**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ THỐNG NHÚNG**

**ĐỀ TÀI: BÃI ĐỖ XE THÔNG MINH**

**Giáo viên hướng dẫn: ThS Nguyễn Văn Nhân**

**ThS Thái Bảo**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Mã sv** | **Họ và tên** | **Lớp** |
| 1 | 1771020477 | Ngô Văn Minh | CNTT 17-01 |
| 2 | 1771020051 | Nguyễn Trung Anh | CNTT 17-01 |
| 3 | 1771020387 | Lý Gia Khánh | CNTT 17-01 |
| 4 | 1771020251 | Vũ Văn Hậu | CNTT 17-01 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: HỆ THỐNG NHÚNG**

**ĐỀ TÀI: BÃI ĐỖ XE THÔNG MINH**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771020477 | Ngô Văn Minh | 17/06/2005 |  |  |
| 2 | 1771020051 | Nguyễn Trung Anh | 30/03/2005 |  |  |
| 3 | 1771020387 | Lý Gia Khánh | 18/02/2005 |  |  |
| 4 | 1771020251 | Vũ Văn Hậu | 10/08/2005 |  |  |

***CÁN BỘ CHẤM THI***

**Hà Nội, năm 2025**

# LỜI NÓI ĐẦU

Việc quản lý bãi đỗ xe đang trở thành yêu cầu quan trọng đối với nhiều khu vực như trường học, bệnh viện, trung tâm thương mại và khu dân cư. Các phương pháp quản lý truyền thống thường gây mất thời gian, khó kiểm soát và dễ xảy ra sai sót khi lượng xe ra vào tăng cao. Điều này làm giảm hiệu quả vận hành và gây bất tiện cho người sử dụng. Vì vậy, cần có một giải pháp hiện đại hơn để tối ưu hoạt động quản lý bãi đỗ xe.

Hệ thống nhúng mang đến giải pháp tự động hóa linh hoạt và có tính ứng dụng cao trong việc xây dựng các mô hình điều khiển thông minh. Khi áp dụng vào bãi đỗ xe, hệ thống nhúng giúp theo dõi số lượng xe, nhận biết xe vào – ra và hỗ trợ hiển thị thông tin cần thiết một cách trực quan. Nhờ đó, việc quản lý trở nên chính xác hơn và giảm bớt sự phụ thuộc vào nhân lực. Điều này góp phần nâng cao chất lượng dịch vụ và cải thiện trải nghiệm của người dùng.

Đề tài “Bãi đỗ xe thông minh” được thực hiện nhằm xây dựng một mô hình minh họa cho quá trình ứng dụng hệ thống nhúng vào quản lý thực tế. Thông qua quá trình thiết kế và triển khai, sinh viên có cơ hội củng cố kiến thức, rèn luyện kỹ năng lập trình và hiểu rõ hơn về cách một hệ thống tự động hoạt động. Đề tài cũng giúp sinh viên phát triển tư duy kỹ thuật và khả năng giải quyết vấn đề hiệu quả hơn.

**MỤC LỤC**

[LỜI NÓI ĐẦU 3](#_Toc214269628)

[CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG: 8](#_Toc214269629)

[1.1. Giới thiệu đề tài và bài toán thực tế: 8](#_Toc214269630)

[1.2. Mục tiêu và yêu cầu của hệ thống: 8](#_Toc214269631)

[1.3. Phạm vi thực hiện dự án: 10](#_Toc214269632)

[CHƯƠNG 2. THIẾT KẾ HỆ THỐNG: 12](#_Toc214269633)

[2.1. Phân tích và lựa chọn phần cứng: 12](#_Toc214269634)

[2.1.1. Vi điều khiển Arduino: 12](#_Toc214269635)

[2.1.2. Cảm biến hồng ngoại IR: 13](#_Toc214269636)

[2.1.3. Động cơ servo SG90: 14](#_Toc214269637)

[2.1.4. Màn hình LCD 16x2 I2C: 15](#_Toc214269638)

[2.2. Thiết kế sơ đồ khối hệ thống: 17](#_Toc214269639)

[2.3. Thiết kế sơ đồ mạch điện: 19](#_Toc214269640)

[2.4. Thiết kế thuật toán và lưu đồ chương trình: 21](#_Toc214269641)

[CHƯƠNG 3. TRIỂN KHAI VÀ KẾT QUẢ: 25](#_Toc214269642)

[3.1. Lập trình và cài đặt hệ thống: 25](#_Toc214269643)

[3.2. Lắp ráp mạch và kiểm tra hoạt động: 32](#_Toc214269644)

[3.3. Chạy thử nghiệm và đánh giá: 35](#_Toc214269645)

[3.3.1. Chạy thử nghiệm: 35](#_Toc214269646)

[3.3.2. đánh giá: 42](#_Toc214269647)

[KẾT LUẬN 45](#_Toc214269648)

[DANH MỤC TÀI LIỆU THAM KHẢO 46](#_Toc214269649)

**MỤC LỤC HÌNH ẢNH**

[Hình 1. Vi điều khiển Arduino 12](#_Toc214271551)

[Hình 2. Cảm biến hồng ngoại IR 13](#_Toc214271552)

[Hình 3. Động cơ servo SG90 14](#_Toc214271553)

[Hình 4. Màn hình LCD 16x2 I2C 16](#_Toc214271554)

[Hình 5. Sơ đồ khối hệ thống bãi đỗ xe thông minh 18](#_Toc214271555)

[Hình 6. Sơ đồ thuật toán điều khiển hệ thống bãi đỗ xe thông minh 22](#_Toc214271556)

[Hình 7. Sơ đồ trạng thái hệ thống xử lý xe vào 24](#_Toc214271557)

[Hình 8. Sơ đồ lắp đặt mạch bãi đỗ xe Arduino 33](#_Toc214271558)

[Hình 9. Mô hình hoạt động cổng ra vào tự động bãi đỗ xe 36](#_Toc214271559)

[Hình 10. Mô tả quá trình đóng thanh chắn tự động 37](#_Toc214271560)

[Hình 11. Mô hình hiển thị trạng thái đóng cổng và cập nhật dữ liệu 38](#_Toc214271561)

[Hình 12. Mô phỏng cơ chế từ chối xe vào khi bãi đỗ xe đầy 39](#_Toc214271562)

[Hình 13. Mô phỏng cơ chế mở thanh chắn tự động cho xe ra 40](#_Toc214271563)

[Hình 14. Mô phỏng cơ chế mở cổng tự động 41](#_Toc214271564)

[Hình 15. Hiển thị thông báo đóng cổng tự động 42](#_Toc214271565)

**MỤC LỤC BẢNG**

[Bảng 1. Chức năng và kết nối của các khối trong hệ thống bãi đỗ xe thông minh 18](#_Toc214272641)

[Bảng 2. Cấu hình chân và loại tín hiệu kết nối các linh kiện trong hệ thống 20](#_Toc214272642)

[Bảng 3. Chức năng các hàm trong chương trình điều khiển hệ thống 23](#_Toc214272643)

[Bảng 4. Trạng thái hệ thống bãi đỗ xe 23](#_Toc214272644)

[Bảng 5. Cấu hình chân và thiết bị trong hệ thống Parking Smart 26](#_Toc214272645)

[Bảng 6. Cấu hình chân và thiết bị servo 26](#_Toc214272646)

[Bảng 7. Giá trị cảm biến slot và ký tự hiển thị 27](#_Toc214272647)

[Bảng 8. Hiển thị LCD theo tình huống 27](#_Toc214272648)

[Bảng 9. Minh họa trạng thái cảm biến và số chỗ trống 28](#_Toc214272649)

[Bảng 10. Quy trình khởi tạo hệ thống Parking Smart 30](#_Toc214272650)

[Bảng 11. Hành động theo số lượng slot trống 31](#_Toc214272651)

