, thiết kế sơ đồ khối, nguyên lý hoạt động, xây dựng phần cứng, lập trình phần mềm, cho đến thử nghiệm và đánh giá kết quả. Thông qua đề tài, nhóm có cơ hội áp dụng kiến thức đã học vào một mô hình gần gũi với thực tế, qua đó rèn luyện tư duy thiết kế hệ thống, xử lý lỗi và tối ưu tính năng. Nội dung báo cáo cũng đề cập các hướng mở rộng để hệ thống hoàn thiện và có thể áp dụng vào mô hình lớn hơn trong tương lai.

1. Giới thiệu về hệ thống

1.1 Đặc tả hệ thống

-Tên sản phẩm: bãi xe thông minh

-Mục đích: quản lý, thực hiện các chức năng của bãi xe .

-Ngõ vào/ra:

1)Ngõ vào: cảm biến hồng ngoại IR, RFID, nút ấn đổi chế độ lưu thẻ.

2) Ngõ ra: màn hình LCD, động cơ SERVO.

-Trường hợp sử dụng:

Chế độ quét và lưu thẻ mới:

- Nút nhấn giữ dùng để đổi giữa 2 chế độ là đọc/lưu thẻ và chế độ bãi giữ xe thông minh.
- Khi trong chế độ đọc/lưu thẻ thì hệ thống sẽ đọc UID của thẻ.
- Sau đó hệ thống sẽ kiểm tra xem ID của thẻ có từng được lưu trước đó không, nếu có thì hiện CARD_EXIST trên LCD, nếu không thì ID mới này sẽ được lưu vào vùng EEPROM-nơi chứa ID thẻ.

Chế độ bãi giữ xe thông minh:

- Chuyển đổi sang chế độ bằng cách nhấn nút lúc này sẽ chuyển sang chế độ quét thẻ cho bãi giữ xe.
- Mỗi mã thẻ được lưu trong EEPROM bao gồm 2 thành phần: ID thẻ và trạng thái thẻ. Trạng thái thẻ ứng với mức 0 tức là thẻ chưa vào bãi giữ xe, mức 1 là đã vào bãi giữ xe, nên chỉ cần 1 RFID ta có thể kiểm soát cả cổng vào ra.
- ID được quét vào sẽ kiểm tra xem ID trên thẻ có đúng với dữ liệu được lưu trong EEPROM không, nếu thẻ lạ cổng vào sẽ không được mở. Nếu thẻ

đúng, cổng vào sẽ được mở ứng với trạng thái thẻ ban đầu là 0, sau khi thẻ được xác nhận đi vào bãi, trạng thái thẻ sẽ chuyển sang 1.

- Phía sau cổng vào (servo) được trang bị cảm biến hồng ngoại, vị trí đặt cảm biến phù hợp để khi xe vừa qua khỏi cổng thì lúc này cảm biến sẽ nhận diện và đóng cổng vào.
- Khi ra ta cũng quét tương tự lúc vào trạng thái thẻ sẽ chuyển từ mức 1 sang mức 0 và xe khi ra cũng có cảm biến ở ngoài để đóng cổng lại.

- Chức năng:

- 1) Đọc/lưu thẻ
- 2) Tự động mở cửa, đóng cửa khi có xe vào ra
- 3) Tự kiểm tra thẻ có hợp lệ hay không, tự quản lý số lượng chỗ trống trong bãi xe

-Hiệu năng: hệ thống quét thẻ liên tục, mở cửa khi thẻ quét hợp lệ, và tự động đóng cửa sau khi xe đi qua với thời gian trễ dưới 1s.

-Công suất: 2W

-Cài đặt: lắp pin và cho mô hình hoạt động.

-Đạt chuẩn: không có.

1.2 Đặc tả về kỹ thuật

-Linh kiện điện tử và vi xử lý

STT	Tên sản phẩm	Mô tả
1	VXL ATmega328p	-Tổng số chân ngõ vào Analog là 6 -Chứa 1 kilobyte EEPROM -Nhiệt độ tối thiểu và tối đa -40 độ C đến 105 độ C. -Tổng số chân I / O kỹ thuật số là 14 chân -Chứa tổng cộng ba bộ định thời, hai 8 bit và một 16 bit -Tổng số chân I / O là 23 chân

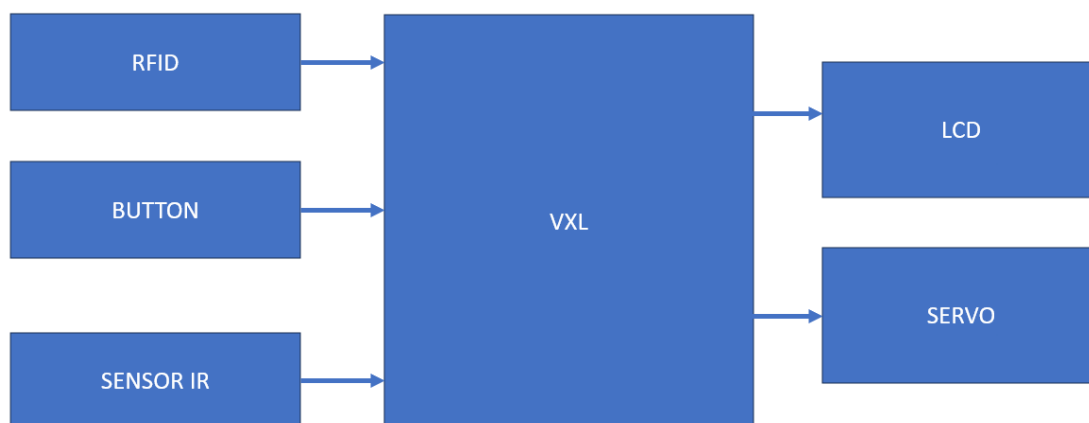
		-Điện áp hoạt động tối thiểu và tối đa từ 1.8V DC đến 5.5V DC
2	LCD 1602 nền vàng xanh chữ đen 5v	-Điện áp hoạt động là 5 V. -Kích thước: 80 x 36 x 12.5 mm -Chữ trắng, nền xanh dương
3	Động cơ SERVO SG90	-Điện áp hoạt động: 4.8-5VDC -Góc quay của trục: 0-180° -Tốc độ: 0.12 sec/ 60 degrees (4.8VDC) -Lực kéo: 1.6KG.CM -Kích thước: 21x12x22mm -Trọng lượng: 9g.
4	Mạch RFID NFC 13.56MHz RC522, tag reader	Nguồn sử dụng: 3.3VDC Dòng điện: 13~26mA Tần số hoạt động: 13.56Mhz Khoảng cách hoạt động: 0~60mm (mifare1 card) Chuẩn giao tiếp: SPI Tốc độ truyền dữ liệu: tối đa 10Mbit/s -Các loại card RFID hỗ trợ: mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire -Kích thước: 40mm × 60mm
5	Nút ấn giữ	-Số chân : 2 chân -Kích thước: 12x12x9.4mm
6	Thạch anh 16Mhz HC-49S	-tần số: 16MHz -số chân: 2 -Điện dung tải: 18pF -kích thước: 11.4 x 4.7 x 4.2mm -nhiệt độ min-max: -20 đến 70 độ c
7	IC LM-7805	- Đầu ra 5V chính xác và cố định

