|  |
| --- |
| HỌC VIỆN KỸ THUẬT MẬT MÃ  **KHOA CÔNG NGHỆ THÔNG TIN**  Logo HvKTMM  **BÁO CÁO MÔN HỌC**  **THIẾT KẾ HỆ THỐNG NHÚNG**  **ĐỀ TÀI**: THIẾT KẾ MÔ HÌNH XE TỰ HÀNH TRÁNH VẬT CẢN  Nhóm sinh viên thực hiện:  **Hoàng Văn Tân – CT030248**  **Nguyễn Đình Trung – CT030254**  **Nguyễn Văn Chiến – CT030206**  Giảng viên hướng dẫn:  TS. **Phạm Văn Hưởng**  Khoa Công nghệ thông tin – Học viện Kỹ thuật Mật mã  Hà Nội, 03-2022 |

**MỤC LỤC**

[DANH MỤC HÌNH VẼ 3](#_Toc98877826)

[DANH MỤC BẢNG BIỂU 4](#_Toc98877827)

[LỜI CẢM ƠN 5](#_Toc98877828)

[LỜI MỞ ĐẦU 6](#_Toc98877829)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 7](#_Toc98877830)

[1.1 Tổng quan về đề tài 7](#_Toc98877831)

[1.1.1 Khảo sát tình hình thực tế 7](#_Toc98877832)

[1.1.2 Một số thông tin về đề tài 9](#_Toc98877833)

[1.2 Tổng quan Proteus và Aduino IDE 10](#_Toc98877834)

[1.2.1 Công cụ Proteus 10](#_Toc98877835)

[1.2.2 Arduino IDE 12](#_Toc98877836)

[CHƯƠNG 2. GIỚI THIỆU CÁC LINH KIỆN ĐƯỢC SỬ DỤNG 15](#_Toc98877837)

[2.1 Arduino UNO R3 15](#_Toc98877838)

[2.2 Module điều khiển động cơ L298 16](#_Toc98877839)

[2.3 Cảm biến siêu âm SRF 04 18](#_Toc98877840)

[2.4 Động cơ điện một chiều 23](#_Toc98877841)

[2.5 Module bluetooth RN-42 trên HC-05 24](#_Toc98877842)

[CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ 26](#_Toc98877843)

[3.1 Sơ đồ khối và nguyên lý hoạt động của mạch 26](#_Toc98877844)

[3.2 Sơ đồ khối và nguyên lý hoạt động của mạch chế độ tự hành 28](#_Toc98877845)

[3.3 Sơ đồ khối và nguyên lý hoạt động của toàn mạch 29](#_Toc98877846)

[3.4 Biểu đồ use-case 30](#_Toc98877847)

[3.5 Biểu đồ tuần tự 33](#_Toc98877848)

[3.6 Thiết kế mạch và mô hình thực tế 34](#_Toc98877849)

[3.7 Chương trình Code 36](#_Toc98877850)

[KẾT LUẬN 43](#_Toc98877851)

[TÀI LIỆU THAM KHẢO 44](#_Toc98877852)

DANH MỤC HÌNH VẼ

[Hình 1. 1 Ngôn ngữ HTML 7](#_Toc95979402)

[Hình 1. 2 Ngôn ngữ CSS 8](#_Toc95979403)

[Hình 1. 3 Ngôn ngữ JavaScript 9](#_Toc95979404)

[Hình 1. 4 Ngôn ngữ NodeJs 10](#_Toc95979405)

[Hình 1. 5 Framework Express JS 11](#_Toc95979406)

[Hình 1. 6 Thư viện ReactJS 12](#_Toc95979407)

[Hình 1. 7 Hình ảnh Socket.io 13](#_Toc95979408)

[Hình 1. 8 Hình ảnh MongoDB 15](#_Toc95979409)

[Hình 2. 1 Hình ảnh Thống kê người dùng trên các mạng xã hội 18](#_Toc95982444)

[Hình 2. 2 Hình ảnh Thống kê người dùng mạng xã hội tại Việt Nam 19](#_Toc95982445)

[Hình 2. 3 Biểu đồ tuần tự chức năng đăng ký 22](#_Toc95982446)

[Hình 2. 4 Biểu đồ tuần tự chức năng đăng nhập 23](#_Toc95982447)

[Hình 2. 5 Biểu đồ tuần tự quản lý thông tin 25](#_Toc95982448)

[Hình 2. 6 Biểu đồ tuần tự thêm bài viết 26](#_Toc95982449)

[Hình 2. 7 Biểu đồ tuần tự xóa bài viết 27](#_Toc95982450)

[Hình 2. 8 Biểu đồ tuần tự chức năng thả cảm xúc 29](#_Toc95982451)

[Hình 2. 9 Biểu đồ tuần tự chức năng bình luận bài viết 30](#_Toc95982452)

[Hình 2. 10 Biểu đồ tuần tự chức năng chia sẻ 31](#_Toc95982453)

[Hình 2. 11 Biểu đồ tuần tự chức năng tìm kiếm 32](#_Toc95982454)

[Hình 2. 12 Biểu đồ tuần tự chức năng theo dõi 33](#_Toc95982455)

[Hình 2. 13 Biểu đồ tuần tự chức năng nhắn tin 34](#_Toc95982456)

[Hình 2. 14 Hình ảnh Giao diện Login 37](#_Toc95982457)

[Hình 2. 15 Hình ảnh Giao diện Register 37](#_Toc95982458)

[Hình 2. 16 Hình ảnh Giao diện trang chủ 38](#_Toc95982459)

[Hình 2. 17 Hình ảnh Giao diện trang tin nhắn 38](#_Toc95982460)

[Hình 2. 18 Hình ảnh Giao diện thông tin cá nhân 39](#_Toc95982461)

[Hình 2. 19 Hình ảnh Giao diện thông báo 39](#_Toc95982462)

[Hình 2. 20 Hình ảnh Chức năng đăng nhập 40](#_Toc95982463)

[Hình 2. 21 Hình ảnh Chức năng đăng ký 40](#_Toc95982464)

[Hình 2. 22 Hình ảnh Chức năng đăng bài viết 41](#_Toc95982465)

[Hình 2. 23 Hình ảnh Chức năng tìm kiếm người dùng 41](#_Toc95982466)

[Hình 2. 24 Hình ảnh Chức năng nhắn tin 42](#_Toc95982467)

[Hình 2. 25 Hình ảnh Chức năng chỉnh sửa thông tin cá nhân 42](#_Toc95982468)

[Hình 2. 26 Hình ảnh Chức năng chia sẻ bài viết 43](#_Toc95982469)

[Hình 2. 27 Hình ảnh Chức năng bình luận và thả cảm xúc bài viết 43](#_Toc95982470)

[Hình 2. 28 Hình ảnh Chức năng theo dõi/hủy theo dõi 43](#_Toc95982471)

[Hình 2. 29 Hình ảnh Chức năng thông báo 44](#_Toc95982472)

[Hình 3. 1 Thực nghiệm đăng ký 45](#_Toc95982473)

[Hình 3. 2 Thực nghiệm đăng nhập 46](#_Toc95982474)

[Hình 3. 3 Thực nghiệm trang chủ 47](#_Toc95982475)

[Hình 3. 4 Thực nghiệm trang thông tin cá nhân 47](#_Toc95982476)

[Hình 3. 5 Thực nghiệm chỉnh sửa thông tin cá nhân 48](#_Toc95982477)

[Hình 3. 6 Thực nghiệm trang nhắn tin 48](#_Toc95982478)

[Hình 3. 7 Thực nghiệm chức năng gọi điện 49](#_Toc95982479)

[Hình 3. 8 Thực nghiệm chế độ ban đêm 49](#_Toc95982480)

DANH MỤC BẢNG BIỂU

[Bảng 2. 1 Đặc tả Use Case chức năng Đăng ký 20](#_Toc95851895)

[Bảng 2. 2 Đặc tả usecase đăng nhập 21](#_Toc95851896)

[Bảng 2. 3 Đặc tả usecase quản lý thông tin cá nhân 23](#_Toc95851897)

[Bảng 2. 4 Bảng Đặc tả usecase thêm bài viết 24](#_Toc95851898)

[Bảng 2. 5 Bảng Đặc tả usecase xóa bài viết 25](#_Toc95851899)

[Bảng 2. 6 Bảng Đặc tả usecase thả cảm xúc 27](#_Toc95851900)

[Bảng 2. 7 Bảng Đặc tả usecase bình luận bài viết 28](#_Toc95851901)

[Bảng 2. 8 Bảng Đặc tả usecase chia sẻ bài viết 29](#_Toc95851902)

[Bảng 2. 9 Bảng Đặc tả usecase tìm kiếm người dùng 30](#_Toc95851903)

[Bảng 2. 10 Bảng Đặc tả usecase theo dõi 31](#_Toc95851904)

[Bảng 2. 11 Bảng Đặc tả usecase nhắn tin 32](#_Toc95851905)

[Bảng 2. 12 Bảng User 33](#_Toc95851906)

[Bảng 2. 13 Bảng Post 34](#_Toc95851907)

[Bảng 2. 14 Bảng Chat 34](#_Toc95851908)

[Bảng 2. 15 Bảng Notify 34](#_Toc95851909)

[Bảng 2. 16 Bảng Chat 35](#_Toc95851910)

[Bảng 2. 17 Bảng Message 35](#_Toc95851911)

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn thầy Phạm Văn Hưởng trong khoa công nghệ thông tin đã tận tình chỉ bảo và giúp đỡ chúng em trong thời gian làm đề tài.