[Bảng 12. Thông số điện áp – dòng tiêu thụ của các thiết bị 32](#_Toc214272652)

**BẢNG CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| **1** | **Iot** | **Internet of Things**  **(Internet vạn vật)** |
| **2** | **I2C** | **Inter-Integrated Circuit**  **(Giao tiếp nối tiếp nội bộ) hoặc**  **(Giao tiếp nối tiếp tích hợp)** |
| **3** | **IR** | **Infrared**  **(Hồng ngoại)** |
| **4** | **SG90** | **Servo SG90**  **(Động cơ servo SG90)** |
| **5** | **IDE** | **Integrated Development Environment**  **(Môi trường phát triển tích hợp)** |
| **6** | **MHz** | **(Megahertz) hoặc (triệu hertz)** |
| **7** | **KB** | **(Kilobyte) hoặc (nghìn byte)** |
| **8** | **SRAM** | **Static Random-Access Memory**  **(Bộ nhớ truy cập ngẫu nhiên tĩnh)** |
| **9** | **PWM** | **Pulse-Width Modulation**  **(Điều chế độ rộng xung)** |
| **10** | **SDA** | **Serial Data**  **(Dữ liệu nối tiếp)** |
| **11** | **GND** | **Ground**  **(Mass) hoặc (Mặt đất điện)** |
| **12** | **VCC** | **Voltage at the Common Collector**  **(Nguồn dương) hoặc (Cực dương)** |
| **13** | **DC** | **Direct Current**  **(Dòng điện một chiều)** |

# CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG:

## Giới thiệu đề tài và bài toán thực tế:

Trong bối cảnh đô thị hóa diễn ra nhanh chóng và số lượng phương tiện cá nhân gia tăng mạnh mẽ, bài toán quản lý bãi đỗ xe đang trở thành một thách thức nghiêm trọng đối với các thành phố lớn. Người lái xe thường phải mất nhiều thời gian lái vòng quanh bãi đỗ để tìm chỗ trống, gây lãng phí thời gian, nhiên liệu và làm tăng ùn tắc giao thông cục bộ. Các nghiên cứu cho thấy, trung bình một người lái xe có thể mất từ 10 đến 20 phút chỉ để tìm được chỗ đỗ xe phù hợp tại các khu vực đông đúc.

Về phía người quản lý bãi đỗ xe, các phương pháp quản lý truyền thống dựa vào nhân lực để quan sát và hướng dẫn xe vào vị trí đỗ không chỉ tốn kém chi phí vận hành mà còn dễ xảy ra sai sót trong việc thống kê số lượng xe ra vào. Việc thiếu thông tin trực quan về số chỗ trống còn lại khiến người lái xe không thể biết trước tình trạng bãi đỗ, dẫn đến tình trạng nhiều xe cùng lúc chen chúc trong bãi tìm chỗ đỗ, gây mất trật tự và tiềm ẩn nguy cơ va chạm.

Xuất phát từ những vấn đề thực tiễn nêu trên, đề tài "Bãi đỗ xe thông minh" được thực hiện nhằm nghiên cứu, thiết kế và triển khai một mô hình hệ thống nhúng5 có khả năng tự động hóa việc quản lý bãi đỗ xe. Hệ thống sử dụng cảm biến hồng ngoại để phát hiện xe ra vào và giám sát trạng thái từng chỗ đỗ, vi điều khiển Arduino để xử lý dữ liệu và điều khiển các thiết bị như động cơ servo (mở/đóng cổng) và màn hình LCD (hiển thị thông tin). Thông qua dự án này, sinh viên sẽ có cơ hội áp dụng kiến thức về hệ thống nhúng, lập trình vi điều khiển và tích hợp phần cứng - phần mềm vào một ứng dụng thực tế.

## Mục tiêu và yêu cầu của hệ thống:

Mục tiêu chính của đề tài là xây dựng một hệ thống bãi đỗ xe thông minh quản lý 4 chỗ đỗ, có khả năng tự động phát hiện xe ra vào, điều khiển cổng, giám sát trạng thái chỗ đỗ và hiển thị thông tin trực quan cho người sử dụng.

**Về chức năng điều khiển xe ra vào:**

Hệ thống sử dụng cảm biến hồng ngoại tại cổng vào (chân 2) để phát hiện khi có xe tiếp cận. Khi phát hiện xe, hệ thống tự động kiểm tra số chỗ trống: nếu còn chỗ thì servo cổng vào quay về góc 0° để mở cổng và hiển thị thông báo trên LCD, nếu hết chỗ thì hiển thị "Parking FULL" và từ chối mở cổng. Sau khi xe đi vào, cổng tự động đóng lại bằng cách servo quay về góc 90°. Đối với cổng ra, hệ thống sử dụng cảm biến hồng ngoại tại chân 3 để phát hiện xe muốn rời bãi. Khác với cổng vào, cổng ra không cần kiểm tra điều kiện số chỗ trống mà tự động mở ngay khi phát hiện xe và đóng lại sau khi xe đi qua. Hệ thống sử dụng kỹ thuật edge detection - phát hiện sườn tín hiệu từ HIGH xuống LOW để tránh đọc lặp lại nhiều lần một sự kiện.

**Về chức năng giám sát và hiển thị:**

Hệ thống sử dụng 4 cảm biến hồng ngoại (chân 4-7) được đặt tại mỗi chỗ đỗ để phát hiện liên tục có xe hay trống. Trạng thái của 4 cảm biến được đọc liên tục và lưu vào mảng, với quy ước HIGH là chỗ trống và LOW là có xe. Hệ thống tự động đếm số chỗ có trạng thái trống để tính tổng số chỗ còn lại, thông tin này được cập nhật theo thời gian thực và sử dụng để kiểm tra trước khi cho xe vào. Màn hình LCD 16x2 với giao tiếp I2C được sử dụng để hiển thị trạng thái các chỗ đỗ theo format rõ ràng với ký hiệu "X" cho chỗ có xe và "O" cho chỗ trống. Màn hình được cập nhật sau mỗi sự kiện xe ra vào hoặc thay đổi trạng thái. Ngoài ra, hệ thống còn in thông tin chi tiết ra Serial Monitor bao gồm trạng thái từng slot và tổng số chỗ trống để phục vụ mục đích giám sát và debug.

**Về yêu cầu phần cứng:**

Hệ thống sử dụng Arduino Uno hoặc Mega làm bộ xử lý trung tâm có đủ số lượng chân I/O và bộ nhớ để xử lý các tác vụ. Hệ thống yêu cầu 6 cảm biến hồng ngoại IR trong đó 2 cảm biến đặt tại cổng vào và cổng ra, 4 cảm biến đặt tại các chỗ đỗ, tất cả được kết nối với chế độ INPUT\_PULLUP để đảm bảo ổn định và tránh nhiễu. Hai động cơ servo SG90 được gắn vào chân 9 và chân 10 để điều khiển cổng vào và cổng ra, với cơ chế 0° là mở cổng và 90° là đóng cổng. Màn hình LCD 16x2 giao tiếp I2C địa chỉ 0x27 được sử dụng, giúp tiết kiệm số lượng chân kết nối. Hệ thống cần nguồn cấp điện ổn định 5V để đảm bảo hoạt động liên tục và chính xác của các linh kiện.