		- Dòng điện đầu ra là 1,5 Ampe -Điện áp đầu vào tối đa là 35V DC
8	IC LM1117T-3.3	-Điện áp ngõ ra là 3,3v -Dòng điện ngõ ra là 800mA -Điện áp đầu vào tối đa là 15v
9	IC 74HC08	-IC cổng AND -Điện áp hoạt động 2V-6V (phổ biến 3,3V-5V)
10	IC 74HC04	-IC cổng NOT -Điện áp hoạt động 5V-6V

-Requirements:

1. Phát hiện xe đi ra/vào bằng cảm biến hồng ngoại IR
2. Đếm số lượng chỗ trống trong bãi xe.
3. Điều khiển servo motor tự động mở/đóng khi có xe đi vào/ra.
4. Hiển thị số xe hiện tại và trạng thái(“còn chỗ” / “hết chỗ”).
- 5.Xử lý và cập nhật số liệu thời gian thực.

-Sơ đồ khối:



1.3 Đặc tả phần cứng

- VXL Atmega328p, LCD 1602(with I2c interface), cảm biến hồng ngoại IR, động cơ SERVO,

- Nguyên lý hoạt động: khi có xe đi tới, RFID sẽ đọc thẻ được quét. Nếu hợp lệ thì hệ thống sẽ tự động tính toán bãi xe còn chỗ trống hay không, nếu còn thì hệ thống sẽ đưa tín hiệu cho SERVO để mở cổng và servo sẽ tự động đóng cổng sau khi xe đi qua bằng tín hiệu từ cảm biến hồng ngoại IR. Nút ấn giữ dùng để đổi giữa 2 chế độ là quét thẻ để cho xe ra vào, cả chế độ đọc/lưu thẻ dùng để lưu thêm thẻ.

1.4 Đặc tả phần mềm

- Các chương trình khởi tạo:

- 1) LCD_I2C_init();
- 2) SPI_master_init();
- 3) rfid_init();
- 4) init_systick_us();
- 5) init_systick_ms();
- 6) pwm_init(void);
- 7) clear_all_eeprom();

-Các chương trình chính:

- 1) lcd_task(): Điều khiển hiển thị những thông tin trên LCD
- 2) servo_in_task(): Điều khiển SERVO khi có xe đi vào
- 3) servo_out_task(): Điều khiển SERVO khi có xe đi ra
- 4) main(): Nơi chương trình chính của hệ thống hoạt động.

1.5 Thao tác kiểm thử

-VOM

-prototype: breadboard

-quá trình test:

1) Sử dụng VOM, chính 2 đầu đen và đỏ vào lần lượt GND và VCC để đo xem thử mạch nguồn sau hạ áp có đúng với 5v hay 3,3v hay không.

2) Kiểm tra khối VXL có hoạt động đúng với yêu cầu, nạp code bằng Arduino IDE hoặc Microchip Studio cho sáng một chân I/O port bất kỳ, nối dây từ port đó ra breadboard, nối tiếp với một điện trở khoảng tầm 220 ohm rồi nối tiếp với LED sau đó nối với GND để kiểm tra xem thử khối VXL có hoạt động hay không. Tiếp tục sau đó kiểm tra nút reset của khối VXL có hoạt động hay không. Bằng cách ấn nút reset xem LED có tắt hay không.

3) Kiểm tra xem thử LCD có hoạt động hay không. Nạp code cho VXL bằng Arduino IDE hoặc Microchip studio, in ra màn hình những chữ cái đơn giản như “Hello” hay tương tự như thế để kiểm tra LCD I2C có giao tiếp được hay không.

4) Kiểm tra xem thử RFID có đọc được thẻ và trả về mã UID hay không. Sử dụng Arduino IDE hoặc Microchip studio để nạp code kiểm tra RFID bằng cách cho LCD in ra UID của thẻ đọc từ RFID.

5) Kiểm tra xem thử nút ấn giữ có đổi được giữa 2 chế độ hay không. Sử dụng Arduino IDE hoặc Microchip Studio để nạp code đọc tín hiệu từ nút ấn giữ rồi sau đó quan sát màn hình LCD và quét thẻ bằng RFID nếu màn hình RFID in ra số UID của thẻ và sau đó hiện chữ “CARD DETECTED” hoặc “CARD EXITS” thì hệ thống đang ở chế độ đọc/lưu thẻ. Ấn nút nhấn thêm một lần nữa để đổi sang chế độ quét. Quét thẻ lên RFID nếu màn hình LCD hiện mã UID của thẻ và sau đó hiện số lượng chỗ trống “REMAINING PARK:...” thì nút ấn giữ đã hoạt động đúng với yêu cầu đề ra.

6) Kiểm tra xem thử SERVO có quay đúng với yêu cầu hay không. Sau khi chuyển qua chế độ quét thì quét thẻ lên RFID, sau đó hệ thống sẽ kiểm tra thử xem còn chỗ trống

hay không, tiếp tục nếu thẻ hợp lệ và bãi xe còn chỗ trống thì sau đó SERVO đầu vô sẽ quay để mở cổng cho xe đi vô. Lúc này mình sẽ kiểm tra xem thử là SERVO đã quay đúng 90 độ hay chưa để có thể chỉnh sửa lại cho đúng với yêu cầu. Tương tự như vậy, khi xe đã đi vô, mình quét thẻ mà đã dừng trước đó quét thêm lần nữa để cho xe đi ra và tiếp tục kiểm tra xem thử SERVO bên còn lại có quay đúng 90 độ và trả lại đúng vị trí ban đầu hay không.