Do thời gian, tài liệu và trình độ còn hạn chế nên không thể tránh khỏi những thiếu sót trong quá trình làm đề tài. Nhóm em mong được sự chỉ dẫn và góp ý của thầy và các bạn.

# LỜI MỞ ĐẦU

Ngày nay,nền khoa học kỹ thuật đang phát triển như vũ bão, rất nhiều các sản phẩm công nghệ, phát minh mới ra đời. Các robot tự động đang là xu thế công nghệ tương lai vì sự tiện dụng, khả năng hoạt động chính xác trong các điều kiện khắc nghiệt với con người. Một yêu cầu cơ bản của các robot tự động là khả năng hoạt động theo chức năng mong muốn mà không cần sự can thiệp của con người trong quá trình làm việc. Đây là cơ hội cũng như là thách thức cho các sinh viên kỹ thuật, công nghệ thông tin nói chung và sinh viên Học Viện Kỹ Thuật Mật Mã nói riêng.

Xe tự hành là một ý tưởng robot cơ bản nhất, có thể sử dụng cho mục địch vận tải trong quy mô nhỏ hoặc kết hợp với các modun khác để thực hiện các nhiệm vụ phức tạp hơn.

Trong thời lượng hạn chế của môn học, với kinh nghiệm còn ít của nhóm. Chúng em không thể tránh khỏi những sai sót trong quá trình thực hiện, rất mong được sự góp ý của thầy và các bạn để đề tài của bọn em được hoàn thiệt hơn.

# CƠ SỞ LÝ THUYẾT

## Tổng quan về đề tài

### Khảo sát tình hình thực tế

Ngày nay, với những ứng dụng của khoa học kỹ thuật tiên tiến, thế giới của chúng ta đã và đang ngày một thay đổi, văn minh và hiện đại hơn. Với sự phát triển vượt bậc của khoa học kỹ thuật đã cho ra đời nhưng sản phẩm vi mạch tiên tiến với tốc độ vượt trội. Những vi mạch đó được áp dụng vào nhưng chiếc xe tự vận hành hay những con Robot để giúp con người hoàn thành những công việc từ đơn giản đến phức tạp trong cuộc sống  
 Theo dự đoán thì sau cuộc cách mạng lớn về internet sẽ là cuộc cách mạng về lĩnh vực robot. Các Robot được ứng dụng trong đời sống ngày càng nhiều như robot vận chuyển hàng hóa, robot kiểm tra nguy hiểm, xe lăn cho người khuyết tật. Robot phục vụ sinh hoạt gia đình… Điểm hạn chế của các robot và xe tự hành hiện tại là tính thiếu linh hoạt và khả năng thích ứng khi làm việc ở những vị trí khác nhau. Từ những lý do đó nảy sinh vấn đề tránh vật cản cho robot tự hành nhằm nâng cao tính linh hoạt cho robot. Hầu hết các robot và xe tự hành hiện đại đều có một kiểu tránh vật cản nào đó giúp robot xác định vật cản và dừng lại ở khoảng cách ngắn so với vật cản để tránh va chạm, đến các thuật toán tinh tế hơn, cho phép robot di chuyển theo đường viền quanh vật cản.

***Giới thiệu về xe tự hành,tự động tránh vật cản*** Xe tự hành hay robot di động (mobile robots, thường được gọi tắt là mobots) được định nghĩa là một loại xe robot có khả năng tự dịch chuyển, tự vận động (có thể lập trình lại được) dưới sự điền khiển tự động để thực hiện thành công công việc được giao. Theo lý thuyết, môi trường hoạt động của robot tự hành có thể là đất, nước, không khí, không gian vũ trụ hay sự tổ hợp giữa chúng. Địa hình bề mặt mà xe di chuyển trên đó có thể bằng phẳng hoặc thay đổi, lồi lõm.Vậy nên việc trang bị cho xe tính năng tự nhận biết vật cản là việc hoàn toàn cần thiết. Xe tự hành sẽ được chia ra làm 2 phần chính: Phần chuyển động cho xe bao gồm các bánh xe các bộ chuyển động. Phần cảm biến vật thể bao gồm: Cảm biến và mạch adruino chứa code.

***Tình hình phát triển***

Theo dự đoán trong vòng 20 năm nữa nhu cầu sử dụng xe tự hành hay robot tăng lên rất nhiều và nó sẽ trở thành tâm điểm của một cuộc cách mạng lớn sau Internet.Với xu hướng hiện nay, cùng các ứng dụng truyền thống khác vai trò của xe tự hành hoặc robot trong các lĩnh vực y tế, giáo dục, giải trí thậm chí là trong an ninh quốc phòng sẽ vô cùng quan trọng và to lớn.

Robot đã có những bước tiến đáng kể trong hơn nửa thế kỉ qua. Nó giúp con người thực hiện những công việc nặng nhọc, nguy hiểm. Do nhu cầu sử dụng ngày càng nhiều và đặc biệt được ứng dụng trong nhiều lĩnh vực nên nó đã trở thành động lực cho việc phát triển của nghành công nghiệp Robot.

Có rất nhiều loại được quan tâm nhiều trong thời gian trở lại đây, đặc biệt là xe tự hành trên mặt đất hay gọi tắt là AGV ( Autonomous Guided Vehicles ).

AGV góp ích rất nhiều trong các lĩnh vực đời sống từ y tế, an ninh quốc phòng cho đến cuộc sống thường ngày trong gia đình. Ví dụ một vài thiết bị sau đây: Xe lăn tự hành cho người khuyết tật, xe dọn vệ sinh trong gia đình, xe di chuyển tham gia trong các vấn đề về vũ trụ vận hành ngoài trái đất…. và rất nhiều ứng dụng trong các lĩnh vực khác nhau.

***Yêu cầu công nghệ và nhu cầu thực tế***

* Yêu cầu công nghệ :
* Để xe tự hành có thể hoạt động được tốt thì trước tiên phải có một hệ thống di chuyển giúp xe có thể di chuyển được trên bề mặt.
* Và tiếp theo đó là một cảm biến siêu âm để xe có thể nhận biết được vật cản phía trước trong một khoảng cách nhất định và dựa vào thuật toán được thiết lập sẵn để đưa ra hướng di chuyển khác giúp tránh vật cản.
* Hiện tại có rất nhiều cảm biến để có thể giúp xe tránh được vật cản như : Cảm biến siêu âm, cảm biến hồng ngoại, cảm biến từ,…..
* Nhu cầu thực tế :
* Xe tự hành hiện nay được áp dụng trong rất nhiều các lĩnh vực trong đời sống
* Về y tế: Xe lăn tự động cho người khuyết tật, xe vận chuyển người bệnh tự động, xe tự hành chở các thiết bị vật tư y tế,….
* Về đời sống: Robot lau dọn nhà thông minh, Robot đo lường khoảng cách trong xây dựng nhà,….
* Về an ninh quốc phòng: Xe tự hành do thám, xe tự hành kiểm tra chất nổ,…
* Về vũ trụ : Xe tự hành trên bề mặt hành tinh, xe tự hành thu thập hình ảnh và các mẫu vật
* Về xây dựng: Xe tự hành chở vật liệu xây dựng,……

Và còn rất nhiều ứng dụng thực tế khác.

### Một số thông tin về đề tài

* **Lý do chọn đề tài**

Hiện nay, đời sống ngày càng được nâng cao, các máy móc tự động có tính ứng dụng cao và ngày càng được sử dụng rộng rãi trong đời sống nhằm phục vụ cho con người với nhiều mục đích khác nhau như: nhà thông minh, nông trại thông minh, máy hút bụi tự động, xe dò đường. Do kiến thức còn hạn chế và trong phạm vi của môn học chúng em xin phép được chọn đề tài làm xe tự động tránh vật cản. Trong thực tế và tương lai để tài này có thể được bổ sung thêm chức năng, nâng cấp để tạo ra những chiếc xe tự hành hay những con robot tự động cao cấp,hiện đại và có tính ứng dụng cao hơn trong cuộc sống.

* **Mục tiêu của đề tài**

Cài đặt và thực nghiệm xe tự động tránh vật cản bằng việc áp dụng cảm biến siêu âm SRF-04, module điều khiển động cơ L298 và vi điều khiển (Arduino Uno R3) cho quá trình xử lý tín hiệu trong hệ thống. Cùng với đó kết hợp kết nối Bluetooth từ điện thoại thông minh tới xe thực nghiệm.