**Về yêu cầu phần mềm:**

Hệ thống được lập trình bằng C/C++ trên Arduino IDE với các thư viện Servo.h để điều khiển servo, Wire.h để giao tiếp I2C và LiquidCrystal\_I2C.h để điều khiển màn hình LCD. Chương trình được tổ chức thành các hàm chức năng rõ ràng bao gồm hàm điều khiển cổng mở/đóng, hàm cập nhật LCD, hàm đếm số chỗ trống và hàm in thông tin debug. Hệ thống sử dụng kỹ thuật edge detection - so sánh trạng thái hiện tại với trạng thái trước đó để phát hiện sườn xuống từ HIGH sang LOW, đảm bảo mỗi sự kiện chỉ được xử lý một lần. Vòng lặp chính có delay 300ms để tránh đọc cảm biến quá nhanh gây nhiễu. Chương trình phải xử lý chính xác các trạng thái, không bị treo trong quá trình hoạt động, thời gian phản hồi từ khi phát hiện xe đến khi cổng mở hoàn toàn dưới 2 giây, và có khả năng mở rộng dễ dàng để thêm chức năng mới trong tương lai.

## Phạm vi thực hiện dự án:

Để đảm bảo tính khả thi và hoàn thành đúng tiến độ, đề tài được giới hạn trong phạm vi cụ thể về quy mô, chức năng và thời gian.

**Về quy mô hệ thống:**

Dự án quản lý 4 chỗ đỗ xe (tương ứng với 4 cảm biến SLOT1-4 trong code), phù hợp cho mục đích nghiên cứu, học tập và demo. Đây là quy mô vừa đủ để minh họa đầy đủ các chức năng cơ bản mà không quá phức tạp. Mô hình được xây dựng dưới dạng thu nhỏ với kích thước khoảng 50x40cm, sử dụng vật liệu như gỗ ép, nhựa acrylic hoặc bìa cứng, dễ dàng di chuyển và trình diễn. Mỗi chỗ đỗ được bố trí rõ ràng với cảm biến đặt ở vị trí phù hợp.

**Về chức năng thực hiện:**

Hệ thống tập trung vào ba chức năng cốt lõi: tự động phát hiện và điều khiển xe vào (kiểm tra số chỗ trống, mở cổng nếu còn chỗ hoặc hiển thị "FULL" nếu hết chỗ); tự động điều khiển xe ra (mở cổng không điều kiện khi phát hiện xe); giám sát và hiển thị trạng thái 4 chỗ đỗ theo thời gian thực trên LCD với ký hiệu "X" cho chỗ có xe và "O" cho chỗ trống. Các chức năng mở rộng như kết nối WiFi/Bluetooth, lưu log dữ liệu, nhận diện biển số xe không nằm trong phạm vi bài tập này nhưng sẽ được đề cập trong phần đề xuất phát triển.

**Về công nghệ và thiết bị:**

Dự án sử dụng các linh kiện phổ biến và dễ tiếp cận: Arduino Uno/Mega, 6 cảm biến hồng ngoại IR (2 cho cổng vào/ra - chân 2-3, 4 cho các slot - chân 4-7), 2 động cơ servo SG90 (chân 9-10), màn hình LCD 16x2 I2C địa chỉ 0x27, breadboard, dây jumper và nguồn 5V. Phần mềm sử dụng Arduino IDE với các thư viện Servo.h, Wire.h, LiquidCrystal\_I2C.h. Tất cả là công cụ mã nguồn mở và linh kiện chuẩn với giá thành hợp lý.

**Về thời gian thực hiện:**

Dự án được chia thành các giai đoạn trong 8 tuần: Tuần 1-2 nghiên cứu tài liệu về Arduino, cảm biến IR, servo và LCD I2C; Tuần 3 thiết kế sơ đồ khối, sơ đồ nguyên lý và chuẩn bị linh kiện; Tuần 4-5 lập trình từng module riêng biệt sau đó tích hợp code hoàn chỉnh; Tuần 6 lắp ráp phần cứng và xây dựng mô hình vật lý; Tuần 7 kiểm thử tổng thể với các kịch bản khác nhau; Tuần 8 hoàn thiện, viết báo cáo và chuẩn bị demo.

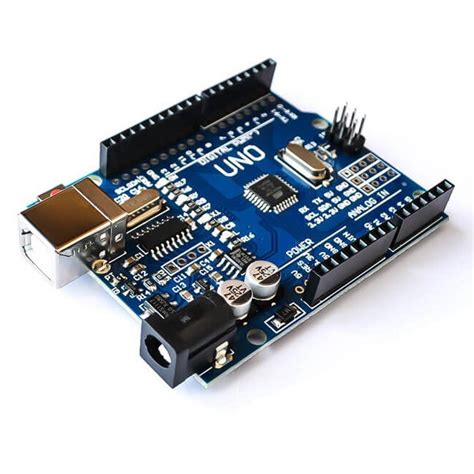
**Về đánh giá và kiểm thử:**

Hệ thống được đánh giá dựa trên các tiêu chí: độ chính xác phát hiện xe và trạng thái slot (>95%), thời gian phản hồi từ khi phát hiện đến khi cổng mở/đóng hoàn toàn (<2 giây), độ ổn định hoạt động liên tục (>2 giờ không bị treo), tính rõ ràng của thông tin hiển thị trên LCD và Serial Monitor. Kết quả kiểm thử được ghi nhận bằng video, hình ảnh và log dữ liệu để đánh giá và cải thiện hệ thống.

# CHƯƠNG 2. THIẾT KẾ HỆ THỐNG:

## 2.1. Phân tích và lựa chọn phần cứng:

### 2.1.1. Vi điều khiển Arduino:



Hình 1. Vi điều khiển Arduino

Trong dự án bãi đỗ xe thông minh này, vi điều khiển Arduino được lựa chọn làm bộ xử lý trung tâm của hệ thống. Arduino là một nền tảng điện tử mã nguồn mở được thiết kế dựa trên phần cứng và phần mềm dễ sử dụng, đặc biệt phù hợp cho các dự án nhúng và IoT.

**Thông số kỹ thuật Arduino Uno:**

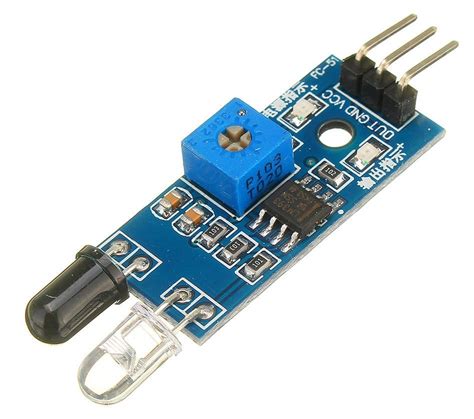
* Vi xử lý: ATmega328P, tần số 16MHz
* Bộ nhớ: Flash 32KB, SRAM 2KB, EEPROM 1KB
* Chân I/O: 14 chân digital (6 chân PWM), 6 chân analog
* Điện áp hoạt động: 5V, điện áp đầu vào 7-12V

Dựa vào code, hệ thống sử dụng tổng cộng 8 chân digital (chân 2-10) để kết nối 6 cảm biến và 2 servo, nằm hoàn toàn trong khả năng cung cấp của Arduino Uno. Arduino Mega có thể được sử dụng nếu cần mở rộng thêm nhiều chỗ đỗ hoặc tính năng phức tạp hơn với 54 chân I/O và bộ nhớ lớn hơn.

**Lý do lựa chọn Arduino:**

* Cộng đồng hỗ trợ lớn với nhiều thư viện sẵn có (Servo.h, Wire.h, LiquidCrystal\_I2C.h)
* Arduino IDE trực quan, lập trình C/C++ đơn giản
* Giá thành phải chăng, dễ mua và thay thế
* Tương thích cao với hầu hết cảm biến và module ngoại vi
* Hỗ trợ các hàm cơ bản như pinMode(), digitalRead(), digitalWrite()

### 2.1.2. Cảm biến hồng ngoại IR:



Hình 2. Cảm biến hồng ngoại IR

Cảm biến hồng ngoại là thiết bị quan trọng nhất, đóng vai trò "mắt" phát hiện xe. Dựa vào code, hệ thống sử dụng 6 cảm biến IR.

