**TRƯỜNG ĐẠI HỌC**

**KHOA ĐIỆN - ĐIỆN TỬ**

**BỘ MÔN**

----------- 🕮 -----------

**ĐỒ ÁN TỐT NGHIỆP**

HÀ NỘI – 2024

**LỜI CAM ĐOAN**

*Tôi xin cam đoan đồ án “” là công trình nghiên cứu được thực hiện bởi chính cá nhân tôi. Tôi đã tự đọc các thông tin nghiên cứu, dịch tài liệu khác nhau và tổng hợp các kiến thức trên tất cả các nền tảng uy tín để hoàn thiện báo cáo đồ án này và đảm bảo rằng không sao chép ở bất cứ nguồn nào khác. Những lý thuyết được trình bày trong báo cáo đều được sử dụng từ các tài liệu mà tôi đã tham khảo và được trích rõ ràng ở phần tài liệu tham khảo đã có trong báo cáo đồ án.*

|  |  |
| --- | --- |
|  | *Người cam đoan* |

**LỜI NÓI ĐẦU**

1. **Tính cấp thiết của đề tài**

Trong thời đại công nghiệp 4.0, các hệ thống tự động hóa ngày càng đóng vai trò quan trọng trong nhiều lĩnh vực, đặc biệt là trong sản xuất công nghiệp, giao thông thông minh và nghiên cứu trí tuệ nhân tạo. Xe dò line tránh vật cản là một trong những ứng dụng cơ bản của robot di động, giúp mô phỏng các nguyên tắc điều khiển tự động, cảm biến môi trường và ra quyết định. Loại xe này được ứng dụng rộng rãi trong các dây chuyền sản xuất, hệ thống kho thông minh và làm nền tảng cho xe tự hành trong tương lai. Tại Việt Nam, ngành tự động hóa và robot đang được chú trọng phát triển để bắt kịp xu hướng toàn cầu, do đó nghiên cứu và chế tạo xe dò line tránh vật cản không chỉ giúp nâng cao trình độ công nghệ mà còn tạo tiền đề cho các ứng dụng thực tiễn. Bên cạnh đó, đề tài này còn mang ý nghĩa khoa học khi nghiên cứu các công nghệ như cảm biến hồng ngoại, siêu âm, thuật toán PID và trí tuệ nhân tạo trong điều khiển robot. Việc phát triển xe dò line tránh vật cản không chỉ góp phần nâng cao hiệu suất làm việc, giảm sự can thiệp của con người mà còn giúp hiện đại hóa ngành công nghiệp và giao thông trong tương lai.

1. **Đối tượng và phạm vi nghiên cứu:**

**1. Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài là xe dò line tránh vật cản, bao gồm:

* Hệ thống cảm biến dùng để nhận diện đường line và phát hiện vật cản
* Bộ điều khiển trung tâm thực hiện xử lý tín hiệu và điều khiển xe di chuyển.
* Cơ cấu truyền động, bao gồm động cơ DC, động cơ servo hoặc động cơ bước để điều khiển bánh xe.
* Thuật toán điều khiển giúp xe di chuyển chính xác theo đường line và tránh vật cản hiệu quả.

**2. Phạm vi nghiên cứu**

Phạm vi nghiên cứu của đề tài tập trung vào các nội dung chính sau:

* **Thiết kế và lắp ráp phần cứng**: Bao gồm thiết kế mô hình khung xe, lắp đặt động cơ, cảm biến và mạch điều khiển.
* **Lập trình điều khiển**: Viết chương trình cho vi điều khiển để xử lý tín hiệu từ cảm biến, điều khiển động cơ và thực hiện thuật toán dò line kết hợp tránh vật cản.
* **Thuật toán điều khiển**:
  + Áp dụng thuật toán PID để điều chỉnh tốc độ và hướng di chuyển của xe khi dò line.
  + Ứng dụng thuật toán tránh vật cản dựa trên cảm biến siêu âm
* **Môi trường hoạt động**:
  + Xe hoạt động trên bề mặt phẳng có vạch dẫn đường
  + Vật cản có kích thước và màu sắc đa dạng, xuất hiện ngẫu nhiên trên đường đi.
  + Tốc độ và phản ứng của xe phải đảm bảo ổn định và chính xác.

**3. Giới hạn nghiên cứu**

* Xe chỉ hoạt động trong môi trường mô phỏng (phòng thí nghiệm hoặc mô hình thử nghiệm) mà chưa triển khai thực tế ngoài trời.
* Vạch dẫn đường có màu sắc cố định (đường đen trên nền trắng hoặc ngược lại).
* Xe sử dụng các cảm biến phổ biến như hồng ngoại, siêu âm, không nghiên cứu các công nghệ cao cấp như thị giác máy tính hay AI phức tạp.

1. **Phương pháp nghiên cứu:**

Để thực hiện đề tài, nhóm nghiên cứu sử dụng phương pháp lý thuyết và thực nghiệm. Trước tiên, nhóm thu thập và nghiên cứu các tài liệu liên quan đến xe dò line và robot tránh vật cản, tìm hiểu nguyên lý hoạt động của cảm biến hồng ngoại, cảm biến siêu âm, cũng như các thuật toán điều khiển như PID. Đồng thời, khảo sát các hệ thống xe dò line hiện có để rút ra ưu, nhược điểm và hướng cải tiến. Sau đó, nhóm tiến hành thực nghiệm bằng cách lựa chọn linh kiện phù hợp, thiết kế mạch điện, mô hình cơ khí và lập trình vi điều khiển để xử lý tín hiệu cảm biến, điều khiển động cơ theo thuật toán đã nghiên cứu. Xe sẽ được thử nghiệm trên nhiều loại đường line và vật cản khác nhau nhằm đánh giá độ chính xác, tốc độ phản ứng và khả năng vận hành thực tế, từ đó tối ưu hệ thống.

1. **Mục tiêu nghiên cứu của đề tài:**

**1. Mục tiêu chung**

Nghiên cứu và chế tạo một mô hình xe dò line có khả năng tránh vật cản, hoạt động ổn định, có độ chính xác cao và phản ứng nhanh trong môi trường thử nghiệm. Hệ thống giúp mô phỏng nguyên lý điều khiển tự động, ứng dụng trong các lĩnh vực như robot di động, xe tự hành và tự động hóa công nghiệp.

**2. Mục tiêu cụ thể**

* **Thiết kế và lắp ráp phần cứng**:
  + Xây dựng mô hình xe với khung gầm chắc chắn, động cơ vận hành ổn định.
  + Tích hợp các cảm biến phù hợp siêu âm, để nhận diện đường đi và phát hiện vật cản.
  + Lựa chọn và thiết kế mạch điều khiển dựa trên vi điều khiển ESP32.
* **Phát triển thuật toán điều khiển**:
  + Xây dựng thuật toán dò line sử dụng cảm biến hồng ngoại
  + Xây dựng thuật toán tránh vật cản bằng cách xử lý dữ liệu từ cảm biến siêu âm,
  + Ứng dụng thuật toán PID để tối ưu hóa tốc độ và hướng di chuyển của xe.
* **Lập trình và kiểm thử hệ thống**:
  + Lập trình vi điều khiển để xử lý tín hiệu từ cảm biến, điều khiển động cơ và thực hiện thuật toán đã thiết kế.
  + Tiến hành các thử nghiệm thực tế trên đường dẫn khác nhau để kiểm tra độ chính xác và tính ổn định.
  + Hiệu chỉnh và tối ưu thuật toán để xe hoạt động hiệu quả trong các điều kiện thử nghiệm.

1. **Kết cấu của đề tài:**

**Chương 1: Tổng quan đề tài**

**Chương 2: Cơ sở lý thuyết**

**Chương 3: Thiết kế mô hình và mạch điện**

**Chương 4: Thiết kế chương trình điều khiển**

**Chương 5: Thử nghiệm và đánh giá kết quả**

Hà Nội, ngày … tháng … năm 2024

**Sinh viên thực hiện**

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 9](#_Toc192453225)

[1.1 Tổng quan về robot 9](#_Toc192453226)

[1.1.1 Lịch sử phát triển 9](#_Toc192453227)

[1.1.2 Nhu cầu sử dụng 9](#_Toc192453228)

[1.2 Một số mô hình robot dò line trong các cuộc thi 10](#_Toc192453229)

[1.3 Hướng nghiên cứu chính của đề tài 11](#_Toc192453230)

[1.3.1 Đặt vấn đề 11](#_Toc192453231)

[1.3.2 Mục tiêu 12](#_Toc192453232)

[1.3.3 Giới hạn đề tài 12](#_Toc192453233)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 13](#_Toc192453234)

[2.1 Cơ sở lý thuyết 13](#_Toc192453235)

[2.1.1 Cấu trúc động học của xe 13](#_Toc192453236)

[2.1.2 Phương án thiết kế cho xe 13](#_Toc192453237)

[2.2 Cơ sở lý thuyết phần điện 16](#_Toc192453238)

[2.2.1 Phương pháp nhận biết vạch kẻ 16](#_Toc192453239)

[2.2.2 Cấu trúc bộ điều khiển 19](#_Toc192453240)

[CHƯƠNG 3. THIẾT KẾ MÔ HÌNH VÀ MẠCH ĐIỀU KHIỂN 21](#_Toc192453241)

[3.1 Thiết kế cơ khí 21](#_Toc192453242)

[3.1.1 Lựa chọn động cơ sử dụng 22](#_Toc192453243)

[3.1.2 Lựa chọn bánh xe 24](#_Toc192453244)

[3.2 Thiết kế mạch điều khiển 24](#_Toc192453245)

[3.2.1 Sơ đồ khối hệ thống 24](#_Toc192453246)

[3.2.2 Khối điều khiển động cơ 24](#_Toc192453247)

[3.2.3 Khối cảm biến hồng ngoại dò line 26](#_Toc192453248)

[3.2.4 Khối cảm biến tránh vật cản 29](#_Toc192453249)

[3.2.5 Khối điều khiển trung tâm 32](#_Toc192453250)

[3.2.6 Thiết kế khối nguồn 35](#_Toc192453251)

[CHƯƠNG 4. THIẾT KẾ CHƯƠNG TRÌNH ĐIỀU KHIỂN 42](#_Toc192453252)

[4.1 Lưu đồ thuật toán 42](#_Toc192453253)

[4.2 Phân tích thiết kế bộ điều khiển PID 43](#_Toc192453254)

[4.3 Lập trình cho hệ thống 48](#_Toc192453255)

[4.3.1 Phần mềm Arduino IDE 48](#_Toc192453256)

[CHƯƠNG 5. THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ 66](#_Toc192453257)

[5.1 Thử nghiệm phần cứng 66](#_Toc192453258)

[5.1.1 Kiểm tra hoạt động của các nguồn 66](#_Toc192453259)

[5.2 Lắp ráp và thử nghiệm toàn bộ 66](#_Toc192453260)

[5.3 Đánh giá kết quả 66](#_Toc192453261)

**Danh mục hình ảnh**

[Hình 2.1: Cảm biến hồng ngoại 18](#_Toc192453318)

[Hình 2.2: Cảm biến quang học 19](#_Toc192453319)

[Hình 2.3: Nhận diện đường đi bằng xư lý ảnh 20](#_Toc192453320)

[Hình 2.4: Cảm biến thời gian bay 21](#_Toc192453321)

[Hình 3.1:Sơ đồ bố trí các bánh xe 24](#_Toc192453322)

