

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC BÀ RỊA-VŨNG TÀU



**BARIA VUNGTAU**  
UNIVERSITY  
CAP SAINT JACQUES

## **ĐỒ ÁN TỐT NGHIỆP**

### **HỆ THỐNG ĐỖ XE THÔNG MINH**

Trình độ đào tạo	: Đại học chính quy
Ngành	: Công nghệ kỹ thuật điện – Điện tử
Chuyên ngành	: Điều khiển và tự động hóa
Giảng viên hướng dẫn	: ThS. Lưu Hoàng
Sinh viên thực hiện	: Ngô Minh Tuấn
MSSV: 19033788	Lớp: DH19TD

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

**PHIẾU GIAO ĐỀ TÀI**  
**ĐỒ ÁN/ KHOÁ LUẬN TỐT NGHIỆP**

*(Đính kèm Quy định về việc tổ chức, quản lý các hình thức tốt nghiệp ĐH, CĐ ban hành kèm theo Quyết định số 585/QĐ-ĐHBRVT ngày 16/7/2013 của Hiệu trưởng Trường Đại học BR-VT)*

**Họ và tên sinh viên:** Ngô Minh Tuấn

**Ngày sinh:** 15/11/2001

**MSSV:** 19033788

**Lớp:** DH19TD

**E-mail:** tuanngm01@gmail.com

**Hệ đào tạo:** Đại học chính quy

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

**1. Tên đề tài:** Hệ thống đỗ xe thông minh

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

**3. Ngày giao đề tài:**

**4. Ngày hoàn thành đồ án/ khoá luận tốt nghiệp:** 30/4/2023

**GIẢNG VIÊN HƯỚNG DẪN**

*(Ký và ghi rõ họ tên)*

ThS. Lưu Hoàng

**TRƯỞNG BỘ MÔN**

*(Ký và ghi rõ họ tên)*

*Bà Rịa-Vũng Tàu, ngày.... tháng.... năm ..*

**SINH VIÊN THỰC HIỆN**

*(Ký và ghi rõ họ tên)*

Ngô Minh Tuấn

**TRƯỞNG KHOA**

*(Ký và ghi rõ họ tên)*

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

- Thái độ, tác phong và nhận thức trong quá trình thực hiện:

.....  
.....  
.....

- Kiến thức chuyên môn:

.....  
.....  
.....

- Hình thức, bố cục trình bày:

.....  
.....  
.....

- Nội dung, kết quả:

.....  
.....  
.....

- Nhận xét khác:

.....  
.....  
.....

Bà Rịa – Vũng Tàu, ngày ... tháng ... năm 2023

**Giảng viên hướng dẫn**  
(Ký và ghi rõ họ tên)

## NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

- Thái độ, tác phong và nhận thức trong quá trình thực hiện:

.....  
.....  
.....

- Kiến thức chuyên môn:

.....  
.....  
.....

- Hình thức, bố cục trình bày:

.....  
.....  
.....

- Nội dung, kết quả:

.....  
.....  
.....

- Nhận xét khác:

.....  
.....  
.....

Bà Rịa – Vũng Tàu, ngày ... tháng ... năm 2023

**Giảng viên phản biện**  
(Ký và ghi rõ họ tên)

**LỜI CẢM ƠN**

Thực tế không có sự thành công nào mà không có sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của mọi người xung quanh. Trong quá trình làm đồ án tốt nghiệp em nhận được sự giúp đỡ nhiệt tình của giảng viên hướng dẫn. Em xin cảm ơn thầy Lưu Hoàng – người đã tận tình hướng dẫn, giúp đỡ cho em trong khi lựa chọn đề tài cũng như trong quá trình thực hiện đề tài này. Khi thực hiện đồ án cũng đã xảy ra nhiều khó khăn, thiếu sót nhưng được sự hỗ trợ và góp ý của Thầy nên bản thân em đã hoàn thành được đồ án.

Trong suốt thời gian được theo học tại trường Đại Học Bà Rịa Vũng Tàu, em đã nhận được nhiều sự quan tâm và giúp đỡ từ thầy cô và bạn bè. Cảm ơn tới hiệu trưởng cùng các quý thầy cô trường Đại Học Bà Rịa – Vũng Tàu đã hỗ trợ tận tình về trang thiết bị, phần mềm, cơ sở vật chất tạo điều kiện hoàn thành đồ án. Với lòng biết ơn sâu sắc, em xin gửi lời cảm ơn tới quý thầy cô, những người đã truyền lại cho em rất nhiều kinh nghiệm và kiến thức quý báu, những sự giúp đỡ ấy đã tiếp thêm động lực cho em vững bước trên con đường mình đã chọn. Và đặc biệt là thầy, cô Khoa KTCN, bộ môn Điện, điện tử đã truyền đạt kiến thức, kinh nghiệm cũng như tạo những điều kiện tốt nhất để nhóm em hoàn thành đề tài.

Xin cảm ơn các bạn cùng khóa, cùng khoa đã động viên, khích lệ, ủng hộ về nhiều mặt góp phần làm nên sự hoàn thiện của đồ án này.

Cảm ơn trường Đại Học Bà Rịa Vũng Tàu! Em xin chân thành cảm ơn!

*Người thực hiện đề tài*

*Ngô Minh Tuấn*

## MỤC LỤC

<b>LỜI CẢM ƠN</b> .....	<b>5</b>
<b>LỜI MỞ ĐẦU</b> .....	<b>8</b>
<b>CHƯƠNG I: TỔNG QUAN</b> .....	<b>9</b>
<b>CHƯƠNG II: CƠ SỞ LÝ THUYẾT</b> .....	<b>13</b>
<b>2.1. Giới thiệu về phân cứng</b> .....	<b>13</b>
<b>2.2. Phần mềm lập trình</b> .....	<b>20</b>
<b>CHƯƠNG III: NỘI DUNG ĐỀ TÀI</b> .....	<b>26</b>
<b>3.1. GIỚI THIỆU</b> .....	<b>26</b>
<b>3.2. TÍNH TOÁN VÀ THIẾT KẾ</b> .....	<b>26</b>
<b>3.2.1. Sơ đồ khối</b> .....	<b>26</b>
<b>3.2.2. Sơ đồ mạch</b> .....	<b>27</b>
<b>3.2.3. Sơ đồ di dây</b> .....	<b>28</b>
<b>3.2.4. Mô hình hoàn thiện</b> .....	<b>29</b>
<b>3.2.5. Thực nghiệm</b> .....	<b>30</b>
<b>CHƯƠNG IV: KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN</b> .....	<b>36</b>
<b>4.1. KẾT LUẬN</b> .....	<b>36</b>
<b>4.2. HƯỚNG PHÁT TRIỂN</b> .....	<b>37</b>
<b>TÀI LIỆU THAM KHẢO</b> .....	<b>38</b>

## MỤC LỤC HÌNH ẢNH

HÌNH 1. TÌNH TRẠNG TẮC NGHỀN GIAO THÔNG.....	9
HÌNH 2.1. SƠ ĐỒ CHÂN ARDUINO UNO R3 .....	13
HÌNH 2.2. MÀN HÌNH LCD 20x4 .....	15
HÌNH 2.3. MẠCH GIAO TIẾP LCD I2C.....	16
HÌNH 2.4. RFID.....	17
HÌNH 2.5. ĐỘNG CƠ SERVO .....	18
HÌNH 2.6. CẢM BIẾN VẬT CẢN HỒNG NGOẠI.....	19
HÌNH 2.7. CÁC PHIÊN BẢN JAVA RUNTIME ENVIRONMENT .....	20
HÌNH 2.8. CÁC PHIÊN BẢN ARDUINO .....	21
HÌNH 2.9. GIAO DIỆN KHI MỞ ARDUINO.....	22
HÌNH 2.10. CHỨC NĂNG CÁC NÚT LỆNH TRONG GIAO DIỆN ARDUINO .....	23
HÌNH 2.11. CÁC BƯỚC CÀI ĐẶT DRIVER.....	25
HÌNH 3.1. SƠ ĐỒ KHỐI.....	26
HÌNH 3.2. SƠ ĐỒ MẠCH .....	27
HÌNH 15. SƠ ĐỒ ĐI DÂY .....	28
HÌNH 16. MÔ HÌNH HOÀN THIỆN.....	29
HÌNH 17. KHI HỆ THỐNG KHÔNG HOẠT ĐỘNG .....	30
HÌNH 18. MÀN HÌNH LCD KHI KHỞI ĐỘNG HỆ THỐNG .....	31
HÌNH 19. KHI HỆ THỐNG VỪA KHỞI ĐỘNG LẠI.....	32
HÌNH 20. XE ĐẾN VỊ TRÍ CẢM BIẾN XE VÀO .....	32
HÌNH 21. KHI DỪNG SAI THỂ ĐỂ VÀO BÃI ĐỖ XE.....	33
HÌNH 22. TRẠNG THÁI KHI XE 1 ĐI VÀO HỆ THỐNG ĐỖ XE.....	33
HÌNH 23. TRẠNG THÁI KHI XE 1 SAU KHI VÀO VỊ TRÍ ĐỖ XE S1 .....	34
HÌNH 27. TÌNH TRẠNG BÃI ĐỖ XE ĐẦY.....	34
HÌNH 28. TÌNH TRẠNG KHI CÓ XE MUỐN VÀO KHI BÃI ĐỖ XE ĐẦY .....	34
HÌNH 29. TRẠNG THÁI XE 1 DỪNG Ở CỬA RA.....	35
HÌNH 30. TRẠNG THÁI KHI XE 1 RỜI ĐI .....	35
HÌNH 34. MÀN HÌNH HIỂN THỊ TRẠNG THÁI KHI TOÀN BỘ XE RỜI ĐI.....	35

**LỜI MỞ ĐẦU**

Hiện nay, đất nước ta đang trong thời kì công nghiệp hoá hiện đại hoá nhằm đưa đất nước tiến kịp với nền kinh tế của các nước trong khu vực và thế giới. khoa học công nghệ ngày càng đạt được những thành tựu to lớn, kéo theo đó là sự phát triển vượt bậc trong các ngành nghề có ứng dụng khoa học kỹ thuật. Đối với một nước mà nền nông nghiệp còn chiếm vai trò to lớn trong nền kinh tế thì việc ứng dụng khoa học công nghệ là điều cấp thiết và cần được mở rộng. Nhằm giải quyết vấn đề này, nhờ sự giúp sức của tiên bộ về khoa học kỹ thuật, các hệ thống giám sát, xử lý, cung ứng quá trình sản xuất... ngày càng hiện đại đã được đưa vào nông nghiệp và đặc biệt là ứng dụng các thiết bị tự động hóa như cảm biến đã góp phần tạo nên một môi trường sản xuất năng động, khoa học và giải phóng sức lao động, tăng năng suất, mang lại hiệu quả kinh tế cao.