7) Kiểm tra xem thử cảm biến hồng ngoại IR có trả về tín hiệu hay không. Bằng cách nạp code từ Arduino IDE hoặc Microchip studio đọc tín hiệu từ chân PD6, sau đó xuất tín hiệu từ 2 chân đó ra LCD để kiểm tra xem thử là cảm biến hồng ngoại có trả về tín hiệu chính xác hay không, nếu cảm biến hồng ngoại không sáng(không có vật gì trong tầm) thì LCD sẽ in ra "PD6: SANG" nếu có vật gì trong tầm thì LCD sẽ in ra "PD6 TAT" từ đó nếu như ta kiểm tra thấy đúng với yêu cầu và hồng ngoại hoạt động ổn định, trả tín hiệu chính xác thì ta xác nhận là hồng ngoại hoạt động đúng với yêu cầu.

8) Kiểm tra xem thử toàn hệ thống có hoạt động đúng với yêu cầu hay không.

+)Đầu tiên ta kiểm tra xem thử hệ thống đang ở chế độ đọc/lưu hay quét bằng cách ấn nút ấn giữ và quét thử thẻ lên RFID nếu màn hình LCD hiện lần lượt UID và dòng chữ "CARD DETECTED" hoặc "CARD EXITS" thì có nghĩa là hệ thống đang ở chế độ đọc/lưu thẻ, còn nếu không hiện những dòng trên thì hệ thống đang ở chế độ quét thẻ, ấn nút ấn giữ thêm một lần nữa để đổi về chế độ đọc/lưu thẻ tiếp tục quét hết những thẻ ta có.






+)Xong sau đó ấn nút ấn giữ thêm một lần nữa, tiếp tục quét thẻ lên RFID, nếu mà lúc này lúc này hệ thống hiện mã UID của thẻ đồng thời với việc SERVO quay và LCD hiện lên thông tin "REMAINING PARK: 1" thì có nghĩa là lúc này đã có xe đi vô bãi xe và bãi xe chỉ còn một chỗ, sau khi xe đi qua cảm biến hồng ngoại IR thì SERVO sẽ tự động đóng cửa lại. Tiếp tục quét tiếp thêm một thẻ mới nữa thì hệ thống sẽ hoạt động tương tự vậy và LCD sẽ cập nhật "REMAINING PARK: 0" thì lúc này có nghĩa là bãi xe đã hết chỗ đậu và nếu còn tiếp tục quét thẻ thì màn hình LCD sẽ hiện ra dòng chữ "No space left" và hệ thống sẽ không mở cửa cho xe mới đi vô.

+) Nếu ta quét những thẻ của những xe đã đi vô thì sau đó hệ thống sẽ mở cửa để xe đi ra và tự đóng lại sau khi xe chạy qua cảm biến hồng ngoại. ngay sau đó sẽ cập nhật lại trên màn hình LCD “REMAINING PARK:” Với chỗ trống tương ứng.

+) Nếu trong suốt quá trình quét thẻ, có một thẻ nào đó chưa được lưu vô hệ thống thì LCD sẽ cho ra mã UID kèm dòng chữ “UNIDENTIFIED” có nghĩa là thẻ ko xác định được cho chưa được hệ thống đọc và lưu. Lúc này nếu muốn lưu thêm thẻ thì ta ấn nút ấn giữ thêm một lần nữa, quét thẻ mới mà ta muốn lưu vô. Nếu màn hình LCD hiện ra mã UID của thẻ cùng với dòng chữ “CARD DETECTED” thì có nghĩa là hệ thống đã đọc và lưu thẻ thành công. Ấn nút ấn giữ thêm một lần nữa để chuyển về chế độ quét thẻ và quét thẻ ta vừa lưu, nếu LCD hiện UID của thẻ, SERVO quay và LCD cập nhật số chỗ trống trong bãi xe thì hệ thống của đã hoạt động đúng với yêu cầu.

1.6 thời gian thực hiện sản phẩm

SƠ ĐỒ GANTT THỰC HIỆN BÀI TẬP LỚN

Công việc	Tuần 43	Tuần 44	Tuần 45	Tuần 46	Tuần 47	Tuần 48
BTL						
Phần mềm						
Prototype breadboard						
Sản phẩm						
Mô hình						

-Phần mềm: lập trình các thư viện EEPROM, giao tiếp I2C, SPI để giao tiếp với các ngoại vi RFID và LCD.

-Prototype breadboard: là quá trình cắm breadboard để kiểm tra phần mềm đã chạy đúng với yêu cầu hay không và fix bugs nếu có xuất hiện.

-Quá trình làm sản phẩm là quá trình hàn mạch và kiểm tra những bug của hệ thống trong, kiểm tra nguồn pin có ổn định và đủ cấp cho hệ thống hay không.

-Quá trình làm mô hình diễn ra sau quá trình hàn mạch và kiểm tra xem có chạy ổn định hay không. Nhóm em tiến hành làm một mô hình bãi xe.

-Đánh giá:

+)Trong quá trình làm mô hình, mô hình vẫn chạy bình thường nhưng vào buổi tối trước hôm báo cáo thì phần nguồn của 5v cấp cho MCU bị vấn đề nên nhóm đã cố gắng sửa nhưng không kịp.

+)Trong quá trình hàn mạch thì do là hàn bằng board lỗ với dây nên dẫn tới là dây dễ đứt, nhóm chúng em đưa ra giải pháp là đổ thêm nhựa thông vào.

+)Trong suốt quá trình thực hiện sản phẩm, các thành viên trong nhóm tương đối phối hợp với nhau để làm BTL nhưng chưa hiệu quả lắm.

2. Issue khi thiết kế hệ thống

Trong quá trình thiết kế và triển khai mô hình bãi xe thông minh, nhóm gặp phải nhiều vấn đề liên quan đến ràng buộc hệ thống, chức năng, thời gian thực, xử lý đồng thời và khả năng phản ứng khi xảy ra lỗi.

2.1 constraint issues

- Chi phí sản xuất < 200K.
- Vòng đời sản phẩm > 1 năm.
- Độ tin cậy 80%.

