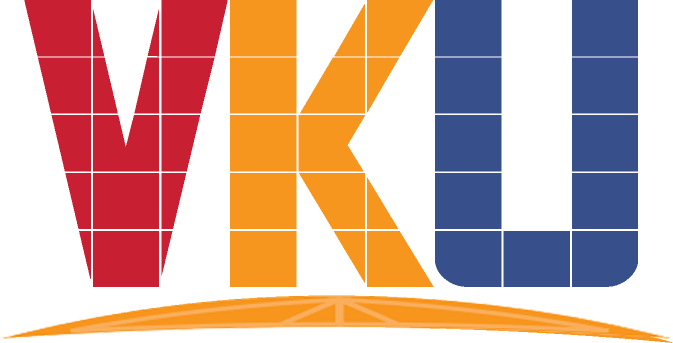
ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN

**KHOA KỸ THUẬT MÁY TÍNH VÀ ĐIỆN TỬ**



**🕮**

**ĐỒ ÁN CƠ SỞ 3**

**ĐỀ TÀI**

# XÂY DỰNG HỆ THỐNG BÃI ĐỔ XE THÔNG MINH

Sinh viên thực hiện: **Phạm Võ Minh Tiến – 23CE.B022**

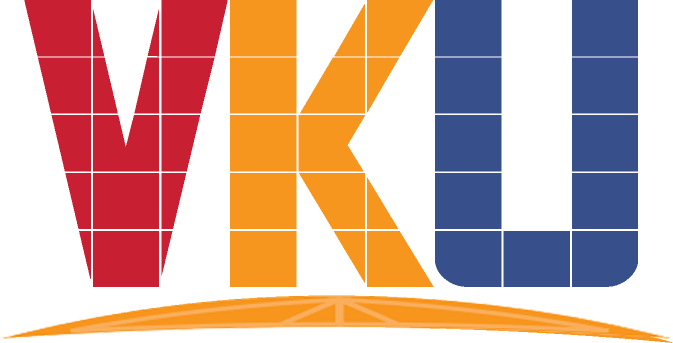
Lớp: **23ES**

Giảng viên hướng dẫn: TS.Trần Thanh

ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN

**KHOA KỸ THUẬT MÁY TÍNH VÀ ĐIỆN TỬ**

**🕮**

**ĐỒ ÁN CƠ SỞ 3**

**ĐỀ TÀI**

# XÂY DỰNG HỆ THỐNG BÃI ĐỔ XE THÔNG MINH

Sinh viên thực hiện: **Phạm Võ Minh Tiến – 23CE.B022**

Lớp: **23ES**

Giảng viên hướng dẫn: TS. Trần Thanh

**NHẬN XÉT & ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN**

*Đà Nẵng , ngày……tháng……năm…….*

Người hướng dẫn

*(Ký và ghi rõ họ tên)*

# LỜI CẢM ƠN

Lời nói đầu tiên chúng em xin gửi lời cảm ơn chân thành đến ban giám hiệu trường Đại học Công nghệ Thông tin và Truyền thông Việt – Hàn đã xây dựng một môi trường học tập lý tưởng với hệ thống cơ sở vật chất đầy đủ, đặc biệt là thư viện trường với nguồn tài liệu phong phú và đa dạng. Những tài liệu quý giá này đã đóng góp không nhỏ vào quá trình nghiên cứu và hoàn thành đồ án cơ sở 3 của chúng em.

Với chúng em, những sinh viên năm hai đang từng bước làm quen và nắm vững các kiến thức chuyên ngành, đồ án này vừa là một thử thách giúp chúng em củng cố kiến thức đã học, vừa là một cơ hội để chúng em tiếp cận, khám phá những khía cạnh mới trong lĩnh vực điện tử. Qua quá trình thực hiện, chúng em nhận thấy sự cần thiết của việc nghiên cứu tài liệu và không ngừng học hỏi để trao dồi kinh nghiệm và hoàn thiện bản thân hơn. Tuy nhiên, do kinh nghiệm của chúng em còn hạn chế, nên chắc chắn vẫn còn những thiếu sót cần được khắc phục.

Chúng em xin trân trọng cảm ơn giáo viên hướng dẫn đã tận tình chỉ dạy và đưa ra những nhận xét quý báu, giúp chúng em rút ra nhiều kinh nghiệm quan trọng. Những góp ý từ thầy / cô sẽ là nguồn động lực để chúng em không ngừng hoàn thiện và đạt được những tiến bộ trong các đồ án và nghiên cứu tiếp theo.

Cuối cùng, chúng em xin kính chúc cô và các thầy cô trong hội đồng chấm thi thật nhiều sức khỏe và ngày càng thành công hơn trong sự nghiệp nghề giáo của mình.

# LỜI MỞ ĐẦU

Ngày nay, xã hội ngày càng phát triển, kéo theo nhu cầu về cơ sở hạ tầng và tiện ích phục vụ đời sống con người cũng không ngừng tăng lên. Trong đó, nhu cầu đi lại bằng phương tiện cá nhân ngày càng phổ biến, đặc biệt tại các khu đô thị lớn, dẫn đến tình trạng thiếu hụt chỗ đỗ xe trầm trọng. Việc đỗ xe không đúng nơi quy định không chỉ gây mất mỹ quan đô thị mà còn ảnh hưởng đến giao thông và tiềm ẩn nguy cơ tai nạn. Trước thực trạng đó, việc xây dựng một hệ thống bãi đỗ xe thông minh là rất cần thiết nhằm tối ưu hóa không gian, nâng cao hiệu quả quản lý và đáp ứng nhu cầu ngày càng cao của người dân.

Nhận thấy vai trò quan trọng của bãi đỗ xe thông minh trong việc giải quyết bài toán giao thông đô thị và nâng cao trải nghiệm người sử dụng, chúng em đã quyết định lựa chọn đề tài "Xây Dựng Bãi Đỗ Xe Thông Minh ". Đây không chỉ là một đề tài mang tính ứng dụng cao mà còn giúp chúng em hiểu rõ hơn về cách áp dụng kiến thức điện tử vào các giải pháp công nghệ trong đời sống thực tiễn.

Mục tiêu của đề tài là tìm hiểu, nghiên cứu và ứng dụng nguyên lý hoạt động của các linh kiện điện tử cơ bản vào việc thiết kế một hệ thống bãi đỗ xe thông minh. Thông qua đề tài, nhóm chúng em mong muốn xây dựng một mô hình có khả năng hoạt động chính xác, ổn định và hiệu quả, từ đó mô phỏng các tình huống thực tế như kiểm soát lượng xe ra vào, phát hiện chỗ trống và hiển thị thông tin cho người sử dụng.

Phạm vi nghiên cứu của đề tài tập trung vào việc thiết kế, xây dựng và mô phỏng hoạt động của một bãi đỗ xe thông minh trên mô hình thực nghiệm, sử dụng các linh kiện điện tử cơ bản.

Đề tài này không chỉ giúp chúng em áp dụng các kiến thức đã học vào thực tế mà còn mở ra cơ hội nghiên cứu và sáng tạo những giải pháp tối ưu góp phần hiện đại hóa hạ tầng giao thông trong tương lai

**MỤC LỤC**

**LỜI CẢM ƠN**

**LỜI MỞ ĐẦU**

**DANH MỤC HÌNH ẢNH 5**

**CHƯƠNG I : TỔNG QUAN VỀ HỆ THỐNG 6**

1.1: Đặt vấn đề 7

1.2: Các công nghệ hiện tại trong hệ thống bãi đỗ xe 8

1.3: Ứng dụng vi điều khiển trong hệ thống bãi đổ xe thông minh 9

**CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 12**

2.1: Các thành phần chính trong mạch 13

*2.1.1: Linh kiện sử dụng* 14

a) Vi Điều Khiển Arduino UNO R3 15

b) Module RFID 23

c) LED 24

d) Cảm biến hồng ngoại 25

e) Màn hình LCD. 26

2.2: Yêu cầu kĩ thuật và chức năng của hệ thống 27

*2.2.1: Yêu cầu kĩ thuật* 29

*2.2.2: Chức năng của hệ thống* 29

**CHƯƠNG III : XÂY DỰNG VÀ TRIỂN KHAI 30**

3.1: Mô phỏng nguyên lí mạch trên 31

3.2:Phân tích code 33

3.3: Mạch hoàn thiện thực tế 43

**CHƯƠNG IV : NHẬN XÉT VÀ ĐÁNH GIÁ 44**

4.1: Kết quả nghiên cứu 45

4.2: Hạn chế và hướng phát triển trong tương lai 45

**Tài liệu tham khảo 48**

# DANH MỤC HÌNH ẢNH

Hình 1: Tổng quan hệ thống ....................................................................................... 12

Hình 2: Sơ đồ khối ...................................................................................................... 13

Hình 3: Vi điều khiển Arduino ................................................................................... 15

Hình 4: Module RFID ................................................................................................. 21

Hình 5: LED đơn ........................................................................................................ 22

Hình 6: Cấu tạo LED .................................................................................................. 23

Hình 7: Nguyên lí hoạt động LED .............................................................................. 23

Hình 8: Cảm biến hồng ngoại ..................................................................................... 24

Hình 9: Màn hình LCD ............................................................................................... 27

Hình 10: Module I2C .................................................................................................. 30

Hình 11: Mạch mô phỏng ........................................................................................... 33

Hình 12: Mạch thực tế ................................................................................................ 43

**CHƯƠNG I : TỔNG QUAN VỀ HỆ THỐNG ĐÈN GIAO THÔNG**

## 1.1: Đặt vấn đề

Cùng với sự phát triển không ngừng của khoa học kỹ thuật, xã hội ngày càng hiện đại và văn minh hơn. Các đô thị đang chuyển mình mạnh mẽ, kéo theo đó là nhu cầu về phương tiện giao thông cá nhân ngày càng tăng cao. Sự gia tăng nhanh chóng của xe ô tô, đặc biệt tại các thành phố lớn như Hà Nội, TP. Hồ Chí Minh, Đà Nẵng,... đã dẫn đến nhiều vấn đề nghiêm trọng trong quản lý và vận hành hạ tầng giao thông.