Sau quá trình học tại trường, em đã được các thầy/cô chỉ bảo tận tình, với những kiến thức bên ngoài đã giúp ích chúng em hiểu thêm nhiệm vụ và nhu cầu thiết yếu của các bộ môn đã vào đời sống thực tiễn. Với kiến thức được chỉ dạy, bản thân em đã nghiên cứu đề tài đó là: “Hệ thống đỗ xe thông minh”.



## CHƯƠNG I: TỔNG QUAN

### 1. Đặt vấn đề

Trong những năm gần đây cùng với sự phát triển của nền kinh tế, thu nhập đời sống của người dân được nâng lên cùng với đó là sự gia tăng chóng mặt của số lượng phương tiện đặc biệt ở các thành phố lớn.

Trong khi đó, theo thống kê, hiện nay số lượng hệ thống đỗ xe có giấy phép ở các thành phố chỉ đáp ứng được 8-10% nhu cầu người dân, dẫn tới tình trạng thiếu hệ thống bãi đỗ xe là vô cùng nghiêm trọng.

Giao thông tĩnh ở các thành phố lớn như Hà Nội, TP. Hồ Chí Minh, Đà Nẵng hiện nay đang là một bài toán khó và cần giải quyết ngay. Hàng ngày, hàng giờ chúng ta vẫn thường xuyên được nghe những thông tin về tình trạng tắc đường tại các tuyến phố, đặc biệt trong những giờ cao điểm.



*Hình 1. Tình trạng tắc nghẽn giao thông*

Và một trong những nguyên nhân gây ra tình trạng đó là việc đỗ, dừng xe không đúng nơi quy định. Các điểm đỗ, dừng xe được tận dụng ở mọi chỗ, mọi nơi: trên vỉa hè, lòng đường, công viên các nơi không phép... điều đó ảnh hưởng không nhỏ đến giao thông nội đô và mỹ quan đô thị.

Chính vì vậy việc áp dụng các mô hình hệ thống bãi đỗ xe thông minh công cộng để giải quyết tình trạng khan hiếm chỗ đỗ xe là vô cùng cần thiết đối với các đô thị lớn của nước ta.

## **2. Mục tiêu đề tài**

- Nghiên cứu kết nối thu thập thông tin dữ liệu xe ra vào từ các cảm biến.
- Nghiên cứu màn hình led LCD và phương thức hiển thị thông tin.
- Nghiên cứu dựa trên kiến thức đã học như: Kỹ thuật số, kỹ thuật cảm biến, linh kiện điện tử....
- Tự học thêm các kiến thức về lập trình Arduino Uno R3.
- Xây dựng mô hình hoàn chỉnh để hiển thị số lượng xe đang đỗ và trạng thái của từng vị trí.

## **3. Nội dung nghiên cứu**

- Tìm hiểu công dụng của từng thiết bị điện, điện tử cụ thể là: Màn hình hiển thị LCD, động cơ servo, cảm biến, RFID 522, dây nối và Arduino Uno R3.
- Tìm hiểu phần mềm để lập trình cho Arduino Uno R3: Ngôn ngữ Arduino, Arduino IDE.
- Đưa ra các phương án nghiên cứu.
- Kiểm tra, đánh giá tính ứng dụng của đề tài.

## **4. Đối tượng, phạm vi nghiên cứu và phương pháp nghiên cứu**

- Đối tượng nghiên cứu:
  - + Về phần cứng: Cảm biến IR Infrared Obstacle Avoidance, màn hình LCD 20x4, mạch giao tiếp LCD2004 sang I2C, động cơ servo SG90, RFID 522, dây nối và Arduino Uno R3.
  - + Về phần mềm: Arduino IDE
- Phạm vi nghiên cứu: Thiết kế mạch liên kết các cảm biến thu thập dữ liệu và RFID với Arduino, từ Arduino sẽ điều khiển động cơ servo và hiển thị dữ liệu lên LCD.

- Cách tiếp cận và phương pháp nghiên cứu: Trong quá trình thực hiện, em đã nghiên cứu, tìm hiểu qua sách, báo, các tài liệu trên internet, nghiên cứu cơ sở lý thuyết. Ứng dụng những kiến thức đã học ở trường cộng thêm học ở mạng như: tập lập trình Arduino Uno R3, liên kết input/output.... Từ cơ sở đó có thể xây dựng được mô hình này qua 3 bước chính. Thiết kế mạch điều khiển, tiến hành lắp ráp mạch và hoàn thiện mô hình.

#### **4. Ý nghĩa khoa học và thực tiễn của đề tài**

❖ Ý nghĩa khoa học:

Xây dựng được cơ sở khoa học về khả năng tự hành trong hệ thống bãi đỗ xe.

❖ Thực tiễn của đề tài:

Tiềm năng ứng dụng của hệ thống đỗ xe thông minh là rất lớn, Hệ thống đỗ xe thông minh đang được phủ sóng rộng rãi tại hầu hết các doanh nghiệp, các tòa nhà trên toàn quốc nhờ những tiện ích nổi bật mà nó mang lại.

#### **5. Bố cục**

- Chương I: Tổng quan

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

- Chương II: Cơ sở lý thuyết

Trình bày các lý thuyết liên quan đến vấn đề mà đề tài sẽ dùng để thực hiện thiết kế cho đề tài.

- Chương III: Nội dung đề tài

Trình bày tổng quan các yêu cầu của đề tài về thiết kế. Trình bày kết quả thi công phần cứng và kết quả hình ảnh trên màn hình hay mô phỏng tín hiệu, kết quả thống kê.

- Chương IV: Kết luận và phương hướng phát triển

Trình bày kết quả đạt được so với mục tiêu đề ra ban đầu, nhận xét và đánh giá kết quả đạt được của đề tài nghiên cứu. Hướng phát triển của đề tài sau này trong quá trình nghiên cứu.

- Tài liệu tham khảo.
- Phụ lục.

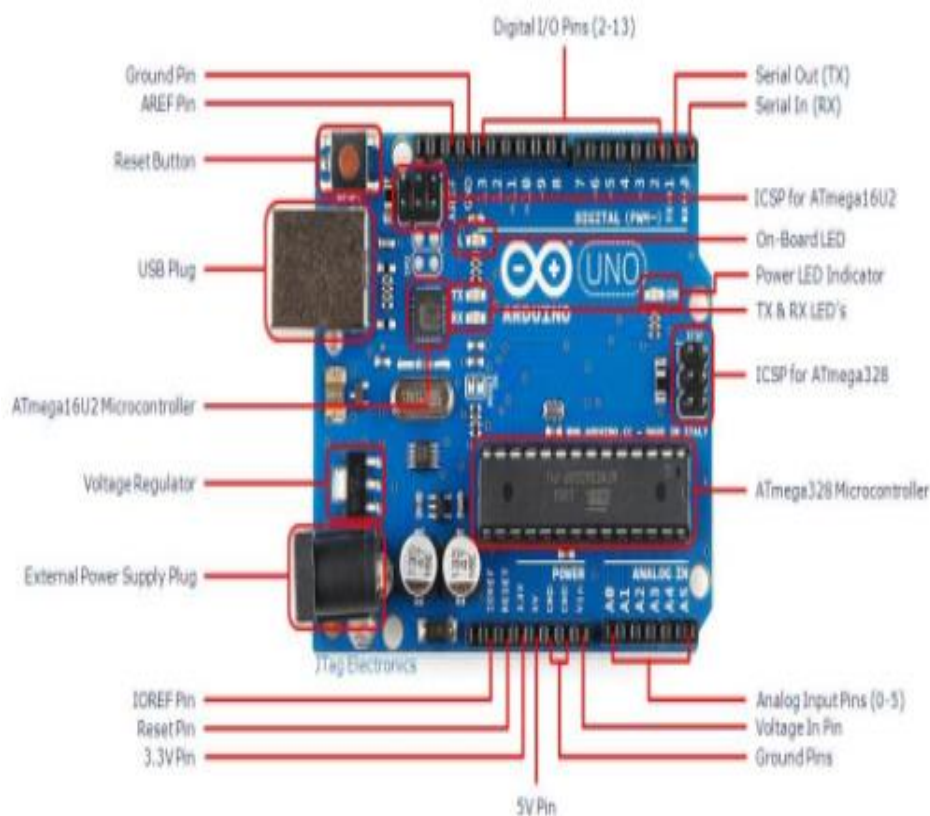
## CHƯƠNG II: CƠ SỞ LÝ THUYẾT

### 1. Giới thiệu về phần cứng

#### 1.1. Arduino Uno R3

Arduino Uno là một bo mạch vi điều khiển dựa trên chip Atmega328P. Uno có 14 chân I/O digital (trong đó có 6 chân xuất xung PWM), 6 chân Input analog, tốc độ xung nhịp 16MHz, 1 cổng USB, 1 jack nguồn DC, 1 nút reset.