[Hình 3.2: Step motor nema 17hs4401 25](#_Toc192453323)

[Hình 3.3: Sơ đồ khối hệ thống 26](#_Toc192453324)

[Hình 3.4:Mô-đun A4988 27](#_Toc192453325)

[Hình 3.5: Sơ đồ chân của mô-đun 27](#_Toc192453326)

[Hình 3.6: Cảm biến TCRRT 5000 29](#_Toc192453327)

[Hình 3.7: Bản vẽ của cảm biến 29](#_Toc192453328)

[Hình 3.8: Mạch nguyên lý của cam biến 30](#_Toc192453329)

[Hình 3.9: Nguyễn lý đo khoảng cách bằng sóng âm 31](#_Toc192453330)

[Hình 3.10: Nguyên lý đo khoảng cách bằng ánh sáng 32](#_Toc192453331)

[Hình 3.11: Cảm biến siêu âm HC-SR04 33](#_Toc192453332)

[Hình 3.12 Mô đun ESP32 DEVKIT V1 36](#_Toc192453333)

[Hình 3.13: Module hạ áp LM2596 38](#_Toc192453334)

[Hình 3.14: Mạch nguyên lý của LM2596 38](#_Toc192453335)

[Hình 3.15: Phần mềm thiết kế mạch Altium 40](#_Toc192453336)

[Hình 3.16: Mạch nguyên lý 41](#_Toc192453337)

[Hình 3.17: Mạch in 2D 42](#_Toc192453338)

[Hình 3.18: Mạch in 3D 43](#_Toc192453339)

[Hình 4.1: Lưu đồ thuật toán 44](#_Toc192453340)

[Hình 4.2: Mô hình đường đi của robot 46](#_Toc192453341)

[Hình 4.3: Mô--đun cảm biến dò line 46](#_Toc192453342)

[Hình 4.4: Mô hình bộ điều khiển PID 47](#_Toc192453343)

[Hình 4.5: Đường kính bánh xe 48](#_Toc192453344)

[Hình 4.6: Mô hình quỹ độ đường đi của xe 49](#_Toc192453345)

[Hình 4.7: Mô tả hoạt động của chương trình Arduino 51](#_Toc192453346)

[Hình 4.8: Giao diện Arduino IDE 51](#_Toc192453347)

[Hình 4.9: Cấu trúc của void setup () 52](#_Toc192453348)

[Hình 4.10: Cấu trúc của void loop() 53](#_Toc192453349)

**Danh mục bảng biểu**

[Bảng 3.1: Bảng mô tả chân của A4988 28](#_Toc192453350)

[Bảng 3.2: Bảng so sáng các dòng vi điều khiển 34](#_Toc192453351)

[Bảng 4.1: Bảng mã hóa sai số 46](#_Toc192453352)

[Bảng 5.1Bảng đo giá trị của nguồn điện: 68](#_Toc192453353)

# TỔNG QUAN ĐỀ TÀI

## Tổng quan về robot

### Lịch sử phát triển

Xe dò line (Line Following Robot) là một dạng robot tự hành có khả năng di chuyển theo vạch kẻ đường (thường là vạch đen trên nền trắng hoặc ngược lại) nhờ các cảm biến nhận diện đường đi.

Ban đầu, xe dò line ban đầu xuất hiện như một phần của nghiên cứu về điều hướng tự động trong robot di động. Những nguyên mẫu đầu tiên chủ yếu được phát triển trong phòng thí nghiệm để thử nghiệm các thuật toán điều hướng cơ bản.

Với sự phát triển của vi điều khiển và cảm biến quang học, xe dò line bắt đầu được ứng dụng nhiều hơn trong giáo dục và nghiên cứu. Các cuộc thi robot như RoboCup, IEEE Robotics, và các cuộc thi cấp trường đã góp phần phổ biến xe dò line trong cộng đồng sinh viên và nhà nghiên cứu.

Xe dò line ngày nay có thể được tích hợp các công nghệ tiên tiến như trí tuệ nhân tạo (AI), camera nhận diện đường đi, thuật toán điều hướng tối ưu và kết nối IoT. Chúng không chỉ còn giới hạn trong giáo dục mà còn được ứng dụng trong công nghiệp và tự động hóa.

### Nhu cầu sử dụng

Xe dò line ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực nhờ khả năng di chuyển tự động theo tuyến đường đã định sẵn. Trong giáo dục và nghiên cứu, xe dò line là công cụ hữu ích giúp sinh viên và kỹ sư thực hành về lập trình, điều khiển cảm biến, xử lý tín hiệu và thuật toán điều hướng. Đây cũng là một trong những bài toán quan trọng trong các cuộc thi robot, giúp nâng cao kỹ năng thiết kế và tối ưu hóa hệ thống điều khiển.

Trong công nghiệp, xe dò line được sử dụng phổ biến trong các hệ thống vận chuyển hàng tự động (AGV - Automated Guided Vehicle) tại nhà máy và kho hàng. Nhờ khả năng di chuyển chính xác theo lộ trình đã thiết lập, chúng giúp tối ưu hóa quá trình vận chuyển, giảm thiểu sự can thiệp của con người và nâng cao hiệu suất làm việc. Ngoài ra, trong dây chuyền sản xuất, xe dò line có thể đảm nhiệm vai trò vận chuyển linh kiện hoặc sản phẩm, giúp tăng tính tự động hóa trong sản xuất.

Bên cạnh đó, xe dò line còn có ứng dụng thực tiễn trong đời sống, điển hình như robot hút bụi thông minh, phương tiện giao thông tự động và các hệ thống điều hướng trong không gian hẹp. Xu hướng phát triển trong tương lai của xe dò line tập trung vào việc tích hợp trí tuệ nhân tạo (AI), thị giác máy tính và kết nối IoT nhằm nâng cao khả năng nhận diện môi trường và tối ưu hóa hoạt động. Nhờ những ưu điểm này, xe dò line ngày càng đóng vai trò quan trọng trong tự động hóa và công nghệ hiện đại.

## Một số mô hình robot dò line trong các cuộc thi

Trên thế giới, có nhiều cuộc thi robot nổi tiếng được tổ chức, trong đó xe dò line là một hạng mục quan trọng, thu hút sự quan tâm của nhiều kỹ sư và sinh viên đam mê công nghệ. Một số cuộc thi tiêu biểu có thể kể đến như RoboCup (RoboCupJunior Line Following), IEEE Robotics Competition (IEEE Region 5 Line Following), Robofest (Lawrence Technological University, USA),...

Dưới đây là tổng hợp một số so sánh mô hình xe dò line từ cơ bản đến nâng cao, giúp bạn hiểu rõ hơn về nguyên lý hoạt động cũng như các cấp độ phát triển của công nghệ này.

Bảng 1‑1 So sánh mô hình xe dò line

|  |  |  |  |
| --- | --- | --- | --- |
| Loại xe | Ưu điểm | Nhược điểm | Ứng dụng |
| Xe dò line cơ bản | Đơn giản, dễ làm, chi phí thấp | Bị nhiễu bởi ánh sáng, tốc độ chậm | Giáo dục, thực hành vi điều khiển |
| Xe dò line PID | Điều hướng chính xác, tốc độ cao hơn | Cần tinh chỉnh thuật toán PID | Cuộc thi robot, xe tự hành đơn giản |
| Xe dò line AI | Xử lý linh hoạt, tránh chướng ngại vật thông minh | Phần cứng mạnh, lập trình phức tạp | Xe tự hành, robot AI, cuộc thi cao cấp |
| Xe dò line từ tính | Ít bị nhiễu, hoạt động ổn định | Cần đường băng từ để hoạt động | AGV, xe tự hành trong công nghiệp |
| Xe dò line tốc độ cao | Chạy nhanh, bám đường tốt | Yêu cầu cảm biến chính xác, thuật toán phức tạp | Cuộc thi robot tốc độ |
| Xe dò line tránh vật cản | Tránh chướng ngại vật, hoạt động linh hoạt | Cần cảm biến siêu âm/LIDAR, lập trình khó hơn | Xe tự hành trong môi trường phức tạp |





Hình 1‑1 Mô hình cơ bản, PID, tránh vật cản

## Hướng nghiên cứu chính của đề tài

### Đặt vấn đề

Trong thời đại công nghệ phát triển nhanh chóng, robot tự hành đang ngày càng được ứng dụng rộng rãi trong công nghiệp và đời sống, đặc biệt là trong các hệ thống vận chuyển tự động (AGV - Automated Guided Vehicle), robot dịch vụ, và các cuộc thi robot. Xe dò line là một trong những mô hình robot tự hành cơ bản, hoạt động dựa trên nguyên lý dò theo vạch kẻ đường bằng các cảm biến quang học, từ tính hoặc camera.

Việc nghiên cứu và chế tạo xe dò line không chỉ giúp sinh viên tiếp cận với các nguyên lý điều khiển tự động, lập trình nhúng mà còn có tiềm năng ứng dụng thực tế trong công nghiệp và giao thông thông minh.

### Mục tiêu

Mục tiêu của đề tài là nghiên cứu, thiết kế và chế tạo một mô hình xe dò line có khả năng tự động di chuyển theo đường dẫn xác định với độ chính xác cao. Đề tài tập trung vào việc tìm hiểu các phương pháp điều khiển như ON/OFF, PID và AI (nếu có điều kiện phù hợp) để tối ưu hóa khả năng bám đường và tốc độ di chuyển. Bên cạnh đó, hệ thống sẽ được trang bị các cảm biến như cảm biến hồng ngoại (IR), cảm biến từ hoặc camera để cải thiện khả năng nhận diện đường đi. Ngoài việc hoàn thiện phần cứng, đồ án cũng hướng đến việc lập trình và tối ưu thuật toán điều khiển trên vi điều khiển Arduino, Raspberry Pi hoặc ESP32 nhằm nâng cao hiệu suất hoạt động. Thông qua quá trình nghiên cứu và thử nghiệm, đề tài không chỉ giúp nâng cao kỹ năng thiết kế cơ khí, lập trình nhúng mà còn tạo tiền đề để phát triển các ứng dụng xe tự hành trong thực tế.

### Giới hạn đề tài

Phạm vi thiết kế: Xe dò line có kích thước nhỏ, phù hợp để thử nghiệm trong môi trường phòng thí nghiệm hoặc mô hình giả lập. Xe hoạt động trên mặt phẳng với đường dẫn cố định (vạch kẻ màu đen trên nền trắng hoặc băng từ). Hệ thống không yêu cầu kết nối IoT hoặc điều khiển từ xa phức tạp.

Giới hạn công nghệ: Chỉ nghiên cứu và áp dụng một số phương pháp điều khiển cơ bản như PID hoặc AI cơ bản (nếu có đủ điều kiện phần cứng và thời gian nghiên cứu). Không tập trung vào các giải thuật AI phức tạp hoặc thị giác máy tính cao cấp do giới hạn tài nguyên phần cứng và thời gian thực hiện.

Giới hạn ứng dụng: Xe được thiết kế cho mục đích nghiên cứu, thử nghiệm và giáo dục, chưa thể áp dụng ngay vào thực tế công nghiệp. Chưa tích hợp đầy đủ các tính năng của xe tự hành cao cấp như tránh vật cản bằng AI hoặc kết nối với hệ thống giao thông thông minh.