* **Phương pháp nghiên cứu và phạm vi nghiên cứu**
* Phương pháp nghiên cứu:

Tìm hiểu về lý thuyết về vi xử lý arduino, cách kết nối, sử dụng cảm biến siêu âm SRF-04 và module điều khiển động cơ L298 với arduino. Sử dụng phần mềm Proteus để thiết kế mạch, từ đó kết nối các linh kiện và sử dụng phần mềm Arduino IDE để viết chương trình điều khiển cho xe hoạt động như ý muốn.

* Phạm vi nghiên cứu: Thực hiện mô hình thử nghiệm thông qua kiến thức bài học.

## Tổng quan Proteus và Aduino IDE

### Công cụ Proteus

Phần mềm vẽ Proteus là phần mềm vẽ mạch điện tử được phát triển bởi công ty Lancenter Electronics. Phần mềm có thể mô tả hầu hết các [Linh Kiện Điện Tử](https://chotroihn.vn/) thông dụng hiện nay, đặc biệt hỗ trợ cho cả các phần mềm như 8051, PIC, Motorola, AVR.

Proteus có khả năng mô phỏng hoạt động của các mạch điện tử bao gồm phần thiết như kế mạch và viết trình điều khiển cho các loại vi điều khiển như MCS-51, AVR, PIC…

Có 2 chương trình trong phần mềm đó là ARES dùng trong vẽ mạch in và ISIS sử dụng cho mô phỏng mạch. Trong 2 chương trình này thì ISIS có phần nổi bật hơn so với ARES. ISIS đã được phát triển trong 12 năm và có tới hơn 12000 người dùng trên khắp thế giới ( chắc chắn con số hiện tại đã tăng hơn rất nhiều. Điểm nổi bật của chúng đó là khả năng mô phỏng hoạt động của các vi điều khiển mà không cần dùng thêm bất kỳ một phần mềm phụ trợ nào khác. Từ phần mềm ISIS có thể dễ dàng chuyển sang ARES hoặc bất kỳ phần mềm vẽ mạch in khác.

* **Các đặc điểm nổi trội là:**
* Có khả năng mô phỏng hầu hết trình điều khiển cho [vi điều khiển](https://chotroihn.vn/mach-nap-st-link-v2-tai-linh-kien-dien-tu-3m)
* Chọn đối tượng và thiết lập thông số cho đối tượng dễ dàng
* Xuất ra file Netlist tương thích với các chương trình làm mạch in thông dụng.
* Xuất file thống kê linh kiện cho mạch
* ISIS tích hợp nhiều công cụ giúp cho việc quản lý mạch điện lớn, mạch điện có thể lên đến hàng ngàn linh kiện phục vụ cho thiết kế mạch chuyên nghiệp.
* Thiết kế theo cấu trúc (hierachical design)
* Khả năng tự động đánh số linh kiện
* **Ưu điểm của Proteus:**
* Dễ dạng tạo ra một sơ đồ nguyên lý từ đơn giản đến phức tạp.
* Dễ dàng chỉnh sữa các đặc tính của linh kiện trên sơ đồ nguyên lý.
* Hỗ trợ kiểm tra lỗi thiết kế trên sơ đồ nguyên lý. Có thể xem và lưu lại phần báo lỗi
* Phần mềm chạy mô phỏng và phân tích các tính chất của một mạch điện một cách chính xác.
* Proteus cung cấp cho người sử dụng công cụ biên dịch cho các họ vi xử lý như MSC51, AVR, HC11, …qua đó tạo ra các tập tin .hex dùng để nạp cho vi xử lý và tập tin .dsi dùng để xem và chạy kiểm tra từng bước trong quá trình mô phỏng.
* Phần mềm cung cấp rất nhiều mô hình linh kiện có chức năng mô phỏng, từ các vi điều khiển thông dụng đến các link kiện ngoại vi như LED, LCD, Keypad, cổng RS232… cho phép người sử dụng mô phỏng từ một hệ vi điều khiển hoàn chỉnh đến việc xây dựng phần mềm cho hệ thống đáp ứng các giao thức vật lý. Ngoài ra, Proteus còn cho phép bạn tự tạo link kiện tương tác động do đó bạn có thể thực hiện các mô phỏng có tương tác giống như hoạt động của một mạch thật.
* **Nhược điểm của Proteus:**
* Phần vẽ mạch có giao diện không được đẹp và hấp dẫn.

### Arduino IDE

Có thể mọi người đã quen lập trình trên PC, với những ngôn ngữ như C, C++, C#, Java, Python, Ruby...

Nhưng có biết rằng phần mềm trên PC chỉ chiếm khoảng 10% sản lượng phần mềm trên thị trường. 90% còn lại là code điều khiển tivi, máy giặt, điều hòa, tủ lạnh... tóm lại là tất cả các thiết bị điện tử xung quanh ta. Lập trình theo hướng này được gọi là embedded computing, hay physical computing, tức là lập trình để con người tương tác với các thiết bị thực.

Để người thiết kế có thể nhanh chóng đưa ra được mẫu thể hiện ý tưởng của mình, rất cần phải có những platform để dễ dàng prototyping. Và một trong những platform đang được sử dụng rất nhiều trong prototyping là Arduino.

Arduino là một bo mạch xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ,... Điểm hấp dẫn ở Arduino là ngôn ngữ cực kì dễ học (giống C/C++), các ngoại vi trên bo mạch đều đã được chuẩn hóa, nên không cần biết nhiều về điện tử, cũng có thể lập trình được những ứng dụng thú vị. Thêm nữa, vì Arduino là một platform đã được chuẩn hóa, nên đã có rất nhiều các bo mạch mở rộng (gọi là shield) để cắm chồng lên bo mạch Arduino, có thể hình dung như là "library" của các ngôn ngữ lập trình. Ví dụ, muốn kết nối Internet thì có Ethernet shield, muốn điều khiển động cơ thì có Motor shield, muốn kết nối nhận tin nhắn thì có GSM shield,... Rất đơn giản, và chỉ phải tập trung vào việc "lắp ghép" các thành phần này và sáng tạo ra các ứng dụng cần thiết.

Có thể kể ở đây một số ứng dụng hay của Arduino:

* Robot: Arduino được dùng để làm bộ xử lý trung tâm của rất nhiều loại robot. Đó là nhờ vào khả năng đọc các thiết bị cảm biến, điều khiển động cơ,... của Arduino.
* Game tương tác: chúng ta có thể dùng Arduino để tương tác với Joystick, màn hình,... để chơi các trò như Tetrix, phá gach, Mario...
* Máy bay không người lái.
* Và nhiều ví dụ khác nữa,...

Ngôn ngữ lập trình của Arduino chính là C/C++, nhưng so với lập trình lập trình trực tiếp với vi điều khiển, lập trình với Arduino đơn giản hơn nhiều vì chỉ phải giao tiếp với phần cứng thông qua các thư viện, có thể xem như các lớp C++ wrapper lên các giao tiếp với phần cứng. Trên website, có khá nhiều các library viết sẵn để điều khiển ngoại vi: LCD, sensor, motor... việc cần làm là kết hợp chúng với nhau để tạo ứng dụng cho riêng mình.

Một chương trình arduino được gọi là sketch, sẽ được upload lên bo mạch Arduino qua cổng USB.

Phân tích chương trình: có 2 method quan trọng nhất là setup() và loop().

* Setup(): làm nhiệm vụ khởi tạo mode cho các ngoại vi của Arduino. Hàm này sẽ được chạy một lần khi bo mạch Arduino được reset. Ở chương trình này, setup() chỉ làm nhiệm vụ đặt các chân 4,5,6,7 của Arduino sang mode output.
* Loop(): là chương trình chính của Arduino. Đoạn code trong loop() sẽ được Arduino chạy vô hạn. Trong chương trình này, có hàm digitalWrite() để đặt các chân (pin) ở mức điện áp cao (HIGH) hay thấp (LOW). Hàm tiếp theo là delay(), nhận đối số là một số nguyên, thẻ hiện số mili giây ta muốn chương trình tạm ngưng.

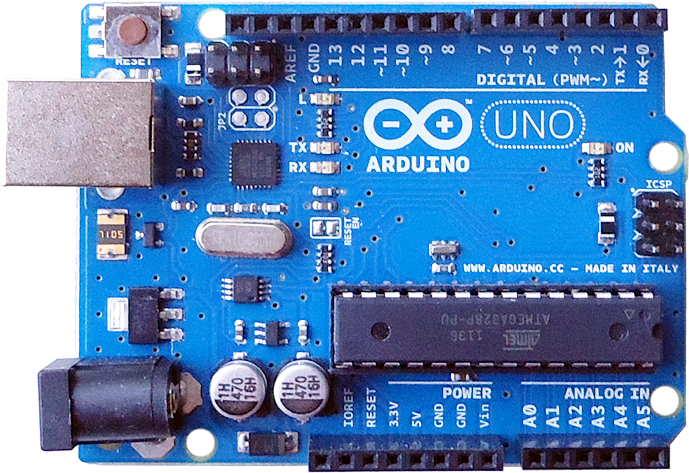


# GIỚI THIỆU CÁC LINH KIỆN ĐƯỢC SỬ DỤNG

## Arduino UNO R3

Arduino board có rất nhiều phiên bản với hiệu năng và mục đích sử dụng khác nhau như: Arduino Mega, Aruino LilyPad... Trong số đó, Arduino Uno R3 là một trong những phiên bản được sử dụng rộng rãi nhất bởi chi phí và tính linh động của nó.