Uno hỗ trợ đầy đủ những thứ cần thiết để chúng ta có thể bắt đầu làm việc.



**Hình 2. Sơ đồ chân Arduino Uno R3**

## ❖ Thông số kỹ thuật:

Vi điều khiển	Atmega328P
Điện áp hoạt động	5V
Điện áp cấp (hoạt động tốt)	7 – 12 V
Điện áp cấp (giới hạn)	6 – 12 V
Chân I/O digital	14 ( có 6 chân xuất xung PWM)
Chân Input analog	6 (A0 – A5)
Dòng điện mỗi chân I/O	20 mA
Dòng điện chân 3.3V	50 mA
Bộ nhớ Flash	32 kB (Atmega328P) – trong đó 0.5 kB dùng cho bootloader.
SRAM	2 kB (Atmega328P)
EEPROM	1 kB (Atmega328P)
Tốc độ xung nhịp	16 MHz
Kích thước	68.6 x 53.4 mm
Trọng lượng	25 g

**❖ Chức năng của các chân trên board Arduino Uno R3:**

Chân 0 (RX): Nhận dữ liệu từ kết nối ngoài vào UART.

Chân 1 (TX): Gửi dữ liệu ra ngoài UART.

Chân 2-13: Điều khiển các thiết bị như LED, còi, motor, cảm biến, ...

Chân 3, 5, 6, 9, 10, 11: Có thể sử dụng chế độ PWM để tạo tín hiệu xung với tần số và mức độ độ rộng khác nhau.

Chân A0-A5: Chân vào analog để đọc giá trị tín hiệu analog từ các sensor hoặc thiết bị đo.

Chân Vin: Điện áp đầu vào từ nguồn ngoài.

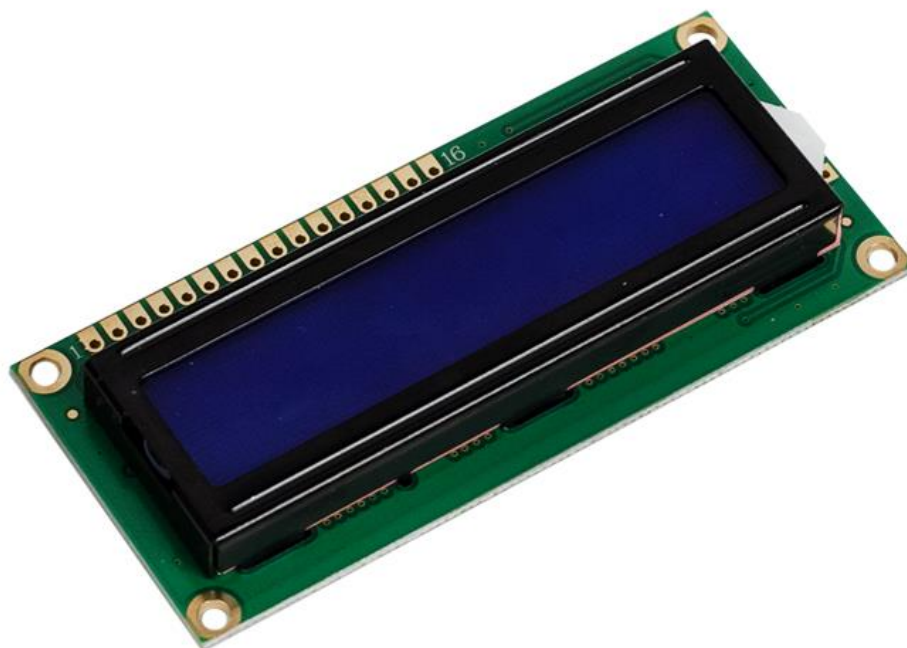
Chân 5V: Điện áp ra 5V.

Chân 3.3V: Điện áp ra 3.3V.

Chân GND: Đất (0V) cho mạch điện tử.

**1.2. Màn hình LCD 20x4**

LCD 20x4 là loại màn hình tinh thể lỏng nhỏ dùng để hiển thị chữ hoặc số trong bảng mã ASCII. Mỗi ô của Text LCD bao gồm các chấm tinh thể lỏng, các chấm này kết hợp với nhau theo trình tự “ẩn” hoặc “hiện” sẽ tạo nên các kí tự cần hiển thị và mỗi ô chỉ hiển thị được một kí tự duy nhất.



***Hình 3. Màn hình LCD 20x4***

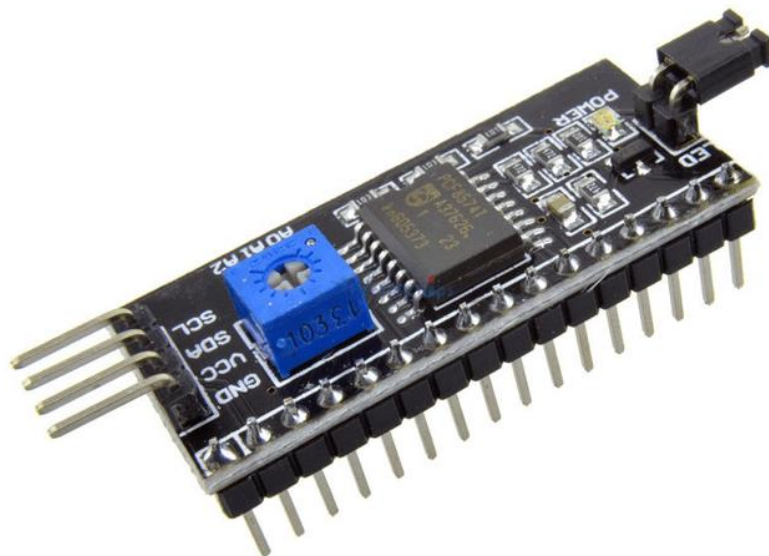
LCD 20x4 là loại LCD có 4 dòng và mỗi dòng chỉ hiển thị được 20 kí tự. Đây là loại màn hình được sử dụng rất phổ biến trong các loại mạch điện.

❖ **Thông số kĩ thuật của LCD 20x4:**

- Điện áp: 5V.
- Ngõ giao tiếp: 16 chân.
- Màu sắc: xanh lá hoặc xanh dương.
- Module hỗ trợ giao tiếp với vi điều khiển: LCD I2C.

### 1.3. Mạch giao tiếp LCD I2C

Mạch chuyển giao tiếp LCD1602, LCD1604, LCD2004 sang I2C sử dụng các loại LCD có driver là HD44780(LCD 1602, LCD 2004, ...), cần có ít nhất 6 chân của MCU kết nối với các chân RS, EN, D7, D6, D5 và D4 để có thể giao tiếp với LCD. Nhưng với module chuyển giao tiếp LCD sang I2C, các bạn chỉ cần 2 chân (SDA và SCL) của MCU kết nối với 2 chân (SDA và SCL) của module là đã có thể hiển thị thông tin lên LCD.



*Hình 4. Mạch giao tiếp LCD I2C*



## 1.4. RFID 552

Mạch RFID RC522 sử dụng IC MFRC522 được sử dụng để đọc và ghi dữ liệu cho thẻ RFID tần số 13.56MHz, mạch có thiết kế nhỏ gọn được sử dụng rất phổ biến hiện nay với Arduino hoặc các loại Vi điều khiển khác trong các ứng dụng cần ghi, đọc thẻ RFID.



*Hình 5. Mạch RFID*

### ❖ Thông số kỹ thuật:

- Nguồn sử dụng: 3.3VDC.
- Dòng điện: 13~26mA.
- Tần số hoạt động: 13.56Mhz.
- Khoảng cách hoạt động: 0~60mm.
- 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.

### 1.5. Động cơ servo SG90

Động cơ RC Servo 9G có kích thước nhỏ, là loại được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng, động cơ RC Servo 9G có tốc độ phản ứng nhanh, các bánh răng được làm bằng nhựa nên cần lưu ý khi nâng tải nặng vì có thể làm hư bánh răng, động cơ RC Servo 9G có tích hợp sẵn Driver điều khiển động cơ bên trong nên có thể dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.