# CƠ SỞ LÝ THUYẾT

## Cơ sở lý thuyết

### Cấu trúc động học của xe

Cấu trúc động học của xe dò line là một yếu tố quan trọng ảnh hưởng đến khả năng di chuyển và hoạt động của xe. Nó bao gồm các thành phần cơ khí và cách chúng tương tác với nhau để tạo ra chuyển động.

Các thành phần cơ bản của cấu trúc động học xe dò line:

* Khung xe: Là bộ phận chịu lực và kết nối các thành phần khác của xe, phải chắc chắn, nhẹ, đảm bảo độ bền và tính thẩm mỹ. Vật liệu: Nhôm, nhựa, gỗ, mica,...
* Động cơ: Tạo ra lực kéo để di chuyển xe. Các loại động cơ như động cơ DC phổ biến, dễ điều khiển, giá thành rẻ; Động cơ Servo điều khiển chính xác, thường dùng cho các ứng dụng phức tạp; động cơ bước độ chính xác cao, thích hợp cho các ứng dụng đòi hỏi độ chính xác về vị trí.
* Hệ thống truyền động: Truyền lực từ động cơ đến bánh xe.

Các loại:

Truyền động trực tiếp: Động cơ gắn trực tiếp vào bánh xe.

Truyền động bằng dây đai: Dây đai kết nối động cơ và bánh xe.

Truyền động bằng xích: Xích kết nối động cơ và bánh xe.

### Phương án thiết kế cho xe

* **Lựa chọn loại và số bánh xe**

Bảng 2‑1 Bảng so sánh các phương án thiết kế xe dò line

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tiêu chí | Xe 2 bánh | Xe 3 bánh | Xe 4 bánh | Xe Omni | Xe Mecaum |
| Ưu điểm | Đơn giản, dễ chế tạo. Di chuyển linh hoạt, xoay tại chỗ. Giá thành rẻ. | Dễ giữ thăng bằng. Di chuyển linh hoạt. Dễ điều khiển. | Ổn định, dễ dàng di chuyển trên địa hình không bằng phẳng. Chở tải trọng lớn. | Di chuyển linh hoạt theo mọi hướng. Xoay tại chỗ. | Di chuyển linh hoạt theo mọi hướng. Xoay tại chỗ. |
| Nhược điểm | Khó giữ thăng bằng, dễ bị lật. Yêu cầu hệ thống điều khiển phức tạp để giữ thăng bằng. | Khả năng di chuyển không bằng xe 2 bánh. Có thể gặp khó khăn khi di chuyển trên địa hình không bằng phẳng. | Khó di chuyển linh hoạt. Cấu tạo phức tạp, giá thành cao. | Cấu tạo phức tạp, giá thành cao. Yêu cầu hệ thống điều khiển phức tạp. | Cấu tạo phức tạp, giá thành cao. Yêu cầu hệ thống điều khiển phức tạp. |
| Độ ổn định | Thấp | Trung bình | Cao | Trung bình | Trung bình |
| Khả năng di chuyển | Cao | Trung bình | Thấp | Cao | Cao |
| Độ phức tạp | Thấp | Trung bình | Cao | Cao | Cao |
| Giá thành | Rẻ | Trung bình | Cao | Cao | Cao |

Chọn xe dò line 3 bánh là lựa chọn tốt cho các đồ án có yêu cầu về tính ổn định và khả năng di chuyển linh hoạt, đồng thời không quá phức tạp về mặt cơ khí và điều khiển.

* Tính ổn định:

Với 3 điểm tiếp xúc với mặt đất, xe 3 bánh có khả năng tự cân bằng tốt hơn so với xe 2 bánh. Điều này giúp xe không bị lật khi di chuyển hoặc thay đổi hướng đột ngột.

Ổn định trên địa hình không bằng phẳng xe 3 bánh ít bị ảnh hưởng bởi các chướng ngại vật nhỏ trên đường đi, giúp xe di chuyển ổn định hơn.

* Khả năng di chuyển:

Xe 3 bánh có thể di chuyển linh hoạt, rẽ trái, rẽ phải và quay đầu dễ dàng.Việc điều khiển xe 3 bánh đơn giản hơn so với xe 2 bánh, đặc biệt là đối với người mới bắt đầu.

* Đơn giản, dễ chế tạo: Cấu trúc của xe 3 bánh không quá phức tạp, giúp việc chế tạo và bảo trì dễ dàng hơn.
* **Lựa chọn loại động cơ**

Bảng 2‑2 Bảng so sánh các loại động cơ sử dụng cho xe dò line

|  |  |  |  |
| --- | --- | --- | --- |
| Loại động cơ | Ưu điểm | Nhươc điểm | Ứng dụng |
| Động cơ bước | Độ chính xác cao. Khả năng điều khiển vị trí và tốc độ. Đơn giản hóa cơ cấu truyền động. Giá thành phải chăng. | Tốc độ quay không cao. Yêu cầu mạch điều khiển phức tạp. | Xe dò line yêu cầu độ chính xác cao. Các ứng dụng robot. Các ứng dụng đo lường. |
| Động cơ DC | Tốc độ quay cao. Momen xoắn lớn. Giá thành rẻ. Dễ điều khiển. | Độ chính xác không cao. Khó điều khiển vị trí và tốc độ. Cần hệ thống truyền động phức tạp.. | Xe dò line tốc độ cao. Các ứng dụng đơn giản. |
| Động cơ Servo | Độ chính xác cao. Khả năng điều khiển vị trí. Tốc độ quay vừa phải | Giá thành cao. Yêu cầu mạch điều khiển phức tạp | Xe dò line yêu cầu độ chính xác cao. Các ứng dụng robot phức tạp |

Chọn động cơ bước dùng để điều khiển xe dò line động cơ bước là lựa chọn tối ưu cho xe dò line trong phạm vi đồ án nhờ độ chính xác cao, khả năng điều khiển vị trí và tốc độ linh hoạt, cơ cấu truyền động đơn giản, giá thành phải chăng.

Bánh chủ động được gắn trực tiếp với động cơ giúp xe đạt độ chính xác và khả năng điều khiển cao nhất. Mỗi bước quay nhỏ của động cơ bước tương ứng với một góc quay nhỏ của bánh xe, loại bỏ sai số từ hệ thống truyền động. Điều này cho phép xe di chuyển, dừng đỗ và thay đổi tốc độ một cách linh hoạt và chính xác, đồng thời đơn giản hóa cơ cấu và giảm thiểu ma sát.

## Cơ sở lý thuyết phần điện

### Phương pháp nhận biết vạch kẻ

***Dùng cảm biến hồng ngoại***

A diagram of a blue circuit board

AI-generated content may be incorrect.

Hình 2.1: Cảm biến hồng ngoại

* Nguyên lý: Cảm biến hồng ngoại phát ra tia hồng ngoại và nhận tín hiệu phản xạ từ mặt đường. Do độ phản xạ của màu trắng và màu đen khác nhau, cảm biến có thể phát hiện được vạch kẻ đường.
* Cấu hình: Thường sử dụng dãy 3, 5 hoặc 7 mắt cảm biến hồng ngoại để quét bề mặt đường
* Ưu điểm:
* Đơn giản, dễ triển khai.
* Chi phí thấp.
* Xử lý nhanh.
* Nhược điểm:
* Bị ảnh hưởng bởi ánh sáng môi trường.
* Không hoạt động tốt trên các bề mặt phản chiếu hoặc có kết cấu phức tạp.

***Dùng cảm biến quang học***

A close-up of a laser scan

AI-generated content may be incorrect.

Hình 2.2: Cảm biến quang học

Nguyên lý: Cảm biến quang học sử dụng ánh sáng LED (thường là đỏ hoặc trắng) để chiếu xuống mặt đường và đo tín hiệu phản xạ.

* Ưu điểm:
* Độ chính xác cao hơn so với cảm biến hồng ngoại.
* Có thể hoạt động tốt hơn trong môi trường sáng mạnh.
* Nhược điểm
  + Giá thành cao hơn.
  + Cần xử lý tín hiệu để loại bỏ nhiễu từ môi trường.

***Dùng Camera và xử lý ảnh:***

A machine on a track

AI-generated content may be incorrect.

Hình 2.3: Nhận diện đường đi bằng xư lý ảnh

* Nguyên lý: Sử dụng camera để thu hình ảnh mặt đường, sau đó dùng thuật toán xử lý ảnh (như OpenCV) để phát hiện vạch kẻ.
* Ưu điểm:
* Có thể nhận diện được nhiều loại vạch kẻ (đứt đoạn, cong, nhiều làn, v.v.).
* Linh hoạt trong các điều kiện đường khác nhau.
* Nhược điểm:
* Yêu cầu vi điều khiển hoặc máy tính có khả năng xử lý mạnh (Raspberry Pi, Jetson Nano).
* Độ trễ xử lý lớn hơn so với cảm biến IR.

***Dùng cảm biến Lidar hoặc Time-of-Flight (ToF)***

A close-up of a cable and a piece of electronics

AI-generated content may be incorrect.

Hình 2.4: Cảm biến thời gian bay

Nguyên lý: Dùng cảm biến đo khoảng cách để phát hiện sự thay đổi độ cao hoặc màu sắc của vạch kẻ đường.

* Ưu điểm:
  + Độ chính xác cao.
  + Hoạt động tốt trong nhiều môi trường.
* Nhược điểm:
  + Chi phí cao.
  + Cần thuật toán xử lý phức tạp.

### Cấu trúc bộ điều khiển

Thuật toán điều khiển xe dò line quyết định cách xe phản ứng với dữ liệu từ cảm biến để bám theo vạch kẻ đường một cách ổn định và chính xác.

***Điều khiển bật – tắt:***

* Nguyên lý: Nếu cảm biến phát hiện xe lệch sang trái hoặc phải, bộ điều khiển sẽ điều chỉnh động cơ để đưa xe trở lại đúng đường.
* Thuật toán:
* Nếu cảm biến giữa nhận thấy vạch đen → đi thẳng.
* Nếu cảm biến trái nhận thấy vạch đen → rẽ phải.
* Nếu cảm biến phải nhận thấy vạch đen → rẽ trái.
* Ưu điểm:
* Đơn giản, dễ triển khai.
* Xử lý nhanh, không yêu cầu tính toán nhiều.
* Nhược điểm:
* Dao động mạnh, dễ mất ổn định ở tốc độ cao.
* Không phù hợp cho đường cong hoặc vạch kẻ phức tạp.

***Điều khiển theo từng mức***

* Nguyên lý**:** Dùng nhiều cảm biến hơn và tạo ra nhiều mức điều khiển thay vì chỉ rẽ trái/phải.
* Thuật toán:
* Dùng dãy cảm biến IR (thường là 5 hoặc 7 cảm biến).
* Nếu cảm biến 1, 2 (trái) nhận vạch đen → quay phải mạnh.
* Nếu cảm biến 3 (giữa) nhận vạch đen → đi thẳng.
* Nếu cảm biến 4, 5 (phải) nhận vạch đen → quay trái mạnh.
* Ưu điểm:
* Giảm hiện tượng dao động so với điều khiển bật tắt.
* Bám đường tốt hơn.
* Nhược điểm:
* Vẫn có hiện tượng dao động khi di chuyển nhanh.