**Phân loại và vị trí:**

* 2 cảm biến cổng: SENSOR\_ENTRY (chân 2), SENSOR\_EXIT (chân 3)
* 4 cảm biến slot: SLOT1-4 (chân 4-7)

**Nguyên lý hoạt động:**

* LED phát hồng ngoại (IR Transmitter) phát tia hồng ngoại
* Photodiode thu hồng ngoại (IR Receiver) nhận tín hiệu phản xạ
* Không có vật cản: đầu ra HIGH
* Có vật cản (xe): đầu ra LOW

**Thông số kỹ thuật:**

* Điện áp hoạt động: 5V DC
* Khoảng cách phát hiện: 2-30cm (có thể điều chỉnh)
* Đầu ra: Digital (HIGH/LOW)
* Dòng tiêu thụ: 20mA

**Lý do lựa chọn:**

* Đầu ra digital dễ đọc bằng digitalRead()
* Tốc độ phản hồi nhanh, phù hợp ứng dụng thời gian thực
* Giá thành rẻ (10,000-20,000 VNĐ/module)
* Kích thước nhỏ gọn, dễ lắp đặt
* Tiêu thụ điện năng thấp

Trong code, cảm biến được cấu hình INPUT\_PULLUP: pinMode(SENSOR\_ENTRY, INPUT\_PULLUP). Chế độ này kích hoạt điện trở kéo lên nội 20-50kΩ, đảm bảo chân luôn ở mức HIGH khi không có tín hiệu, tránh nhiễu và đọc sai.

### 2.1.3. Động cơ servo SG90:



Hình 3. Động cơ servo SG90

Động cơ servo điều khiển cổng vào/ra của bãi đỗ. Code sử dụng 2 servo kết nối chân 9 và 10.

**Thông số kỹ thuật SG90:**

* Điện áp: 4.8-6V (tương thích 5V Arduino)
* Dòng tiêu thụ: 100-250mA tùy tải
* Moment xoắn: 1.8kg.cm ở 5V
* Tốc độ: 0.1s/60° ở 4.8V
* Góc quay: 0-180°
* Trọng lượng: 9g
* Kích thước: 23x12.2x29mm

**Điều khiển trong code:**

* Khai báo: Servo servoEntry; Servo servoExit;
* Gắn chân: servoEntry.attach(SERVO\_ENTRY\_PIN);
* Logic đảo ngược: 0° = mở cổng, 90° = đóng cổng
* Hàm openGate(): gate.write(0); delay(1500);
* Hàm closeGate(): gate.write(90); delay(1500);

**Lý do lựa chọn:**

* Điều khiển góc quay chính xác
* Thư viện Servo.h hỗ trợ sẵn, lập trình đơn giản
* Giá thành rẻ (30,000-50,000 VNĐ/con)
* Kích thước nhỏ gọn
* Không cần driver bổ sung

**Lưu ý nguồn**: 2 servo có thể tiêu thụ 500mA, vượt khả năng chân 5V Arduino. Nên dùng nguồn ngoài 5V/2A riêng cho servo, đấu chung GND với Arduino.

### 2.1.4. Màn hình LCD 16x2 I2C:



Hình 4. Màn hình LCD 16x2 I2C

Màn hình LCD là thiết bị hiển thị thông tin đầu ra cho người dùng trong hệ thống bãi đỗ xe thông minh. LCD 16x2 với module I2C được sử dụng để hiển thị trạng thái các chỗ đỗ xe một cách trực quan và dễ theo dõi. LCD 16x2 có nghĩa là màn hình có khả năng hiển thị 16 ký tự trên mỗi dòng với tổng cộng 2 dòng.

**Thông số kỹ thuật LCD 16x2 I2C:**

* Kích thước hiển thị: 16 ký tự x 2 dòng (tổng 32 ký tự)
* Giao tiếp: I2C (Inter-Integrated Circuit)
* Chip chuyển đổi: PCF8574 hoặc PCF8574A
* Địa chỉ I2C: 0x27 (hoặc 0x3F tùy module)
* Điện áp hoạt động: 5V DC
* Đèn nền: LED xanh lá/xanh dương
* Số chân kết nối: 4 chân (VCC, GND, SDA, SCL)

**Các hàm điều khiển LCD được sử dụng trong code:**

* lcd.init(); - Khởi tạo màn hình LCD trong hàm setup()
* lcd.backlight(); - Bật đèn nền để màn hình sáng và dễ đọc
* lcd.clear(); - Xóa toàn bộ nội dung trên màn hình
* lcd.setCursor(col, row); - Đặt vị trí con trỏ tại cột col, dòng row (bắt đầu từ 0)
* lcd.print("text"); - In chuỗi ký tự hoặc biến lên màn hình

**Lý do lựa chọn LCD 16x2 I2C:**

* Giao tiếp I2C tiết kiệm chân I/O của Arduino, chỉ cần 2 chân (SDA, SCL) thay vì 6-8 chân
* Thư viện LiquidCrystal\_I2C.h dễ sử dụng, có nhiều tài liệu hướng dẫn chi tiết
* Hiển thị rõ ràng, dễ đọc ngay cả trong điều kiện ánh sáng yếu nhờ có đèn nền LED
* Giá thành hợp lý (khoảng 70,000 - 100,000 VNĐ), phù hợp với ngân sách dự án
* Tiêu thụ điện năng thấp (~50mA), hoạt động ổn định với nguồn 5V từ Arduino
* Kích thước nhỏ gọn, dễ lắp đặt vào mô hình

**Lưu ý:** nếu LCD không hoạt động sau khi kết nối, cần sử dụng chương trình I2C Scanner để quét và xác định đúng địa chỉ I2C của module, vì một số module sử dụng địa chỉ 0x3F thay vì 0x27. Biến trở trên mặt sau module cho phép điều chỉnh độ tương phản (contrast) của màn hình để hiển thị rõ ràng nhất.RetryClaude can make mistakes. Please double-check responses.

## 2.2. Thiết kế sơ đồ khối hệ thống:

Sơ đồ khối mô tả tổng quan các thành phần chính của hệ thống và mối quan hệ giữa chúng. Hệ thống được chia thành 5 khối chức năng: khối cảm biến vào/ra, khối cảm biến chỗ đỗ, khối xử lý trung tâm, khối điều khiển cổng và khối hiển thị.

**Sơ đồ khối tổng quan:**

A diagram of a computer system

AI-generated content may be incorrect.

Hình 5. Sơ đồ khối hệ thống bãi đỗ xe thông minh

**Bảng mô tả chức năng các khối:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên khối** | **Chức năng** | **Kết nối** |
| 1 | Khối cảm biến vào/ra | Phát hiện xe tại cổng vào và cổng ra bằng cảm biến hồng ngoại IR | Pin 2, 3 |
| 2 | Khối cảm biến chỗ đỗ | Phát hiện trạng thái 4 chỗ đỗ xe (trống hoặc có xe) | Pin 4, 5, 6, 7 |
| 3 | Khối xử lý trung tâm | Xử lý logic, điều khiển servo, cập nhật dữ liệu lên màn hình LCD | Arduino Uno |
| 4 | Khối điều khiển cổng | Mở/đóng cổng vào và cổng ra tự động bằng servo SG90 | Pin 9, 10 (PWM) |
| 5 | Khối hiển thị | Hiển thị trạng thái 4 chỗ đỗ lên màn hình LCD | I2C (A4/A5) |

Bảng 1. Chức năng và kết nối của các khối trong hệ thống bãi đỗ xe thông minh

**Nguyên lý hoạt động:**

**Bước 1:** Các cảm biến IR liên tục giám sát cổng vào, cổng ra và 4 chỗ đỗ xe

**Bước 2:** Arduino đọc tín hiệu từ tất cả cảm biến (sử dụng INPUT\_PULLUP)

**Bước 3:** Khi có xe tại cổng vào → kiểm tra số chỗ trống

**Bước 4:**

* Nếu có chỗ trống → mở servo cổng vào (0°)
* Nếu đầy → hiển thị "Parking FULL" trên LCD

**Bước 5:** LCD cập nhật realtime trạng thái 4 chỗ (X = có xe, O = trống)

**Bước 6:** Khi có xe ở cổng ra → mở servo cổng ra tự động

## 2.3. Thiết kế sơ đồ mạch điện:

Sơ đồ nguyên lý trình bày chi tiết cách kết nối các linh kiện với Arduino Uno, bao gồm phân bổ chân I/O, cấu hình tín hiệu và nguồn cấp điện để đảm bảo hệ thống hoạt động ổn định.