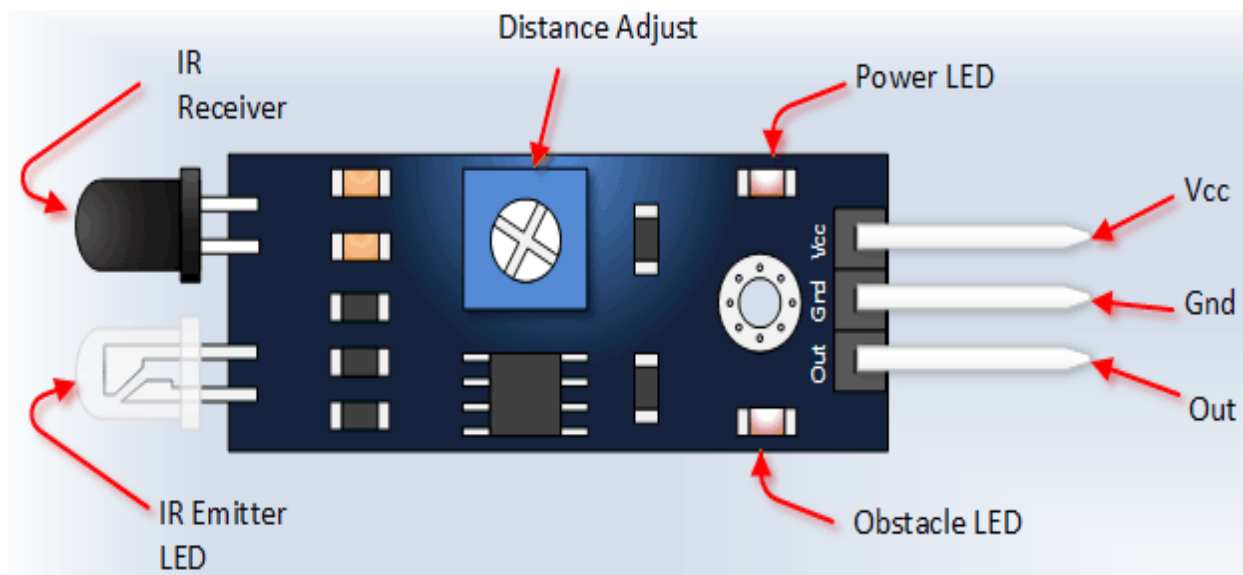
*Hình 6. Động cơ servo*

#### **Thông số kỹ thuật:**

- Điện áp hoạt động: 4.8-5VDC.
- Tốc độ: 0.12 sec/ 60 degrees (4.8VDC).
- Lực kéo: 1.6KG.CM.
- Kích thước: 21x12x22mm.
- Trọng lượng: 9g.

## 1.6. Cảm biến vật cản hồng ngoại IR Infrared Obstacle Avoidance

Cảm biến vật cản hồng ngoại IR Infrared Obstacle Avoidance được sử dụng để nhận biết vật cản bằng ánh sáng hồng ngoại, cảm biến có cách sử dụng đơn giản với biến trở chỉnh khoảng cách nhận biết vật cản, ngõ ra dạng Digital dễ dàng giao tiếp và lập trình với vi điều khiển.



*Hình 7. Cảm biến vật cản hồng ngoại*

### ❖ Thông số kỹ thuật:

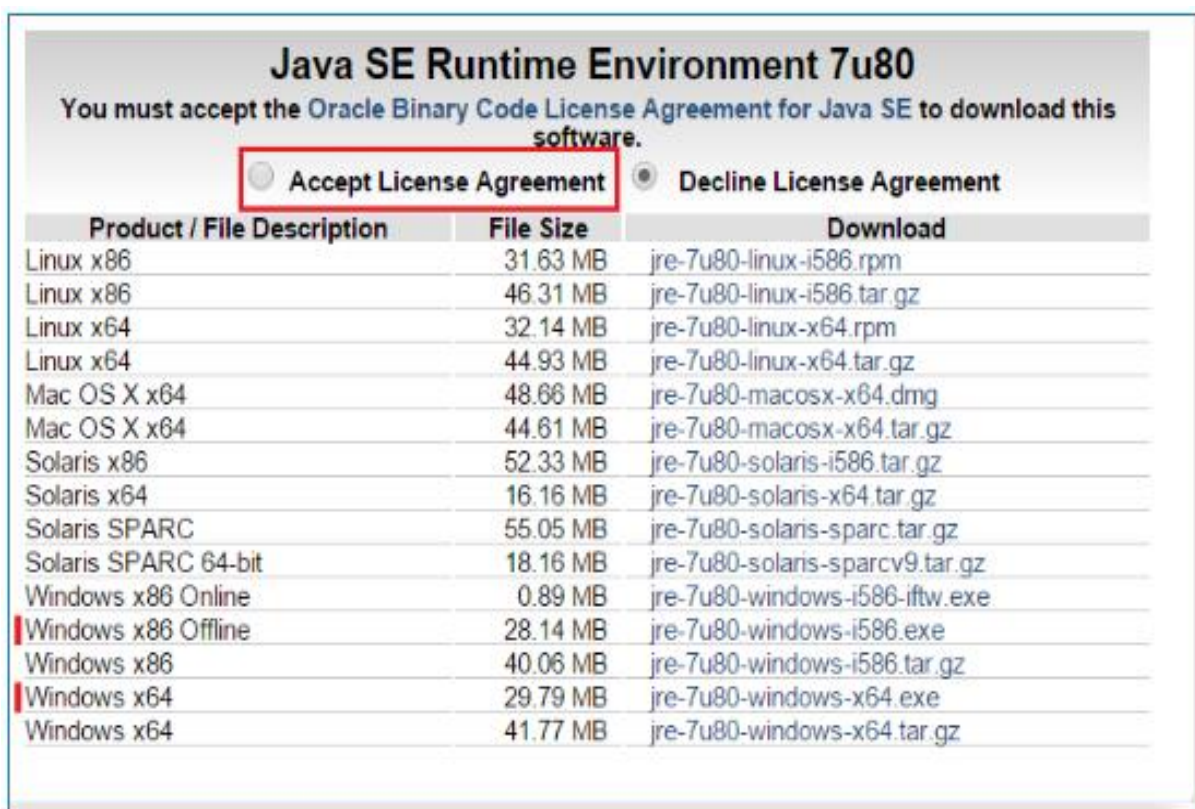
- Điện áp sử dụng: 3.3~5vDC.
- Nhận biết vật cản bằng ánh sáng hồng ngoại.
- Ngõ ra: Digital TTL.
- Tích hợp biến trở chỉnh khoảng cách nhận biết vật cản.
- Kích thước: 3.2 x 1.4cm.

## 2. Phần mềm lập trình

Chương trình được lập trình trên phần mềm Arduino.

Vì Arduino IDE được viết trên Java nên bạn cần phải cài đặt JRE trước Arduino IDE.

### ❖ Cài đặt Java Runtime Environment (JRE)



**Hình 8. Các phiên bản Java Runtime Environment**

Hai bản JRE phổ biến nhất là bản dành cho Windows 32bit (x86) và Windows 64bit (x64). Chọn "Accept License Agreement".

## - Cài đặt Arduino IDE

Bước 1: Truy cập địa chỉ <https://www.arduino.cc/en/software>

## Downloads



### Arduino IDE 2.1.0

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

#### DOWNLOAD OPTIONS

**Windows** Win 10 and newer, 64 bits  
**Windows** MSI installer  
**Windows** ZIP file

**Linux** AppImage 64 bits (X86-64)  
**Linux** ZIP file 64 bits (X86-64)

**macOS** Intel, 10.14: "Mojave" or newer, 64 bits  
**macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

## Legacy IDE (1.8.X)



### Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

#### DOWNLOAD OPTIONS

**Windows** Win 7 and newer  
**Windows** ZIP file

**Windows app** Win 8.1 or 10 

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

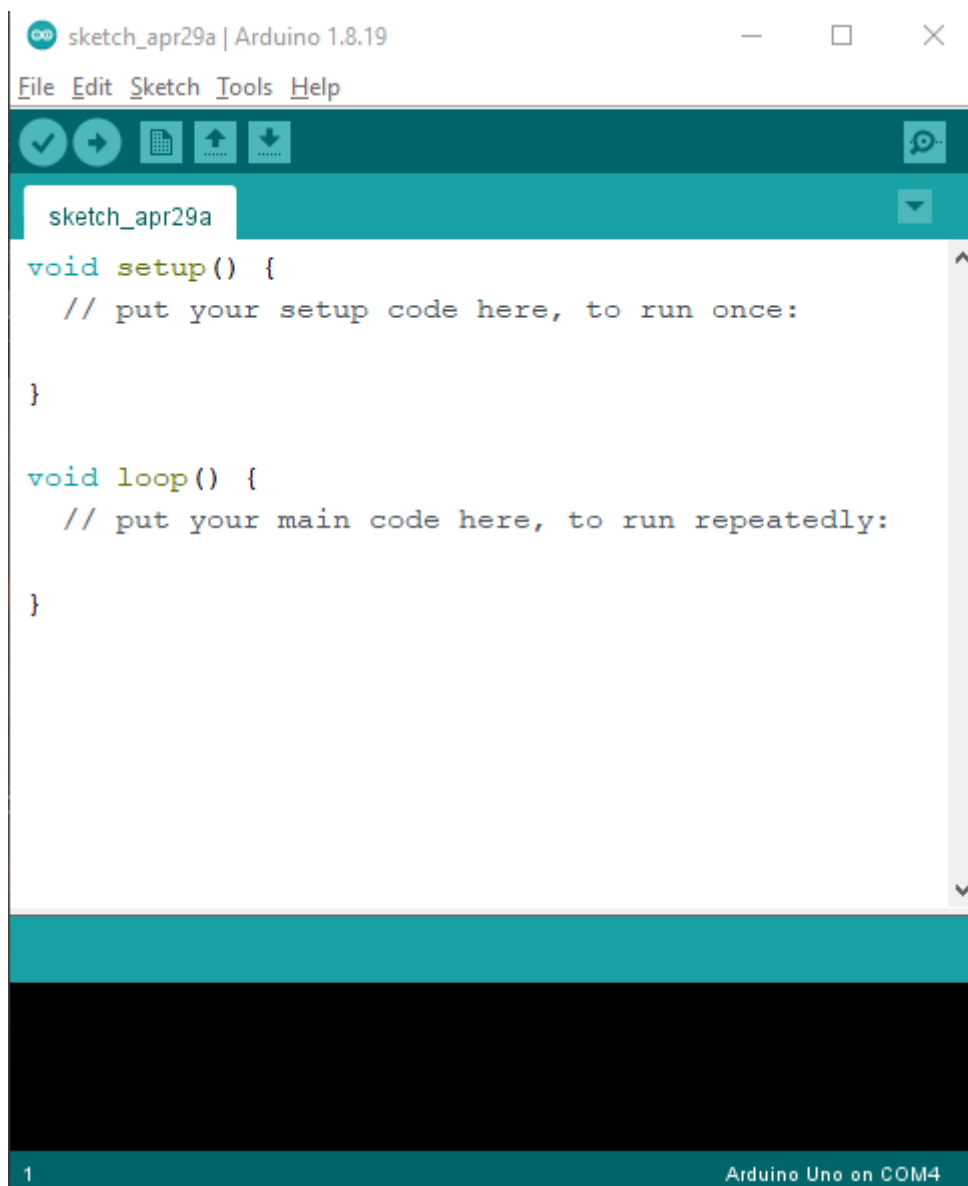
*Hình 9. Các phiên bản Arduino*

Đây là nơi lưu trữ cũng như cập nhật các bản IDE của Arduino. Tải phiên bản phù hợp với hệ điều hành của bản thân.