Do Arduino có tính mở về phần cứng, chính vì vậy bản thân Arduino Uno R3 cũng có những biến thể của để phù hợp cho nhiều đối tượng khác nhau.



Hình 2.1 Arduino UNO R3

Vì vậy, để có thể tận dụng tối đa sức mạnh của Arduino board, ta cần phải biết rõ được thông số kỹ thuật của từng loại, nắm được các khác biệt giữa các phiên bản từ đó có giải pháp xử lý phù hợp.

**Một vài thông số của Arduino UNO R3**

|  |  |
| --- | --- |
| Vi điều khiển | ATmega 328 họ 8bit |
| Điện áp hoạt động | 5V DC (chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16MHz |
| Dòng tiêu thụ | Khoảng 30mA |
| Điện áp vào khuyên dùng | 7 – 12 V DC |
| Điện áp vào giới hạn | 6 – 20 V DC |
| Số chân Digital | 14 (6 chân hardware PWM) |
| Số chân Analog | 6 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa (3.3V) | 50 mA |
| Bộ nhớ flash | 32 KB (ATmega 328) với 0.5 KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega 328) |
| EEPROM | 1 KB (ATmega 328) |

## Module điều khiển động cơ L298

**Thông số kỹ thuật**

Driver: L298 tích hợp hai mạch cầu H.

Điện áp điều khiển: +5V ~ +12V.

Dòng tối đa cho mỗi cầu H là: 2A.

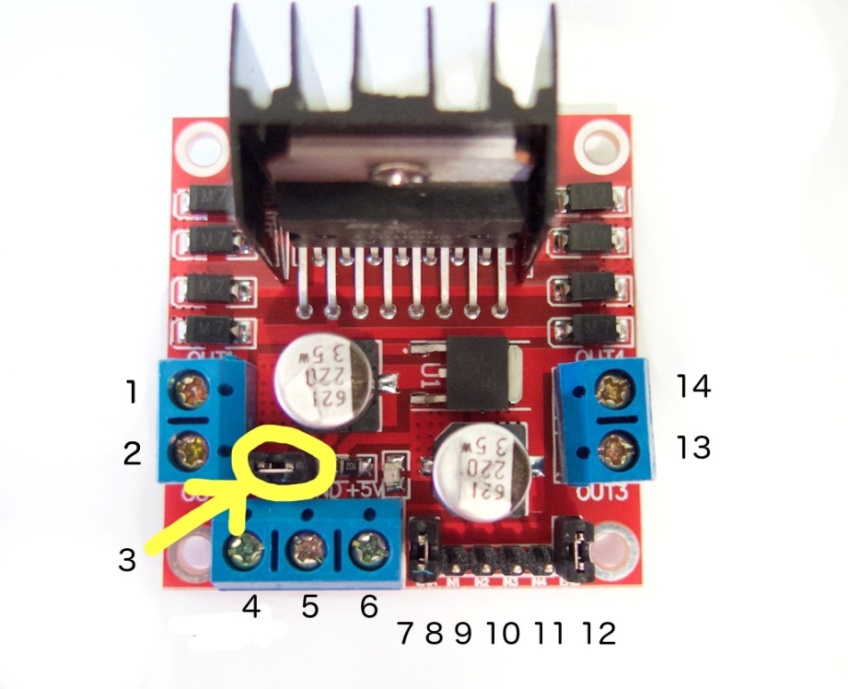
Điện áp của tín hiệu điều khiển: +5V ~ +7V.

Dòng của tín hiệu điều khiển: 0 ~ 36mA.

Công suất hao phí: 20W (khi nhiệt độ T = 75oC).

Nhiệt độ bảo quản : -25 oC ~ +130 oC.

**Các chân tín hiệu**



1. DC motor 1 “+” hoặc stepper motor A+
2. DC motor 1 “-” hoặc stepper motor A-
3. 12V jumper – tháo jumper ra nếu sử dụng nguồn trên 12V. Jumper này dùng để cấp nguồn cho IC ổn áp tạo ra nguồn 5V nếu nguồn trên 12V sẽ làm cháy IC nguồn.
4. Cắm dây nguồn cung cấp điện áp cho motor vào đây từ 6V đến 35V.
5. Cắm chân GND của nguồn.
6. Ngõ ra nguồn 5V, nếu jumper đầu vào không rút ra.
7. Chân Enable của motor 1, chân này dùng để cấp xung PWM cho motor nếu dùng vi điều khiển thì rút jumper ra và cắm chân PWM vào. Giữ nguyên khi dùng với động cơ bước.
8. IN1
9. IN2
10. IN3
11. IN4
12. Chân Enable của Motor 2, chân này dùng để cấp xung cho motor nếu dùng vi điều khiển thì rút jumper ra và cắm chân PWM vào. Giữ nguyên khi dùng với động cơ bước.
13. DC motor 2 “+”, hoặc stepper motor B+
14. DC motor 2 “-“, hoặc stepper motor B-

## Cảm biến siêu âm SRF 04

**Sóng siêu âm**

Sóng siêu âm (sonar) là một loại sóng cao tầng mà con người không thể nghe thấy được. Tuy nhiên, ta có thể thấy được sự hiện diện của sóng siêu âm ở khắp mọi nơi trong tự nhiên. Ta có các loài động vật như dơi, cá heo ... dùng sóng siêu âm để liên lạc với nhau, để săn mồi hay định vị trong không gian.

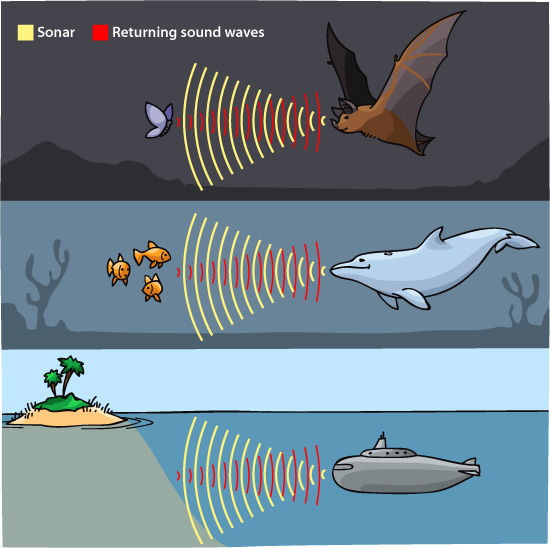
Dựa trên việc quan sát các quan sát hoạt động của chúng, ta thấy được nguyên tắc mà các loài vật sử dụng sóng âm để định vị rất đơn giản, có thể tóm gọn trong 3 bước sau:

Vật chủ phát ra sóng âm. Sóng âm này va chạm với môi trường xung quanh và phản xạ lại.

Dựa vào thời gian phát / thu, khoảng cách giữa vật chủ và môi trường xung quanh được tính ra.

Việc tính toán khoảng cách cũng còn phụ thuộc rất nhiều vào môi trường truyền dẫn, ví dụ như sóng âm truyền trong môi trường nước hay kim loại sẽ nhanh hơn rất nhiều so với sóng âm được truyền trong môi trường không khí.

Cảm biến siêu âm SRF04 cũng hoạt động theo như nguyên tắc ở trên, thiết bị gồm có 2 loa thu và phát cùng với 5 chân để kết nối với Arduino. Theo tài liệu của nhà sản xuất thì tầm hoạt động tối đa của cảm biến này nằm trong khoảng 5m.



Hình 2. 14 Hình ảnh Giao diện Login

**Giới thiệu chung**

Cảm biến SRF 04 là một loại cảm biến khoảng cách dựa trên nguyên lý thu phát siêu âm. Cảm biến gồm một bộ phát và một bộ thu sóng siêu âm. Sóng siêu âm từ đầu phát truyền đi trong không khí, gặp vật cản (vật cản đo khoảng cách tới) sẽ phản xạ ngược trở lại và được đầu thu ghi lại. Khoảng cách đo được của SRF 04 nằm trong phạm vi từ 4cm đến 300cm.



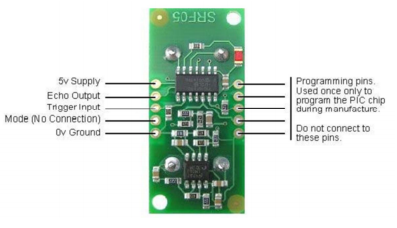
Hình 2. 15 Hình ảnh Giao diện Register

**Nguyên lý làm việc**

SRF 04 có thể thiết lập 2 mode hoạt động khác nhau thông qua các chân điều khiển. Nối hoặc không nối chân OUT xuống GND cho phép cảm biến được điều khiển thông qua giao tiếp dùng 1 chân hay 2 chân IO.