Theo thống kê của Bộ Giao thông Vận tải, tính đến năm 2023, Việt Nam có hơn 72 triệu phương tiện giao thông đăng ký, trong đó xe ô tô chiếm tỷ lệ không ngừng tăng, kéo theo đó là nhu cầu đỗ xe ngày càng trở nên cấp thiết. Tuy nhiên, hệ thống bãi đỗ xe hiện nay còn nhiều bất cập như diện tích hạn chế, quản lý thủ công, thiếu tính minh bạch và gây lãng phí thời gian cho người sử dụng. Việc đậu xe không đúng nơi quy định cũng làm gia tăng tình trạng ùn tắc giao thông, gây mất mỹ quan đô thị và ảnh hưởng đến trật tự an toàn giao thông.

Trước thực trạng đó, việc ứng dụng công nghệ vào xây dựng hệ thống bãi đỗ xe thông minh là một giải pháp mang tính cấp thiết và hiệu quả. Hệ thống này không chỉ giúp quản lý bãi đỗ xe một cách khoa học và chính xác, mà còn hỗ trợ người dùng dễ dàng tìm kiếm vị trí trống, rút ngắn thời gian chờ đợi và giảm thiểu xung đột trong giao thông đô thị.

Xuất phát từ nhu cầu cấp thiết và tiềm năng ứng dụng thực tiễn cao, nhóm chúng em quyết định lựa chọn thực hiện đề tài Xây dựng hệ thống bãi đỗ xe thông minh sử dụng vi điều khiển Arduino và cảm biến. Đề tài giúp chúng em không chỉ hiểu rõ hơn về nguyên lý hoạt động của các linh kiện điện tử như cảm biến hồng ngoại, RFID, servo motor, mà còn rèn luyện kỹ năng phân tích – thiết kế hệ thống điều khiển trong thực tế, góp phần vào giải pháp công nghệ cho các vấn đề xã hội hiện đại

## Phạm Vi Ứng Dụng

## Bãi đỗ xe tư nhân, khu chung cư, trung tâm thương mại.

## Hệ thống kiểm soát ra vào dựa trên thẻ RFID.

## Hiển thị thông tin chỗ đỗ còn trống trên màn hình LCD.

## Cảnh báo bằng còi buzzer khi có xe vào/ra hoặc thẻ không hợp lệ

## 1.2: Các công nghệ hiện tại trong hệ thống bãi đỗ xe thông minh

## Sự phát triển mạnh mẽ của khoa học công nghệ đã tạo điều kiện cho các hệ thống bãi đỗ xe thông minh ra đời, góp phần giải quyết hiệu quả những bất cập trong việc quản lý và sử dụng không gian đỗ xe tại các đô thị lớn. Nếu như trước đây việc quản lý bãi đỗ xe chủ yếu được thực hiện theo phương pháp thủ công như sử dụng vé giấy và nhân viên kiểm soát, thì hiện nay, nhiều công nghệ hiện đại đã được tích hợp để tự động hóa quy trình vận hành, tiết kiệm nhân lực và nâng cao hiệu quả quản lý.

## Một trong những công nghệ tiêu biểu được ứng dụng phổ biến là cảm biến phát hiện chỗ trống, bao gồm cảm biến hồng ngoại hoặc siêu âm. Các cảm biến này được gắn tại từng vị trí đỗ để xác định xem vị trí đó đang trống hay có xe. Khi có phương tiện vào hoặc rời khỏi bãi, cảm biến sẽ gửi tín hiệu đến bộ điều khiển trung tâm nhằm cập nhật trạng thái chỗ đỗ theo thời gian thực, từ đó hiển thị thông tin cho người điều khiển phương tiện biết số chỗ trống còn lại hoặc hướng dẫn đến vị trí phù hợp. Một số trung tâm thương mại lớn tại TP. Hồ Chí Minh và Hà Nội đã triển khai thành công hệ thống này để nâng cao trải nghiệm người dùng.

## Tiếp theo là công nghệ RFID (Radio Frequency Identification) – một phương pháp nhận diện bằng sóng vô tuyến. Mỗi xe được gắn một thẻ RFID chứa mã định danh riêng biệt. Khi phương tiện đi qua cổng vào hoặc ra, đầu đọc RFID sẽ nhận diện thẻ này để tự động thực hiện các thao tác như mở rào chắn, ghi nhận thời gian ra vào và trừ phí nếu hệ thống tích hợp với chức năng thanh toán tự động. Ưu điểm của RFID là tốc độ xử lý nhanh, không yêu cầu tiếp xúc vật lý và đặc biệt phù hợp với các mô hình bãi đỗ xe không cần người điều khiển.

## Ngoài ra, trí tuệ nhân tạo (AI) kết hợp với xử lý hình ảnh cũng đang được tích cực áp dụng trong các hệ thống cao cấp. AI giúp nhận dạng biển số xe, dự đoán nhu cầu đỗ xe vào các thời điểm cao điểm, và phân bổ chỗ đỗ một cách tối ưu. Camera giám sát tích hợp AI còn có khả năng phát hiện các hành vi bất thường như đỗ sai quy định, va chạm hoặc nghi vấn an ninh, giúp cảnh báo kịp thời cho bộ phận quản lý. Tại nhiều quốc gia phát triển như Nhật Bản và Hàn Quốc, các hệ thống này đã được áp dụng hiệu quả tại các bãi xe tầng, bãi ngầm và trung tâm thương mại lớn.

## Song song với đó là sự phát triển của công nghệ Internet of Things (IoT) – cho phép kết nối tất cả thiết bị như cảm biến, bộ điều khiển, bảng hiển thị và camera thành một hệ thống thống nhất, vận hành đồng bộ và có thể giám sát từ xa qua Internet. Với IoT, người quản lý có thể theo dõi toàn bộ hoạt động của bãi xe theo thời gian thực trên các thiết bị thông minh như điện thoại hay máy tính. Người dùng cũng có thể tra cứu thông tin, đặt chỗ, hoặc thanh toán qua ứng dụng tích hợp.

## Cuối cùng, hệ thống đèn LED và màn hình LCD I2C được sử dụng để hiển thị trạng thái hoạt động như số chỗ còn trống, thông báo khi bãi đầy hoặc chỉ dẫn hướng đi cho người điều khiển phương tiện. Công nghệ LED hiện đại giúp tăng độ sáng, giảm tiêu thụ năng lượng và kéo dài tuổi thọ thiết bị so với bóng đèn truyền thống.

## Tóm lại, việc ứng dụng đồng bộ các công nghệ tiên tiến vào hệ thống bãi đỗ xe thông minh không chỉ nâng cao chất lượng dịch vụ và trải nghiệm người dùng mà còn góp phần quan trọng vào việc giảm thiểu ùn tắc, tiết kiệm chi phí vận hành và thúc đẩy quá trình xây dựng các đô thị thông minh, văn minh và hiện đại hơn trong tương lai.

### 1.3: Ứng dụng vi điều khiển trong hệ thống bãi đỗ xe thông minh

Trong thời đại công nghệ số và xu hướng phát triển đô thị thông minh, việc tự động hóa và số hóa các hoạt động hạ tầng đô thị là một yếu tố then chốt. Một trong những ứng dụng điển hình thể hiện rõ điều này là hệ thống bãi đỗ xe thông minh – nơi mà các vi điều khiển đóng vai trò trung tâm trong quá trình vận hành.