Bước 2: Sau khi download xong, các bạn bấm chuột phải vào file vừa download và chọn “Extract here” để giải nén.







Bước 3: Copy thư mục vừa giải nén đến nơi lưu trữ.

Bước 4: Chạy file trong thư mục để khởi động Arduino IDE.



*Hình 10. Giao diện khi mở Arduino*

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

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

**Hình 11. Chức năng các nút lệnh trong giao diện Arduino**

### **Cài đặt Driver**

Để máy tính của bạn và board Arduino giao tiếp được với nhau, chúng ta cần phải cài đặt driver trước tiên. Nếu bạn dùng Windows 8, trong một số trường hợp Windows không cho phép bạn cài Arduino driver (do driver không được kí bằng chữ kí số hợp lệ). Do vậy bạn cần vào Windows ở chế độ Disable driver signature enforcement thì mới cài được driver.

Bước 1: Chạy file arduino-1.8.12\drivers\dpinst-x86.exe (Windows x86) hoặc arduino-1.8.12\drivers\dpinst-amd64.exe (Windows x64).

Bước 2: Khi cửa sổ “Device Driver Installation Wizard” hiện ra, các bạn chọn Next để tiếp tục.

Bước 3: Khi có yêu cầu xác nhận cài đặt driver, chọn “Install”.

Bước 4: Đợi khoảng 10 giây trong lúc quá trình cài đặt diễn ra ...

Bước 5: Quá trình cài đặt đã hoàn tất. Bấm “Finish” để thoát.







*Hình 12. Các bước cài đặt Driver*

## CHƯƠNG III: NỘI DUNG ĐỀ TÀI

### 1. GIỚI THIỆU

Dựa trên tài liệu tham khảo và giáo trình, giảng viên hướng dẫn đã tạo cơ hội cho bản thân em nghiên cứu về đề tài. Sau một khoảng thời gian tìm hiểu, mặc dù kiến thức của bản thân còn nhiều hạn chế, em đã cố gắng tìm hiểu nhiều nhất trong khả năng của mình, cùng với đó em đã ứng dụng các kiến thức để làm ra mô hình thực tế để giúp mọi người thấy được thành quả của em. Mô hình thiết kế phải đáp ứng được các yêu cầu:

- Hiện thị được trạng thái của các xe.
- Mô hình đạt được sự ổn định và tính chính xác cao.
- Phù hợp với điều kiện kinh tế.

### 2. TÍNH TOÁN VÀ THIẾT KẾ

#### 2.1. Sơ đồ khối



*Hình 13. Sơ đồ khối*

**Khối điều khiển trung tâm:** khối điều khiển trung tâm sử dụng Arduino

Uno R3 dùng để điều khiển các thiết bị khác trong hệ thống:

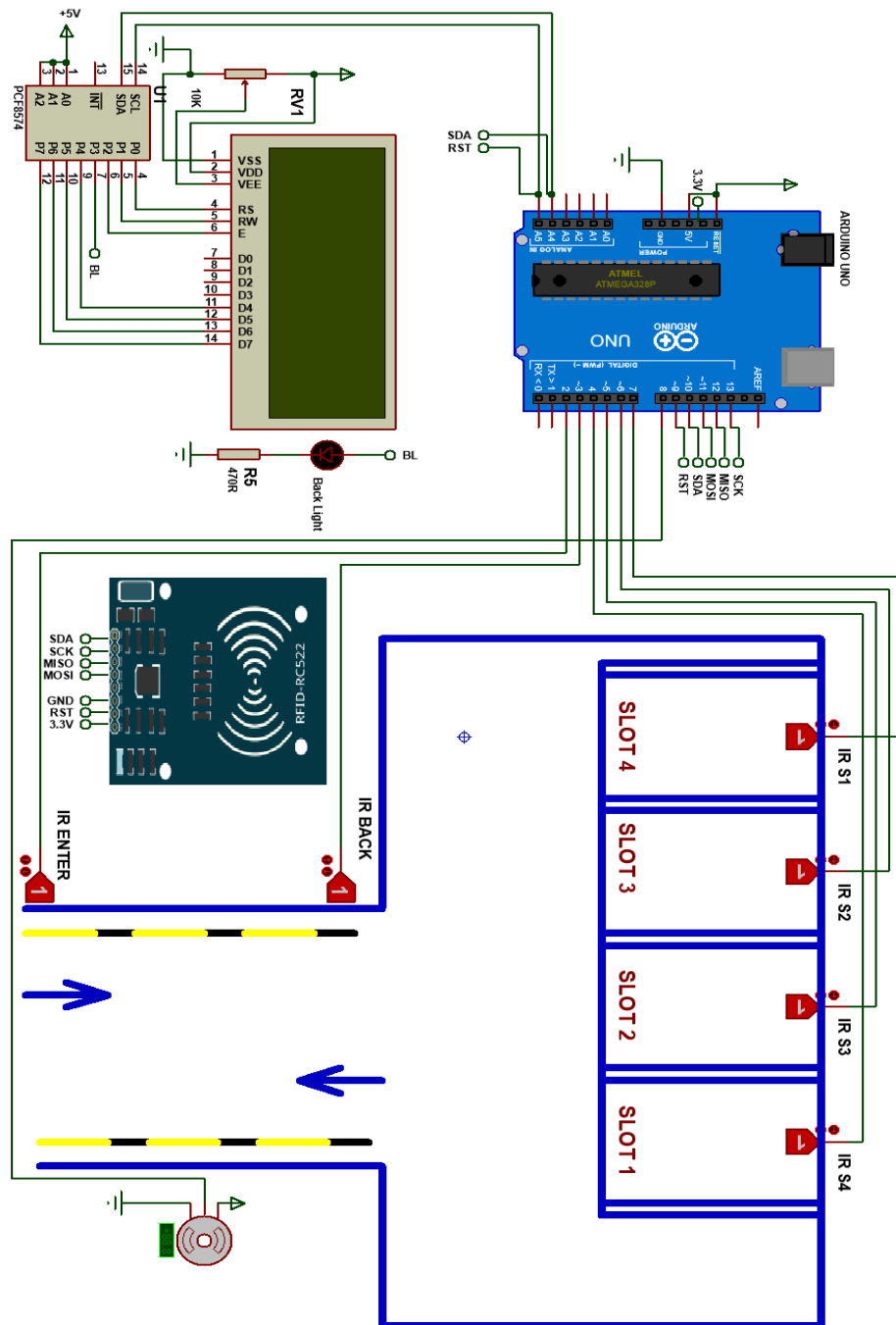
- Điều khiển đọc dữ liệu từ cảm biến.
- Truyền dữ liệu qua màn hình hiển thị của thiết bị.
- Điều khiển động cơ servo

**Khối cảm biến:** bao gồm các cảm biến vật cản hồng ngoại dùng để thu thập dữ liệu về trạng thái xe ở từng vị trí đỗ xe, từ đó đưa tín hiệu về khối điều khiển trung tâm.

**Khởi hiển thị:** Khởi hiển thị sử dụng màn hình Led LCD 20x4 dùng để hiển thị số liệu đọc được từ khối cảm biến và các thông tin đã được xử lý của khối điều khiển trung tâm.

**Khởi nguồn:** Khởi nguồn là khối quan trọng giúp cung cấp điện cho toàn bộ hệ thống. Vì vậy cần tính toán hợp lý để khởi nguồn có thể cung cấp đủ dòng và áp để mạch có thể hoạt động tốt và ổn định.