***Điều khiển PID (bộ điều khiển vi tích phân tỉ lệ)***

* Nguyên lý: Tính toán sai số vị trí của xe so với vạch kẻ đường, từ đó điều chỉnh tốc độ động cơ để giữ xe ổn định.
* Công thức điều khiển:

Trong đó:

* Thuật toán:
* Xác định vị trí vạch kẻ từ cảm biến.
* Tính sai số so với vị trí mong muốn.
* Áp dụng công thức PID để điều chỉnh tốc độ động cơ.
* Ưu điểm:
* Điều khiển mượt, ít dao động.
* Phù hợp với tốc độ cao và đường cong.
* Nhược điểm:
* Cần tinh chỉnh thông số
* Tính toán phức tạp hơn.

***Điều khiển dựa trên Machine Learning:***

* Nguyên lý: Dùng camera và thuật toán học máy (CNN, Deep Learning) để nhận diện đường và tự học cách điều khiển.
* Ưu điểm:
* Linh hoạt, có thể thích nghi với nhiều loại đường khác nhau.
* Hiệu suất cao, đặc biệt với xe tự hành.
* Nhược điểm:
* Yêu cầu phần cứng mạnh (Raspberry Pi, Jetson Nano).
* Cần thu thập và huấn luyện dữ liệu.

# THIẾT KẾ MÔ HÌNH VÀ MẠCH ĐIỀU KHIỂN

## Thiết kế cơ khí

Chọn xe di chuyển bằng 3 bánh( 2 chủ động và 1 bị động) sử dụng step motor cho việc điều khiển và gắn trực tiếp vào bánh xe, bánh chủ động sẽ được in 3D, tấm mica sẽ là phần khung của xe.

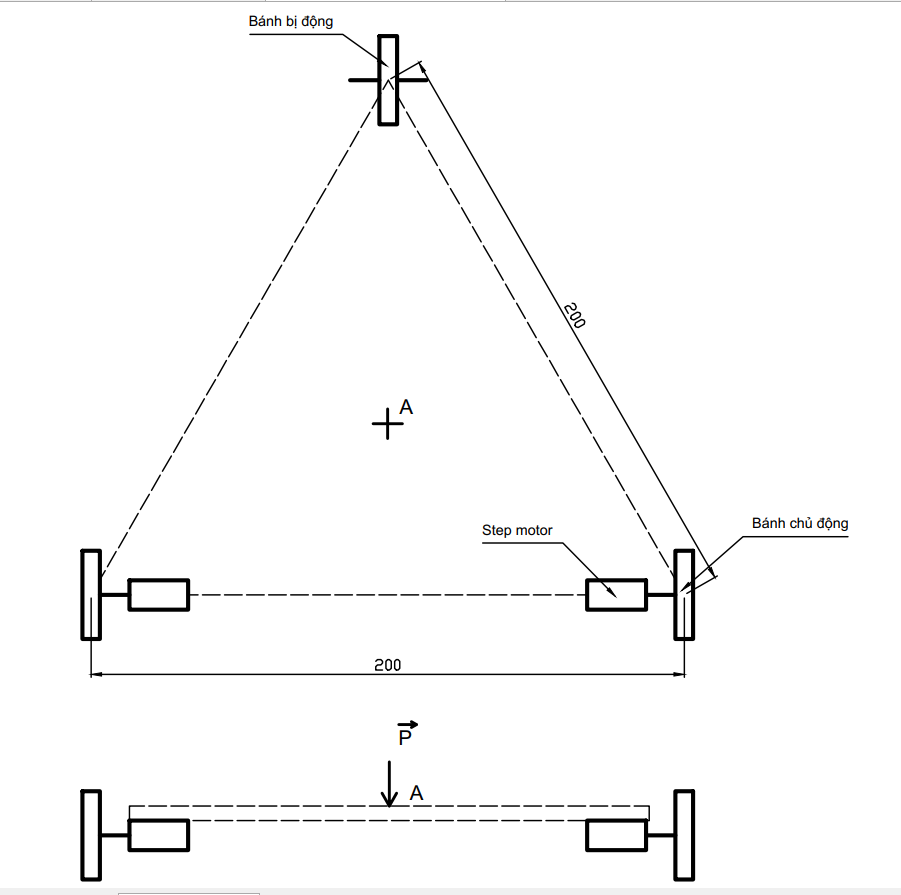
### Lựa chọn động cơ sử dụng

Phương án được chọn là cơ cấu xe di chuyển bằng 2 bánh chủ động phía sau và có 1 bánh dẫn hướng.

Ta có thông số để lựa chọn kích thước bánh xe bị động:

Bán kính bánh xe tự lựa:

Khối lượng tải tối đa :



Hình 3.1:Sơ đồ bố trí các bánh xe

Khối lượng tải: , khối lượng mỗi bánh xe chủ động chịu tải

Lực đẩy cần thiết để di chuyển vật

Với

:khối lượng vật cần chở

(gia tốc trọng trường)

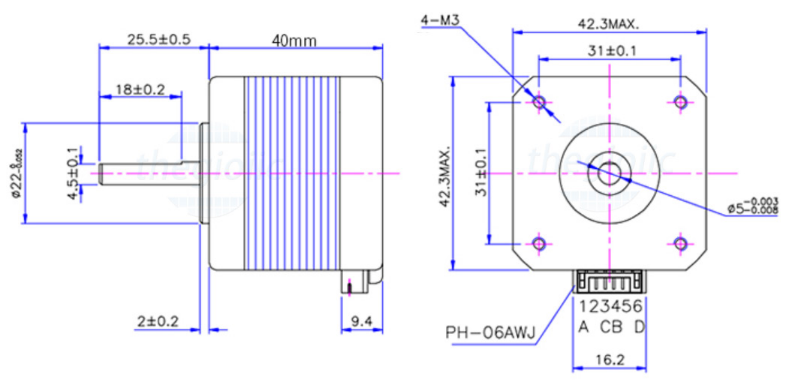
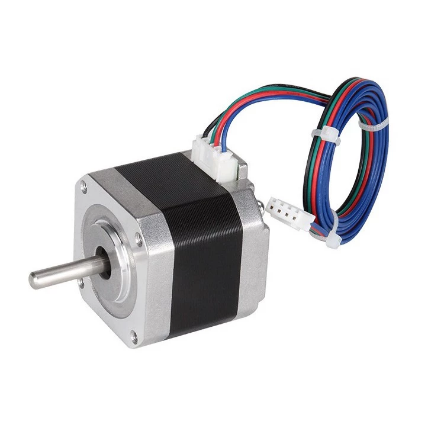
(giả sử hệ số ma sát lăn của bánh xe cao su trên mặt phẳng nhẵn)

bán kính bánh xe chủ động

Mô-men xoắn cần thiết để bánh xe di chuyển có thể tính theo công thức:

Momen chọn động cơ

Ta chọn step motor nema 17hs4401 với momen xoắn giữ



Hình 3.2: Step motor nema 17hs4401

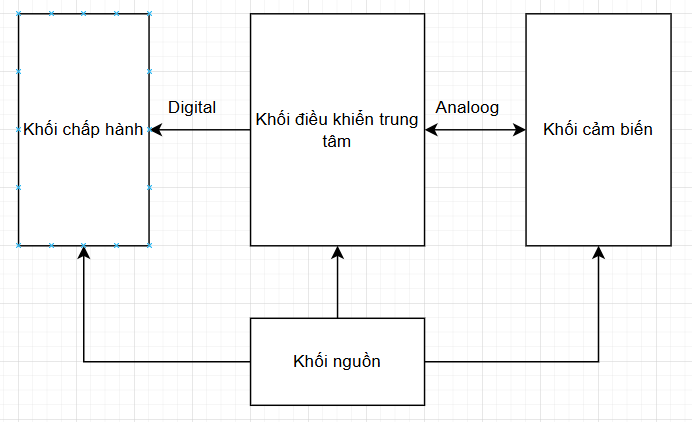
Bảng 3‑1 Thông số động cơ

|  |  |
| --- | --- |
| Thông số | Giá tri |
| Loại động cơ | Động cơ bước (Step Motor) |
| Kích thước | NEMA 17 (42x42 mm) |
| Góc bước | 1.8°/step (200 bước/vòng) |
| Dòng điện định mức | 1.7A / pha |
| Mô-men xoắn giữ | 40 N.cm (0.4 Nm) |
| Mô-men xoắn động | ~20-25 N.cm khi quay |

### Lựa chọn bánh xe

## Thiết kế mạch điều khiển

### Sơ đồ khối hệ thống



Hình 3.3: Sơ đồ khối hệ thống

Trong mạch điều khiển của em sẽ chia ra làm 4 phần:

* Khối nguồn bao gồm mạch hạ áp 5V và mạch ổn áp tuyến tính 3V3, có nhiệm vụ cung cấp nguồn cho khối điều khiển trung tâm và khối ngoại vi và đảm bảo điện áp mức logic điều khiển cho các khối còn lại
* Khối điều khiển trung tâm sẽ làm nhiệm vụ đọc dữ liệu từ cảm biến, xử lý dữ liệu và điều khiển khối chấp hành
* Khối cảm biến bao gồm 8 mắt cảm biến dùng để nhận biết đường line
* Khối chấp hành bao gồm module điều khiển động cơ và động cơ bước

### Khối điều khiển động cơ

A4988 Step Driver có thể dùng cho Reprap trong máy in 3D kèm với board RAMPS 1.4. Dùng với board CNC Shield v3 cho CNC mini. Hoặc dùng điều khiển động cơ bước.

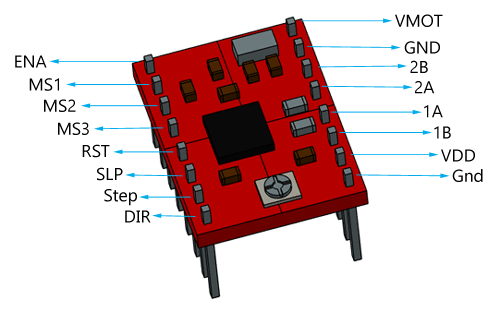


Hình 3.4:Mô-đun A4988

**Ưu điểm:**

* + Giao thức điều khiển số bước và chiều quay rất đơn giản.
  + 5 cấp điều chỉnh bước: 1; 1/2; 1/4; 1/8 và 1/16 bước.
  + Điều chỉnh dòng định mức cấp cho động cơ bằng triết áp.
  + Có chức năng bảo vệ ngắn mạch, bảo vệ quá nhiệt, bảo vệ tụt áp và chống dòng ngược.

Bên cạnh đó, A4988 còn sở hữu các PIN như hình 2.15 dưới đây:



Hình 3.5: Sơ đồ chân của mô-đun

Bảng 3.1: Bảng mô tả chân của A4988

|  |  |
| --- | --- |
| **Pin Name** | **Description** |
| VDD & GND | Được kết nối với 5V và GND của Bộ điều khiển |
| VMOT & GND | Sử dụng để cung cấp năng lượng cho động cơ |
| 1A, 1B, 2A, 2B | Kết nối với 4 cuộn dây của động cơ |
| DIRECTION | Chốt điều khiển hướng động cơ |
| STEP | Pin điều khiển các bước |
| MS1, MS2, MS3 | Microstep Selection Pins |
| SLEEP | Chân để kiểm soát trạng thái điện |
| RESET | Cài lại |
| ENABLE | Cho phép |

**Thông số kĩ thuật:**

* + Điện áp cấp tối thiểu: 8 V
  + Điện áp cấp cực đại: 35 V
  + Dòng cấp liên tục cho mỗi pha: 1 A (không cần tản nhiệt, làm mát)
  + Dòng cấp liên tục cho mỗi pha: 2 A (khi có làm mát, tản nhiệt)
  + Điện áp logic 1 tối thiểu: 3 V
  + Điện áp logic 1 tối đa: 5.5 V
  + Độ phân giải: full, 1/2, 1/4, 1/8, và 1/16

Em dùng chế độ vi bước x8 nên kết nối 2 chân MS1 và MS2 lên 5V

### Khối cảm biến hồng ngoại dò line

Việc lựa chọn cảm biến hồng ngoại cho các ứng dụng dò line cho robot là một yếu tố quan trọng để đảm bảo độ chính xác và hiệu quả. Để nhận biết được đường line em sử dụng cảm biến TCRT5000.