**Tổng quan hệ thống bãi đỗ xe**

Trong bối cảnh bãi đỗ xe, các mạch này được thiết kế để thu thập, xử lý và phản hồi tín hiệu từ các cảm biến đầu vào như cảm biến hồng ngoại (dò xe), RFID (xác thực phương tiện), sau đó xuất tín hiệu điều khiển đến cổng servo, màn hình LCD I2C, hoặc hệ thống LED chỉ dẫn. Khi kết hợp với vi điều khiển Arduino UNO R3, toàn bộ hệ thống có thể vận hành hoàn toàn tự động, với khả năng phản ứng theo thời gian thực và tùy biến linh hoạt.

**Cấu trúc hoạt động và luồng xử lý tín hiệu**

Về mặt cấu trúc, hệ thống gồm các thành phần chính như sau:

* **Cảm biến hồng ngoại (IR sensor):** Được lắp tại từng chỗ đỗ, giúp phát hiện sự hiện diện của xe bằng việc cắt chùm tia hồng ngoại. Khi một xe tiến vào, cảm biến sẽ trả về giá trị logic LOW, báo hiệu chỗ đó đã bị chiếm.
* **Module RFID:** Mỗi xe được gắn một thẻ từ định danh duy nhất. Khi xe đến cổng vào, đầu đọc RFID đọc mã thẻ và gửi dữ liệu về Arduino để kiểm tra quyền truy cập.
* **Vi điều khiển Arduino UNO R3:** Là bộ não trung tâm, điều phối toàn bộ hoạt động. Dựa trên tín hiệu từ cảm biến và RFID, Arduino sẽ quyết định có mở cổng hay không, cập nhật số lượng chỗ trống và hiển thị trạng thái bãi đỗ trên LCD.
* **Servo Motor:** Được điều khiển bằng tín hiệu PWM từ Arduino, đóng/mở rào chắn một cách chính xác theo quy trình.
* **LCD I2C 16x2:** Hiển thị số lượng chỗ trống còn lại, thông báo "x" khi bãi đầy hoặc "v" khi xe được phép vào.
* **Đèn LED chỉ báo:** Dùng để hỗ trợ tín hiệu cho người điều khiển phương tiện – như đèn xanh cho phép vào, đèn đỏ khi đầy.

**Ưu điểm nổi bật của hệ thống sử dụng vi điều khiển**

* **Tính tự động cao:** Hệ thống hoàn toàn tự vận hành không cần sự can thiệp của con người, từ nhận diện xe đến đóng/mở cổng.
* **Độ chính xác tuyệt đối:** Sử dụng cảm biến số giúp loại bỏ sự sai lệch do thao tác thủ công. Việc xử lý bằng logic nhị phân loại bỏ lỗi do nhiễu hay môi trường.
* **Dễ dàng mở rộng:** Có thể tích hợp thêm camera nhận diện biển số, kết nối WiFi để quản lý qua cloud, hoặc mở rộng lên các tầng bãi xe phức tạp.
* **Tiết kiệm năng lượng và chi phí:** Vi điều khiển hoạt động với điện áp thấp (5V), tiêu tốn điện năng rất nhỏ so với hệ thống cơ điện truyền thống.
* **Thân thiện và trực quan:** Màn hình hiển thị LCD giúp người dùng dễ dàng tương tác. Hệ thống có thể hỗ trợ cả âm thanh cảnh báo, chỉ dẫn LED, thậm chí qua app trên smartphone.

**Triển vọng ứng dụng và mở rộng trong thực tế**

Tại các đô thị lớn như Hà Nội, TP.HCM, Đà Nẵng... nhiều bãi đỗ xe đang từng bước chuyển đổi sang hình thức quản lý thông minh. Việc sử dụng vi điều khiển giúp tiết kiệm chi phí vận hành, giảm nhân sự, đồng thời tăng hiệu suất sử dụng mặt bằng đỗ xe lên đến 30-40% so với mô hình truyền thống.

Trong tương lai gần, hệ thống này hoàn toàn có thể tích hợp với hệ sinh thái đô thị thông minh, đồng bộ với các ứng dụng như quản lý phương tiện công cộng, cảnh báo giao thông, bản đồ đỗ xe thời gian thực và thanh toán điện tử qua QR code.

* **Kết luận**

Việc ứng dụng vi điều khiển vào hệ thống bãi đỗ xe thông minh không chỉ là bước tiến kỹ thuật trong lĩnh vực IoT mà còn là giải pháp thiết thực cho các đô thị đang đối mặt với áp lực hạ tầng giao thông ngày càng lớn. Hệ thống này hội tụ đầy đủ các yếu tố: hiện đại – tiết kiệm – hiệu quả – dễ triển khai, và là một hướng đi tất yếu trong công cuộc số hóa giao thông và hạ tầng đô thị tại Việt Nam.

# CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

***Tổng quan về hệ thống***

A computer circuit board with a screen

AI-generated content may be incorrect.

*Hình 1 : Tổng quan về hệ thống*

## A diagram of a system AI-generated content may be incorrect.

*Hình 2 : Sơ đồ khối hệ thống*

Chức năng các khối trong hệ thống bãi đỗ xe thông minh (sử dụng Arduino RFID, màn hình LCD, còi buzzer và servo):

Khối điều khiển trung tâm: Sử dụng Arduino Uno R3 làm bộ não của hệ thống. Khối này chịu trách nhiệm xử lý và điều khiển các yêu cầu từ hệ thống, bao gồm quản lý trạng thái các bãi đỗ xe, giao tiếp với các cảm biến RFID, động cơ servo và các thiết bị khác trong hệ thống.

Khối hiển thị (LCD 16x2, LED, màn hình): Màn hình LCD 16x2 và các đèn LED sẽ hiển thị thông tin từ bộ xử lý trung tâm, giúp người dùng biết được tình trạng của các chỗ đỗ xe, thông báo trạng thái (có hay không có xe đỗ) và các thông tin liên quan.

## Khối cảm biến RFID: Sử dụng module RFID RC522 để nhận diện thẻ RFID khi xe vào/ra bãi đỗ, giúp xác nhận và ghi nhận các xe vào/ra tự động.

## Khối động cơ servo: Động cơ servo sẽ được điều khiển để mở hoặc đóng các cổng đỗ xe, tùy thuộc vào trạng thái của bãi đỗ.

## Khối âm thanh (Còi Buzzer): Còi buzzer phát ra âm thanh cảnh báo khi có sự kiện đặc biệt xảy ra, chẳng hạn như khi bãi đỗ đầy, hoặc có xe đỗ sai quy định.

## Khối cảm biến hồng ngoại (IR): Các cảm biến hồng ngoại được sử dụng để phát hiện sự có mặt của xe trong bãi đỗ. Khi có xe vào vị trí đỗ, cảm biến hồng ngoại sẽ truyền tín hiệu đến bộ điều khiển trung tâm để cập nhật trạng thái của bãi đỗ. Cảm biến này cũng có thể dùng để phát hiện các vật cản hoặc tình trạng lối đi.

**Nguồn cấp cho các linh kiện :**

| **STT** | **Tên linh kiện / Module** | **Điện áp cung cấp (V)** |
| --- | --- | --- |
| 1 | Arduino Uno R3 | 5V DC (qua cổng USB hoặc jack nguồn) |
| 2 | Màn hình LCD 16x2 (I2C) | 5V DC |
| 3 | Module RFID RC522 | 3.3V DC |
| 4 | Còi Buzzer | 5V DC |
| 5 | Động cơ Servo (SG90) | 5V DC |
| 6 | Cảm biến hồng ngoại IR | 5V DC |

## 2.1: Các thành phần chính trong mạch

Hệ thống bãi đỗ xe thông minh sử dụng Arduino được cấu thành từ nhiều thành phần phần cứng điện tử khác nhau. Mỗi linh kiện trong hệ thống đều đảm nhận một vai trò cụ thể, đảm bảo hệ thống hoạt động hiệu quả và tự động. Dưới đây là mô tả chi tiết các linh kiện chính được sử dụng trong mạch:

**2.1.1: Linh kiện sử dụng**

#### a) Vi điều khiển Arduino

Tổng quan về Arduino :

Arduino Uno là bo mạch vi xử lý hoạt động dựa trên ATmega328. Bo mạch này có 14 chân input/output digital (trong đó có 6 chân được dùng cho điều chế xung đầu ra PWM), 6 đầu vào analog, tần số giao động thạch anh là 16MHz, kết nối USB, jack cắm nguồn, chân tiêu đề ICSP, một nút reset. Bo mạch này chứa tất cả các tính năng cần thiết để hỗ trợ kết nối với các vi điều khiển khác. Nguồn sử dụng cho bo mạch có thể qua USB, sử dụng pin hoặc nguồn thông qua bộ chuyển đổi AC–DC.