### 3. Sơ đồ mạch



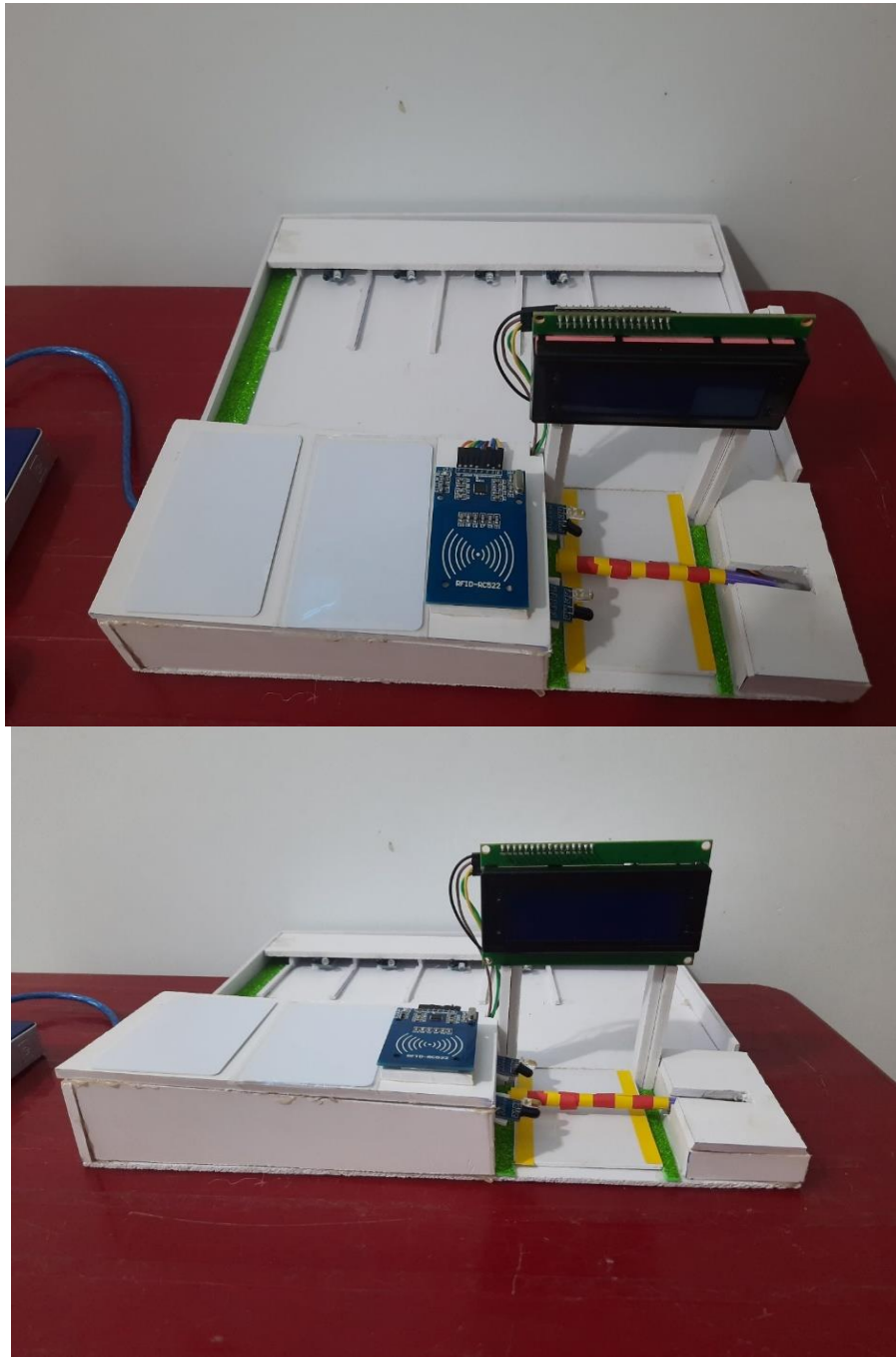
Hình 14. Sơ đồ mạch

## 3.1. Sơ đồ đi dây

Chân	Chức năng	Kết nối đến
0	RX	
1	TX	
2	Digital I/O	Sensor 1
3	Digital I/O	Sensor 2
4	Digital I/O	Sensor 3
5	Digital I/O	Sensor 4
6	Digital I/O	Sensor 5
7	Digital I/O	Sensor 6
8	Digital I/O	Motor
9	Digital I/O	RFID_RST
10	Digital I/O	RFID_SDA
11	Digital I/O	RFID_MOSI
12	Digital I/O	RFID_MISO
13	Digital I/O	RFID_SCK
A0	Analog input	
A1	Analog input	
A2	Analog input	
A3	Analog input	
A4	I2C SDA	LCD, RFID
A5	I2C SCL	LCD, RFID
Vin	Power input	
GND	Ground	
5V	Power output	

Hình 15. Sơ đồ đi dây

3.2. Mô hình hoàn thiện



*Hình 16. Mô hình hoàn thiện*

### 3.3. Thực nghiệm

- ❖ Khi hệ thống đang không hoạt động hoặc đang hoạt động mà bị mất nguồn điện:



Hình

17. Khi hệ thống không hoạt động

❖ Khi khởi động hệ thống, màn hình lcd hiển thị:



*Hình 18. Màn hình LCD khi khởi động hệ thống*

- ❖ Khi hệ thống khởi động lại, căn cứ vào tín hiệu các trạng thái vị trí đỗ xe của các cảm biến S1, S2, S3, S4 để tính toán số vị trí đỗ xe đang còn trống.



***Hình 19. Khi hệ thống vừa khởi động lại***

- ❖ Quá trình khi xe vào bãi đỗ xe:
  - Khi xe đến vị trí cảm biến xe vào:



***Hình 20. Xe đến vị trí cảm biến xe vào***

- Quẹt thẻ ID để cửa mở:



+ Khi quét sai thẻ ID: Khi sử dụng sai thẻ: Màn hình sẽ hiển thị “Wrong ID” và không nâng thanh chắn.



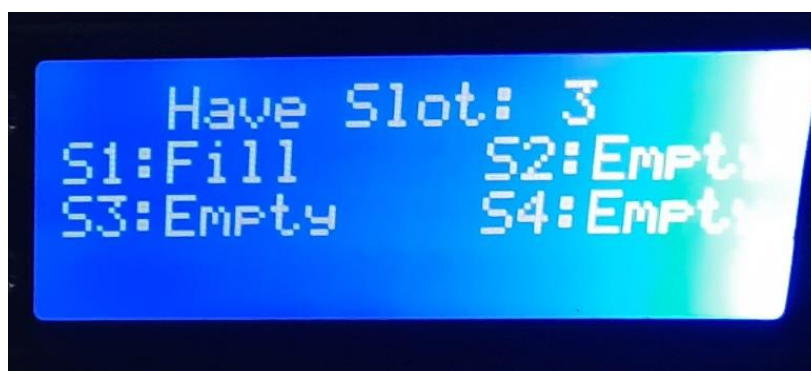
**Hình 21. Khi dùng sai thẻ để vào bãi đỗ xe**

- Khi quét đúng thẻ ID: Thanh chắn nâng lên cho xe vào và hiện lên dòng chữ “Welcome”.



**Hình 22. Trạng thái khi xe 1 đi vào hệ thống đỗ xe**

- Khi xe vào vị trí đỗ thì cảm biến vị trí chuyển trạng thái của S1 từ Empty thành Fill và số vị trí trống giảm 1.



**Hình 23. Trạng thái khi xe 1 sau khi vào vị trí đỗ xe S1**

Khi bãi đỗ xe đầy và nếu có xe 5 muốn vào đỗ thì màn hình sẽ hiển thị “Sorry, parking full”:



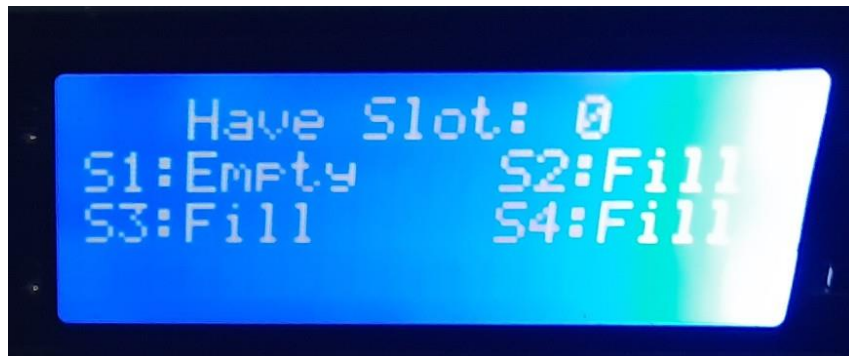
**Hình 24. Tình trạng bãi đỗ xe đầy**



**Hình 25. Tình trạng khi có xe muốn vào khi bãi đỗ xe đầy**

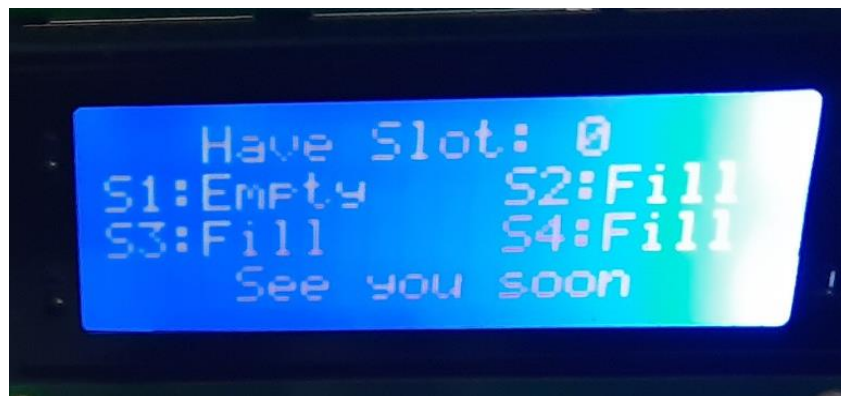
❖ Quá trình xe ra ngoài:

- Khi xe 1 đang dừng ở cảm biến đi ra, trạng thái S1 chuyển từ Fill sang Empty.



*Hình 26. Trạng thái xe 1 dừng ở cửa ra*

- Sau khi quét thẻ: Thanh chắn nâng lên, màn hình LCD hiện lên “See you soon”. Sau khi thanh chắn hạ xuống, số vị trí trống được cộng thêm 1.