Hình 3.6: Cảm biến TCRRT 5000

Thông số kĩ thuật:

* TCRT5000 gồm LED phát và LED thu hồng ngoại.
* Kích thước cơ khí của TCRT5000:

A blueprint of a mechanical device

AI-generated content may be incorrect.

Hình 3.7: Bản vẽ của cảm biến

Nguyên lý hoạt động:

Cảm biến phản xạ hồng ngoại TCRT5000 Infrared Reflective Sensor làm việc dựa trên nguyên lý khi ánh sáng hồng ngoại từ LED hồng ngoại phát ra nếu có vật phản xạ hồng ngoại thì LED thu hồng ngoại sẽ hoạt động. Để sử dụng cảm biến TCRT5000 em sẽ kết nối như sau:

A diagram of a circuit

AI-generated content may be incorrect.

Hình 3.8: Mạch nguyên lý của cam biến

Trong sơ đồ trên em sử dụng điện trở R9 để hạn dòng cho LED hồng ngoại trong cảm biến TCRT5000:

Điện áp nguồn 3.3V, LED có điện áp thuận = 1.2V

Phù hợp với ngưỡng an toàn cho LED (5-20mA)

Điện trở R10 đóng vai trò điện trở kéo lên/ kéo xuống cho phototransistor trong TCRT5000.

Giá trị 10k giúp hạn chế dòng qua phototransistor khi dẫn, đồng thời tạo mức logic rõ rang (cao/ thấp) khi có ánh sáng phản xạ.

Em sẽ kết nối tương tự với 5 cảm biến TCRT5000 để tạo thành hệ thống mắt dò line

### Khối cảm biến tránh vật cản

Để nhận biết được vật cản trên đường di chuyển em tiến hành tìm hiểu các phương pháp đo khoảng cách. Trong đó các phương pháp hiệu quá và có chi phí thấp hiện nay bao gồm: phương pháp sử dụng sóng âm, phương pháp sử dụng ánh sáng.

***Phương pháp sóng âm***

A diagram of a device

AI-generated content may be incorrect.

Hình 3.9: Nguyễn lý đo khoảng cách bằng sóng âm

Cảm biến sẽ phát ra một tín hiệu siêu âm, khi gặp vật cản sóng siêu âm sẽ dội lại. Cảm biến sẽ có một bộ thu để thu nhận sóng phản xạ lại và ghi nhận thời gian từ lúc phát đến lúc thu được tín hiệu từ đó tính được khoảng cách từ cảm biến đến vật thể.

Trong đó:

khoảng cách

vận tốc âm thanh

thời gian

Ưu điểm:

* Chi phí thấp
* Không bị ảnh hưởng bởi ánh sáng
* Độ chính xác ở mức tương đối
* Làm việc tốt trong môi trường bụi bẩn

Nhược điểm:

* Bị ảnh hưởng bởi nhiệt độ và độ ẩm
* Không hoạt động tốt ở những vật hấp thụ âm thanh
* Phạm vi đo hạn chế
* Góc quét hẹp
* Thời gian đáp ứng chậm

***Phương pháp ánh sáng:***

A diagram of a distance

AI-generated content may be incorrect.

Hình 3.10: Nguyên lý đo khoảng cách bằng ánh sáng

Nguyên lý hoạt động:

* Cảm biến sẽ phát ra một chùm tia laser hẹp ( thường là ánh sáng hồng ngoại hoặc ánh sáng khả kiến)
* Khi tia sáng laser gặp vật thể chúng phản xạ lại.
* Cảm biến sẽ ghi nhận thời gian phản hồi
* Từ đó tính toán được khoảng cách dựa trên tốc độ ánh sáng.

Ưu điểm:

* Độ chính xác cao (có thể lên tới vài mm)
* Khoảng cách đo xa
* Không bị ảnh hưởng bởi ánh sáng môi trường.
* Tốc độ quét nhanh

Nhược điểm

* Giá thành cao
* Không hoạt động tốt trong điều kiện sương mù
* Cần thuật toán xử lý phức tạp

***Phương pháp nhận biết vật cản bằng thị giác máy tính:***

Nguyên lý hoạt động: hệ thống sẽ tiến hành thu thập hình ảnh theo thời gian thực, sau đó sử dụng các thuật toán trí tuệ nhân tạo tiến hành phân tích và nhận diện vật thể.

Ưu điểm:

* Nhận diện thông minh
* Không cần phát tín hiệu

Nhược điểm:

* Bị ảnh hưởng bởi điều kiện ánh sáng
* Xử lý phức tạp
* Tốc độ tính toán chậm

Từ những phân tích trên em sử dụng phương pháp sóng âm để nhận biết vật cản vì phương pháp sóng âm có giá thành rẻ, độ chính xác tương đối và thời gian đáp ứng nhanh không bị ảnh hưởng bởi điều kiện ánh sáng.

Em sử dụng cảm biến HC-SR04:

A blue circuit board with two round speakers

AI-generated content may be incorrect.

Hình 3.11: Cảm biến siêu âm HC-SR04

***Thông số kĩ thuật:***

* Điện áp: 5V DC
* Dòng hoạt động: < 2mA
* Mức cao: 5V
* Mức thấp: 0V
* Góc tối đa: 15 độ
* Khoảng cách: 2cm – 450cm (4.5m)
* Độ chính xác: 3mm

### Khối điều khiển trung tâm

Để tính toán và xử lý số liệu cũng như giao tiếp với ngoại vi và để chọn vi điều khiển cho khối xử lý em đánh giá dựa trên hiệu suất làm việc, cách thức sử dụng, giá thành và số lượng IO mà vi điều khiển đáp ứng được.

Số chân IO cần sử dụng trong mạch bao gồm:

* Khối chấp hành : 6 chân
* Khối cảm biến: 8 chân

Tổng cộng cần sử dụng 14 I/O để điều khiển. Em tiến hành khảo sát một số dòng chip phổ biến mọi người hay sử dụng hiện nay:

Bảng 3.2: Bảng so sáng các dòng vi điều khiển

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tính năng** | | **STM32L431CCT6** | | **ATmega328P** | **ESP32** | **STM32F103C8T6** |
| Kiến trúc | ARM Cortex-M4 | | 8-bit AVR | | 32-bit Xtensa LX6 | ARM Cortex-M3 |
| Tốc độ xung nhịp | 80 MHz | | 16 MHz | | 160-240 MHz | 72 MHz |
| Điện áp hoạt động | 1.71V - 3.6V | | 1.8V - 5.5V | | 3.0V - 3.6V | 2.0V - 3.6V |
| Điện năng tiêu thụ | Rất thấp (tối đa 6.72mA) | | Thấp (tối đa 9mA/1MHz | | Cao (tối đa 300mA) | Trung bình(tối đa 5mA) |
| Flash | 256 KB | | 32 KB | | 4 MB | 128 KB |
| SRAM | 64 KB | | 2 KB | | 520 KB | 20 KB |
| Số chân I/O | 37 | | 23 (14 Digital, 6 Analog) | | 34 | 37 |
| Ngoại vi | 1 ADC 12bit với 16 kênh đầu vào, 2 DAC, 3 cổng USART, 3 cổng I2C, 3 cổng SPI 1 cổng CAN | | 1 ADC 10bit 6 kênh đầu vào, 1 UART, 1 SPI, 1 I2C | | Wi-Fi, Bluetooth, 2 UART, SPI, 1 I2C, PWM, 6 ADC, 1 DAC | 3 UART,2 SPI, 2 I2C, Timer, 2 ADC, 1 CAN |

Vì vậy em quyết định chọn ESP32 làm MCU để điều khiển mạch của em.

ESP32 là một vi điều khiển đa năng được sản xuất bởi Espressif, nổi bật với khả năng kết nối Wi-Fi và Bluetooth tích hợp, phù hợp cho các ứng dụng IoT và mạng không dây. Vi điều khiển này trang bị lõi ARM Cortex-M4, với bộ nhớ Flash lên đến 16 MB và RAM lên đến 520 KB, cho phép xử lý mạnh mẽ các ứng dụng phức tạp. Một trong những ưu điểm của ESP32 là khả năng tiết kiệm năng lượng, đồng thời hỗ trợ tốc độ xử lý cao, lý tưởng cho các dự án yêu cầu giao tiếp không dây và điều khiển các thiết bị ngoại vi. Với các tính năng tích hợp như Bluetooth Low Energy (BLE), các giao tiếp SPI, I2C, UART và ADC, ESP32 dễ dàng được ứng dụng trong các sản phẩm thông minh và hệ thống mạng.

Để tiết kiệm thời gian thiết kế mạch điện, đồng thời tăng hiệu suất của mạch em sử dụng ESP32 DEV KIT V1

A close-up of a microchip

Description automatically generated

Hình 3.12 Mô đun ESP32 DEVKIT V1

**Thông số kỹ thuật:**

* IC chính: Wifi BLE SoC ESP32 ESP-WROOM-32
* Điện áp sử dụng: 2.2V~3.6VDC
* Dòng điện sử dụng: ~90mA.
* Nhân xử lý trung tâm: ESP32-D0WDQ6 Dual-core low power Xtensa® 32-bit LX6 microprocessors.
* Bộ nhớ ROM: 448KBytes
* Bộ nhớ SRAM: 520 KBytes
* WiFi: 802.11 b/g/n/d/e/i/k/r (802.11n up to 150 Mbps)
* Bluetooth v4.2 BR/EDR and BLE specification
* Wi-Fi mode Station/softAP/SoftAP+station/P2P
* Bảo mật WPA/WPA2/WPA2-Enterprise/WPS
* Mã hóa AES/RSA/ECC/SHA
* IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
* Giao tiếp: SD-card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S , IR, GPIO, cảm biến chạm, ADC, DAC, cảm biến Hall, cảm biến nhiệt độ.