**Bảng phân bổ chân kết nối chi tiết:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Linh kiện** | **Chân Arduino** | **Kết nối** | **Loại tín hiệu** |
| Cảm biến IR vào | Digital Pin 2 | OUT → D2, VCC → 5V, GND → GND | Digital Input (Pull-up) |
| Cảm biến IR ra | Digital Pin 3 | OUT → D3, VCC → 5V, GND → GND | Digital Input (Pull-up) |
| Cảm biến Slot 1 | Digital Pin 4 | OUT → D4, VCC → 5V, GND → GND | Digital Input (Pull-up) |
| Cảm biến Slot 2 | Digital Pin 5 | OUT → D5, VCC → 5V, GND → GND | Digital Input (Pull-up) |
| Cảm biến Slot 3 | Digital Pin 6 | OUT → D6, VCC → 5V, GND → GND | Digital Input (Pull-up) |
| Cảm biến Slot 4 | Digital Pin 7 | OUT → D7, VCC → 5V, GND → GND | Digital Input (Pull-up) |
| Servo cổng vào | Digital Pin 9 (PWM) | Signal → D9, VCC → 5V, GND → GND | PWM Output (50 Hz) |
| Servo cổng ra | Digital Pin 10 (PWM) | Signal → D10, VCC → 5V, GND → GND | PWM Output (50 Hz) |
| LCD I2C 16x2 | A4 (SDA), A5 (SCL) | SDA → A4, SCL → A5, VCC → 5V, GND → GND | I2C Bus (địa chỉ 0x27) |

Bảng 2. Cấu hình chân và loại tín hiệu kết nối các linh kiện trong hệ thống

**Mô tả kết nối từng phần:**

1. Khối đầu vào (Input):

* 6 cảm biến IR được cấu hình INPUT\_PULLUP
* Mỗi cảm biến có 3 chân: VCC (5V), GND, OUT (tín hiệu)
* Trạng thái: LOW = có vật cản (xe), HIGH = không có xe
* Điện trở pull-up nội bộ của Arduino được sử dụng

1. Khối điều khiển cổng (Output):

* 2 động cơ servo SG90
* Góc quay: 0° = mở cổng, 90° = đóng cổng
* Tín hiệu PWM tần số 50Hz
* Lưu ý: Nên dùng nguồn ngoài 5V/2A cho servo để tránh quá tải Arduino

1. Khối hiển thị I2C:

* LCD 16x2 với module I2C (địa chỉ 0x27)
* Kết nối: SDA → A4, SCL → A5
* Tốc độ truyền: 100 kHz (chuẩn I2C)
* Module I2C đã tích hợp điện trở pull-up 4.7kΩ

1. Nguồn cấp điện:

* Arduino: USB 5V hoặc nguồn DC 7-12V
* Cảm biến IR: 5V từ Arduino (tiêu thụ ~20mA/sensor)
* Servo: 5V (khuyến nghị dùng nguồn ngoài, mỗi servo ~500mA)
* LCD: 5V từ Arduino (tiêu thụ ~50mA)

**Lưu ý quan trọng:**

1. Nguồn servo: Nên dùng nguồn ngoài 5V/2A riêng cho 2 servo để tránh quá tải Arduino
2. GND chung: Kết nối GND của Arduino và nguồn ngoài với nhau
3. Cảm biến IR: Điều chỉnh biến trở trên module để phù hợp khoảng cách phát hiện (5-10cm)
4. Chống nhiễu: Đặt tụ điện 100nF gần chân VCC của mỗi cảm biến
5. Dây nối servo: Sử dụng dây đủ tiết diện (≥22AWG) để tránh sụt áp

## 2.4. Thiết kế thuật toán và lưu đồ chương trình:

Thuật toán mô tả logic xử lý của chương trình từ khâu khởi tạo, đọc cảm biến, xử lý điều kiện đến điều khiển servo và cập nhật LCD. Lưu đồ được thiết kế rõ ràng để dễ hiểu và lập trình cài đặt.

**Lưu đồ thuật toán tổng quát:**

A diagram of a computer

AI-generated content may be incorrect.

Hình 6. Sơ đồ thuật toán điều khiển hệ thống bãi đỗ xe thông minh

**Bảng phân tích các hàm chính:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên hàm** | **Tham số** | **Chức năng** | **Giá trị trả về** |
| openGate() | Servo &gate | Mở cổng (servo về góc 0°), hiển thị LCD “Opening” | void |
| closeGate() | Servo &gate | Đóng cổng (servo về góc 90°), hiển thị LCD “Closing” | void |
| updateLCD() | Không | Cập nhật trạng thái 4 slot lên màn hình LCD | void |
| getFreeSlots() | Không | Đếm số chỗ trống trong bãi đỗ (tín hiệu HIGH = trống) | int |
| printSlotStatusSerial() | Không | In trạng thái các slot lên Serial Monitor | void |

Bảng 3. Chức năng các hàm trong chương trình điều khiển hệ thống

**Bảng mã trạng thái:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Trạng thái** | **Giá trị cảm biến** | **Ý nghĩa** | **Hiển thị LCD** |
| **Slot trống** | HIGH | Không có xe | O |
| **Slot có xe** | LOW | Có xe đỗ | X |
| **Cổng mở** | Servo = 0° | Cho phép xe vào/ra | Gate Opening... |
| **Cổng đóng** | Servo = 90° | Chặn xe | Gate Closing... |
| **Bãi đầy** | freeSlots = 0 | Không còn chỗ trống | Parking FULL! |

Bảng 4. Trạng thái hệ thống bãi đỗ xe

**Sơ đồ trạng thái hệ thống:**

A diagram of a computer program

AI-generated content may be incorrect.

Hình 7. Sơ đồ trạng thái hệ thống xử lý xe vào

**Ưu điểm của thuật toán:**

* Đơn giản: Logic rõ ràng, dễ hiểu và bảo trì
* Thời gian thực: Cập nhật trạng thái liên tục mỗi 300ms
* An toàn: Kiểm tra chỗ trống trước khi cho xe vào
* Chống dội (debouncing): Sử dụng biến trạng thái entryState, exitState
* Phản hồi trực quan: LCD hiển thị rõ ràng cho người dùng

# CHƯƠNG 3. TRIỂN KHAI VÀ KẾT QUẢ:

## 3.1. Lập trình và cài đặt hệ thống:

Sau khi hoàn thành thiết kế phần cứng, bước tiếp theo là lập trình phần mềm để điều khiển toàn bộ hệ thống. Chương trình được viết bằng ngôn ngữ C/C++ trên Arduino IDE, sử dụng các thư viện hỗ trợ để giao tiếp với servo và LCD I2C.

**Khai báo thư viện và khởi tạo:**

A screenshot of a computer program

AI-generated content may be incorrect.