*Hình 27. Trạng thái khi xe 1 rời đi*

- Sau khi tất cả xe rời đi:



*Hình 28. Màn hình hiển thị trạng thái khi toàn bộ xe rời đi*

## CHƯƠNG IV: KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

### 1. KẾT LUẬN

Sau khi nghiên cứu và hoàn thành đề tài, em nhận thấy mô hình đã hiệu quả được 80%. Trong thời gian nghiên cứu và thực hiện đề tài, bản thân em đã học hỏi và tìm hiểu thêm được nhiều kiến thức mới cũng như củng cố lại kiến thức đã học giúp hoàn thành đề tài này.

Vì đây là đề tài hướng đến việc giúp cho bãi đỗ xe hoạt động ra vào một cách tự hành nên phải chú trọng độ ổn định và chính xác dẫn đến nhiều khó khăn trong quá trình lập trình. Đồng thời nhờ sự hướng dẫn của giảng viên hướng dẫn và các tài liệu tham khảo thì em đã giải quyết được tương đối yêu cầu của đề tài.

Trong quá trình thực hiện đề tài về phần cứng, em đã tìm hiểu được chức năng của các chân IO của board Arduino, kết hợp với các cảm biến để đọc giá trị, sử dụng chân TX-RX để giao tiếp, ...Việc kết hợp các module này lại với nhau để làm việc ổn định mất nhiều thời gian nhưng cũng học hỏi được rất nhiều trong quá trình làm.

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

## **2. HƯỚNG PHÁT TRIỂN**

Mặc dù đề tài cơ bản đáp ứng được những yêu cầu cơ bản đặt ra tuy nhiên để sản phẩm hoàn thiện được hơn nữa thì đòi hỏi cần được cải tiến và nghiên cứu thêm.

Và dưới đây là một số hướng phát triển của đề tài này:

- Sử dụng các công nghệ mới: Để tăng tính hiệu quả và tiện ích của bãi đỗ xe thông minh, các công nghệ mới như IoT, AI, Machine Learning, Computer Vision... đang được ứng dụng để phát triển các hệ thống bãi đỗ xe thông minh. Các giải pháp này có thể cung cấp thông tin về việc đỗ xe, trống hay đầy, địa chỉ xe đã đỗ, thời gian đỗ xe và hỗ trợ cho việc định vị xe, tìm kiếm chỗ đỗ xe trống, tối ưu hóa lộ trình và phân bổ nguồn lực.
- Xây dựng hệ thống giám sát thông minh: Một hướng phát triển khác là xây dựng hệ thống giám sát thông minh để đảm bảo an toàn và tính bảo mật của hệ thống bãi đỗ xe. Các hệ thống giám sát này có thể sử dụng các cảm biến và camera để phát hiện sự cố an ninh, giám sát lưu lượng xe và cung cấp dữ liệu cho các bên liên quan.

**TÀI LIỆU THAM KHẢO**

- [1] **ANSAR M (2021)**. Github.com/embeddedlab786.
- [2] **HSHOP (2023)**. hshop.vn/products/mach-rfid-rc522-nfc-13-56mhz.
- [3] **HSHOP (2022)**. hshop.vn/products/arduino-uno-r3.
- [3] **HSHOP (2023)**. hshop.vn/products/dong-co-rc-servo-9g.
- [3] **HSHOP (2023)**. hshop.vn/products/lcd-text-lcd2004-xanh-duong.
- [3] **HSHOP (2023)**. hshop.vn/products/mach-chuyen-giao-tiep-lcd1602-lcd1604-lcd2004-sang-i2c.
- [4] **NGUYỄN QUỐC BẢO (2014)**. arduino.vn/bai-viet/68-cai-dat-driver-va-arduino-ide.
- [5] **Nshop (2023)**. nshopvn.com/product/cam-bien-vat-can-hong-ngoai
- [6] **NGUYỄN THỊ PHƯƠNG HÀ, HUỲNH THÁI HOÀNG (2005)**. LÝ THUYẾT ĐIỀU KHIỂN TỰ ĐỘNG. NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH.
- [7] **ThS. LƯU HOÀNG (2022)**. TÀI LIỆU KỸ THUẬT SỐ.
- [8] **ThS. NGUYỄN TẤN PHƯỚC (2007)**. CẢM BIẾN ĐO LƯỜNG VÀ ĐIỀU KHIỂN. NHÀ XUẤT BẢN HỒNG ĐỨC.

## Phụ lục

## ❖ Code hệ thống:

```
// Khai báo thư viện SPI.
#include <SPI.h>

// Khai báo thư viện Servo.
#include <Servo.h>

// Khai báo thư viện Wire.
#include <Wire.h>

// Khai báo thư viện LiquidCrystal_I2C.
#include <LiquidCrystal_I2C.h>

// Khai báo thư viện MFRC522.
#include <MFRC522.h>

// Khởi tạo đối tượng LiquidCrystal_I2C với địa chỉ I2C là 0x27, số cột là 20 và
số hàng là 4.
LiquidCrystal_I2C lcd(0x27, 20, 4);

// Khởi tạo đối tượng Servo.
Servo myservo;

// Định nghĩa hằng số ir_enter là 2.
#define ir_enter 2

// Định nghĩa hằng số ir_back là 3.
#define ir_back 3

// Định nghĩa hằng số ir_car1 là 4.
#define ir_car1 4

// Định nghĩa hằng số ir_car2 là 5.
#define ir_car2 5
```

```
// Định nghĩa hằng số ir_car3 là 6.
#define ir_car3 6
// Định nghĩa hằng số ir_car4 là 7.
#define ir_car4 7
// Định nghĩa hằng số RFID_RST_PIN là 9.
#define RFID_RST_PIN 9
// Định nghĩa hằng số RFID_SS_PIN là 10.
#define RFID_SS_PIN 10
    // Khởi tạo đối tượng MFRC522 với chân SS_PIN là RFID_SS_PIN và
    // chân RST_PIN là RFID_RST_PIN.
MFRC522 rfid(RFID_SS_PIN, RFID_RST_PIN);
// Khởi tạo mảng validIDs chứa 4 ID của thẻ RFID hợp lệ.
String validIDs[4] = {"ADAE5F83", "13CB3417", "239B3E17", "4390B819"};
// Khởi tạo biến S1, S2, S3, S4 với giá trị ban đầu là 0.
int S1=0, S2=0, S3=0, S4=0;
// Khởi tạo biến flag1, flag2 với giá trị ban đầu là 0.
int flag1=0, flag2=0;
// Khởi tạo biến slot với giá trị ban đầu là 4.
int slot = 4;
void Read_Sensor(){
    // Khởi tạo biến S1, S2, S3, S4 và gán giá trị ban đầu bằng 0.
    S1=0, S2=0, S3=0, S4=0;
    // Kiểm tra trạng thái của cảm biến ir_car1, nếu đang nhận được tín hiệu thì gán
    S1 = 1.
    if(digitalRead(ir_car1) == 0){S1=1;}
    // Kiểm tra trạng thái của cảm biến ir_car2, nếu đang nhận được tín hiệu thì gán
    S2 = 1.
    if(digitalRead(ir_car2) == 0){S2=1;}
```



```
// Kiểm tra trạng thái của cảm biến ir_car3, nếu đang nhận được tín hiệu thì gán
S3 = 1
if(digitalRead(ir_car3) == 0){S3=1;}
// Kiểm tra trạng thái của cảm biến ir_car4, nếu đang nhận được tín hiệu thì gán
S4 = 1.
if(digitalRead(ir_car4) == 0){S4=1;}
void setup(){
//Khởi tạo giao tiếp serial với tốc độ truyền 9600 bit/giây.
Serial.begin(9600);
//Khởi tạo giao tiếp SPI.
SPI.begin();
//Khởi tạo module đọc thẻ RFID.
rfid.PCD_Init();
//Thiết lập chân ir_car1 là input.
pinMode(ir_car1, INPUT);
//Thiết lập chân ir_car2 là input.
pinMode(ir_car2, INPUT);
//Thiết lập chân ir_car3 là input.
pinMode(ir_car3, INPUT);
//Thiết lập chân ir_car4 là input.
pinMode(ir_car4, INPUT);
//Thiết lập chân ir_enter là input.
pinMode(ir_enter, INPUT);
//Thiết lập chân ir_back là input.
pinMode(ir_back, INPUT);
//Gán chân 8 cho Servo.
myservo.attach(8);
//Đặt vị trí ban đầu cho Servo là 90 độ.
myservo.write(90);
// Khởi tạo LCD.
```

```

lcd.init();
// bật đèn nền LCD
lcd.backlight();
//Di chuyển con trỏ đến hàng 1, cột 0.
lcd.setCursor (0,1);
//In chuỗi " Car Parking System " trên LCD.
lcd.print(" Car Parking System ");
//Di chuyển con trỏ đến hàng 2, cột 0.
lcd.setCursor (0,2);
// In chuỗi " Welcome " trên LCD.
lcd.print("    Welcome    ");
//Dừng chương trình trong 2 giây.
delay (2000);
//Xóa màn hình LCD.
lcd.clear();
//Đọc trạng thái các cảm biến và lưu vào biến tương ứng.
Read_Sensor();
//Tổng số xe hiện tại trong bãi đỗ xe.
int total = S1+S2+S3+S4;
// Số chỗ trống còn lại trong bãi đỗ xe. slot = slot-total;
void loop(){
    //Gọi hàm Read_Sensor() để đọc trạng thái của các cảm biến và cập nhật giá trị
    của biến S1, S2, S3, S4 tương ứng.
    Read_Sensor();
    //Đặt con trỏ của LCD đến hàng 0, cột 0.
    lcd.setCursor (0,0);
    //: In ra LCD chuỗi ký tự " Have Slot: " để hiển thị thông tin có bao nhiêu chỗ
    đỗ xe còn trống.
    lcd.print("  Have Slot: ");