### Thiết kế khối nguồn

Để lựa chọn khối nguồn phù hợp, em tính toán tổng công suất cần thiết mà khối nguồn cần phải cấp:

* ESP32: Tối đa 300mA – 3V3
* A4988: Tối đa 15mA – 5V
* Cảm biến siêu âm HC-SR04: 10mA – 5V
* Cảm biến TCRT5000: 1mA – 3V3

Dòng tối đa: 350mA, công suất tiêu thụ tối đa: 1.5W

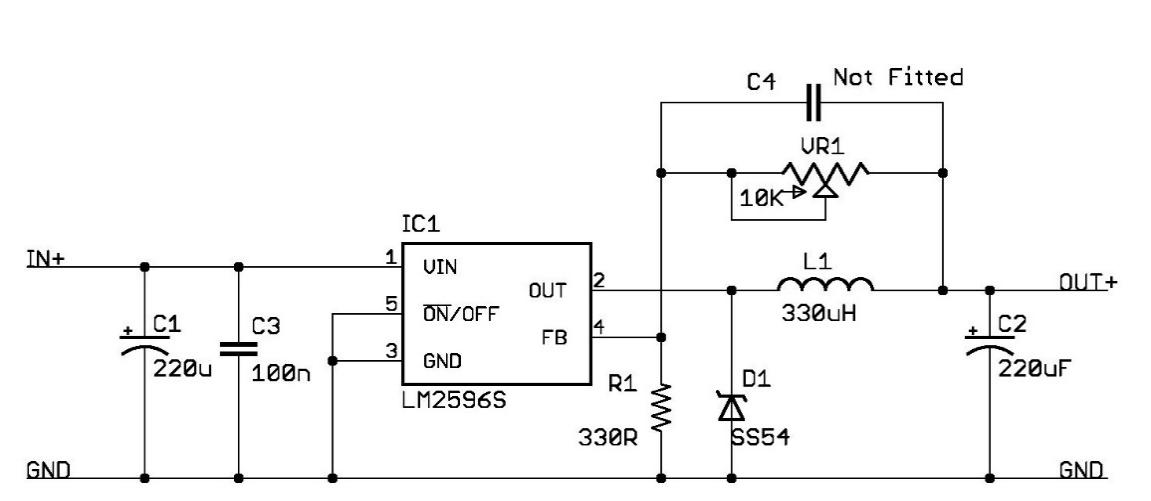
Từ những dữ liệu trên, em sẽ lựa chọn sử dụng khối 3 pin lithium-ion 18650 3V3 – 2200mAh để làm nguồn cấp chính cho hệ thống (do động cơ bước làm cần cung cấp nguồn 12V để hoạt động), thông qua mạch hạ áp tạo ra nguồn điện 5V để cung cấp cho hệ thống.

Để hạ áp từ 12V xuống 5V em sử dụng IC LM2596. IC hạ áp LM2596 là một thiết bị sử dụng chip điều khiển đặc biệt LM2596 để cung cấp giải pháp giảm áp (điện áp hạ áp) trong các ứng dụng điện tử. LM2596 là một bộ giảm áp DC-DC (Switching Regulator) với khả năng điều chỉnh điện áp đầu ra. Module được sử dụng trong đồ án để giảm điện áp từ adapter 12VDC xuống 5VDC giúp các thành phần của hệ thống hoạt động.

A close-up of a circuit board

AI-generated content may be incorrect.

Hình 3.13: Module hạ áp LM2596



Hình 3.14: Mạch nguyên lý của LM2596

Các tính năng nổi bật của LM2596:

* Điều chỉnh điện áp đầu ra: IC này cho phép người dùng điều chỉnh điện áp đầu ra thông qua một viền điều chỉnh hoặc thông qua cách kết nối khác nhau, giúp đáp ứng nhu cầu về điện áp đầu ra khác nhau.
* Dải điện áp đầu ra rộng: Các LM2596 có thể cung cấp một dải điện áp đầu ra rộng, cho phép điều chỉnh từ một giá trị cụ thể đến một giá trị tối đa thích hợp.
* Hiệu suất cao: Bộ giảm áp LM2596 thường có hiệu suất chuyển đổi cao, giúp giảm thiểu tổn thất năng lượng và tản nhiệt.
* Bảo vệ quá tải và quá nhiệt: Các LM2596 thường có các tính năng bảo vệ quá tải và quá nhiệt để đảm bảo an toàn khi sử dụng.
* Ổn định và đáng tin cậy: LM2596 được sử dụng rộng rãi và đã được kiểm tra và xác minh độ ổn định và đáng tin cậy trong các ứng dụng điện tử.
* Ứng dụng rộng rãi: Các module hạ áp LM2596 có thể được sử dụng trong nhiều ứng dụng khác nhau, chẳng hạn như trong các thiết bị điện tử, mô-đun Arduino, dự án DIY, và các ứng dụng công nghiệp.

3.2.7 Sơ đồ mạch thiết kế Altium

Altium Designer trước kia có tên gọi là Protel DXP, là một trong những công cụ vẽ mạch điện tử mạnh nhất hiện nay. Được phát triển bởi hãng Altium Limited. Altium Designer là một phần mềm chuyên ngành được sử dụng trong thiết kế điện tử. Nó là một phần mềm mạnh với nhiều tính năng thú vị, tuy nhiệ phần mềm này ít được biết đến hơn so với các phần mềm cùng chức năng khác như ORCAD hay PROTEUS.

A picture containing qr code

Description automatically generated

Hình 3.15: Phần mềm thiết kế mạch Altium

Altium Designer có một số đặc trưng sau:

* + Giao diện thiết kế, quản lí và chỉnh sửa thân thiện, dêc dàng biên dịch, quản lí file, quản lí phiên bản các tài liệu thiết kế.
  + Hỗ trợ mạnh mẽ cho việc thiết kế tự động, đi dây tự động theo thuật toán tối ưu, phân tích lắp ráp linh kiện. Hỗ trợ cho việc tìm các giải pháp thiết kế hoặc chỉnh sửa mạch, linh kiện có sẵn trước theo các tham số mới.
  + Mở, xem và in các file thiết kế mạch dễ dàng với đầy đủ các thông tin linh kiện, dữ liệu bản vẽ, kích thước và số lượng.
  + Hệ thống các thư viện linh kiện phong phú, chi tiết và hoàn chỉnh bao gồm tất cả các linh kiện nhúng, số và tương tự.
  + Đặt và sửa đối tượng trên các lớp cơ khí, định nghĩa các luật thiết kế, tùy chỉnh các lớp mạch in, chuyển từ schematic sang PCB, đặt vị trí linh kiện trên PCB.
  + Mô phỏng mạch PCB 3D, đem lại hình ảnh ,mạch điện trung thực trong không gian 3 chiều, hỗ trợ MCAD-ECAD, liên kết trực tiếp với mô hình STEP, kiểm tra khoảng cách cách điện , cấu hình cả 2D và 3D.
  + Hỗ trợ thiết kế PCB sang FFGA và ngược lại.

A diagram of a computer

AI-generated content may be incorrect.

Hình 3.16: Mạch nguyên lý

A computer screen shot of a circuit board

AI-generated content may be incorrect.

Hình 3.17: Mạch in 2D

A computer screen shot of a circuit board

AI-generated content may be incorrect.

Hình 3.18: Mạch in 3D

# THIẾT KẾ CHƯƠNG TRÌNH ĐIỀU KHIỂN

## Lưu đồ thuật toán

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 4.1: Lưu đồ thuật toán

**Giải thích chi tiết:**

* **Khởi tạo hệ thống:**
  + Cấu hình các chân I/O cho motor, cảm biến
  + Kết nối với Blynk qua WiFi
  + Khởi tạo PWM cho motor và Servo
  + Khởi động Serial Monitor
* **Vòng lặp chính:**
  + Luôn cập nhật Blynk và timer
  + Kiểm tra trạng thái điều khiển từ Blynk
* **Trạng thái OFF:**
  + Dừng motor ngay lập tức
  + Reset các biến PID và giá trị khoảng cách
* **Trạng thái ON:**
  + Kiểm tra vật cản phía trước bằng cảm biến siêu âm
  + Nếu phát hiện vật cản:
    - Dừng motor
    - Quay Servo để đo khoảng cách 2 bên
    - Quyết định hướng rẽ ưu tiên
    - Thực hiện thao tác rẽ
  + Nếu không có vật cản:
    - Đọc giá trị 5 cảm biến dò line
    - Tính toán giá trị PID
    - Điều chỉnh tốc độ motor bằng PWM dựa trên PID

## Phân tích thiết kế bộ điều khiển PID

Bộ điều khiển vi tích phân tỉ lệ sẽ tính toán giá trị sai số là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn rồi dùng thuật toán PID để tính toán giá trị điều khiển đầu vào sao cho sai số là nhỏ nhất

Trong ứng dụng với xe dò line sai số sẽ là vị trị của line đến cảm biến trung tâm của robot.

A diagram of a sensor

AI-generated content may be incorrect.

Hình 4.2: Mô hình đường đi của robot

Theo tính toán ở phần phía trên em lựa chọn cảm biển TCRT5000 và thiết kế 5 mắt cảm biến liên tiếp mỗi mắt cách nhau 2cm.

A close-up of a ruler and a circuit board

AI-generated content may be incorrect.

Hình 4.3: Mô--đun cảm biến dò line

Từ đó em xây dựng bảng mã hóa sai số của xe và đường line theo giá trị nhận được từ cảm biến.

Bảng 4.1: Bảng mã hóa sai số

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| s1 | s2 | s3 | s4 | s5 | Sai số |
| 0 | 1 | 1 | 1 | 1 | 4 |
| 0 | 0 | 1 | 1 | 1 | 3 |
| 1 | 0 | 1 | 1 | 1 | 2 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | -1 |
| 1 | 1 | 1 | 0 | 1 | -2 |
| 1 | 1 | 1 | 0 | 0 | -3 |
| 1 | 1 | 1 | 1 | 0 | -4 |

Với S1, S2, S3, S4 và S5 tương đương với mắt cảm biến 1 đến 5 tính từ trái qua phải. Giá trị cảm biến trả về là “0” khi mắt cảm biến ở trong line, giá trị cảm biến trả về là “1” khi cảm biến ở ngoài line.

Từ đó em tiến hành tính toán bộ điều khiển với giá trị đặt sao cho xe luôn ở chính giữa line tương đương với giá trị đặt bằng 0.

Mô hình thuật toán PID:

A diagram of a machine

AI-generated content may be incorrect.

Hình 4.4: Mô hình bộ điều khiển PID

Giá trị đầu ra của thuật toán PID:

Tuy nhiên hiết kế một bộ điều khiển PID kỹ thuật số trên một Vi điều khiển (MCU) hoặc thiết bị FPGA yêu cầu dạng chuẩn của bộ điều khiển PID phải được rời rạc hóa Vi phân bậc một được xác định bằng sai [phân hữu hạn](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A2n_h%E1%BB%AFu_h%E1%BA%A1n&action=edit&redlink=1" \o "Phân hữu hạn (trang không tồn tại)) lùi. Khâu tích phân được rời rạc hóa, với thời gian lấy mẫu Δ, như sau:

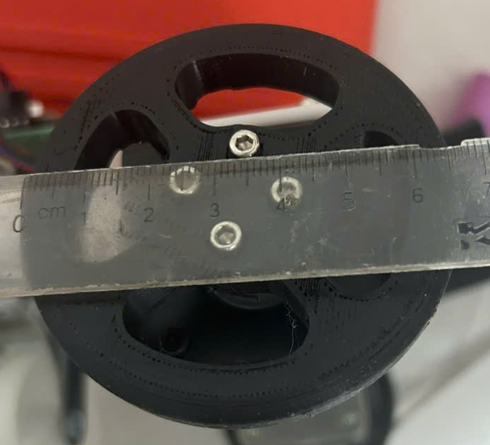
Khâu vi phân sẽ xác định bởi:

Từ đó em tính được giá trị PID của vận tốc trong miền rời rạc:

Bộ điều khiển PID có nhiệm vụ làm giảm sai lệch về 0 tương đương với việc điều khiển để xe đi thẳng với đường line. Để điều khiển hướng của xe em sẽ điều khiển tốc độ của 2 động cơ.