**

*Hình 3: Vi điều khiển Arduino Uno*

Arduino Uno khác với tất cả các bo mạch khác ở chỗ nó không sử dụng chip điều khiển nối tiếp FTDI USB. Thay vào đó các tính năng của ATmega16U2 được lập trình để chuyển đổi USB nối tiếp. 15 Ở phiên bản thứ hai: bo mạch Uno có điện trở nối đường 8U2 HWB với đất, do đó ta dễ dàng hơn trong việc thiết lập chế độ DFU. Ở phiên bản sửa đổi thứ 3 của bo mạch có các tính năng mới dưới đây: - Sơ đồ chân 1.0: Thêm các chân SDA và SCL gần với chân AREF và 2 chân mới được đặt gần chân RESET, IOREF cho phép Shield nhận nguồn cấp từ bo mạch. Trong tương lai, Shield sẽ tương thích với các bo mạch có sử dụng AVR, có điện áp hoạt động là 5V và Arduino Due hoạt động ở 3.3V. Một số chân trong bo mạch không được kết nối gì để dành cho mục đích trong tương lai. - Mạch Reset mạnh mẽ hơn. - ATmega 16U2 thay thế cho 8U2. "Uno" là từ trong tiếng Ý và được đặt tên để đánh dấu việc phát hành phiên bản Arduino 1.0. Uno và phiên bản 1.0 sẽ là phiên bản tham khảo của Arduino, và luôn có sự cải tiến. Uno là một trong những bo mạch mới nhất trong một loạt các bo mạch USB Arduino, và là mô hình tham chiếu nền tảng của Arduino, để so sánh với phiên bản trước đó, xem các thông số của bo mạch Arduino.

| Thông số | Giá trị |
| --- | --- |
| Vi xử lý chính | ATmega328P |
| Xung nhịp | 16 MHz |
| Bộ nhớ Flash | 32 KB (trong đó 0.5 KB dành cho bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Số chân Digital I/O | 14 chân (trong đó có 6 chân hỗ trợ PWM) |
| Số chân Analog Input | 6 chân |
| Giao tiếp | UART, SPI, I2C |
| Điện áp hoạt động | 5V |
| Nguồn vào khuyến nghị | 7V – 12V (qua DC Jack hoặc cổng USB) |

#### *Bảng 1. Tóm tắt các thông số chính của bo mạch Arduino Uno*

#### Arduino Uno có thể được cấp nguồn thông qua kết nối USB hoặc với một nguồn cung cấp điện bên ngoài. Nguồn điện được chọn một cách tự động. Nguồn cấp bên ngoài (không phải là USB) có thể lấy từ bộ chuyển đổi AC-DC hoặc nguồn pin. Các bộ chuyển đổi có thể được kết nối bằng cách cắm chân cắm đường kính 2.1mm vào lỗ cắm điện trên bo mạch. Nếu nguồn lấy từ pin có thể được lắp vào 2 đầu GND và Vin chân tiêu đề của kết nối POWER. Bo mạch có thể hoạt động với các nguồn cấp ngoài từ 6 đến 20 volt. Nếu nguồn ít nhất thường là 7V, tuy nhiên, các chân 5V có thể được cấp nguồn bé hơn 5V nhưng khi đó mạch có thể hoạt động không ổn định. Nếu sử dụng hơn 12V, bộ ổn áp bị nóng và hỏng mạch, khuyến nghị nên sử dụng ở khoảng 7 đến 12 volts. Nguồn cấp của các chân như sau:

#### - Vin. Điện áp đầu vào của bo mạch Arduino khi nó sử dụng nguồn cấp ngoài (khác với các nguồn cấp 5V từ kết nối USB và các nguồn điện theo quy định). Ta có thể cấp nguồn qua chân Vin hoặc cấp nguồn thông qua các jack cắm kết nối với chân này.

#### - 5V. Chân đầu ra được quy đinh là 5V. Bo mạch có thể có thể được cấp nguồn điện từ các jack (7-12V), kết nối USB (5V), hoặc chân Vin của bo mạch (7- 12V). Cung cấp điện áp thông qua chân 5V hoặc 3.3V bỏ qua các khuyến cáo có thể gây hỏng mạch. Không nên sử dụng nó.

#### - 3.3V. Nguồn cấp 3.3V được quy định trên bo mạch. Dòng cấp tối đa là 50mA.

#### - GND Chân nối đất.

#### - IOREF. Chân này cấp điện áp tham chiếu cho vi điểu khiển hoạt động. Bộ hỗ trợ cấu hình chuẩn đọc điện áp trên chân OIREF và lựa chọn nguồn cấp thích hợp hoặc kích hoạt dịch điện áp trên đầu ra để làm việc với các nguồn 5V hoặc 3.3V.

Trên bo mạch có 14 chân digital có thể được sử dụng cho mạch đích vào hoặc ra, sử dụng các hàm pinMode(), digitalWrite(), và digitalRead(). Chúng hoạt động ở mức điện áp 5V. mỗi chân có thể cung cấp hoặc nhận dòng cực đại là 40mA và được nối với điện trở mặc định từ 20-50 KΩ. Ngoài ra còn có một số chân có chức năng đặc biệt:

- Chân nối tiếp: 0 (RX) và 1 (TX). Sử dụng để nhận và truyền dữ liệu TTL. Chân này được nối với chân tương ứng của ATmega82U.

- Chân ngắt ngoài: 2 và 3. Chân này có thể được thiết lập để thực hiện ngắt khi điện áp quá thấp hoặc thay đổi đột biến giá trị điện áp. Thường sử dụng hàm ngắt attachInterrupt().

- PWM: 3, 5, 6, 9 và 10. Xung PWM có độ rộng là 8 bits. Khi điều chế xung đầu ra sử dụng hàm analogWrite().

- SPI: 10(SS), 11 (MOSI), 12 (MISO), 13 (SCK). Các chân này hỗ trợ truyền dẫn SPI sử dụng thư viện SPI.

- LED: 13. Có một đèn LED được nối với chân 13. Khi điện áp ở mức cao LED sáng và ở mức thấp thì LED tắt. Uno có 6 đầu ra analog, có nhãn là A0 đến A5, mỗi chân được cung cấp 10 bits dữ liệu (tương ứng với 1024 giá trị khác nhau). Trên bo mạch Ethernet có 6 chân đầu vào tương tự, có nhãn từ A0 đến A5, mỗi chân sử dụng 10 bits (tức là có 1024 giá trị khác nhau). Nguồn cấp cho các chân này là từ 0-5V, mặc dù nó có thể thay đổi phạm vi hoạt động của chúng bằng cách sử dụng chân AREF và hàm analogReference(). Ngoài ra, một sô chân có chức năng chuyên biệt:

- TWI: A4 (SDA) và A5 (SCL). Hỗ trợ truyền thông TWI sử dụng thư viện Wire. Còn có hai chân khác trên bo mạch là:

- AREF: Điện áp tham chiếu cho đầu vào tương tự. Sử dụng với hàm analogReference().

- Reset. Được dùng để thiết lập lại vi điều khiển. Thường sử dụng nút reset để hỗ trợ các khối trên bo mạch.

***Lập trình.***

Arduino Uno có thể được lập trình với các phần mềm Arduino. Chọn " Arduino Uno từ thanh công cụ Tools → Board menu (theo vi điều khiển trên bo mạch). Để biết chi tiết , xem tài liệu tham khảo và hướng dẫn. ATmega328 trên Arduino Uno đi kèm với một bộ nạp khởi động preburned cho phép tải code mới lên mà không cần sử dụng lập trình phần cứng bên ngoài. Nó giao tiếp bằng cách sử dụng giao thức ban đầu STK500 (tham chiếu, tập tin tiêu đề viết bằng ngôn ngữ C). Ta cũng có thể bỏ qua bộ nạp khởi động và chương trình vi điều khiển thông qua ICSP (In-Circuit Serial Programming). Ở các dòng ATmega16U2 (hoặc 8U2) mã nguồn phần mềm có sẵn. Các ATmega16U2/8U2 được nạp với một bộ nạp khởi động DFU, mà có thể được kích hoạt bằng cách:

- Trên thế hệ bo mạch thứ 1: kết nối jumper được hàn ở mặt sau của bo mạch và sau đó cài đặt lại 8U2.

- Trên thế hệ bo mạch thứ 2 hoặc cuối cùng: có một điện trở nối HWB 8U2/16U2 với đất, làm cho nó dễ dàng hơn trong việc đặt chế độ DFU. Sau đó có thể sử dụng phần mềm FLIP Atmel (Windows) hoặc lập trình DFU (Mac OS X và Linux) để tải một phần mềm mới. Hoặc có thể sử dụng header ISP lập trình ngoài (ghi đè lên các bộ nạp khởi động DFU)Trong hệ thống bãi đỗ xe, Arduino UNO đóng vai trò như một bộ xử lý trung tâm (central controller). Nó thực hiện các chức năng như:

#### Nhận tín hiệu từ cảm biến hồng ngoại để phát hiện xe đang đi vào/ra bãi.

#### Đọc dữ liệu từ thẻ RFID thông qua module RC522 và xác minh thông tin truy cập.

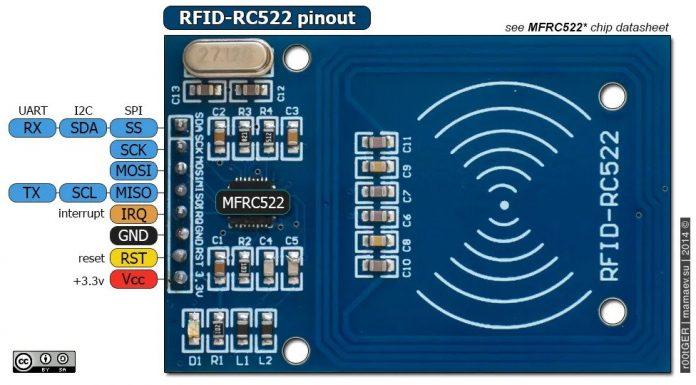
#### Điều khiển động cơ servo đóng/mở cổng barrier khi điều kiện hợp lệ.