```

// In ra LCD số lượng chỗ đỗ xe trống được lưu trong biến slot.

```
lcd.print(slot);
```

// In ra LCD chuỗi ký tự rỗng để tạo khoảng cách giữa thông tin số chỗ đỗ xe trống và trạng thái của các cảm biến.

```
lcd.print("  ");
```

//Đặt con trỏ của LCD đến hàng 1, cột 0.

```
lcd.setCursor (0,1);
```

//Kiểm tra giá trị biến S1 để xác định trạng thái của cảm biến đỗ xe S1, nếu bằng 1 thì hiển thị chuỗi "S1:Fill " để thể hiện cảm biến đã đầy, ngược lại thì hiển thị chuỗi "S1:Empty".

```
if(S1==1){lcd.print("S1:Fill ");}
```

```
else{lcd.print("S1:Empty");}
```

//Đặt con trỏ của LCD đến hàng 1, cột 12 để hiển thị thông tin của cảm biến S2.

```
lcd.setCursor (12,1);
```

//Kiểm tra giá trị biến S2 để xác định trạng thái của cảm biến đỗ xe S2, nếu bằng 1 thì hiển thị chuỗi "S2:Fill " để thể hiện cảm biến đã đầy, ngược lại thì hiển thị chuỗi "S2:Empty".

```
if(S2==1){lcd.print("S2:Fill ");}
```

```
else{lcd.print("S2:Empty");}
```

//Đặt con trỏ của LCD đến hàng 2, cột 0 để hiển thị thông tin của cảm biến S3.

```
lcd.setCursor (0,2);
```

//Kiểm tra giá trị biến S3 để xác định trạng thái của cảm biến đỗ xe S3, nếu bằng 1 thì hiển thị chuỗi "S3:Fill " để thể hiện cảm biến đã đầy, ngược lại thì hiển thị chuỗi "S3:Empty".

```
if(S3==1){lcd.print("S3:Fill ");}
```

```
else{lcd.print("S3:Empty");}
```

//Đặt con trỏ của LCD đến hàng 2, cột 12 để hiển thị thông tin của cảm biến S4.

```
lcd.setCursor (12,2);
```

//Kiểm tra giá trị biến S4 để xác định trạng thái của cảm biến đỗ xe S3, nếu bằng 1 thì hiển thị chuỗi "S4:Fill " để thể hiện cảm biến đã đầy, ngược lại thì hiển thị chuỗi "S4:Empty".

```
if(S4==1){lcd.print("S4:Fill ");}
```

```
else{lcd.print("S4:Empty");}
```

///// Kiểm tra xem có thẻ RFID trong phạm vi đọc hay không.

```
if ( ! rfid.PICC_IsNewCardPresent() || ! rfid.PICC_ReadCardSerial() ) {
```

```
    delay(50);
```

```
    return;
```

```
}
```

/////Đọc ID của thẻ RFID.

```
String content= "";
```

```
byte letter;
```

```
for (byte i = 0; i < rfid.uid.size; i++) {
```

```
    content.concat(String(rfid.uid.uidByte[i] < 0x10 ? "0" : ""));
```

```
    content.concat(String(rfid.uid.uidByte[i], HEX));
```

```
}
```

```
content.toUpperCase();
```

// Kiểm tra xem ID có nằm trong danh sách ID hợp lệ hay không.

```
bool isValidID = false;
```

```
for (int i = 0; i < 4; i++) {
```

```
    if (content == validIDs[i]) {
```

```
        isValidID = true;
```

```
        break;
```

```
    }
```

```
}
```

```
////Hiển thị kết quả và điều khiển servo.
//Kiểm tra nếu ID đọc được từ thẻ RFID hợp lệ, thực hiện các lệnh bên trong.
if (isValidID) {
    //Kiểm tra nếu cảm biến IR xe vào báo hiệu đang được nhận và biến flag1 = 0
    (để đảm bảo thực hiện 1 lần), thực hiện các lệnh bên trong.
    if (digitalRead(ir_enter) == 0 && flag1 == 0 ) {

        /// Kiểm tra xem còn chỗ để đậu xe hay không (slot > 0 và slot < 5).
        if (slot > 0&&slot<5) {
            //Gán giá trị flag1 bằng 1 để đảm bảo không thực hiện lại lệnh này nếu vẫn
            nhận được tín hiệu từ cảm biến IR xe vào.
            flag1 = 1;
            //Kiểm tra xem biến flag2 có bằng 0 hay không, nếu có thực hiện các lệnh
            bên trong.
            if (flag2 == 0) {
                //Thiết lập vị trí con trỏ trên màn hình LCD là dòng thứ 4, cột đầu tiên
                (dòng và cột bắt đầu từ 0).
                lcd.setCursor (0,3);
                //In chuỗi ký tự " Welcome " lên màn hình LCD.
                lcd.print("    Welcome    ");
                //Điều khiển servo quay ở góc 180 độ (mở cổng vào).
                myservo.write(180);
                // Giảm số chỗ đậu xe còn trống đi 1 đơn vị.
                slot = slot-1;
                // Đợi 2 giây.
                delay(2000);
```

//Gán giá trị flag2 bằng 1 để đảm bảo không thực hiện lại lệnh này nếu vẫn nhận được tín hiệu từ cảm biến IR xe vào.

```
flag2=1;
```

```
}
```

// Thiết lập vị trí con trỏ trên màn hình LCD là dòng thứ 4, cột đầu tiên (dòng và cột bắt đầu từ 0).

```
lcd.setCursor(0, 3);
```

```
//Xóa chuỗi ký tự hiện tại trên màn hình LCD.
```

```
lcd.print("          ");
```

```
}
```

```
//Nếu không còn chỗ đậu xe, thực hiện các lệnh bên trong.
```

```
else {
```

//Thiết lập vị trí con trỏ trên màn hình LCD là dòng thứ 4, cột đầu tiên (dòng và cột bắt đầu từ 0).

```
lcd.setCursor(0, 3);
```

```
//In chuỗi ký tự "Sorry, Parking full" lên màn hình LCD.
```

```
lcd.print("Sorry, Parking full");
```

```
//Đợi 1,5 giây.
```

```
delay(1500);
```

//Thiết lập vị trí con trỏ trên màn hình LCD là dòng thứ 4, cột đầu tiên (dòng và cột bắt đầu từ 0).

```
lcd.setCursor(0, 3);
```

```
//Xóa chuỗi ký tự hiện tại trên màn hình LCD.
```

```
lcd.print("          ");
```

```
}
```

//Kiểm tra nếu cảm biến IR xe ra báo hiệu đang được nhận và biến flag2 = 0 (để đảm bảo thực hiện 1 lần), thực hiện các lệnh bên trong.

```
if(digitalRead (ir_back) == 0 && flag2==0){
```

//Gán giá trị flag2 bằng 1 để đảm bảo không thực hiện lại lệnh này nếu vẫn nhận được tín hiệu từ cảm biến IR xe ra.

```
flag2=1;
```

//Kiểm tra xem biến flag1 có bằng 0 hay không, nếu có thực hiện các lệnh bên trong.

```
if(flag1==0){
```

```
    //Điều khiển servo quay ở góc 180 độ.
```

```
    myservo.write(180);
```

```
    //Tăng số chỗ đậu xe còn trống lên 1 đơn vị.
```

```
    slot = slot+1;
```

//Thiết lập vị trí con trỏ trên màn hình LCD là dòng thứ 4, cột đầu tiên (dòng và cột bắt đầu từ 0).

```
    lcd.setCursor(0, 3);
```

```
    //Hiển thị thông báo "See you soon" trên màn hình LCD.
```

```
    lcd.print("  See you soon  ");
```

```
    //Đợi 2 giây.
```

```
    delay(2000);
```

//Gán giá trị flag1 bằng 1.

```
    flag1=1;
```

```
}
```

//Thiết lập vị trí con trỏ trên màn hình LCD là dòng thứ 4, cột đầu tiên (dòng và cột bắt đầu từ 0).

```
    lcd.setCursor(0, 3);
```

```
    //Xóa chuỗi ký tự hiện tại trên màn hình LCD.
```

```
    lcd.print("          ");
```

```
}
```

//Khi giá trị của flag1 bằng 1 và giá trị của flag2 bằng 1 thì thực hiện các lệnh bên trong.

```
if(flag1==1 && flag2==1){
```

```
    //Đợi 2 giây.
```

```

delay (2000);
    //Điều khiển servo quay ở góc 90 độ.
    myservo.write(90);
    //Gán giá trị flag1 bằng 0 và giá trị flag2 bằng 0.
    flag1=0, flag2=0;
//Nếu thẻ RFID không hợp lệ được quét, thực hiện các lệnh bên trong.
else { if ((digitalRead(ir_enter) == 0)|| (digitalRead(ir_back) == 0)){
    //Thiết lập vị trí con trỏ trên màn hình LCD là dòng thứ 4, cột đầu tiên (dòng
và cột bắt đầu từ 0).
    lcd.setCursor (0,3);
    //In chuỗi ký tự "Wrong ID" lên màn hình LCD.
    lcd.print(" Wrong ID ");
    //Đợi 2 giây.
    delay(2000);
}
    //Thiết lập vị trí con trỏ trên màn hình LCD là dòng thứ 4, cột đầu tiên (dòng
và cột bắt đầu từ 0).
    lcd.setCursor(0, 3);
    //Xóa chuỗi ký tự hiện tại trên màn hình LCD.
    lcd.print(" ");
}
//Đợi 1 giây để bắt đầu 1 vòng lặp mới.
delay(1000);
}

```