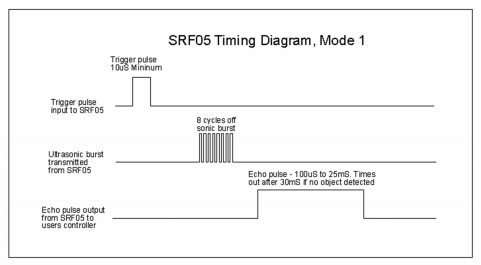
Mode 1: Tách chân Trig và chân Echo dùng riêng:

Trong mode này, SRF 04 sử dụng cả 2 chân trig và echo cho việc giao tiếp với vi điều khiển. Để xử dụng mode này, ta chỉ cần để trống chân OUT của module, điện trở bên trong module sẽ kéo chân pin này lên mức 1



**Hình 2.8 Cấu hình SRF 04 ở mode 1**

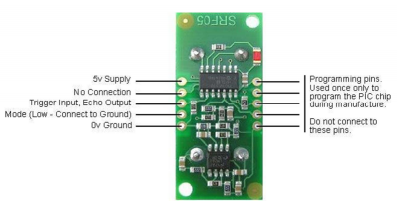
Để điều khiển SRF 04, ta cẩn cấp cho chân trig một xung điều khiển với độ rộng tối thiểu 10us. Sau đó một khoảng thời gian, đầu phát sóng siêu âm sẽ phát ra sóng siêu âm, vi xử lý tích hợp trên module sẽ tự xác định thời điểm phát sóng siêu âm và thu sóng siêu âm. Vi xử lý tích hợp này sẽ đưa kết qua thu được ra chân echo. Độ rộng xung vuông tại chân Echo tỉ lệ với khoảng cách từ cảm biến tới vật thể.



**Hình 2.9 Nguyên lý hoạt động của SRF 04 ở mode 1**

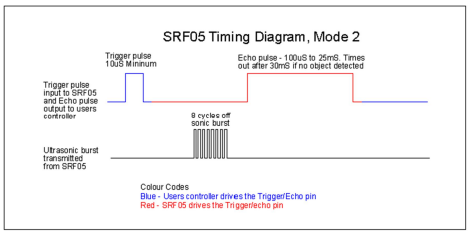
Mode 2: Chân Trig và echo dùng chung

Được thiết kế nhằm cho mục đích tiết kiệm chân pin cho vi điều khiển, nên trong mode này, SRF 04 chỉ sử dụng 1 chân pin cho 2 chức năng trig và echo. Để sử dụng mode này, ta kết nối chân OUT xuống GND.



**Hình 2.10 Cấu hình SRF 04 ở mode 2**

Để điều khiển SRF 04, đầu tiên xuất một xung với độ rộng tối thiểu 10us vào chân trig – echo. Sau đó vi xử lý tích hợp trên cảm biến sẽ phát ra tín hiệu điều khiển đầu phát siêu âm. Sau 700us kể từ lúc kết thúc tín hiệu điều khiển, từ chân trig – echo có thể đọc ra một xung mà độ rộng tỉ lệ với khoảng cách từ cảm biến tới vật cản.



**Hình 2.11 Nguyên lý hoạt động của SRF-04 ở mode 2**

**Lập trình điều khiển**

Với cảm biến SRF-04, ta sẽ minh hoạ việc sử dụng qua việc lập trình cho cảm biến cứ mỗi chu kì 1s ta tiến hành kích hoạt cảm biến và kiểm tra xem có vật cản ở xung quanh hay không.

Thực hiện mỗi chu kì 1s.

Kích hoạt cảm biến bằng việc bật PIN Trigger theo thứ tự LOW - HIGH - LOW qua hàm digitalWrite.

Tính toán khoảng cách thu được bằng việc sử dụng hàm pulseIn và các công thức tính.

Ta giả định nếu khoảng cách trả về < 0.5m thì sẽ in ra thông báo có vật cản.

Lặp lại chu trình này.

## Động cơ điện một chiều



Hình 2.12 Động cơ DC

Động cơ điện một chiều sử dụng dòng điện một chiều. Động cơ điện một chiều được sử dụng trong các ứng dụng đặc biệt yêu cầu mô men khởi động cao hoặc yêu cầu tăng tốc êm ở một dải tốc độ rộng.

Động cơ điện một chiều gồm 3 thành phần chính sau:

* Cực từ: Tương tác giữa hai từ trường tạo ra sự quay trong động cơ một chiều. Động cơ một chiều có các cực từ đứng yên và phần ứng quay trong không gian giữa cực từ. Một động cơ một chiều đơn giản có hai cực từ: cực bắc và cực nam. Các đường sức từ chạy theo khoảng mở từ cực bắc tới cực nam. Với những động cơ phức tạp và lớn hơn, có một vài nam châm điện. Những nam châm này được cấp điện từ bên ngoài đóng vai trò hình thành cấu trúc từ trường.
* Phần ứng: Khi có dòng điện đi qua, phần ứng sẽ trở thành một nam châm điện. Phần ứng, có dạng hình trụ, được nối với trụ ra để kéo tải. Với động cơ một chiều nhỏ, phần ứng quay trong từ trường do các cực tạo ra, cho đến khi cực bắc và cực nam của nam châm hoán đổi vị trí tương ứng với góc quay của phần ứng. Khi sự hoán đổi hoàn tất, dòng điện đảo chiều để xoay chiều các cực bắc và nam của phần ứng.
* Cổ góp: Bộ phần này thường có ở động cơ một chiều. Cổ góp có tác dụng đảo chiều của dòng điện trong phần ứng. Cổ gió cũng hỗ trợ sự truyền điện giữa phần ứng và nguồn điện.

Ưu điểm của động cơ điện một chiều là khả năng điều khiển tốc độ mà không làm ảnh hưởng tới chất lượng điện cung cấp.

Có thể điều khiển động cơ loại này bằng các điều chỉnh:

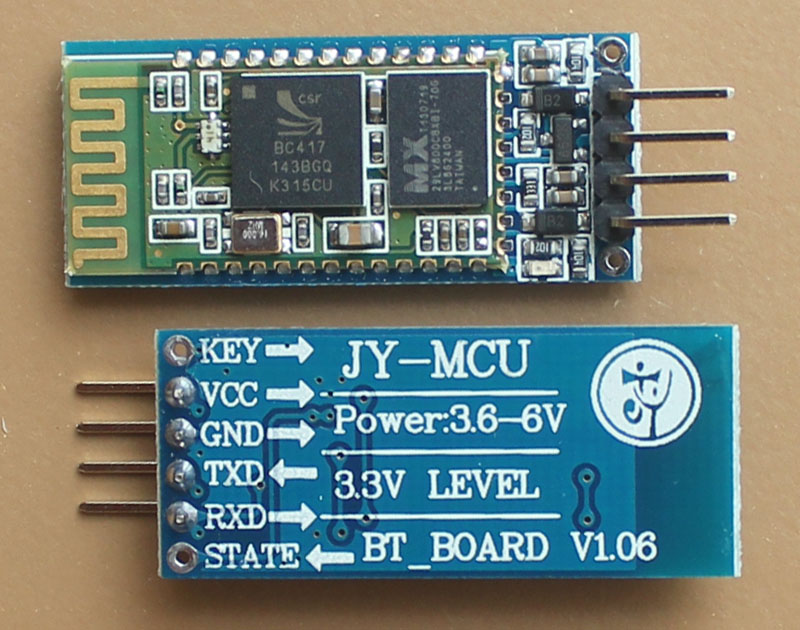
* Điện áp phần ứng: tăng điện áp phần ứng sẽ làm tăng tốc độ.
* Dòng kích thích: Giảm dòng kích thích sẽ làm tăng tốc độ.

Động cơ điện một chiều có nhiều loại khác nhau, nhưng những động cơ loại này thường được sử dụng giới hạn ở những thiết bị tốc độ chậm, công suất thấp đến trung bình. Ở công suất lớn, động cơ một chiều có thể gặp trục trặc với các cổ góp cơ. Các động cơ này cũng bị hạn chế chỉ sử dụng ở những khu vực sạch, không độc hại vì nguy cơ đánh lửa ở chổi than.

## Module bluetooth RN-42 trên HC-05

**Giới thiệu**

* Điện áp hoạt động: 3.3VDC 30mA(hỗ trợ IC 5.0V)
* Dòng điện khi hoạt động : Khi Pairing 30mA , sau khi pairing hoạt động truyền nhận bình thường 8mA
* Baudrate : 1200,2400,4800,9600,19200,38400,57600,11520
* Dải tần hoạt động : 2.4GHz
* Kích thước : 26.9mm x 13mm x 2.2mm
* Giao tiếp : Bluetooth serial port
* Nhiệt độ làm việc : -20℃ ~ +75℃
* Tốc độ : - Asynchronous : 2.1Mbps(Max)/160kbps
* Synchronous : 1Mbps/1Mbps



**Nguyên lý hoạt động**

Giao tiếp với module bằng giao tiếp nối tiếp không đồng bộ qua 2 đường RX và TX, vì vậy các bạn có thể sử dụng PC với chuẩn RS232 hoặc các dòng vi điều khiển để giao tiếp.