2.2 Functional issues

Nhận diện xe chưa chính xác tuyệt đối, người đi bộ có thể bị nhận diện nhầm là xe. Giới hạn chức năng: Chỉ quản lý số lượng xe, chưa phân biệt loại xe. Chưa có chức năng nhận dạng biển số hoặc thẻ RFID nâng cao.

2.3 Real-time issues

Hệ thống soft real-time: khi có sự cố xảy ra không ảnh hưởng đến toàn hệ thống và vẫn đảm bảo phản hồi kịp thời khi có thể được đọc và xe ra vào.

2.4 Concurrent issues

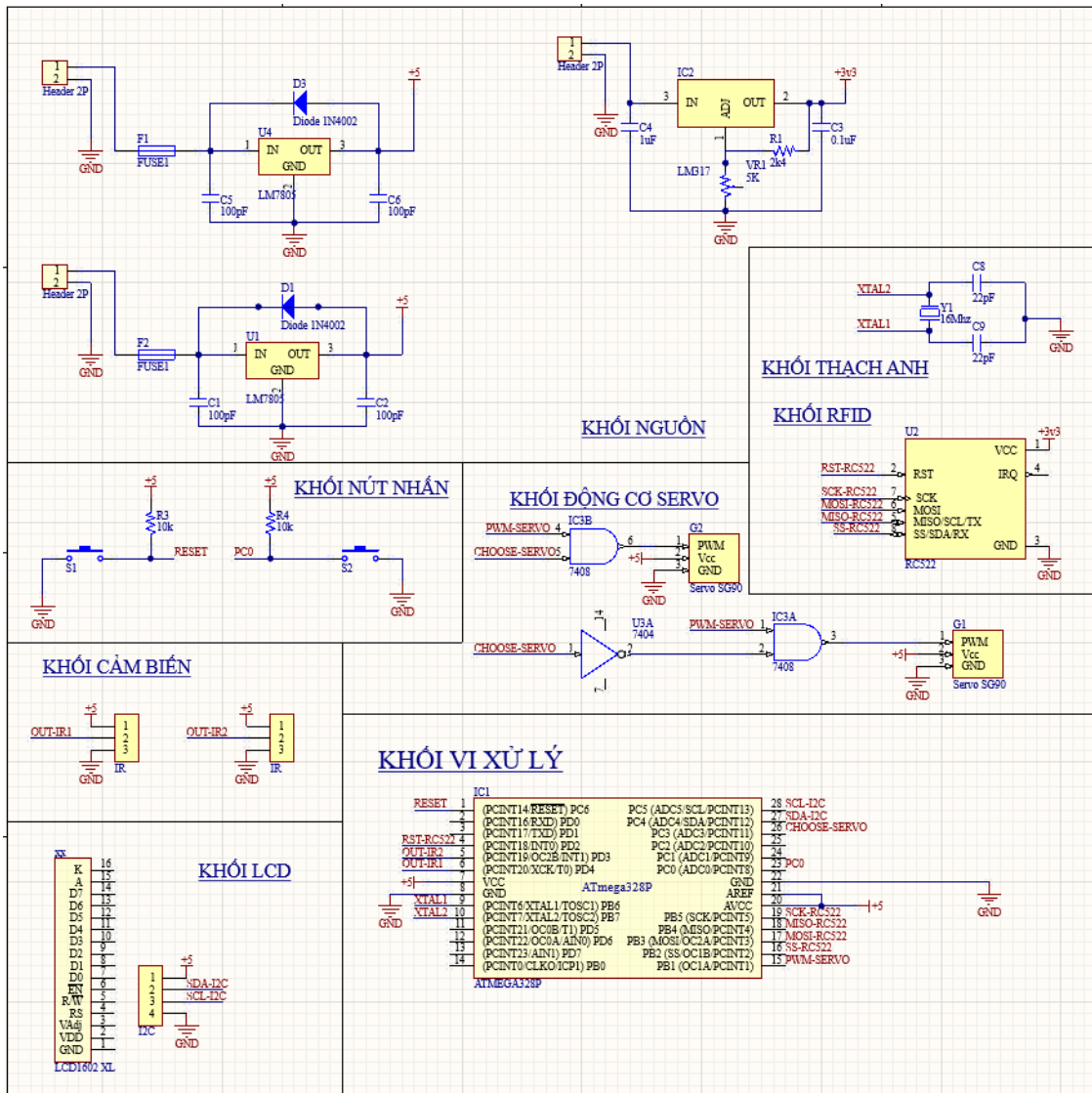
Hệ thống có timer riêng cho các ngoại vi: Timer cho SERVO, Timer cho LCD. Nên hệ thống có thể chạy đồng thời các tác vụ mà không bị ảnh hưởng đến hệ thống.

2.5 Reactive issues

Hệ thống sẽ treo tại chỗ khi xảy ra lỗi và LCD sẽ in ra thông báo lỗi, chưa có chức năng tự khởi động lại khi bị lỗi.

3. Phần cứng của hệ thống

-Sơ đồ nguyên lý:



-thông tin phần cứng:

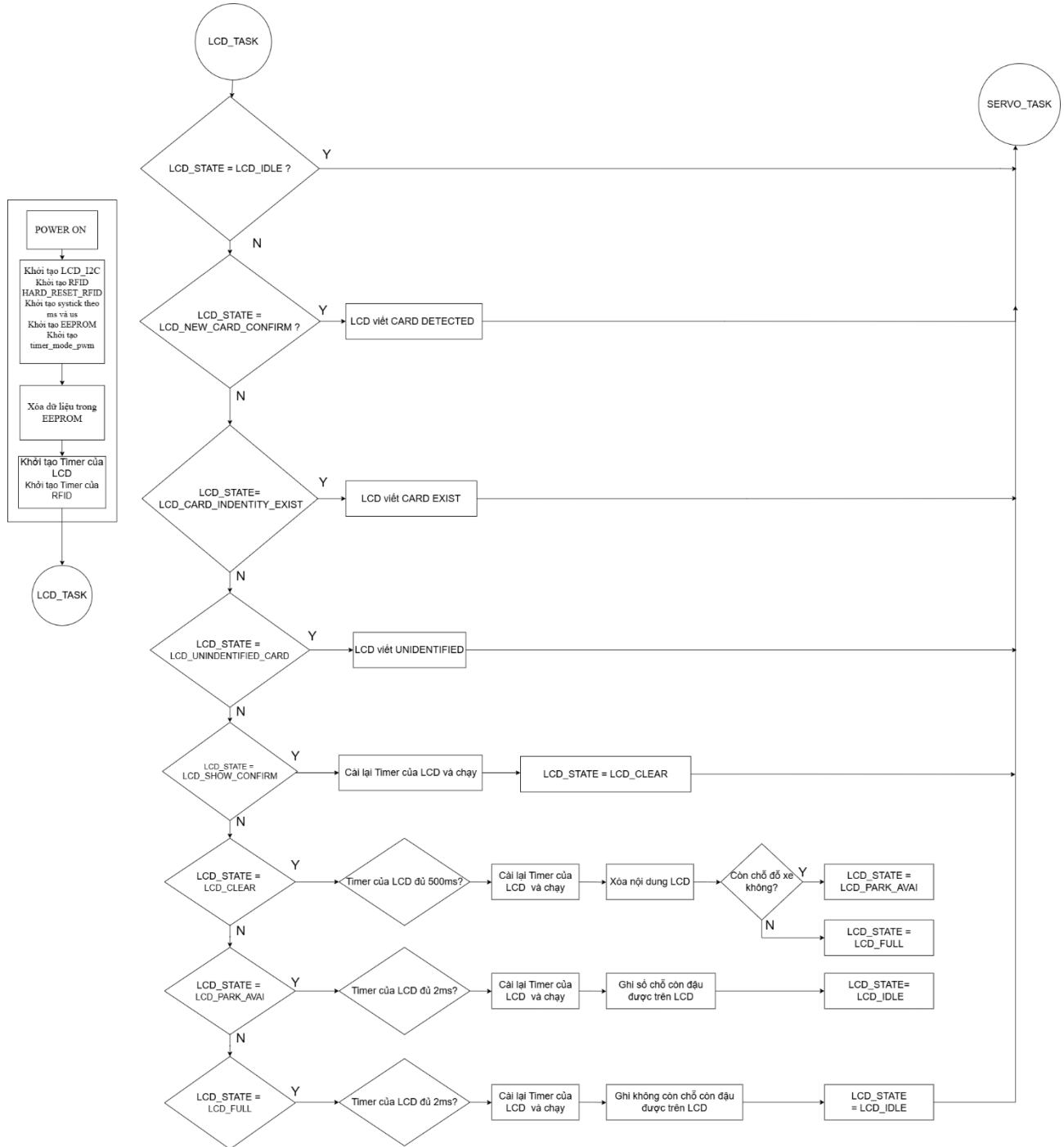
- 1/VXL ATmega328p: Có chức năng xử lý các tín hiệu input từ các ngoại vi: cảm biến hồng ngoại IR, RFID, nút ấn giữ. Từ đó xử lý xem thẻ đọc từ RFID có hợp lệ hay không, từ đó xuất tín hiệu cho servo chạy thực hiện chức năng đóng mở cổng.
- 2/ LCD: Có chức năng hiển thị thông tin của UID, hiển thị thông tin bãi xe còn bao nhiêu chỗ trống và hiện trạng thái của hệ thống
- 3/Nút ấn: Chuyển đổi giữa hai chế độ đọc/ lưu và quét thẻ.