* Sử dụng 3 thư viện chính: Servo.h (điều khiển servo), Wire.h (giao tiếp I2C), LiquidCrystal\_I2C.h (điều khiển LCD)
* Khởi tạo đối tượng LCD với địa chỉ I2C 0x27 và kích thước 16×2
* Dùng #define để đặt tên các chân kết nối, giúp code dễ đọc và không chiếm RAM
* Tạo 2 đối tượng servo riêng biệt cho cổng vào và cổng ra
* Khởi tạo biến entryState, exitState để lưu trạng thái trước, phát hiện cạnh xuống
* Mảng slotState[4] lưu trạng thái realtime của 4 chỗ đỗ

**Bảng phân bổ chân:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên biến** | **Chân** | **Thiết bị** | **Loại tín hiệu** |
| SENSOR\_ENTRY | 2 | Cảm biến IR cổng vào | Digital Input |
| SENSOR\_EXIT | 3 | Cảm biến IR cổng ra | Digital Input |
| SLOT1-4 | 4–7 | Cảm biến IR 4 slot | Digital Input |
| SERVO\_ENTRY\_PIN | 9 | Servo cổng vào | PWM Output |
| SERVO\_EXIT\_PIN | 10 | Servo cổng ra | PWM Output |

Bảng 5. Cấu hình chân và thiết bị trong hệ thống Parking Smart

**Hàm điều khiển cổng:**

A computer code with text

AI-generated content may be incorrect.

* Hàm nhận tham chiếu Servo &gate để tiết kiệm RAM (không tạo bản sao)
* lcd.clear() xóa nội dung cũ trên màn hình
* lcd.setCursor(0, 0) đặt con trỏ về vị trí đầu tiên (cột 0, hàng 0)
* Hiển thị thông báo "Gate Opening..." hoặc "Gate Closing..." cho người dùng
* gate.write(0) quay servo về 0° để mở cổng, gate.write(90) để đóng cổng
* delay(1500) chờ 1.5 giây để servo hoàn tất động tác và người dùng đọc thông báo

**Bảng góc servo:**

|  |  |  |
| --- | --- | --- |
| **Góc servo** | **Trạng thái cổng** | **Hàm sử dụng** |
| 0° | Cổng MỞ | openGate() |
| 90° | Cổng ĐÓNG | closeGate() |

Bảng 6. Cấu hình chân và thiết bị servo

**Hàm cập nhật màn hình LCD:**

A computer screen shot of a program

AI-generated content may be incorrect.

* Xóa toàn bộ màn hình LCD trước khi hiển thị trạng thái mới
* Hàng 0 hiển thị trạng thái Slot 1 và Slot 2
* Hàng 1 hiển thị trạng thái Slot 3 và Slot 4
* Toán tử ba ngôi (slotState[i] == LOW ? "X" : "O") kiểm tra trạng thái slot
* LOW (có xe) → hiển thị "X", HIGH (trống) → hiển thị "O"
* Định dạng ngắn gọn "S1:X S2:O" để vừa với LCD 16×2

**Bảng quy ước hiển thị:**

|  |  |  |
| --- | --- | --- |
| **Giá trị slotState** | **Ý nghĩa** | **Ký tự hiển thị** |
| LOW | Có xe đỗ | **X** |
| HIGH | Slot trống | **O** |

Bảng 7. Giá trị cảm biến slot và ký tự hiển thị

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Tình huống** | **Hàng 0 LCD** | **Hàng 1 LCD** |
| Có 2 xe (slot 1, 3) | S1:X S2:O | S3:X S4:O |
| Bãi đầy | S1:X S2:X | S3:X S4:X |
| Bãi trống | S1:O S2:O | S3:O S4:O |

Bảng 8. Hiển thị LCD theo tình huống

**Hàm đếm số chỗ trống:**

A computer code with white text

AI-generated content may be incorrect.

* Khởi tạo biến đếm freeCount = 0
* Vòng lặp for duyệt qua 4 phần tử của mảng slotState[]
* Kiểm tra từng slot: nếu slotState[i] == HIGH (trống) thì tăng biến đếm
* Trả về giá trị freeCount từ 0 đến 4
* Giá trị 0 nghĩa là bãi đầy, giá trị 4 nghĩa là bãi trống hoàn toàn
* Hàm này quyết định có cho phép xe vào bãi hay không

**Ví dụ:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **slotState[0]** | **slotState[1]** | **slotState[2]** | **slotState[3]** | **freeCount** | **Ý nghĩa** |
| LOW | LOW | HIGH | HIGH | 2 | Có 2 chỗ trống |
| HIGH | HIGH | HIGH | HIGH | 4 | Bãi trống |
| LOW | LOW | LOW | LOW | 0 | Bãi đầy |

Bảng 9. Minh họa trạng thái cảm biến và số chỗ trống

**Hàm in trạng thái Serial Monitor:**

A computer screen with white text and orange letters

AI-generated content may be incorrect.

* In chuỗi "Slots: " ra Serial Monitor để bắt đầu dòng thông tin
* Dùng toán tử ba ngôi để in trạng thái từng slot dạng "[S1:X]" hoặc "[S1:O]"
* In ký tự phân cách " | Free: " để dễ đọc
* Gọi hàm `getFreeSlots()` để lấy số chỗ trống và in ra
* `println()` xuống dòng sau khi in xong
* Hàm này phục vụ debug và giám sát hệ thống realtime

**Hàm Setup() - Khởi tạo hệ thống:**

A screenshot of a computer program

AI-generated content may be incorrect.

* Serial.begin(9600) khởi tạo giao tiếp Serial với tốc độ 9600 baud
* Cấu hình 6 chân cảm biến là INPUT\_PULLUP (mặc định HIGH, có xe → LOW)
* servoEntry.attach() và servoExit.attach() gắn servo vào chân PWM
* Gọi closeGate() cho cả 2 servo để đóng cổng lúc khởi động
* lcd.init() khởi tạo LCD, lcd.backlight() bật đèn nền
* updateLCD() hiển thị trạng thái ban đầu của 4 slot lên màn hình

**Bảng quy trình:**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Lệnh** | **Kết quả** |
| 1 | Serial.begin(9600) | Sẵn sàng debug |
| 2 | pinMode(..., INPUT\_PULLUP) | 6 chân mặc định HIGH |
| 3 | servo.attach() | Gắn servo vào PWM |
| 4 | closeGate() | Cổng đóng (90°) |
| 5 | lcd.init() + backlight() | LCD sẵn sàng |
| 6 | updateLCD() | Hiển thị trạng thái |

Bảng 10. Quy trình khởi tạo hệ thống Parking Smart

**Hàm Loop() - Phần 1: Xử lý cổng vào:**

A screenshot of a computer program

AI-generated content may be incorrect.

* Đọc trạng thái 4 cảm biến slot và lưu vào mảng slotState[]
* Gọi printSlotStatusSerial() để in trạng thái ra Serial Monitor
* Tính số chỗ trống bằng getFreeSlots()
* Đọc cảm biến cổng vào, kiểm tra cạnh xuống (HIGH→LOW)
* Nếu đầy (freeSlots == 0): hiển thị "Parking FULL!" và từ chối xe vào
* Nếu còn chỗ: mở cổng vào → đóng cổng vào
* Cập nhật LCD hiển thị trạng thái mới
* Lưu trạng thái hiện tại vào entryState

**Bảng xử lý:**

|  |  |  |  |
| --- | --- | --- | --- |
| **freeSlots** | **Hành động** | **LCD hiển thị** | **Thời gian** |
| 0 | Từ chối vào | Parking FULL! Entry Denied | 1.5s |
| 1–4 | Mở cổng | Gate Opening... Gate Closing... | 3.0s |

Bảng 11. Hành động theo số lượng slot trống

**Hàm Loop() - Phần 2: Xử lý cổng ra:**

A computer screen shot of a program code

AI-generated content may be incorrect.