#### Điều khiển còi buzzer phát ra âm thanh cảnh báo khi có sự cố hoặc truy cập trái phép.

#### Gửi dữ liệu lên màn hình LCD để hiển thị thông tin cần thiết cho người dùng.

#### Việc sử dụng Arduino trong hệ thống giúp tiết kiệm chi phí, dễ dàng nâng cấp và mở rộng các chức năng mới trong tương lai.

#### b) Module RFID RC522



Hình 4 : Module RFID

Module RFID RC522 sử dụng sóng radio để giao tiếp với các thẻ RFID, hoạt động trên tần số 13.56 MHz theo chuẩn ISO/IEC 14443. RC522 có thể đọc và ghi dữ liệu vào các loại thẻ MIFARE, với khoảng cách đọc hiệu quả từ 0 – 5 cm.

#### Thông số kỹ thuật:

| Thông số | Giá trị |
| --- | --- |
| Điện áp hoạt động | 3.3V |
| Giao tiếp | SPI (có thể lập trình lại cho UART/I2C) |
| Tần số hoạt động | 13.56 MHz |
| Chuẩn thẻ hỗ trợ | ISO/IEC 14443 A, MIFARE |
| Khoảng cách đọc | 0 – 5 cm |
| Tốc độ truyền dữ liệu | Lên đến 424 kbps |
| Kích thước module | 40mm x 60mm |

#### Chức năng trong hệ thống:

#### Nhận diện xe bằng UID thẻ RFID gắn trên phương tiện.

#### Kiểm soát quyền truy cập vào/ra khỏi bãi.

#### Tăng độ bảo mật hệ thống khi chỉ cho phép xe đã đăng ký trước.

#### Quy trình hoạt động:

#### Người dùng đưa thẻ RFID đến gần đầu đọc.

#### Module phát sóng RF để kích hoạt thẻ.

#### Thẻ gửi mã UID về cho module → truyền tiếp về Arduino.

#### Arduino so sánh → xử lý mở cổng hoặc từ chối.

#### Ưu điểm:

#### Độ chính xác cao, không cần tiếp xúc vật lý.

#### Dễ dàng tích hợp với vi điều khiển nhờ giao tiếp SPI.

#### Tốc độ nhận dữ liệu nhanh, tiêu tốn năng lượng thấp

#### LED

#### *A group of leds with different colors Description automatically generated*

*Hình 5 : LED đơn*

LED được viết tắt từ light Light-Emitting-Diode, có nghĩa là điốt phát quang. Về bản chất LED là một điốt, nó chứa một chip bán dẫn có pha các tạp chất để tạo ra một tiếp giáp P-N, kênh P chứa lỗ trống, kênh N chứa điện tử, dòng điện truyền từ A-nốt (kênh P) đến K-tốt (kênh N). Khi điện tử lấp đầy chỗ trống nó sinh ra bức xạ ánh sáng nhìn thấy.

Do cấu tạo của các chất bán dẫn khác nhau mà tạo ra ánh sáng có bước sóng khác nhau. Hay nói cách khác là tạo ra ánh sáng đơn sắc có màu khác nhau A diagram of a circuit

Description automatically generated

*Hình 6 : cấu tạo của LED*

#### Nguyên lý hoạt động của đèn LED dựa trên hiệu ứng quang điện được gọi là tái phân tử một chất bán dẫn. Đèn LED bao gồm một hoặc nhiều lớp chất bán dẫn, thường là các hợp chất như gallium arsenide (GaAs) hoặc gallium nitride (GaN). Khi áp dụng điện áp vào đèn LED, electron và lỗ trống (hole) di chuyển trong chất bán dẫn và kết hợp với nhau. Trong quá trình này, năng lượng được giải phóng dưới dạng ánh sáng.

**Diagram of a diagram of a band

Description automatically generated**

*Hình 7 : Nguyên lí hoạt động của LED*

#### d) Cảm biến hồng ngoại (IR Sensor)

#### Cảm Biến Vật Cản Hồng Ngoại

*Hình 8 : Cảm biến hồng ngoại*

#### Cảm biến hồng ngoại được sử dụng để phát hiện vật thể bằng cách đo phản xạ của tia hồng ngoại từ bề mặt vật thể quay trở lại photodiode thu.

#### Thông số kỹ thuật cơ bản:

| Thông số | Giá trị |
| --- | --- |
| Nguồn hoạt động | 3.3V – 5V |
| Khoảng cách phát hiện | 2 – 30 cm (tùy loại) |
| Tín hiệu đầu ra | Digital (0 hoặc 1) |
| Dòng tiêu thụ | ~20 mA |
| Độ nhạy | Điều chỉnh được bằng chiết áp |

#### Chức năng trong hệ thống:

#### Phát hiện xe tiến vào vùng quét thẻ.

#### Cập nhật trạng thái xe đã vào hoặc ra khỏi bãi.

#### Là điều kiện khởi động cho các quy trình xử lý tiếp theo (như kiểm tra thẻ).

#### Ưu điểm:

#### Phản hồi nhanh, độ chính xác cao ở khoảng cách ngắn.

#### Hoạt động ổn định trong nhiều điều kiện ánh sáng.

#### Không bị ảnh hưởng bởi vật liệu không phản xạ ánh sáng.

#### e) Màn hình LCD

**LCD 16x2**

LCD 16x2 là một loại màn hình tinh thể lỏng rất phổ biến trong các dự án nhúng và vi điều khiển nhờ khả năng hiển thị linh hoạt và hiệu quả. Màn hình này cho phép hiển thị 2 dòng, mỗi dòng gồm 16 ký tự. Nó có khả năng hiển thị nhiều loại ký tự bao gồm chữ cái, chữ số, ký hiệu đặc biệt và một số ký tự đồ họa đơn giản. Chính nhờ ưu điểm trực quan, dễ sử dụng, tiết kiệm tài nguyên hệ thống và có giá thành rẻ, LCD 16x2 được ứng dụng rộng rãi trong nhiều hệ thống điều khiển nhúng như máy đo, bộ điều khiển tự động, và trong đồ án này là hệ thống bãi đỗ xe thông minh.

**Thông số kỹ thuật**

**Điện áp hoạt động**: từ 2.7V đến 5V DC, phù hợp với nhiều loại vi điều khiển khác nhau, bao gồm Arduino.

**Dòng tiêu thụ**: khoảng 350µA đến 600µA, rất thấp nên phù hợp với hệ thống tiêu thụ điện năng thấp.

**Nhiệt độ hoạt động**: từ -30°C đến 75°C, đảm bảo hoạt động ổn định trong môi trường ngoài trời hoặc nhà xe.

**Cấu trúc chân kết nối**: LCD 16x2 có tổng cộng 16 chân:

**8 chân dữ liệu (D0 – D7)**: dùng để truyền dữ liệu từ vi điều khiển đến LCD.

**3 chân điều khiển (RS, RW, EN)**:

**RS (Register Select)**: chọn giữa chế độ lệnh (command) và chế độ dữ liệu.

**RW (Read/Write)**: chọn chế độ đọc hoặc ghi dữ liệu (thường nối GND để chỉ ghi).

**EN (Enable)**: kích hoạt việc ghi dữ liệu vào LCD.

**5 chân còn lại**: bao gồm VSS (GND), VDD (nguồn), VO (điều chỉnh độ tương phản), Anode và Cathode cho đèn nền.

**Chế độ hoạt động**

LCD 16x2 có thể hoạt động ở hai chế độ: **8-bit** và **4-bit**:

**Chế độ 8-bit** sử dụng cả 8 chân dữ liệu (D0-D7), truyền dữ liệu nhanh hơn nhưng tốn nhiều chân kết nối trên vi điều khiển.

**Chế độ 4-bit** chỉ sử dụng 4 chân dữ liệu (D4–D7), tiết kiệm được số lượng chân I/O nhưng cần truyền dữ liệu làm hai lần (2 nửa byte).