Module HC05 có thể nhận 1 trong 3 chức năng: Master, Slave, Loopback (có thể lựa chọn các chức năng bằng lệnhAT).

Bằng cách thay đổi trạng thái chân 34 (KEY), bạn có thể cấu hình chế độ hoạt động cho module:

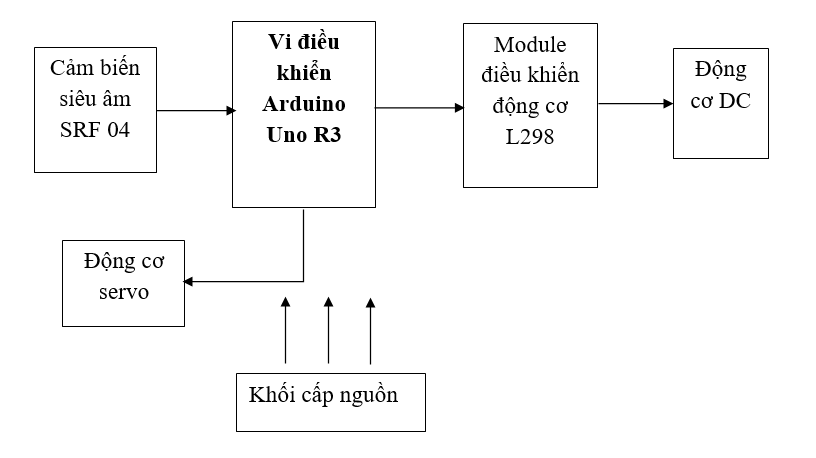
Để module làm việc ở chế độ kết nối tự động: KEY phải ở trạng thái Floating (trạng thái không kết nối).

Để module làm việc ở chế độ đáp ứng theo lệnh: KEY = ‘0’ (kết nối xuống đất) Cấp nguồn cho module chuyểnKEY = ‘1’ (kết nối lên VCC) lúc này có thể sử dụng các lệnh AT để giao tiếp.

# THIẾT KẾ VÀ THỰC NGHIỆM

## Sơ đồ khối và nguyên lý hoạt động của mạch

* **Sơ đồ khối điều khiển**



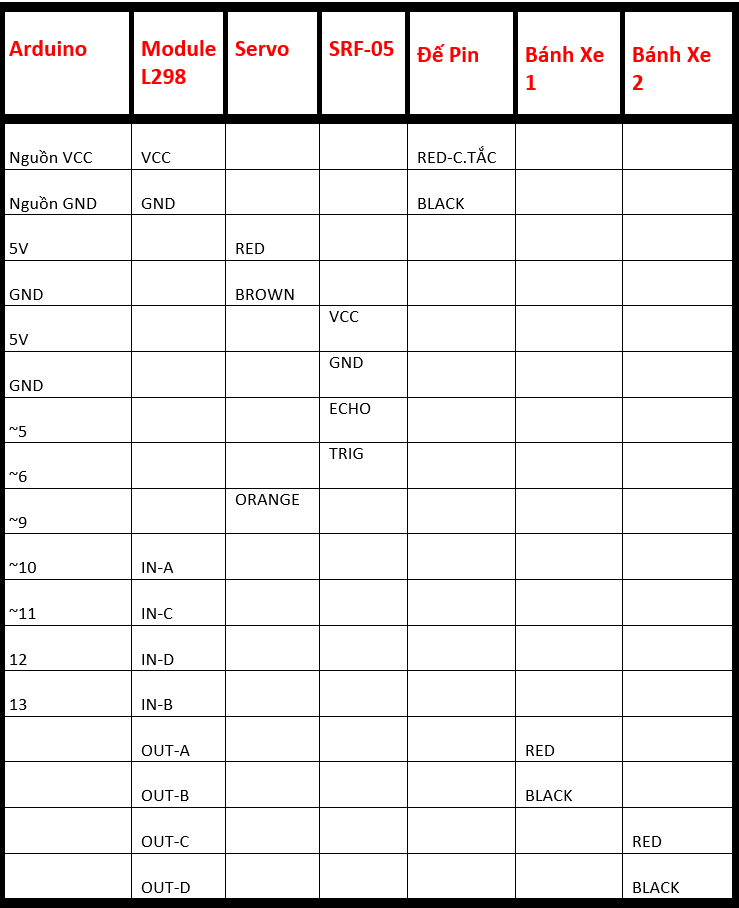
**Sơ đồ khối tổng quát**

* Arduino R3: Xử lý mọi tín hiệu từ module bluetooth hoặc cảm biến siêu âm gửi về để phát tín hiệu điều khiển cho module điểu khiển động cơ để điều khiển động cơ hoạt động theo chương trình.
* Cảm biến siêu âm SRF 04: cảm biến phát hiện vật cản gửi tín hiệu về cho vi điều khiển.
* Module điều khiển động cơ L298: nhận tín hiệu từ vi điều khiển để điều khiển động cơ hoạt động chính xác.
* Khối nguồn: Cung cấp nguồn cho mạch hoạt động.

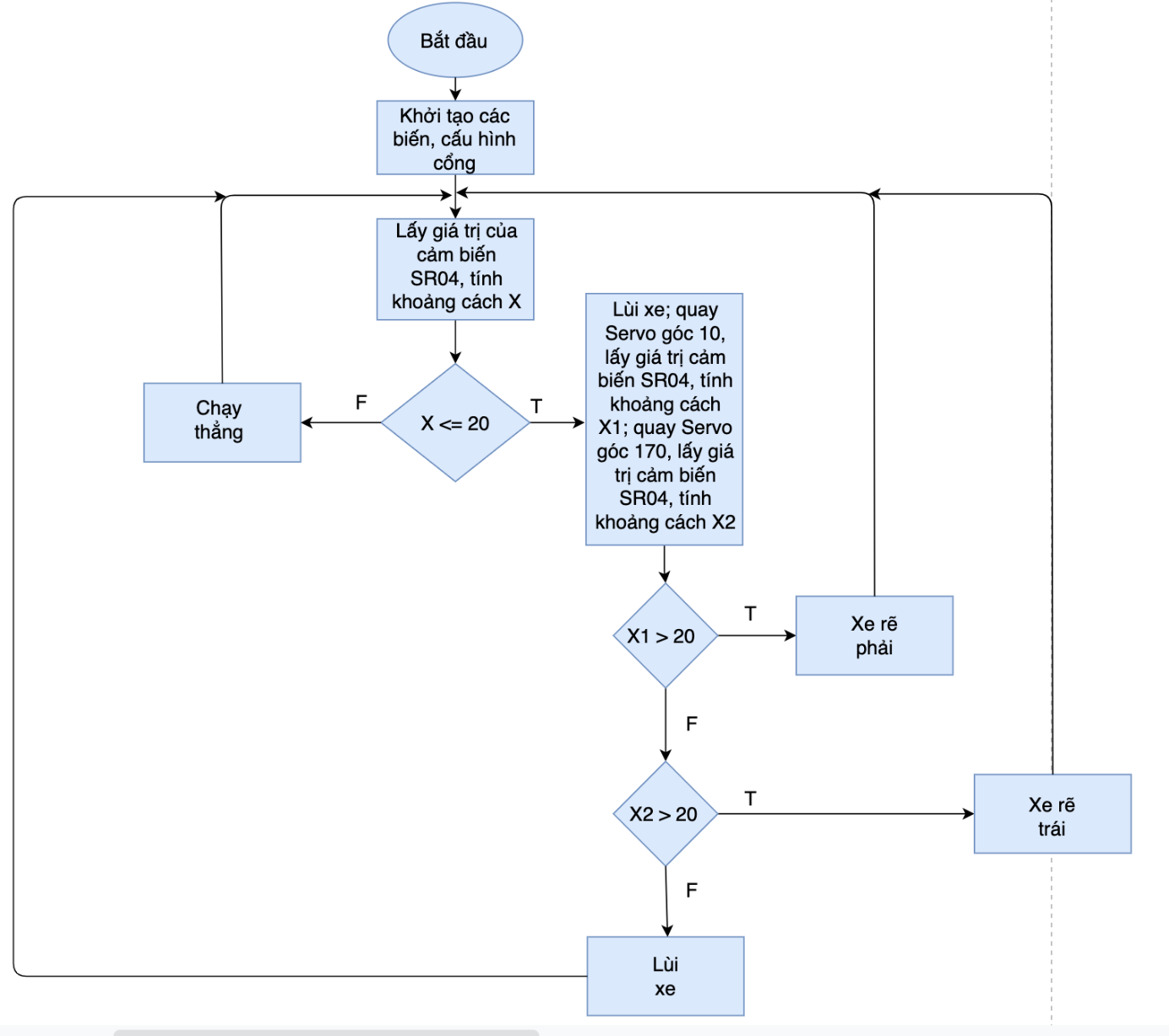
##### **Nguyên lý hoạt động của mạch:**

Khi cấp nguồn cho mạch xe tự khởi động chế độ chạy tránh vật cản. Khi đó arduino nhận tín hiệu từ cảm biến siêu âm để xử lý điều khiển module điều khiển động cơ cho xe chạy một cách hợp lý. Khi không có vật cản ở phía trước xe sẽ chạy thẳng. Nếu gặp vật cản xe sẽ dừng lại. Servo hoạt động quay 180o bắt đầu từ trái qua phải để tìm đường di chuyển. Xe sẽ rẽ về hướng không có vật cản. Quá trình tiếp diễn như vậy đến khi tắt nguồn.