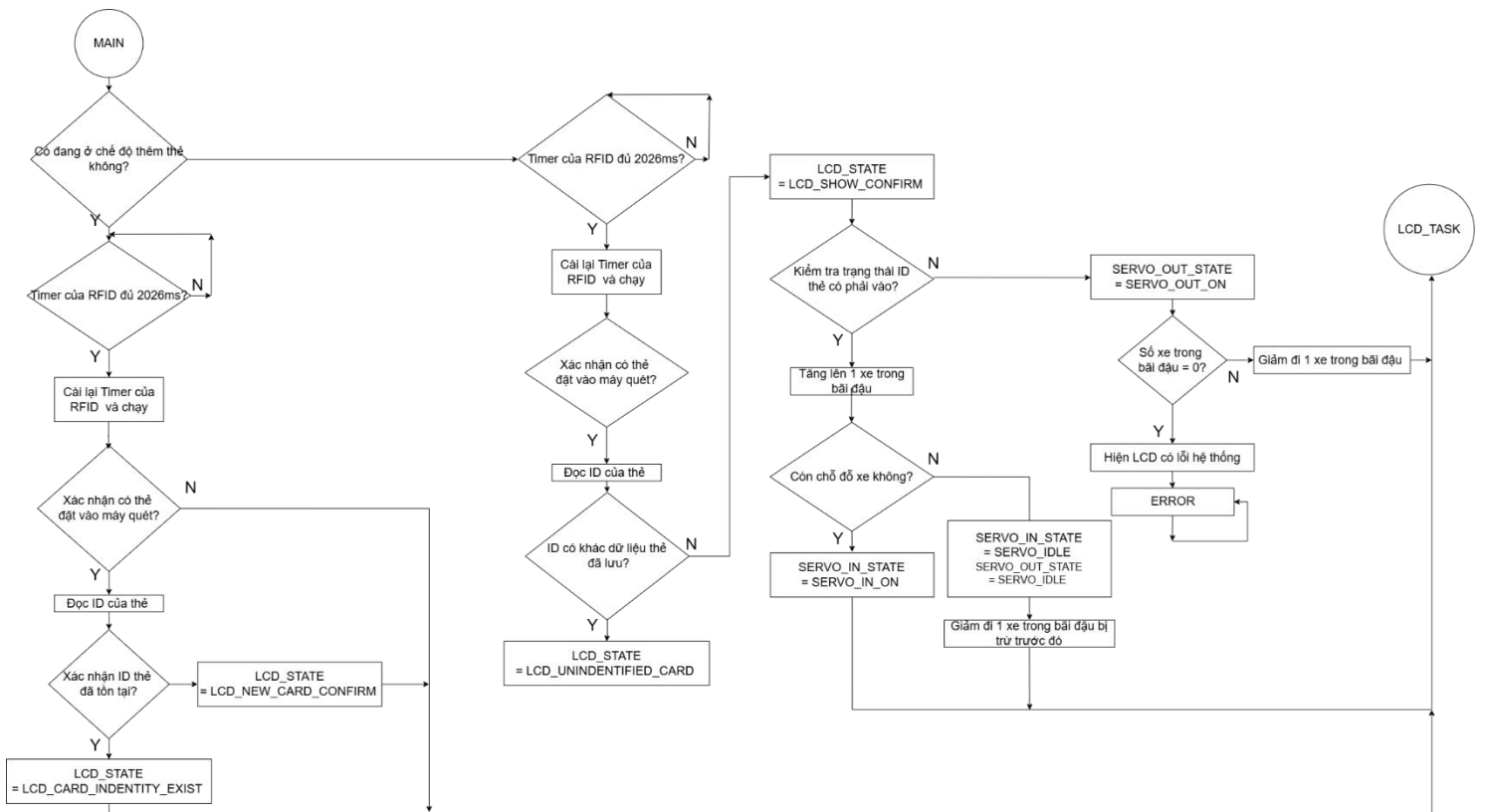
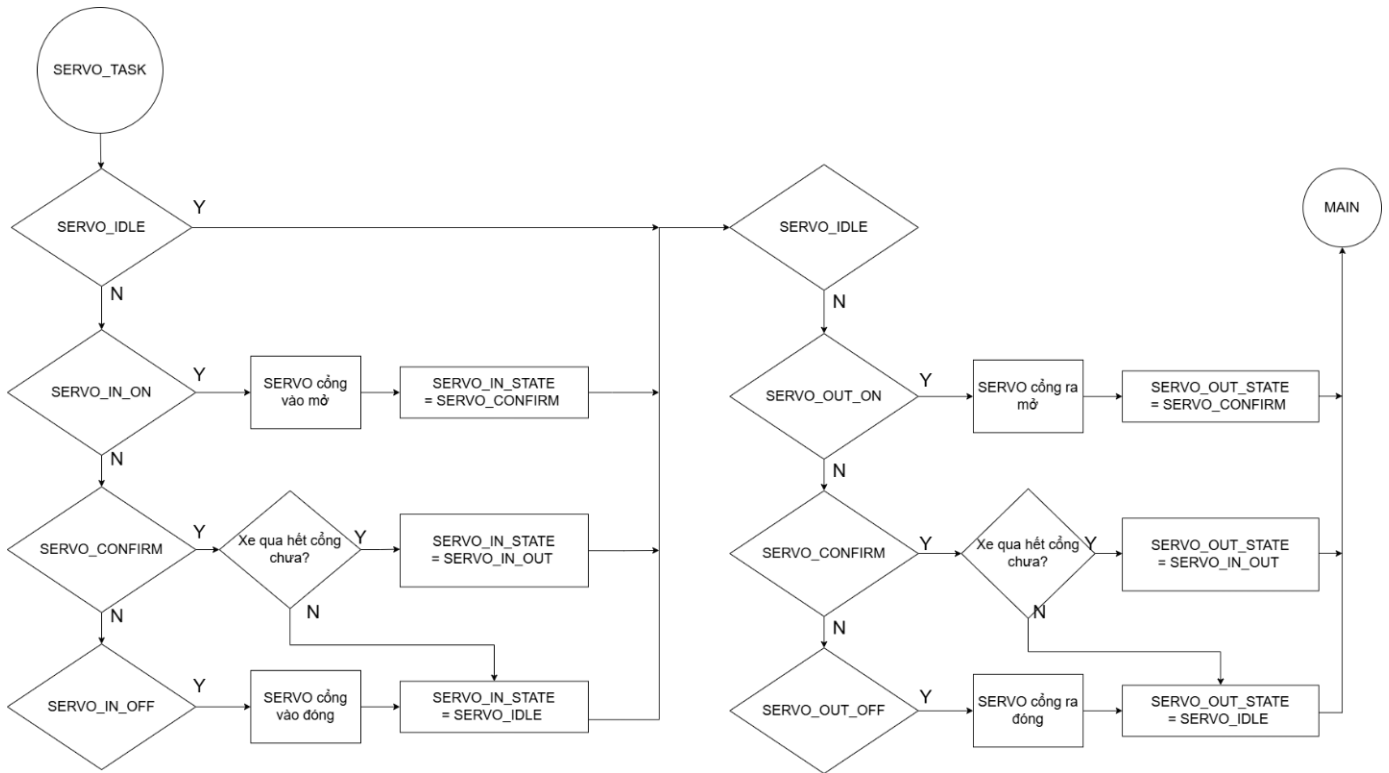
4/RFID: Đọc mã thẻ và gửi tín hiệu về cho VXL xử lý thông tin.

5/ Khối nguồn: Cấp nguồn pin 9v qua hạ áp thành 3,3v và 5v để vận hành hệ thống.

4. Phần mềm của hệ thống

4.1 lưu đồ giải thuật





4.2 code chương trình

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/eeprom.h>
#include <stdint.h>
#include "SPI_MASTER.h"
#include "I2C_Master.h"
#include "Microcontroller.h"
#include "EEPROM.h"

#define CHOOSE_SERVO (PC3)
#define CHOOSE_SERVO_DR (DDRC)
#define CHOOSE_SERVO_PORT (PORTC)
#define READ_STATE (PD2) //CHANGE TO SCAN AND SAVE NEW ID
#define LED_TEST1 (PC4)
#define LED_TEST2 (PC3)

#define BUZZER (PC2) //output
// #define TRIG (PD7) //delete
// #define ECHO (PB0) //delete

#define IN_SENSOR_1 (PD7) //IR SENSOR, SERVO IN
#define IN_SENSOR_2 (PD6) //IR SENSOR, SERVO OUT
#define IN_SERVO (1)
#define OUT_SERVO (2)
#define Parked (2)

typedef enum {
    LCD_IDLE,
    LCD_SHOW_CONFIRM,
    LCD_UNIDENTIFIED_CARD,
    LCD_NEW_CARD_CONFIRM,
    LCD_CARD_IDENTITY_EXIST,
    LCD_CLEAR,
    LCD_PARK_AVAI,
    LCD_FULL
} lcd_state_t;
```

```

typedef enum {
    SERVO_IDLE,
    SERVO_IN_ON,
    SERVO_IN_OFF,
    SERVO_OUT_ON,
    SERVO_OUT_OFF,
    SERVO_CONFIRM
} servo_state_t;

typedef enum{
    HCSR04_IDLE,
    START_PULSE,
    HIGH_PULSE,
    LOW_PULSE,
    CHECK_ANY_VEHICLE
}HCSR04_state_t;

lcd_state_t lcd_state = LCD_IDLE;
uint32_t lcd_timer = 0;
uint32_t taken_spot = 0;
servo_state_t servo_in_state = SERVO_IDLE;
servo_state_t servo_out_state = SERVO_IDLE;
uint16_t eeprom_address = 0x01;
uint8_t data[4] = {};

void pwm_init(void)
{
    DDRB |= (1 << PB1);