Trong ứng dụng hệ thống bãi đỗ xe thông minh này, LCD 16x2 thường được cấu hình ở **chế độ 4-bit** để tiết kiệm tài nguyên chân I/O của Arduino, đặc biệt khi hệ thống còn phải kết nối với nhiều module khác như RFID, cảm biến, servo, buzzer,…

**Tính ứng dụng trong hệ thống**

LCD 16x2 đóng vai trò là thiết bị **hiển thị thông tin chính** cho người dùng và quản trị hệ thống. Nó hiển thị các trạng thái như:

"Xe đã vào/ra bãi"

"Bãi đỗ đầy"

“Số lượng chỗ còn trống”

Nhờ giao diện đơn giản và khả năng cập nhật thông tin nhanh chóng, LCD 16x2 giúp người dùng dễ dàng tương tác và giám sát tình trạng của hệ thống bãi đỗ xe.

A close-up of a green electronic device

Description automatically generated

*Hình 9: Màn hình LCD*

#### Module I2C

#### LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển.

#### Module I2C LCD ra đời và giải quyết vấn để này.

#### Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16×2 (RS, EN, D7, D6, D5 và D4) thì module I2C bạn chỉ cần tốn 2 chân (SCL, SDA) để kết nối.

#### Module I2C hỗ trợ các loại LCD sử dụng driver HD44780 (LCD 16×2, LCD 20×4, …) và tương thích với hầu hết các vi điều khiển hiện nay.

#### Ưu điểm:

#### Tiết kiệm chân cho vi điều khiển.

#### Dễ dàng kết nối với LCD.

#### Thông số kỹ thuật:

#### Điện áp hoạt động: 2.5-6V DC.

#### Hỗ trợ màn hình: LCD 1602, 1604, 2004 (driver HD44780).

#### Giao tiếp: I2C. Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).

#### Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt.

#### Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

#### A close-up of a microchip Description automatically generated

#### *Hình 10 : Module I2C*

# 

**2.2: YÊU CẦU KĨ THUẬT VÀ CHỨC NĂNG CỦA HỆ THỐNG**

**2.2.1: Yêu cầu kỹ thuật**

Bo mạch điều khiển:

* + Sử dụng Arduino Uno để điều khiển toàn bộ hệ thống.
  + Đảm bảo có đủ chân GPIO để kết nối các thiết bị: RFID, LED, buzzer, servo, LCD và cảm biến hồng ngoại.

RFID:

* + Dùng mô-đun RFID RC522 để nhận dạng thẻ.
  + Giao tiếp qua SPI, khoảng cách đọc từ 0–5 cm.

Màn hình LCD:

* + Dùng LCD 16x2 (giao tiếp I2C) để hiển thị thông tin trạng thái hệ thống như "Thẻ hợp lệ", "Cổng mở", "Hết chỗ"...

LED:

* + LED chỉ thị trạng thái:

Buzzer:

* + Buzzer dùng để cảnh báo bằng âm thanh khi có thẻ không hợp lệ hoặc khi hệ thống có lỗi.

Servo:

* + Servo motor SG90 hoặc tương đương để mở/đóng cổng tự động khi thẻ RFID hợp lệ.

Cảm biến hồng ngoại (IR sensor):

* + Dùng để phát hiện xe ra vào hoặc kiểm tra xem có xe trong bãi hay không.
  + Giúp đếm số lượng xe hiện có để xác định bãi còn trống hay đã đầy.
  + Có thể dùng cảm biến IR obstacle avoidance hoặc IR reflective sensor.

**2.2.2: Chức năng của hệ thống**

Xác thực thẻ RFID:

* + Người dùng quét thẻ RFID.
  + Nếu hợp lệ: hiển thị thông báo, mở cổng bằng servo.
  + Nếu không hợp lệ: buzzer cảnh báo và từ chối truy cập.

Hiển thị thông tin bằng LCD:

* + LCD hiển thị trạng thái như: “Cổng mở”, “Thẻ hợp lệ”, “Hết chỗ”, “Xe vào”...

Điều khiển cổng bằng Servo:

* + Mở cổng khi thẻ hợp lệ và bãi còn trống.
  + Cổng tự đóng sau một khoảng thời gian hoặc sau khi xe đi qua (dựa trên cảm biến IR).

Báo hiệu bằng LED:

Cảnh báo bằng Buzzer:

* + Cảnh báo âm thanh khi có lỗi hoặc truy cập trái phép.

Cảm biến hồng ngoại (IR sensor):

* + Phát hiện xe đi vào hoặc đi ra.
  + Tăng/giảm số lượng xe trong bãi.
  + Dữ liệu dùng để xác định bãi đã đầy hay còn trống.

# CHƯƠNG III : XÂY DỰNG VÀ TRIỂN KHAI

## 3.1: Mô phỏng nguyên lí mạch

A diagram of a circuit board

AI-generated content may be incorrect.

*Hình 11: Mạch mô phỏng*

**Mục tiêu hệ thống:**

Quản lý xe ra vào bằng thẻ RFID.

Phát hiện xe đậu bằng cảm biến hồng ngoại.

Hiển thị trạng thái các vị trí đỗ và số chỗ trống trên LCD.

Điều khiển cổng vào/ra bằng servo.

Cảnh báo bằng buzzer khi bãi đầy.

| **Linh kiện** | **Chức năng chính** |
| --- | --- |
| Arduino Uno | Bộ điều khiển trung tâm |
| RFID RC522 | Đọc thẻ RFID để xác nhận xe |
| Servo motor | Mở và đóng thanh chắn |
| LCD 16x2 (I2C) | Hiển thị thông tin (số chỗ, trạng thái) |
| Cảm biến I3 | Phát hiện xe tại từng vị trí đỗ (S1, S2, S3) |
| Buzzer | Cảnh báo khi bãi đầy |
| LED | Nhấp nháy báo hiệu hệ thống hoạt động |

Hệ thống bãi đỗ xe thông minh được thiết kế nhằm tự động hóa quá trình quản lý xe ra vào một cách hiệu quả, nhanh chóng và chính xác. Ngay khi được cấp nguồn, hệ thống sẽ khởi động và hiển thị tên hệ thống lên màn hình LCD, đồng thời đèn LED sẽ chớp tắt liên tục để thông báo rằng hệ thống đang hoạt động bình thường. Servo motor được sử dụng như một thanh chắn tự động, ban đầu ở trạng thái đóng để kiểm soát cổng vào/ra.

Hệ thống sử dụng ba cảm biến hồng ngoại được bố trí tại ba vị trí đỗ xe nhằm giám sát trạng thái thực tế của từng chỗ đỗ. Khi không có vật cản, cảm biến sẽ phản hồi tín hiệu cho biết chỗ đỗ đang trống và được hiển thị trên màn hình dưới dạng ký hiệu “V”. Ngược lại, khi có xe đỗ tại vị trí đó, cảm biến phát hiện có vật cản và hiển thị “X” tương ứng. Trạng thái của từng chỗ đỗ sẽ được cập nhật thường xuyên và hiển thị liên tục trên màn hình LCD giúp người dùng dễ dàng quan sát và lựa chọn vị trí đỗ phù hợp.

Khi một phương tiện muốn vào bãi, người dùng sẽ quét thẻ RFID tại đầu vào. Hệ thống sẽ đọc và kiểm tra mã thẻ. Nếu đây là thẻ mới (chưa có trong danh sách các xe đang đỗ) và bãi vẫn còn chỗ trống, hệ thống sẽ chấp nhận cho xe vào bằng cách lưu thông tin thẻ, hiển thị thông báo “Welcome!” trên màn hình, đồng thời kích hoạt servo để mở cổng. Còi buzzer sẽ phát ra hai tiếng ngắn để xác nhận thao tác thành công. Sau vài giây, servo sẽ tự động đóng lại để đảm bảo an toàn và kiểm soát. Trường hợp bãi đã đầy, khi người dùng quét thẻ, hệ thống sẽ từ chối quyền vào bãi, đồng thời hiển thị thông báo “Full slot!” và kích hoạt còi kêu dài để cảnh báo người dùng không được vào.

Tương tự, khi xe muốn rời khỏi bãi, người dùng cần quét lại đúng thẻ đã dùng trước đó. Nếu thẻ được xác nhận là đã từng đăng ký, hệ thống sẽ xóa thông tin thẻ khỏi danh sách xe đang đỗ, mở cổng cho xe ra và phát ra hai tiếng còi ngắn, đồng thời hiển thị lời chào tạm biệt cùng số chỗ trống mới nhất trên màn hình. Sau một khoảng thời gian ngắn, servo sẽ đóng lại để sẵn sàng cho lượt tiếp theo.

Ngoài ra, hệ thống còn tích hợp một còi buzzer đóng vai trò cảnh báo âm thanh. Buzzer kêu hai lần ngắn khi có xe vào hoặc ra thành công và phát ra một tiếng dài khi người dùng cố gắng vào lúc bãi đã đầy. Hệ thống LED nhấp nháy liên tục trong suốt quá trình vận hành giúp người dùng biết rằng hệ thống đang trong trạng thái hoạt động tốt.