##### **Bảng kết nối chân**



## Sơ đồ khối và nguyên lý hoạt động của mạch chế độ tự hành



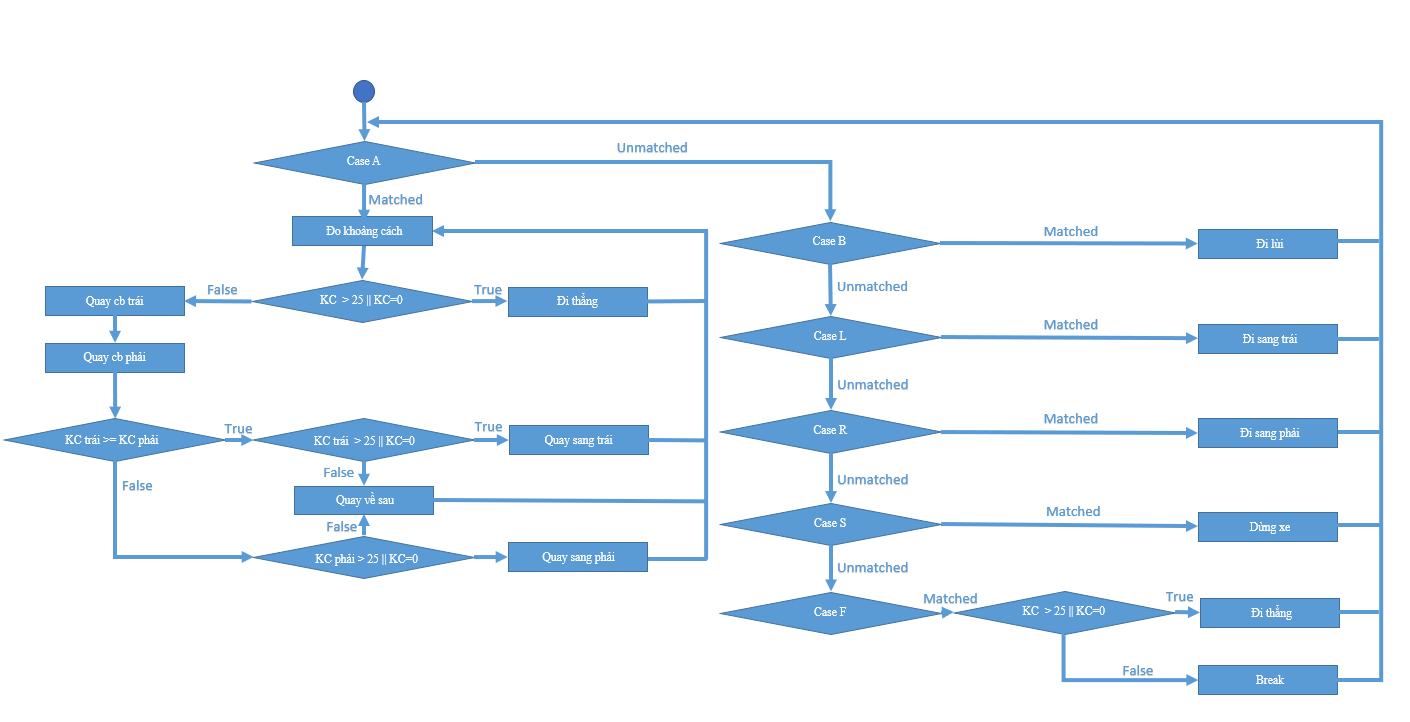
**Sơ đồ khối của chương trình chính**

Giải thích sơ đồ:

1. Bắt đầu vào
2. Khởi tạo biến và cấu hình các cổng.
3. Lấy giá trị của cảm biến SR04, tính khoảng cách X.
4. Thực hiện phép so sánh: X <= 20, nếu sai thì sang bước 5, đúng thì sang bước 6.
5. Xe chạy thẳng rồi quay về bước 3.
6. Lùi xe, thực hiện quay Servo góc 10 độ, lấy giá trị SR04, tính X21. Sau đó quay Servo góc 170 độ, lấy giá trị SR04, tính X2.
7. Thực hiện so sánh: X1 > 20. Nếu đúng thì sang bước 8, nếu sai thì sang bước 9.
8. Thực hiện xe rẽ phải rồi quay về bước 3.
9. Thực hiện so sánh: X2 >20. Nếu đúng thì sang bước 10, nếu sai sang bước 11.
10. Thực hiện xe rẽ trái rồi quay về bước 3.
11. Thực hiện xe đi lùi rồi quay về bước 3.

## Sơ đồ khối và nguyên lý hoạt động của toàn mạch

Mô tả toàn bộ quá trình xử lý của hệ thống điều khiển. Giúp kiểm tra tính khả thi của việc lập trình, nhanh chóng đưa ra những giải thuật để viết chương trình một cách nhanh chóng và hiệu quả.



Khi người điều khiển kết nối thành công Bluetooth từ module bluetooth HC05, khi đó dòng thông báo “ĐẪ KẾT NỐI BLUETOTH!”. Người điều khiển có thể sử dụng các button trên điện thoại để điều khiển xe từ xa.

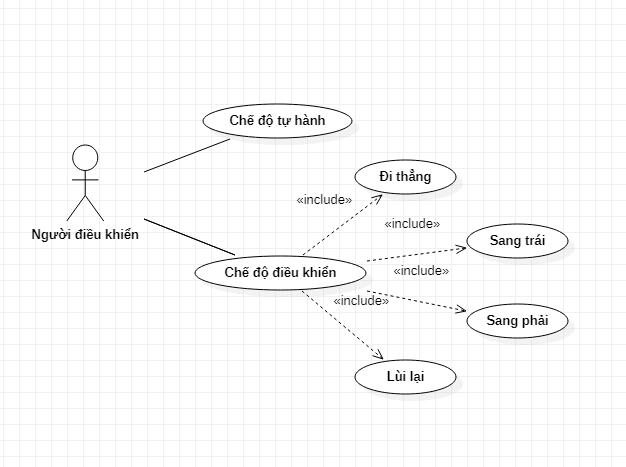
Khi người điều khiển nhấn “Arrow up icon” app sẽ gửi đến Arduino tín hiệu “F” thông qua Bluetooth, khi đó xe sẽ đi thẳng. Khi gặp vật cản trong khoảng cách 25cm, người điều khiển cố tình cho xe chạy thẳng thì khi đó xe sẽ rơi vào trạng thái đứng yên và bỏ qua lệnh đi thẳng của người điều khiển.

Khi người điều khiển nhấn “Arrow left icon” app sẽ gửi đến Arduino tín hiệu “L” thông qua Bluetooth, khi đó xe sẽ quay sang trái. Tương tự như vậy khi người dùng nhấn “Arrow down icon”, “Arrow right icon”,…

Khi người điều khiển nhấn “CHẾ ĐỘ TỰ HÀNH” xe sẽ vào chế độ tự hành tránh vật cản khi gặp vật cản trong khoảng 25cm.

Người điều khiển nhấn “HỦY GHÉP NỐI” để ngắt kết nối Bluetooth giữa app và xe khi người dùng hoàn tất quá trình điều khiển

## Biểu đồ use-case



**Hình 3.3 Biểu đồ use-case**

Đặc tả các ca sử dụng

|  |  |
| --- | --- |
| Use - Case | Tiến lên |
| Actor | Module HC-SR04 |
| Brief Description | Xe tự điều khiển tiến lên phía trước dò đường và cảm biến siêu âm đo khoảng cách tránh vật cản |
| Pre - condition |  |
| Basic Flows | 1. Cảm biến đo khoảng cách từ xe đến vật cản ở phía trước.  2. Cảm biến gửi thông tin khoảng cách vật cản tới Arduino.  3. Nếu khoảng cách giữa xe và vật cản lớn hơn 20cm thì xe có thể tiến lên phía trước |
| Alternative Flows | 3.1 Nếu khoảng cách giữa xe và vật cản nhỏ hơn hoặc bằng 20 cm thì xe sẽ tự hành tránh sang phía khác. |

Bảng 3.1 Use – case tiến lên

|  |  |
| --- | --- |
| Use - Case | Lùi lại |
| Actor | Module HC-SR04 |
| Brief Description | Xe tự động lùi về sau, cảm biến siêu âm đo khoảng cách vật cản để tìm đường tránh. |
| Pre - condition |  |
| Basic Flows | 1. Cảm biến đo khoảng cách từ xe đến vật cản ở phía trước.  2. Cảm biến gửi thông tin khoảng cách vật cản tới Arduino.  3. Nếu khoảng cách giữa xe và vật cản lớn hơn 20 cm thì xe có thể tiến lên phía trước |
| Alternative Flows | 3.1 Nếu khoảng cách giữa xe và vật cản nhỏ hơn hoặc bằng 20 cm thì xe sẽ tự hành tránh sang phía khác. |