    TCCR1A = (1 << COM1A1) | (1 << WGM11);
    TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11); // Prescaler =
8

    ICR1 = 39999;
}

void pwm_set_duty_ms(float pulse_ms)
{
    OCR1A = (uint16_t)(pulse_ms * 2000.0);
}

```

```

void pwm_set_angle(float angle)
{
    if (angle < 0) angle = 0;
    if (angle > 180) angle = 180;

    float pulse_ms = 1.0 + (angle / 180.0) * 1.0; // 1 -> 2 ms
    pwm_set_duty_ms(pulse_ms);
}

void lcd_task()
{
    switch (lcd_state)
    {
        case LCD_IDLE:
            break;

        case LCD_NEW_CARD_CONFIRM:
            LCD_I2C_write_string("CARD DETECTED",1,1);
            break;

        case LCD_CARD_IDENTITY_EXIST:
            LCD_I2C_write_string("CARD EXIST",1,1);
            break;

        case LCD_UNIDENTIFIED_CARD:
            LCD_I2C_write_string("UNIDENTIFIED",1,1);
            break;

        case LCD_SHOW_CONFIRM:
            lcd_timer = systick_ms();
            lcd_state = LCD_CLEAR;
            break;

        case LCD_CLEAR:
            if (systick_ms() - lcd_timer >= 500)
            {
                lcd_timer = systick_ms();
            }
    }
}

```

```

        LCD_Clear();
        if(taken_spot > Parked)
        {
            lcd_state = LCD_FULL;
        }
        else
        {
            lcd_state = LCD_PARK_AVAI;
        }
    }
    break;

case LCD_PARK_AVAI:
    if (systick_ms() - lcd_timer >= 2)
    {
        lcd_timer = systick_ms();
        LCD_I2C_write_string("Remaining park",1,1);
        LCD_I2C_write_number(Parked - taken_spot,2,1);
        lcd_state = LCD_IDLE;
    }
    break;

case LCD_FULL:
    if (systick_ms() - lcd_timer >= 2)
    {
        lcd_timer = systick_ms();
        LCD_I2C_write_string("No space left",1,1);
        uint8_t dummy_state = 0;
        read_eeprom(data,&dummy_state);
        lcd_state = LCD_IDLE;
    }
    break;

default:
    break;
}
}

void servo_in_task()

```

```

{
    switch (servo_in_state)
    {
        case SERVO_IDLE:
            break;

        case SERVO_IN_ON:
            digital_write(&CHOOSE_SERVO_PORT,CHOOSE_SERVO,1);
            pwm_set_duty_ms(2.2);
            digital_write(&PORTC,LED_TEST1,1);
            servo_in_state = SERVO_CONFIRM;
            break;

        case SERVO_CONFIRM:
            if(digital_read(PIND,IN_SENSOR_1) == 0) //xai cam bien chuyen
            {
                servo_in_state = SERVO_IN_OFF;
            }
            break;

        case SERVO_IN_OFF:
            digital_write(&CHOOSE_SERVO_PORT,CHOOSE_SERVO,1);
            pwm_set_duty_ms(1);
            servo_in_state = SERVO_IDLE;
            break;

        default:
            break;
    }
}

void servo_out_task()
{
    switch (servo_out_state)
    {
        case SERVO_IDLE:
            break;

        case SERVO_OUT_ON:

```

```

        digital_write(&CHOOSE_SERVO_PORT,CHOOSE_SERVO,0);
        pwm_set_duty_ms(2);
        servo_out_state = SERVO_CONFIRM;
        break;

        case SERVO_CONFIRM:
            if(digital_read(PIND,IN_SENSOR_2) == 0) //xai cam bien chuyen
ve 1
            {
                servo_out_state = SERVO_OUT_OFF;
            }
            break;

        case SERVO_OUT_OFF:
            digital_write(&CHOOSE_SERVO_PORT,CHOOSE_SERVO,0);
            pwm_set_duty_ms(1);
            servo_out_state = SERVO_IDLE;
            break;

        default:
            break;
    }
}

int main(void)
{
    //-----
    //initialize
    LCD_I2C_init();
    LCD_I2C_set();
    SPI_master_init();
    RFID_hard_reset();
    rfid_init();
    init_systick_us();
    init_systick_ms();
    eeprom_address = eeprom_read_byte((uint8_t*)0x00);
    if (eeprom_address == 0xFF) eeprom_address = 0x01;

    CHOOSE_SERVO_DR = CHOOSE_SERVO_DR | (1 <<
    CHOOSE_SERVO);

```

```

DDRD = DDRD & ~(1 << READ_STATE));
DDRD = DDRD & ~(1 << IN_SENSOR_1)) & ~(1 << IN_SENSOR_2));

DDRC |= (1 << BUZZER);

//-----
uint32_t rfid_time = systick_ms();
uint32_t write_eeprom_time = systick_ms();
uint32_t count_tag = 0;
uint8_t check_state = 0;
pwm_init();
clear_all_eeprom();
while (1)
{
    lcd_task();
    servo_in_task();
    servo_out_task();

    if(digital_read(PIND,READ_STATE) == 0)
    {
        if(systick_ms() - write_eeprom_time > 2026)
        {
            write_eeprom_time = systick_ms();
            if(READ_UID_BCC_CMD(data) == SUCCESS)
            {
                uint8_t check_exist =
write_eeprom(eeprom_address,data);
                if(check_exist == 0)
                {
                    lcd_state =
LCD_NEW_CARD_CONFIRM;
                    count_tag++;
                }
                else if(check_exist == DATA_EXIST)
                {
                    lcd_state =
LCD_CARD_INDENTITY_EXIST;
                }
            }
        }
    }
}

```