* Đọc trạng thái cảm biến cổng ra và kiểm tra điều kiện readExit == LOW && exitState == HIGH để phát hiện cạnh xuống (HIGH→LOW) khi có xe mới xuất hiện tại cổng ra.
* Khi phát hiện xe ra, gọi hàm openGate(servoExit) để mở cổng ra ngay lập tức mà không cần kiểm tra số chỗ trống vì xe luôn được phép rời khỏi bãi.
* Gọi closeGate(servoExit) để đóng cổng lại sau 1.5 giây, sau đó gọi updateLCD() để cập nhật trạng thái các slot lên màn hình LCD.
* Lưu trạng thái hiện tại của cảm biến vào biến exitState để phục vụ việc phát hiện cạnh xuống ở lần lặp tiếp theo, sau đó thực hiện delay(300) để tạm dừng 300ms.
* Với delay 300ms giữa các vòng lặp, hệ thống cập nhật trạng thái khoảng 3 lần mỗi giây (1000ms / 300ms ≈ 3.3 lần/giây), đủ nhanh để phản hồi realtime.

**Nguồn cấp điện:**

* Arduino Uno cấp nguồn qua USB 5V hoặc DC 7-12V
* 6 cảm biến IR tiêu thụ ~20mA mỗi sensor, cấp từ Arduino 5V
* 2 servo SG90 tiêu thụ ~500mA mỗi servo, nên dùng nguồn ngoài 5V/2A
* LCD I2C tiêu thụ ~50mA, cấp từ Arduino 5V
* GND của Arduino và nguồn ngoài phải kết nối chung
* Điều chỉnh biến trở trên cảm biến IR để phù hợp khoảng cách 5-10cm

**Bảng yêu cầu nguồn:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thiết bị** | **Điện áp** | **Dòng tiêu thụ** | **Nguồn cấp** |
| Arduino Uno | 5V | ~50mA | USB / DC 7–12V |
| Cảm biến IR (×6) | 5V | ~120mA | Arduino 5V |
| Servo (×2) | 5V | ~1000mA | Nguồn ngoài 5V / 2A |
| LCD I2C | 5V | ~50mA | Arduino 5V |

Bảng 12. Thông số điện áp – dòng tiêu thụ của các thiết bị

## 3.2. Lắp ráp mạch và kiểm tra hoạt động:

***3.2.1. lắp ráp mạch*.**

-Sơ đồ lắp ráp mạch:

A diagram of a car system

AI-generated content may be incorrect.

Hình 8. Sơ đồ lắp đặt mạch bãi đỗ xe Arduino

- Các bước lắp ráp:

Thiết lập nguồn điện: Kết nối dây GND của nguồn 5V ngoài với chân GND của Arduino.

Kết nối LCD I2C: Hoàn thành 4 kết nối (GND, VCC, SDA $\rightarrow$ A4, SCL $\rightarrow$ A5).

**Kết nối Servo Motors:**

* Cấp nguồn VCC (Đỏ) và GND (Nâu) cho cả hai Servo từ nguồn ngoài 5V.
* Dây tín hiệu (Cam) Cổng Vào $\rightarrow$ D9.
* Dây tín hiệu (Cam) Cổng Ra $\rightarrow$ D10.

**Kết nối Cảm biến:**

Cấp nguồn VCC và GND cho tất cả 6 cảm biến từ nguồn ngoài 5V.

**Dây tín hiệu:**

* Cổng vào kết nối D2.
* Cổng ra kết nối D3.
* Vị trí 1 kết nối D4.
* Vị trí 2 kết nối D5.
* Vị trí 3 kết nối D6.
* Vị trí 4 kết nối D7.

Sau khi lắp ráp, bạn có thể tải code lên Arduino và quan sát hoạt động của hệ thống qua màn hình LCD và cửa sổ Serial Monitor

***3.2.2. Kiểm tra hoạt động.***

- Khởi động và Trạng thái ban đầu

Cắm nguồn và tải code: Khi bạn cấp nguồn cho Arduino và hệ thống khởi động, hai động cơ Servo (cổng vào và cổng ra) phải đóng (góc 90°).

Kiểm tra màn hình: Màn hình LCD 16x2 I2C phải hiển thị trạng thái "O" (Open/Trống) cho cả 4 vị trí đỗ. Ví dụ: S1:O S2:O, S3:O S4:O.

Kiểm tra Serial Monitor: Mở Serial Monitor với tốc độ 9600 baud. Nó phải liên tục in ra trạng thái: Slots: [S1:O] [S2:O] [S3:O] [S4:O] | Free: 4.

- Kiểm tra Luồng Xe Vào (Entry)

1. Cho phép vào (Có chỗ trống):

Kích hoạt cảm biến vào: Dùng tay hoặc mô hình xe che cảm biến cổng vào (SENSOR\_ENTRY - D2).

Quan sát Servo: Hệ thống sẽ in ra Serial "Slot available -> Opening Entry Gate", sau đó rào chắn vào (Servo Entry) phải mở (góc 0°), duy trì trong 1.5 giây, và sau đó đóng (góc 90°) trở lại.

Kiểm tra chỗ đỗ: Sau khi xe vào (giả định), bạn hãy che một cảm biến vị trí đỗ bất kỳ, ví dụ SLOT1 (D4).

- Cập nhật hiển thị:

* LCD: Trạng thái của vị trí 1 phải chuyển thành "X" (Occupied/Đầy).
* Serial: Phải in ra Slots: [S1:X] [S2:O]... | Free: 3.

1. Từ chối vào (Bãi đầy):

* Làm đầy bãi đỗ: Lặp lại bước A.1-A.4 cho 3 vị trí còn lại (che SLOT2, SLOT3, và SLOT4) cho đến khi cả 4 vị trí đều hiển thị "X" và Serial in ra | Free: 0.
* Kích hoạt cảm biến vào lần nữa: Che cảm biến cổng vào (SENSOR\_ENTRY - D2).

**Quan sát phản ứng:**

* Serial: Hệ thống phải in ra "Parking FULL – Entry Denied".
* LCD: Màn hình phải hiển thị thông báo "Parking FULL!" và "Entry Denied" trong 1.5 giây.
* Servo Entry phải giữ nguyên trạng thái đóng (90°).

- Kiểm tra Luồng Xe Ra (Exit)

Kích hoạt cảm biến ra: Khi bãi đang đầy (hoặc có ít nhất một xe), che cảm biến cổng ra (SENSOR\_EXIT - D3).

Quan sát Servo: Hệ thống sẽ in ra Serial "Xe ra -> Mo cong ra". Rào chắn ra (Servo Exit) phải mở (góc 0°), sau đó đóng (góc 90°) trở lại.

Cập nhật chỗ đỗ: Ngay sau khi xe ra khỏi vị trí, bạn hãy bỏ che cảm biến vị trí đỗ mà chiếc xe đó vừa rời đi (ví dụ: SLOT1 - D4).

**Cập nhật hiển thị:**

* LCD: Trạng thái của vị trí đó phải chuyển thành "O" (Trống).
* Serial: Phải in ra | Free: 1 (hoặc số chỗ trống mới).

## 3.3. Chạy thử nghiệm và đánh giá:

### 3.3.1. Chạy thử nghiệm:

Khi xe chưa vào hiện thị các ô còn trống:

A toy car on a white cardboard structure

AI-generated content may be incorrect.

Hình 9. Mô hình hoạt động cổng ra vào tự động bãi đỗ xe

Cổng tự động mở:

A hand holding a small electronic device

AI-generated content may be incorrect.

Hình 10. Mô tả quá trình đóng thanh chắn tự động

Khi xe vào cổng tự động đóng:

A close-up of a device

AI-generated content may be incorrect.

Hình 11. Mô hình hiển thị trạng thái đóng cổng và cập nhật dữ liệu

Khi xe vào chỗ sẽ tải dữ liệu lên lcd:

A white model of a building with a blue display

AI-generated content may be incorrect.

Hình 12. Mô phỏng cơ chế từ chối xe vào khi bãi đỗ xe đầy

Khi hết chỗ để xe sẽ báo hết chỗ và không cho xe mới vào:

A model of a car with a display

AI-generated content may be incorrect.