Tổng thể, hệ thống bãi đỗ xe thông minh này là sự kết hợp giữa các công nghệ như RFID, cảm biến hồng ngoại, LCD I2C, servo motor và cảnh báo âm thanh nhằm tạo nên một giải pháp hiện đại, tiết kiệm nhân lực, nâng cao trải nghiệm người dùng và giúp quản lý bãi đỗ xe một cách tự động, hiệu quả và an toàn.

**Hoạt động của buzzer:**

| Tình huống | Hành động buzzer |
| --- | --- |
| Xe vào/ra thành công | Buzzer kêu 2 tiếng ngắn |
| Bãi đầy khi quẹt thẻ vào | Buzzer kêu dài 1.5 giây |

## 3.2: Phân tích Code

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <SPI.h>

#include <MFRC522.h>

#include <Servo.h>

#define SS\_PIN 10

#define RST\_PIN 9

#define SERVO\_PIN 2

#define BUZZER\_PIN A0

#define LED\_PIN 7

#define IR\_SENSOR\_1 4

#define IR\_SENSOR\_2 5

#define IR\_SENSOR\_3 6

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

Servo servo;

LiquidCrystal\_I2C lcd(0x27, 16, 2);

int availableSlots = 3;

String parkedRFIDs[3] = {"", "", ""};

void setup() {

Serial.begin(9600);

SPI.begin();

mfrc522.PCD\_Init();

servo.attach(SERVO\_PIN);

servo.write(0);

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(LED\_PIN, OUTPUT);

pinMode(IR\_SENSOR\_1, INPUT);

pinMode(IR\_SENSOR\_2, INPUT);

pinMode(IR\_SENSOR\_3, INPUT);

pinMode(LED\_PIN, OUTPUT);

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("VKU SmartParking");

delay(2000);

displayParkingStatus();

}

void loop() {

displayParkingStatus();

if (mfrc522.PICC\_IsNewCardPresent() && mfrc522.PICC\_ReadCardSerial()) {

String rfid = getRFID();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("RFID: " + rfid);

int index = findRFID(rfid);

if (index != -1) {

parkedRFIDs[index] = "";

availableSlots = min(availableSlots + 1, 3);

servo.write(90);

lcd.setCursor(2, 1);

lcd.print("See you again!");

buzzBuzzerTwice();

delay(2000);

displayAvailableSlots();

servo.write(0);

}

else if (availableSlots > 0) {

int emptyIndex = findEmptySlot();

if (emptyIndex != -1) {

parkedRFIDs[emptyIndex] = rfid;

availableSlots = max(availableSlots - 1, 0);

servo.write(90);

lcd.setCursor(4, 1);

lcd.print("Welcome!");

buzzBuzzerTwice();

delay(2000);

displayAvailableSlots();

servo.write(0);

}

}

else {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Full slot!");

buzzBuzzerLong();

delay(2000);

}

}

delay(500);

digitalWrite(LED\_PIN,HIGH);

delay(500);

digitalWrite(LED\_PIN,LOW);

delay(500);

}

void displayParkingStatus() {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("VKU SmartParking");

lcd.setCursor(0, 1);

lcd.print("S1:" + String(digitalRead(IR\_SENSOR\_1) == LOW ? "X " : "V "));

lcd.print("S2:" + String(digitalRead(IR\_SENSOR\_2) == LOW ? "X " : "V "));

lcd.print("S3:" + String(digitalRead(IR\_SENSOR\_3) == LOW ? "X " : "V "));

delay(2000);

}

void displayAvailableSlots() {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Available: " + String(availableSlots) );

delay(2000);

}

int findRFID(String rfid) {

for (int i = 0; i < 3; i++) {

if (parkedRFIDs[i] == rfid) {

return i;

}

}

return -1;

}

int findEmptySlot() {

for (int i = 0; i < 3; i++) {

if (parkedRFIDs[i] == "") {

return i;

}

}

return -1;

}

String getRFID() {

String rfid = "";

for (byte i = 0; i < mfrc522.uid.size; i++) {

rfid.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : ""));

rfid.concat(String(mfrc522.uid.uidByte[i], HEX));

}

mfrc522.PICC\_HaltA();

return rfid;

}

void buzzBuzzer(unsigned int duration) {

digitalWrite(BUZZER\_PIN, HIGH);

delay(duration);

digitalWrite(BUZZER\_PIN, LOW);

}

void buzzBuzzerTwice() {

for (int i = 0; i < 2; i++) {

digitalWrite(BUZZER\_PIN, HIGH);

delay(100);

digitalWrite(BUZZER\_PIN, LOW);

delay(300);

}

}

void buzzBuzzerLong() {

digitalWrite(BUZZER\_PIN, HIGH);

delay(1500); // Kêu 1.5 giây

digitalWrite(BUZZER\_PIN, LOW);

}

**Thư viện và định nghĩa phần cứng**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <SPI.h>

#include <MFRC522.h>

#include <Servo.h>

Wire.h và LiquidCrystal\_I2C.h: hỗ trợ giao tiếp với màn hình LCD I2C để hiển thị thông tin.

SPI.h và MFRC522.h: cho phép giao tiếp với mô-đun đọc thẻ RFID.

Servo.h: điều khiển servo đóng/mở cổng bãi xe.

#define SS\_PIN 10

#define RST\_PIN 9

#define SERVO\_PIN 2

#define BUZZER\_PIN A0

#define LED\_PIN 7

#define IR\_SENSOR\_1 4

#define IR\_SENSOR\_2 5

#define IR\_SENSOR\_3 6

Gán các chân kết nối phần cứng: RFID, servo, buzzer, đèn LED và cảm biến hồng ngoại (3 vị trí bãi xe).

**Khai báo đối tượng và biến**

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

Servo servo;

LiquidCrystal\_I2C lcd(0x27, 16, 2);

int availableSlots = 3;

String parkedRFIDs[3] = {"", "", ""};

**MFRC522**: đọc thẻ từ.

**Servo**: điều khiển thanh chắn cổng.

**LCD**: hiển thị thông tin bãi đỗ.

**availableSlots**: số chỗ trống còn lại (ban đầu là 3).

**parkedRFIDs[]**: mảng lưu RFID của các xe đã vào bãi.

**Hàm setup() – Cấu hình ban đầu**

void setup() {

...

}

Khởi tạo kết nối Serial, SPI, màn hình LCD, servo và cảm biến.

Cài đặt chế độ cho các chân I/O.

Hiển thị tên hệ thống trên LCD trong 2 giây.

Gọi hàm displayParkingStatus() để hiển thị trạng thái ban đầu của bãi đỗ.

**Hàm loop() – Vòng lặp chính**

void loop() {

displayParkingStatus();

...

digitalWrite(LED\_PIN,HIGH);

delay(500);

digitalWrite(LED\_PIN,LOW);

delay(500);

}

Luôn hiển thị trạng thái của ba chỗ đỗ bằng cảm biến hồng ngoại.

Khi người dùng quẹt thẻ, hệ thống đọc mã RFID:

Nếu xe đã đỗ (đã có trong mảng): xe rời → xóa ID, tăng số chỗ trống, mở cổng, còi kêu, hiện lời chào tạm biệt.

Nếu là xe mới và còn chỗ → lưu ID, giảm chỗ trống, mở cổng, còi kêu, hiện “Welcome!”.

Nếu đầy → hiển thị “Full slot!” và còi kêu dài cảnh báo.

Cuối cùng, LED chớp nháy để báo hiệu hệ thống đang hoạt động.

**Hàm displayParkingStatus() – Hiển thị trạng thái chỗ đỗ**

void displayParkingStatus() {

lcd.print("S1:X/V S2:X/V S3:X/V");

}

Dựa vào tín hiệu từ cảm biến hồng ngoại, trạng thái từng chỗ (trống “V” hoặc đầy “X”) sẽ được hiển thị trên LCD.

**Hàm displayAvailableSlots() – Hiển thị số chỗ trống**

void displayAvailableSlots() {

lcd.print("Available: x");

}

Hiển thị số lượng chỗ trống còn lại sau khi xe vào hoặc ra.

**Hàm tìm kiếm RFID**

int findRFID(String rfid) {...}

int findEmptySlot() {...}

**findRFID()**: trả về vị trí thẻ trong danh sách nếu xe đã đỗ.

**findEmptySlot()**: tìm vị trí còn trống để thêm thẻ mới.

**Hàm getRFID() – Đọc mã thẻ RFID**

String getRFID() {

...

}

Ghép từng byte của mã UID thành chuỗi hex để xác định ID duy nhất của thẻ.

**Hàm điều khiển còi buzzer**

void buzzBuzzer(), buzzBuzzerTwice(), buzzBuzzerLong() {...}

**buzzBuzzer(duration)**: kêu với thời gian tùy chọn.

**buzzBuzzerTwice()**: phát hai tiếng ngắn khi xe vào/ra thành công.

**buzzBuzzerLong()**: kêu dài cảnh báo khi bãi đầy.

## 3.3: Mạch hoàn thiện thực tế

A close up of a device

AI-generated content may be incorrect.