Loại động cơ sử dụng theo như tính toán là STEP NEMA17, số bước trên 1 vòng là 200 bước. Để làm giảm độ rung của động cơ cũng như để thuận tiện trong quá trình điều khiển em sử dụng số bước chia là 8 trên module A4988 tương đương với việc động cơ cần 1600 bước để hoàn thành 1 vòng.

Với đường kính bánh xe 6.4cm



Hình 4.5: Đường kính bánh xe

Từ đó 1 vòng xe di chuyển được quãng đường xấp xỉ là 20cm.

Để xe rẽ trái hay rẽ phải em sẽ điều chỉnh tốc độ của 2 bên động cơ như sau nếu muốn rẽ phải em sẽ tăng tốc động cơ bên phải và giảm tốc độ động cơ bên trái, ngược lại nếu muốn rẽ trái em sẽ làm ngược lại.

Từ bảng sai số được mã hóa trên kết hợp với nguyên tắc điều khiển trái phải. Khi cảm biến cho ra sai số là dương em sẽ rẽ trái, còn cảm biển cho sai số là âm em sẽ rẽ phải. Qua đó em có công thức tính tốc độ của các động cơ:

Và để điều khiển tốc độ của xe em sẽ điều khiển số bước đi được trong 1 giây từ đó em có được góc rẽ như sau:

A graph of a line

AI-generated content may be incorrect.

Hình 4.6: Mô hình quỹ độ đường đi của xe

Với X là số vòng quay trong 1 giây của động cơ bên phải, Y là số vòng quay trong 0.1 giây của động cơ bên trái, góc rẽ là A, l là chiều rộng của xe

Góc rẽ xẽ được tính như sau:

Suy ra

Để xe có thể quay góc 90 độ và để xe có thể đáp ứng tốt, em tiến hành thử nghiệm và chọn hệ số PID, từ đó tính ra được các hệ số giúp xe ổn định nhất là , số vòng quay ban đầu là 1900 vòng/s.

## Lập trình cho hệ thống

### Phần mềm Arduino IDE

**Giao diện phần mềm Arduino IDE**

Môi trường phát triển tích hợp Arduino IDE là một ứng dụng đa nền tảng được viết bằng Java, và được dẫn xuất từ IDE cho ngôn ngữ lập trình xử lý và các dự án lắp ráp. Nó bao gồm một trình soạn thảo mã với các tính năng như làm nổi bật cú pháp, khớp dấu ngoặc khối chương trình, thụt đầu dòng tự động và cũng có khả năng biên dịch và tải lên các chương trình vào board mạch với một nhấp chuột duy nhất. Một chương trình hoặc mã viết cho Arduino được gọi là “sketch”.

Chương trình Arduino được viết bằng C hoặc C++. Arduino IDE đi kèm với một thư viện phần mềm được gọi là “Writing” từ dự án lắp ráp ban đầu, cho hoạt động đầu vào/đầu ra phổ biến trở nên dễ dàng hơn nhiều. Người sử dụng chỉ cần định nghĩa hai hàm để thực hiện một chương trình điều hành theo chu kỳ.

Khi chúng ta bật điện bảng mạch Arduino, reset hay nạp chương trình mới, hàm setup() sẽ được gọi đến đầu tiên. Sau khi xử lý xong hàm setup(), Arduino sẽ nhảy đến hàm loop() và lặp vô hạn hàm này cho đến khi tắt điện board mạch Arduino. Chu trình đó có thể mô tả trong hình dưới đây:

A diagram of a system

Description automatically generated

Hình 4.7: Mô tả hoạt động của chương trình Arduino

Arduino IDE sử dụng GNU toolchain và AVR libc để biên dịch chương trình và sử dụng avrdule để tải lên các chương trình vào board mạch chủ. Do nền tảng Arduino sử dụng vi điều khiển Atmel, môi trường phát triển Atmel, AVR Studio hoặc Atmel Studio mới hơn, cũng có thể được sử dụng để phát triển phần mềm cho Arduino.

Arduino IDE là nơi để soạn thảo code, kiểm tra lỗi và upload code cho Arduino.

A screenshot of a computer

Description automatically generated

Hình 4.8: Giao diện Arduino IDE

**Cấu trúc lệnh của một chương trình trong phần mềm IDE**

*Phần 1*: Khai báo biến

Đây là phần khai báo kiểu biến, tên các biến, định nghĩa các chân trên board, một số kiểu khai báo biến thông dụng:

- #define : Nghĩa của từ define là định nghĩa, hàm #define có tác dụng định nghĩa, hay còn gọi là gán, tức là gán một chân, một ngõ ra nào đó với một cái tên. Ví dụ #define led 13. (Chú ý: sau #define thì không có dấu “;”).

*Phần 2*: Thiết lập “void setup()”

Phần này dùng để thiết lập cho chương trình:

A white background with black text

Description automatically generated

Hình 4.9: Cấu trúc của void setup ()

Cấu trúc của void setup () có dấu ngặc nhọn ở đầu và cuối, nếu phần này bị thiếu khi chạy chương trình thì chương trình sẽ báo lỗi. Trong câu lệnh này dùng các lệnh để thiết lập các tốc độ truyền dữ liệu, kiểu chân IN/OUT. Trong đó:

Serial.begin(19200): Dùng để truyền dữ liệu từ board Arduino lên máy tính.

pinMode(biến, INPUT/OUTPUT): Dùng để xác định kiểu chân là vào hay ra.

*Phần 3:* Vòng lặp

Dùng để viết các câu lệnh trong chương trình để mạch Arduino thực hiện các nhiệm vụ mà người lập trình mong muốn, được bắt đầu bằng “void loop”.

A white background with black text

Description automatically generated

Hình 4.10: Cấu trúc của void loop()

Với Arduino IDE, ta có thể viết mã bằng ngôn ngữ lập trình dựa trên C/C++ và sử dụng các thư viện có sẵn để tương tác với các linh kiện điện tử kết nối với board Arduino. Sau khi viết xong mã, có thể biên dịch chương trình và tải lên board Arduino thông qua kết nối USB.

Arduino IDE cũng cung cấp các công cụ gỡ lỗi đơn giản để giúp người sử dụng theo dõi và sửa lỗi trong chương trình của mình.

Ngoài ra, cộng đồng Arduino rất lớn và hoạt động tích cực. Có thể tìm thấy rất nhiều tài liệu, ví dụ mã nguồn, và ví dụ ứng dụng của Arduino trên trang web và diễn đàn của Arduino.cc và các nguồn thông tin khác.

**Thuật toán điều khiển**

#define BLYNK\_TEMPLATE\_ID "TMPL6sv4EUUKx"

#define BLYNK\_TEMPLATE\_NAME "smartdoor"

#define BLYNK\_AUTH\_TOKEN "9QNPdXlYXu2TSKjXZW5bTiFlllEdWp4X"

#include <Servo\_ESP32.h>

#include <BlynkSimpleEsp32.h>

#include <HCSR04.h>

#include <WiFi.h>

#include <WiFiClient.h>

#define ON\_OFF   V0

#define PID\_BLK  V1

#define LD    2

#define LE    16

#define LS    4

#define RD    5

#define RS    17

#define RE    18

#define S1    36

#define S2    34

#define S3    35

#define S4    32

#define S5    33

static const int servoPin = 13; //printed G14 on the board

Servo\_ESP32 servo1;

const int PWM\_CHANNEL = 0;    // ESP32 has 16 channels which can generate 16 independent waveforms

const int PWM\_FREQ = 1900;     // Recall that Arduino Uno is ~490 Hz. Official ESP32 example uses 5,000Hz

const int PWM\_RESOLUTION = 8; // We'll use same resolution as Uno (8 bits, 0-255) but ESP32 can go up to 16 bits

int PWM\_FREQ\_L;

int PWM\_FREQ\_R;

// The max duty cycle value based on PWM resolution (will be 255 if resolution is 8 bits)

const int MAX\_DUTY\_CYCLE = (int)(pow(2, PWM\_RESOLUTION) - 1);

const int LCHANNEL = 0U;

const int RCHANNEL = 2U;

const int DELAY\_MS = 4;  // delay between fade increments

const int trig = 21;     // chân trig của HC-SR04

const int echo = 19;     // chân echo của HC-SR04

int error; int previous\_error = 0;

int P; int Kp = 50;

int I; int Ki =10;

int D; int Kd =0;

int PID\_value;

char auth[] = BLYNK\_AUTH\_TOKEN;                // Gán mã token của app blynk vào biến auth

char ssid[] = "Tang8";          // Khai báo tên wifi kết nối và gán vào biến ssid

char pass[] = "123456789@";                      // Khai báo mật khẩu wifi và gán vào biến pass

BlynkTimer timer;

BLYNK\_CONNECTED() {

  Blynk.syncVirtual(ON\_OFF);

  Blynk.syncVirtual(PID\_BLK);

}

uint8\_t bt\_state\_blk;

BLYNK\_WRITE(ON\_OFF)

{

  bt\_state\_blk=param.asInt();

}

BLYNK\_WRITE(V2)

{

  Kp=param.asInt();

}

BLYNK\_WRITE(V3)

{

  Ki=param.asInt();

}

BLYNK\_WRITE(V4)

{

  Kd=param.asInt();

}

uint8\_t s[5];

void Read\_sensor()

{

  s[0] = digitalRead(S1);

  s[1] = digitalRead(S2);

  s[2] = digitalRead(S3);

  s[3] = digitalRead(S4);

  s[4] = digitalRead(S5);

  Serial.print(s[0] ); Serial.print(s[1] ); Serial.print(s[2] ); Serial.print(s[3] ); Serial.print(s[4] );

  Serial.println();

  if(s[0] == 1 && s[1] == 1 && s[2] == 1 && s[3] == 1 && s[4] == 1 && previous\_error > 0) {error = 5;}

  if(s[0] == 0 && s[1] == 1 && s[2] == 1 && s[3] == 1 && s[4] == 1) {error = 4;}

  if(s[0] == 0 && s[1] == 0 && s[2] == 1 && s[3] == 1 && s[4] == 1) {error = 3;}

  if(s[0] == 1 && s[1] == 0 && s[2] == 1 && s[3] == 1 && s[4] == 1) {error = 2;}

  if(s[0] == 1 && s[1] == 0 && s[2] == 0 && s[3] == 1 && s[4] == 1) {error = 1;}

  if(s[0] == 1 && s[1] == 1 && s[2] == 0 && s[3] == 1 && s[4] == 1) {error = 0;}

  if(s[0] == 1 && s[1] == 1 && s[2] == 0 && s[3] == 0 && s[4] == 1) {error = -1;}

  if(s[0] == 1 && s[1] == 1 && s[2] == 1 && s[3] == 0 && s[4] == 1) {error = -2;}

  if(s[0] == 1 && s[1] == 1 && s[2] == 1 && s[3] == 0 && s[4] == 0) {error = -3;}

  if(s[0] == 1 && s[1] == 1 && s[2] == 1 && s[3] == 1 && s[4] == 0) {error = -4;}

  if(s[0] == 1 && s[1] == 1 && s[2] == 1 && s[3] == 1 && s[4] == 1 && previous\_error < 0) {error = -5;}

  if(s[0] == 1 && s[1] == 1 && s[2] == 0 && s[3] == 1 && s[4] == 0) {error = -10;}

  if(s[0] == 1 && s[1] == 0 && s[2] == 0 && s[3] == 0 && s[4] == 1) {error = 10;}

}

void pid\_calculation(){

 P = error;

 I = I + error;

 D = D - previous\_error;