Hình 13. Mô phỏng cơ chế mở thanh chắn tự động cho xe ra

Cửa sẽ tự động mở khi xe ra:

A white structure with a screen and a sign

AI-generated content may be incorrect.

Hình 14. Mô phỏng cơ chế mở cổng tự động

Và tự động đóng :

A hand holding a white box with a screen and a small display

AI-generated content may be incorrect.

Hình 15. Hiển thị thông báo đóng cổng tự động

### 3.3.2. đánh giá:

Hệ thống quản lý bãi đỗ xe nhỏ (4 chỗ). Nó thể hiện việc áp dụng tốt các nguyên tắc cơ bản của Arduino, I2C LCD, Servo và cảm biến kỹ thuật số. Cấu trúc code rõ ràng, có sử dụng hàm riêng biệt cho các tác vụ cụ thể (như openGate, closeGate, updateLCD).

- Đánh giá chi tiết về Mã nguồn (Code Review)

Cấu trúc và Tính dễ đọc (Structure and Readability)

**Ưu điểm:**

* Sử dụng #define cho các chân PIN giúp code dễ đọc và dễ thay đổi khi cần.
* Có các hàm riêng biệt cho các chức năng chính (openGate, closeGate, updateLCD, getFreeSlots), làm cho hàm loop() gọn gàng và dễ bảo trì.
* Sử dụng Serial Monitor (printSlotStatusSerial) để gỡ lỗi và theo dõi trạng thái, đây là một thực hành tốt.
* Chú thích Tiếng Việt rõ ràng (// HIGH = trống, // mở = 0°).

**Điểm cần cải thiện:**

Các biến trạng thái toàn cục (entryState, exitState) được sử dụng để phát hiện sự kiện thay đổi trạng thái (cạnh xuống). Đây là cách làm truyền thống, nhưng có thể thay thế bằng ngắt ngoài (Interrupts) cho SENSOR\_ENTRY và SENSOR\_EXIT để xử lý sự kiện tức thời và hiệu quả hơn, thay vì polling liên tục trong loop().

- Logic Hệ thống và Độ trễ (System Logic and Timing)

Logic Servo: Hàm openGate và closeGate bao gồm delay(1500). Điều này ngăn chặn các hoạt động khác của Arduino trong suốt $1.5$ giây. Trong một hệ thống nhỏ thì chấp nhận được, nhưng trong các hệ thống lớn hơn, nên sử dụng phương pháp non-blocking (không dùng delay) với hàm millis() để Servo mở/đóng mà không làm gián đoạn việc đọc cảm biến khác.

**Logic Chỗ đỗ xe:**

Sử dụng INPUT\_PULLUP là phù hợp, và việc xác định LOW là Đầy ("X") và HIGH là Trống ("O") là logic chuẩn khi dùng cảm biến IR (thường đóng) với trở kéo lên.

Hệ thống không theo dõi xe nào đỗ ở slot nào, nó chỉ kiểm tra TỔNG số chỗ trống. Đây là thiết kế đơn giản và hiệu quả.

- Hiển thị (Display):

* LCD I2C: Việc sử dụng thư viện I2C là hợp lý vì nó giúp tiết kiệm chân Arduino (chỉ cần A4/A5).
* Thông báo trên LCD: Thông báo Parking FULL! và Entry Denied là rất trực quan.
* Trạng thái Slot: Hiển thị S1:X S2:O rất dễ hiểu.

- Đánh giá về Kết nối và Phần cứng (Hardware/Wiring)

Cấp nguồn: Việc sử dụng Servo (SERVO\_ENTRY\_PIN D9, SERVO\_EXIT\_PIN D10) đòi hỏi phải có nguồn điện ngoài 5V ổn định. Nếu không có nguồn ngoài, Arduino có thể bị reset hoặc hoạt động không ổn định khi Servo hoạt động.

**Kết nối cảm biến:**

* Sử dụng chân từ D2 đến D7 cho cảm biến là hợp lý và tuân thủ code.
* Việc cài đặt INPUT\_PULLUP cho tất cả các chân cảm biến là chính xác để tránh nhiễu và sử dụng logic LOW khi phát hiện xe.
* Sơ đồ tổng thể: Hệ thống này có tính Modular cao, dễ dàng thay thế hoặc mở rộng các thành phần.

- Cải tiến và mở rộng:

* Sử dụng millis() thay vì delay(): Loại bỏ các delay(1500) trong các hàm openGate và closeGate để hệ thống phản ứng nhanh hơn với các cảm biến khác (ví dụ: một xe có thể đang ở cổng ra trong khi cổng vào đang chờ Servo đóng).
* Ngắt ngoài (Interrupts): Áp dụng ngắt ngoài cho SENSOR\_ENTRY (chân D2) và SENSOR\_EXIT (chân D3) để đảm bảo không bỏ lỡ tín hiệu khi xe đi qua nhanh.
* Tối ưu hóa hiển thị: Thay vì lcd.clear() liên tục trong openGate, chỉ cần thay đổi dòng thông báo để tránh nhấp nháy màn hình.
* Mở rộng: Thêm một biến đếm xe tổng thể và hiển thị "Available Slots: X" thay vì chỉ hiển thị trạng thái từng slot trên LCD.

# KẾT LUẬN

Hệ thống bãi đỗ xe thông minh mang lại nhiều ưu điểm nổi bật như khả năng tự động hóa quá trình quản lý, giảm sai sót của con người và tiết kiệm thời gian cho người sử dụng. Việc áp dụng cảm biến và bộ điều khiển giúp theo dõi và hiển thị chính xác số lượng xe trong bãi. Đồng thời, mô hình còn giúp nâng cao hiệu quả vận hành nhờ khả năng hoạt động ổn định và dễ dàng triển khai trong thực tế. Đây là cơ sở quan trọng để phát triển các ứng dụng tự động hóa trong môi trường đô thị hiện đại.

Bên cạnh những ưu điểm, hệ thống vẫn tồn tại một số hạn chế nhất định, chủ yếu do quy mô mô hình và mức độ đơn giản của thiết bị. Các cảm biến có thể gặp sai số trong môi trường nhiều nhiễu hoặc khi khoảng cách nhận diện không ổn định. Ngoài ra, mô hình chưa tích hợp nhiều tính năng nâng cao như bảo mật, kiểm soát vé xe hay nhận dạng biển số, khiến khả năng ứng dụng thực tiễn còn bị hạn chế. Những yếu tố này cần được cải thiện để hệ thống đáp ứng tốt hơn nhu cầu thực tế.

Trong tương lai, hệ thống có thể được phát triển theo hướng mở rộng chức năng và nâng cao độ chính xác của các thiết bị cảm biến. Việc bổ sung các công nghệ như nhận dạng biển số, kết nối IoT và ứng dụng điều khiển qua điện thoại sẽ giúp tối ưu hóa trải nghiệm người dùng. Mô hình cũng có thể được mở rộng thành một hệ thống quản lý bãi đỗ xe hoàn chỉnh, phù hợp với nhu cầu của các khu vực đông phương tiện. Đây sẽ là hướng đi quan trọng để nâng cao tính thông minh và hiệu quả cho các bãi đỗ xe hiện đại.

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Trần Minh Hoàng (2018), *Giáo trình Hệ thống nhúng và ứng dụng*, NXB Khoa học & Kỹ thuật.
2. Lê Văn Dũng (2019), *Điện tử cơ bản dành cho sinh viên công nghệ*, NXB Giáo dục Việt Nam.
3. Phạm Quốc Khánh (2020), *Cảm biến và kỹ thuật đo lường trong tự động hóa*, NXB Bách Khoa.
4. Nguyễn Thị Ngọc Anh (2021), *Kỹ thuật vi điều khiển AVR và Arduino*, NXB Lao động – Xã hội.