Hình 12 : Mạch thực tế

# CHƯƠNG IV : NHẬN XÉT VÀ ĐÁNH GIÁ

# *4.1. Kết quả nghiên cứu*

# Sau một quá trình tìm hiểu, thiết kế và lắp đặt mô hình, nhóm chúng em đã xây dựng thành công hệ thống bãi đỗ xe thông minh ứng dụng các công nghệ cơ bản nhưng hiệu quả như vi điều khiển Arduino Uno R3, cảm biến hồng ngoại, màn hình LCD I2C, đầu đọc thẻ RFID RC522, động cơ servo và còi báo buzzer. Mô hình mô phỏng được quá trình xe vào và ra khỏi bãi đỗ một cách tự động và chính xác, đồng thời giúp giám sát tình trạng chỗ đỗ trong thời gian thực.

# Kết quả nghiên cứu cho thấy mô hình hoạt động ổn định và đạt được nhiều tiêu chí đã đề ra. Cụ thể:

# Ứng dụng công nghệ RFID để xác thực phương tiện ra vào: Mỗi phương tiện được cấp một thẻ từ riêng biệt. Khi người dùng quét thẻ lên đầu đọc RFID RC522, vi điều khiển Arduino sẽ kiểm tra mã thẻ, so sánh với danh sách hợp lệ. Nếu thẻ hợp lệ và hệ thống còn chỗ trống, hệ thống sẽ cho phép mở cổng. Ngược lại, nếu thẻ không hợp lệ hoặc bãi đã đầy, hệ thống sẽ thông báo từ chối truy cập.

# Tự động điều khiển cổng bằng động cơ servo: Sau khi xác thực thẻ, Arduino điều khiển động cơ servo để mở thanh chắn, cho phép xe vào hoặc ra. Sau một khoảng thời gian định trước, thanh chắn sẽ tự động đóng lại, đảm bảo việc kiểm soát phương tiện ra vào được chặt chẽ.

# Hiển thị thông tin trạng thái qua màn hình LCD I2C: Màn hình LCD I2C giúp hiển thị các thông báo quan trọng như: “Thẻ hợp lệ”, “Thẻ không hợp lệ”, “Bãi đầy”, số lượng chỗ trống còn lại,... Nhờ đó người sử dụng có thể dễ dàng quan sát và tương tác với hệ thống.

# Giám sát vị trí đỗ xe bằng cảm biến hồng ngoại: Tại mỗi chỗ đỗ xe, một cảm biến hồng ngoại được gắn để phát hiện sự hiện diện của xe. Dữ liệu từ các cảm biến này được tổng hợp lại và hiển thị trên màn hình LCD, cho phép người quản lý dễ dàng theo dõi số lượng chỗ trống theo thời gian thực. Hệ thống sẽ không cho phép xe mới vào khi tất cả các cảm biến báo là đã có xe.

# Cảnh báo bằng còi buzzer: Còi buzzer được sử dụng để phản hồi nhanh cho người dùng. Ví dụ, khi quét thẻ thành công, còi sẽ phát ra tiếng “bíp” ngắn; nếu thẻ sai hoặc bãi đầy, còi sẽ phát âm báo dài hơn để cảnh báo.

# Từ quá trình thiết kế và thử nghiệm mô hình, nhóm chúng em đã học được cách tích hợp nhiều loại linh kiện điện tử khác nhau thành một hệ thống tự động hóa thống nhất. Việc phối hợp giữa phần cứng và phần mềm (lập trình Arduino) giúp nhóm hiểu rõ hơn về cách xử lý tín hiệu, điều khiển thiết bị ngoại vi, cũng như tổ chức logic hoạt động cho một hệ thống hoàn chỉnh. Đây là cơ sở quan trọng để áp dụng vào các dự án lớn hơn hoặc triển khai thực tế trong tương lai.

# Ngoài ra, mô hình bãi đỗ xe thông minh còn cho thấy tiềm năng lớn trong ứng dụng thực tế, đặc biệt trong bối cảnh đô thị hoá và số lượng phương tiện cá nhân ngày càng tăng nhanh. Việc triển khai hệ thống như vậy trong các khu dân cư, trường học, trung tâm thương mại hoặc tòa nhà văn phòng sẽ giúp giảm thiểu tình trạng kẹt xe, nâng cao hiệu quả quản lý và tiết kiệm thời gian cho người sử dụng.

# *4.2. Hạn chế và hướng phát triển trong tương lai*

# *Hạn chế của đề tài*

# Mặc dù đã đạt được nhiều kết quả tích cực, đề tài vẫn tồn tại một số hạn chế do điều kiện về thời gian, kinh phí và trình độ kỹ thuật:

# Mô hình chỉ ở mức quy mô nhỏ: Số lượng chỗ đỗ xe trong mô hình chỉ giới hạn từ 2 đến 4 vị trí, chưa thể mô phỏng đầy đủ cho các bãi xe lớn với hàng chục hoặc hàng trăm chỗ. Điều này làm giới hạn khả năng đánh giá hiệu suất của hệ thống trong điều kiện tải cao.

# Thiếu khả năng lưu trữ và xử lý dữ liệu nâng cao: Hiện tại, hệ thống chưa tích hợp chức năng lưu trữ dữ liệu ra/vào hoặc nhật ký sử dụng thẻ. Điều này gây khó khăn trong việc kiểm tra lại lịch sử hoạt động hoặc xử lý các trường hợp tranh chấp.

# Chưa có kết nối mạng: Hệ thống hoạt động hoàn toàn độc lập, không có kết nối với mạng Internet hoặc giao diện quản trị từ xa. Do đó, người quản lý phải quan sát trực tiếp màn hình LCD thay vì truy cập từ xa qua ứng dụng hoặc trang web.

# Khả năng mở rộng còn hạn chế: Nếu muốn tăng số lượng chỗ đỗ, hệ thống cần phải được lập trình lại để đọc nhiều cảm biến hơn và xử lý dữ liệu phức tạp hơn, điều này đòi hỏi kiến thức nâng cao về lập trình Arduino và phần cứng mở rộng.

# *Hướng phát triển trong tương lai*

# Để phát triển hệ thống bãi đỗ xe thông minh theo hướng thực tiễn và chuyên nghiệp hơn, nhóm đề xuất một số hướng mở rộng như sau:

# Tích hợp lưu trữ và xử lý dữ liệu: Sử dụng module thẻ nhớ SD hoặc kết nối với hệ thống máy chủ để lưu thông tin các xe ra vào, thời gian đỗ xe,... Từ đó có thể triển khai thêm tính năng tính phí, báo cáo thống kê, hoặc phân tích hành vi sử dụng.

# Kết nối với Internet và điều khiển từ xa (IoT): Ứng dụng công nghệ Internet of Things để người dùng có thể kiểm tra số chỗ trống còn lại ngay trên điện thoại qua ứng dụng hoặc trang web. Quản trị viên cũng có thể quản lý hệ thống từ xa, nhận cảnh báo khi có sự cố hoặc truy cập lịch sử hoạt động.

# Tối ưu độ chính xác của cảm biến hồng ngoại: Nâng cấp loại cảm biến để hoạt động ổn định hơn trong mọi điều kiện ánh sáng, môi trường nhiều bụi, hoặc dùng kết hợp với các loại cảm biến khác như siêu âm để tăng độ chính xác.

# Bổ sung các tính năng nâng cao: Như nhận diện biển số xe bằng camera, kết hợp AI để phân tích thời gian đỗ, gợi ý vị trí còn trống, hoặc triển khai tính năng mở cổng tự động mà không cần thẻ nếu đã đăng ký từ trước.

# Mở rộng quy mô mô hình: Xây dựng mô hình với số lượng chỗ đỗ nhiều hơn, bố trí trong không gian thực tế như sân trường, bãi gửi xe công cộng, nhằm đánh giá khả năng vận hành của hệ thống ở quy mô lớn.

# Bền vững và tiết kiệm năng lượng: Kết hợp hệ thống với nguồn năng lượng mặt trời, sử dụng các linh kiện tiết kiệm điện để đảm bảo vận hành liên tục trong môi trường ngoài trời và giảm chi phí sử dụng điện năng.

# Tài liệu tham khảo

**C Embedded:**

Books: *C Programming for Embedded Systems, Making Embedded Systems, Embedded Systems Architecture.*

**Books:**

* Arduino Cookbook, Michael Margolis.
* Programming Arduino: Getting Started with Sketches, Simon Monk.
* Exploring Arduino, Jeremy Blum.

**[1]** Arduino UNO Reference Manual.  
**[2]** Arduino IDE Documentation.  
**[3]** Instructables Arduino Projects.  
**[4]** All About Circuits: <https://www.allaboutcircuits.com/>  
**[5]** Arduino Official Website: <https://www.arduino.cc/>  
**[6]** Hackster.io Arduino Projects: https://www.hackster.io/arduino