```

    }
    else
    {
        if(systick_ms() - rfid_time >= 2026)
        {
            rfid_time = systick_ms();
            if(READ_UID_BCC_CMD(data) == SUCCESS)
            {
                if(read_eeprom(data,&check_state) == SUCCESS)
                {
                    lcd_state = LCD_SHOW_CONFIRM;
                    if(check_state == 0)
                    {
                        taken_spot++;
                        if(taken_spot == (Parked + 1))
                        {
                            servo_in_state = SERVO_IDLE;
                            servo_out_state = SERVO_IDLE;
                            taken_spot--;
                        }
                        else
                        {
                            servo_in_state = SERVO_IN_ON;
                        }
                    }
                    else if(check_state == 1)
                    {
                        servo_out_state = SERVO_OUT_ON;
                        if (taken_spot == 0)
                        {
                            LCD_I2C_write_string("ERROR...",2,1);
                            while(1)
                            {
                                }
                            }
                            taken_spot--;
                        }
                    }
                }
            }
        }
    }
}

```

```

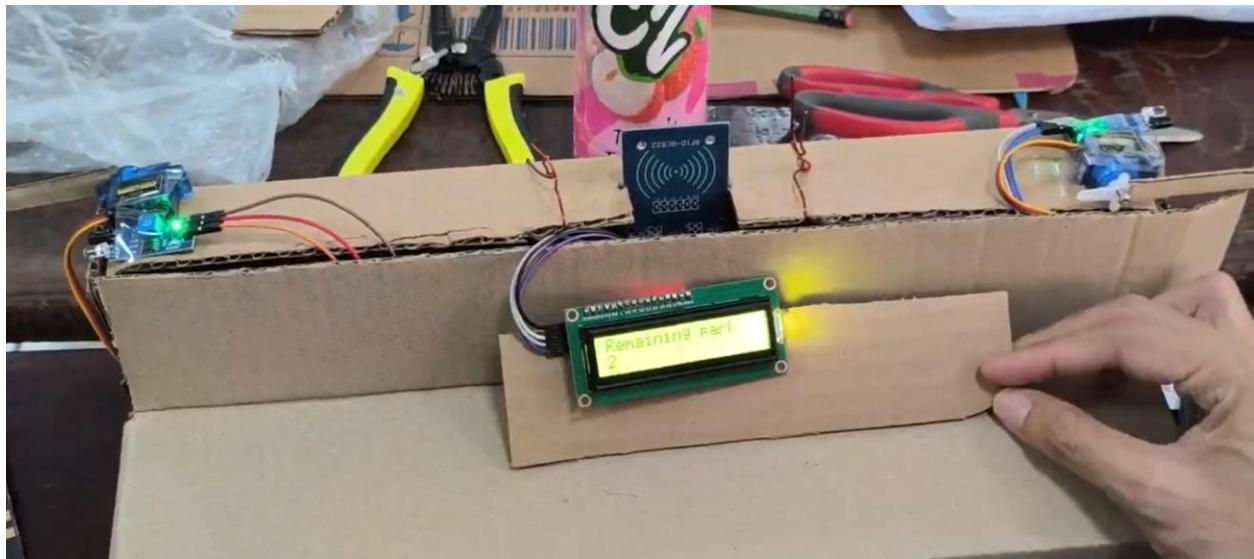
else
{
    lcd_state = LCD_UNIDENTIFIED_CARD;
}
}
}
}
}
}
}
}
}
}

```

5. Sản phẩm cuối cùng

5.1 Mô hình bãi xe thông minh

-Mô hình bãi xe thông minh:



5.2 kiểm tra và đánh giá hệ thống vận hành

-Kiểm tra khối nguồn: Nguồn 5v cho khối MCU, LCD, SERVO, IC, nút bấm, khi nối nguồn đầu vào 9v qua bộ hạ áp còn 5v thì những khối trên hoạt động đúng với yêu cầu. và kiểm tra nguồn 3,3v cho khối RFID và IR, nguồn đầu vào 9v qua hạ áp còn 3,3v thì khối RFID và cảm biến hồng ngoại IR hoạt động bình thường.

-Kiểm tra khối VXL: Khối VXL đọc input từ những ngoại vi: nút bấm, cảm biến hồng ngoại IR, RFID sau đó xử lý giữa 2 chế độ đọc/lưu thẻ và chế độ quét thẻ, đưa thông tin về số chỗ trống trong bãi xe, thẻ có được lưu hay chưa, thẻ này có hợp lệ để quét hay không, đưa tín hiệu để SERVO quay mở cổng và đóng cổng.

- Kiểm tra khối LCD: LCD hiện rõ những thông tin như mã UID đọc từ RFID, số chỗ trống trong bãi xe.

-Kiểm tra khối động cơ SERVO: Nếu thẻ quét hợp lệ và còn chỗ trống thì động cơ SERVO sau khi nhận tín hiệu từ khối VXL thì quay 90 độ, mở cổng cho xe đi vào, sau đó tiếp tục nhận tín hiệu từ khối VXL để quay 90 độ đóng cổng, tương tự với quét ra. Nếu thẻ quét không hợp lệ hoặc bãi xe không còn chỗ trống thì SERVO sẽ không quay, chờ đến khi được quét thẻ hợp lệ hoặc bãi xe có chỗ trống thì mới quay để mở cổng.

-Kiểm tra khối RFID: đọc được mã UID từ thẻ để có thể trả tín hiệu về cho VXL xử lý.

KẾT LUẬN

Sau khoản thời gian làm và hoàn thiện bài tập lớn nhóm chúng em đã kết hợp những kiến thức đã được học trên lớp và hoàn thành được một mô hình về bãi xe thông minh, đọc quét thẻ bằng RFID và hiện những thông tin cần kiểm tra cho LCD.

Trong quá trình thực hiện sản phẩm, nhóm chúng em đã học được nhiều điều. Từ quá trình lên kế hoạch, ý tưởng, viết các đặc tả, ràng buộc,... Cho đến việc bắt tay vào thiết kế, thử nghiệm cuối cùng là hoàn thiện hệ thống. thông qua quá trình thực hiện hệ thống, nhóm đã tìm hiểu thêm về vi xử lý ATMEGA 328P và có được cái nhìn tổng quát về cách để có thể hoàn thiện một hệ thống nhúng. Cũng như là học được thêm về cách phân chia công việc, làm việc nhóm, cách để giải quyết những vấn đề trong suốt quá trình thực hiện sản phẩm.

Videos kiểm tra hệ thống hoạt động:

<https://drive.google.com/drive/folders/1X-8VSGQEHZhN04-JYkm2xYVY1bG1itku>