 PID\_value = Kp\*P + Ki\*I + Kd\*D;

 previous\_error = error;

//Serial.print("PID: ");Serial.println(PID\_value);

}

int initial\_motor\_speed = 50;

int left\_motor\_speed;

int right\_motor\_speed;

void speedControl(){

  digitalWrite(RE, 0);

  digitalWrite(LE, 0);

  digitalWrite(RD, 0);

  digitalWrite(LD, 0);

  PWM\_FREQ\_L = PWM\_FREQ - PID\_value ;

  PWM\_FREQ\_R = PWM\_FREQ + PID\_value ;

  if(PWM\_FREQ\_L < 800) PWM\_FREQ\_L = 800;

  if(PWM\_FREQ\_L > 3000) PWM\_FREQ\_L = 3000;

  if(PWM\_FREQ\_R < 800) PWM\_FREQ\_R = 800;

  if(PWM\_FREQ\_R > 3000) PWM\_FREQ\_R = 3000;

  ledcSetup(LCHANNEL, PWM\_FREQ\_L, PWM\_RESOLUTION);

  ledcSetup(RCHANNEL, PWM\_FREQ\_R, PWM\_RESOLUTION);

  ledcWrite(LCHANNEL, 120);

  ledcWrite(RCHANNEL, 120);

  Serial.print(PWM\_FREQ\_L); Serial.print("   "); Serial.println(PWM\_FREQ\_R);

}

int measure\_distance()

{

  unsigned long duration; // biến đo thời gian

    int distance;           // biến lưu khoảng cách

    /\* Phát xung từ chân trig \*/

    digitalWrite(trig,0);   // tắt chân trig

    delayMicroseconds(2);

    digitalWrite(trig,1);   // phát xung từ chân trig

    delayMicroseconds(5);   // xung có độ dài 5 microSeconds

    digitalWrite(trig,0);   // tắt chân trig

    /\* Tính toán thời gian \*/

    // Đo độ rộng xung HIGH ở chân echo.

    duration = pulseIn(echo,HIGH);

    // Tính khoảng cách đến vật.

    distance = int(duration/2/29.412);

  return distance;

}

void turn\_right()

{

  /\* Motor right stop motor left full speed \*/

  digitalWrite(RE, 0);

  digitalWrite(LE, 0);

  digitalWrite(RD, 0);

  digitalWrite(LD, 0);

  ledcSetup(LCHANNEL, PWM\_FREQ, 1200);

  ledcSetup(RCHANNEL, PWM\_FREQ\_R, 1200);

  digitalWrite(RD, 0);

  digitalWrite(LD, 0);

  ledcWrite(LCHANNEL, 120);

  ledcWrite(RCHANNEL, 0);

  delay(1000);

  ledcWrite(LCHANNEL, 120);

  ledcWrite(RCHANNEL, 120);

  delay(1000);

  ledcWrite(LCHANNEL, 0);

  ledcWrite(RCHANNEL, 120);

  delay(1000);

  digitalWrite(RE, 1);

  digitalWrite(LE, 1);

}

void turn\_left()

{

  /\* Motor right stop motor left full speed \*/

  digitalWrite(RE, 0);

  digitalWrite(LE, 0);

  digitalWrite(RD, 0);

  digitalWrite(LD, 0);

  ledcSetup(LCHANNEL, PWM\_FREQ, 1200);

  ledcSetup(RCHANNEL, PWM\_FREQ\_R, 1200);

  digitalWrite(RD, 0);

  digitalWrite(LD, 0);

  ledcWrite(LCHANNEL, 0);

  ledcWrite(RCHANNEL, 120);

  delay(1000);

  ledcWrite(LCHANNEL, 120);

  ledcWrite(RCHANNEL, 120);

  delay(1000);

  ledcWrite(LCHANNEL, 120);

  ledcWrite(RCHANNEL, 0);

  delay(1000);

  digitalWrite(RE, 1);

  digitalWrite(LE, 1);

}

int Distance\_C = 0, Distance\_L = 0, Distance\_R = 0;

uint8\_t checking()

{

  servo1.write(90);

  Distance\_C =  measure\_distance();

  Serial.print("Kc trung tam: "); Serial.println(Distance\_C);

  if(Distance\_C < 16 && Distance\_C > 5)

  {

    digitalWrite(RE, 1);

    digitalWrite(LE, 1);

    delay(400);

    servo1.write(180);

    Distance\_L = measure\_distance();

    Serial.print("Kc ben trai: "); Serial.println(Distance\_L);

    delay(400);

    servo1.write(0);

    Distance\_R = measure\_distance();

    Serial.print("Kc ben phai: "); Serial.println(Distance\_R);

    delay(400);

    if(Distance\_L < 16 && Distance\_R > 16 && Distance\_L > 5 && Distance\_R > 5)

    {

      // re phai

      return 1;

    }

    else if(Distance\_L > 16 && Distance\_R < 16 && Distance\_L > 5 && Distance\_R > 5)

    {

      // re trai

      return 2;

    }

    else if(Distance\_L > 16 && Distance\_R > 16 && Distance\_L > 5 && Distance\_R > 5)

    {

      // vat can o giua

      return 3;

    }

    else

    {

      return 4;

      // vat ca 3 phia nghi chay

    }

  }

  return 0;

}

void setup() {

  // put your setup code here, to run once:

  pinMode(LS,OUTPUT);

  pinMode(LE,OUTPUT);

  pinMode(LD,OUTPUT);

  pinMode(RS,OUTPUT);

  pinMode(RE,OUTPUT);

  pinMode(RD,OUTPUT);

  pinMode(S1, INPUT);

  pinMode(S2, INPUT);

  pinMode(S3, INPUT);

  pinMode(S4, INPUT);

  pinMode(S5, INPUT);

  pinMode(trig,OUTPUT);   // chân trig sẽ phát tín hiệu

  pinMode(echo,INPUT);    // chân echo sẽ nhận tín hiệu

  Blynk.begin(auth, ssid, pass); // Chạy blynk với các thông tin ta đã cung cấp

  // Sets up a channel (0-15), a PWM duty cycle frequency, and a PWM resolution (1 - 16 bits)

  // ledcSetup(uint8\_t channel, double freq, uint8\_t resolution\_bits);

  ledcSetup(LCHANNEL, PWM\_FREQ, PWM\_RESOLUTION);

  ledcSetup(RCHANNEL, PWM\_FREQ, PWM\_RESOLUTION);

  // ledcAttachPin(uint8\_t pin, uint8\_t channel);

  ledcAttachPin(RS, LCHANNEL);

  ledcAttachPin(LS, RCHANNEL);

  servo1.attach(servoPin, 5, 0, 180, 544, 2400);

  Serial.begin(115200);

}

void loop() {

  // put your main code here, to run repeatedly:

  Blynk.run();

  timer.run();

  uint8\_t isTurn = 0;

  if(bt\_state\_blk == 0)

  {

      digitalWrite(LE, 1);

      digitalWrite(RE, 1);

      PID\_value = 0;

      isTurn = 0;

    Distance\_C = 0;

    Distance\_L = 0;

    Distance\_R = 0;

  }

  else

  {

    Distance\_C = 0;

    Distance\_L = 0;

    Distance\_R = 0;

    isTurn = checking();

    if(isTurn == 1)

    {

      turn\_right();

      digitalWrite(LE, 1);

      digitalWrite(RE, 1);

      isTurn = 0;

    Distance\_C = 0;

    Distance\_L = 0;

    Distance\_R = 0;

          PID\_value = 0;

      delay(1000);

    }

    else if(isTurn == 2 || isTurn == 3)

    {

      turn\_left();

      digitalWrite(LE, 1);

      digitalWrite(RE, 1);

      isTurn = 0;

    Distance\_C = 0;

    Distance\_L = 0;

    Distance\_R = 0;

     PID\_value = 0;

      delay(1000);

    }

    else if(isTurn == 4)

    {

      digitalWrite(LE, 1);

      digitalWrite(RE, 1);

      isTurn = 0;

    Distance\_C = 0;

    Distance\_L = 0;

    Distance\_R = 0;

          PID\_value = 0;

    }

    else if(isTurn == 0)

    {

      digitalWrite(LE, 0);

      digitalWrite(RE, 0);

      Read\_sensor();

      pid\_calculation();

      speedControl();

      Blynk.virtualWrite(PID\_BLK , PID\_value);

    }

  }

}

# THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

Mục đích của chương 5 là giúp em đánh giá được độ hoàn thiện của thiết bị, bên cạnh đó kiểm tra tính ổn định của mạch điện được thiết kết. Các việc kiểm tra bao gồm:

## Thử nghiệm phần cứng

### Kiểm tra hoạt động của các nguồn

Bảng 5.1Bảng đo giá trị của nguồn điện:

|  |  |  |  |
| --- | --- | --- | --- |
| Thông số cần đo | Giá trị tính toán | Giá trị trung bình sau 10 lần đo | Sai só |
| Điện áp nguồn | 5 | 4.99 | 0.1% |
| Điện áp cảm biến | 3.3 | 3.29 | 0.1% |
| Điện áp A4988 | 5 | 4.99 | 0.1% |

## Lắp ráp và thử nghiệm toàn bộ

## Đánh giá kết quả

Qua những gì em đã đo đạc, kiểm tra và thử nghiệm mạch điện, em đã thu được những kết quả sau:

Mạch điện hoạt động ổn định, các giá trị điện áp thu được đúng so với với mong muốn

Xe chạy ổn định, cảm biến dò line hoạt động chính xác, bộ điều khiển PID hoạt động tốt xe chạy không có hiện tượng giật.

Hạn chế: Tuy nhiên do thời gian còn hạn chế và nghiên cứu chưa đủ, sản phẩm của em vẫn còn một vài hạn chế như:

+) Tốc độ xe chạy chưa được nhanh:

+) Khả năng tránh vật cản còn hạn chế

+) Xe vẫn chưa được kiểm nghiệm khả năng mang tải.

Phương hướng phát triển:

Trong tương lai, em sẽ tiếp tục nghiên cứu và phát triển các version để hoạt động được ổn định hơn, lâu hơn có khả năng mang tải và di chuyển ở tốc độ cao, không phải dừng để tránh vật cản.

**TÀI LIỆU THAM KHẢO**

[1]. Vũ Quang Hồi, giáo trình kỹ thuật cảm biến, nhà xuất bản giáo dục Việt Nam, 2015.

[2]. Mr.M.Anantha Guptha – Assistant Professor “*Internet of things & Its Applications*”, Malla Reddy College of Engineering & Technology.

[3]. Gauri R Choudhari et al “*IoT-Base Smart Gardening System*”, Joural of Physical: Conference Series, 2023.

[4]. Yan Sen Tan, Sean Zhuang Tan, Li Wei Chew, and Yi Xuen Tan “*IoT – base Smart Farming System*”, Internartional Joural of Emerging Multidisciplinaries: Computer Science and Artificial Intelligence, 2024.

[5]. Vinoth Kumar.P, K.C Ramya, Abishek.J.S, Arundhathy.T.S, Bhavvya.B,  
Gayathri.V, Smart Garden Monitoring and Control System with Sensor Technology, 2021 3rd Int. Conference on Signal Processing and Communication, 2021, pp. 93-97.

[6]. <https://nshopvn.com/>

[7]. https://vi.wikipedia.org/wiki/Internet\_Vạn\_Vật

[8]. <https://khuenguyencreator.com/lap-trinh-esp32-tu-a-toi-z/>