Bảng 3.2 Use – case lùi về

|  |  |
| --- | --- |
| Use - Case | Sang phải |
| Actor | Module HC-SR04 |
| Brief Description | Cảm biến siêu âm đo khoảng cách từ xe đến vật cản, sau đó xe tự động quay sang phải và đi theo hướng đó. |
| Pre - condition |  |
| Basic Flows | 1. Cảm biến đo khoảng cách từ xe đến vật cản ở phía trước.  2. Cảm biến gửi thông tin khoảng cách vật cản tới Arduino.  3. Nếu khoảng cách giữa xe và vật cản nhỏ hơn hoặc bằng 20cm thì thực hiện quay Servo góc 10 độ.  4. Cảm biến tiếp tục đo khoảng cách từ xe đến vật cản theo góc quay servo vừa quay.  5. Nếu khoảng cách giữa xe và vật cản lớn hơn 20cm thì xe có thể quay góc bên phải và đi tiến về phía đó. |
| Alternative Flows | 3.1 Nếu khoảng cách giữa xe và vật cản lớn hơn 20cm thì xe sẽ tự hành tiến về phía trước  5.1 Nếu Nếu khoảng cách giữa xe và vật cản nhỏ hơn hoặc bằng 20cm thì Servo sẽ quay và kiểm tra góc bên trái |

Bảng 3.3 Use – case sang phải

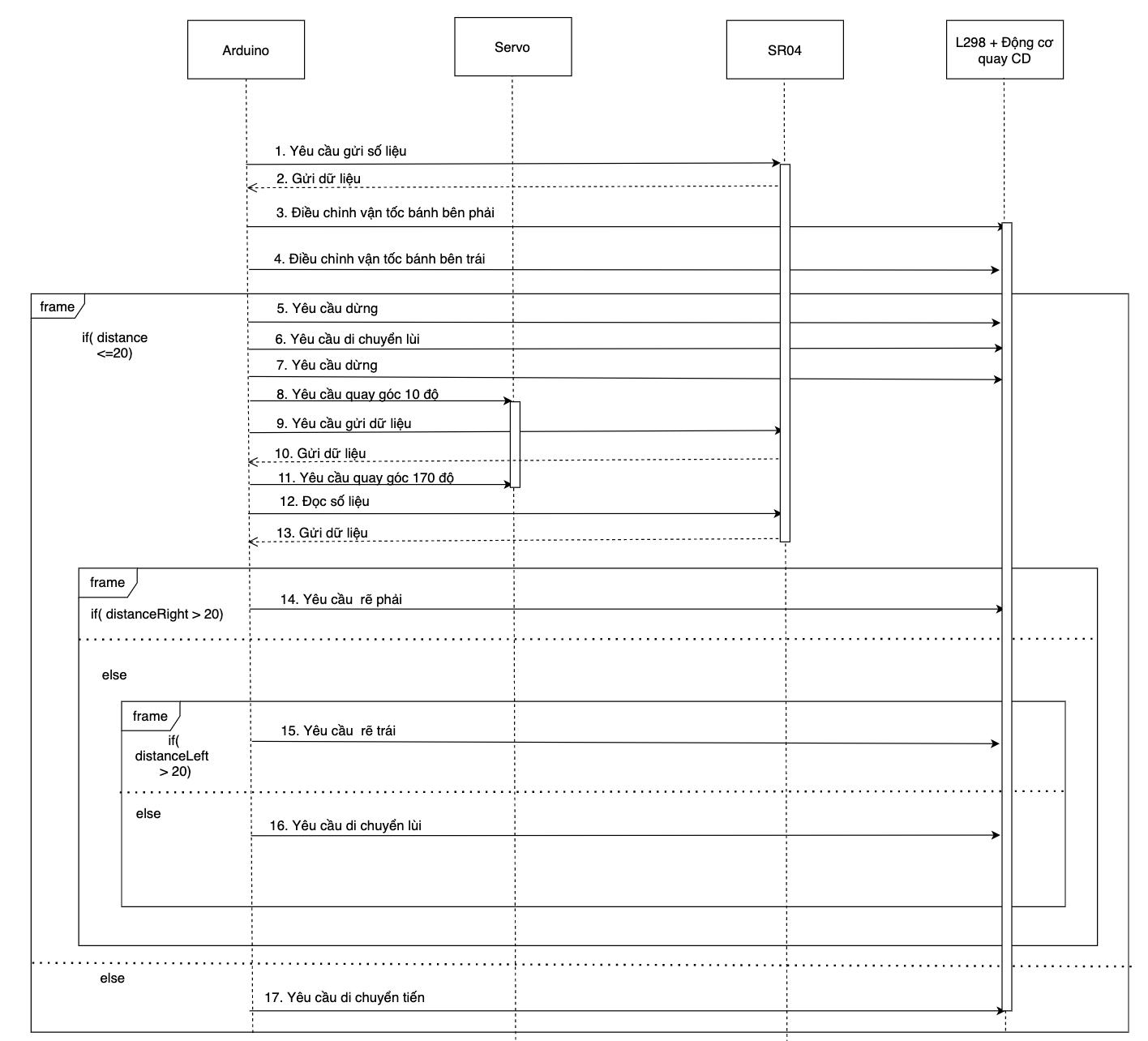
|  |  |
| --- | --- |
| Use - Case | Sang trái |
| Actor | Module HC-SR04 |
| Brief Description | Xe tự điều khiển tiến lên phía trước dò đường và cảm biến siêu âm đo khoảng cách tránh vật cản |
| Pre - condition |  |
| Basic Flows | 1. Cảm biến đo khoảng cách từ xe đến vật cản ở phía trước.  2. Cảm biến gửi thông tin khoảng cách vật cản tới Arduino.  3. Nếu khoảng cách giữa xe và vật cản nhỏ hơn hoặc bằng 20cm thì thực hiện quay Servo góc 170 độ.  4. Cảm biến tiếp tục đo khoảng cách từ xe đến vật cản theo góc quay servo vừa quay.  5. Nếu khoảng cách giữa xe và vật cản lớn hơn 20cm thì xe có thể quay góc bên trái và đi tiến về phía đó. |
| Alternative Flows | 3.1 Nếu khoảng cách giữa xe và vật cản nhỏ hơn hoặc bằng 20cm thì xe sẽ tự hành tránh sang phía khác.  5.1 Nếu khoảng cách giữa xe và vật cản nhỏ hơn hoặc bằng 20cm thì xe sẽ lùi sau |

Bảng 3.4 Use – case sang trái

|  |  |
| --- | --- |
| Use - Case | Tránh vật cản |
| Actor | Module HC-SR04 |
| Brief Description | Cảm biến siêu âm đo khoảng cách để tránh vật cản |
| Pre - condition |  |
| Basic Flows | 1. Cảm biến đo khoảng cách từ xe đến vật cản ở phía trước.  2. Cảm biến gửi thông tin khoảng cách vật cản tới Arduino.  3. Nếu khoảng cách giữa xe và vật cản lớn hơn 20cm thì xe có thể tiến lên phía trước |
| Alternative Flows | 3.1 Nếu khoảng cách giữa xe và vật cản nhỏ hơn hoặc bằng 20cm thì xe sẽ tự hành tránh sang phía khác. |

Bảng 3.5 Use – case tránh vật cản

## Biểu đồ tuần tự



Hình 3.4 Biểu đồ tuần tự

Giải thích:

1. Arduino yêu cầu lấy dữ liệu của SR04.
2. SR04 gửi dữ liệu về Arduino.
3. Arduino yêu cầu L298 điều chỉnh tốc độ động cơ CD bên trái.
4. Arduino yêu cầu L298 điều chỉnh tốc độ động cơ CD bên phải.

* Nếu distance <= 20 thực hiện các bước sau:

1. Arduino yêu cầu L298 tạm dừng 2 động cơ CD.
2. Arduino yêu cầu L298 điều chỉnh 2 động cơ CD lùi.
3. Arduino yêu cầu L298 tạm dừng 2 động cơ CD.
4. Arduino yêu cầu Servo quay góc 10 độ.
5. Arduino yêu cầu lấy dữ liệu của SR04.
6. SR04 gửi dữ liệu về Arduino.
7. Arduino yêu cầu Servo quay góc 170 độ.
8. Arduino yêu cầu lấy dữ liệu của SR04.
9. SR04 gửi dữ liệu về Arduino.

* Nếu distanceRight > 20, thực hiện các bước sau:

1. Arduino yêu cầu L298 điều khiển động cơ serve để xe rẽ phải.

* Ngược lại, nếu distance Right< 20 thì thực hiện bước sau:
* Nếu distanceLeft > 20, thực hiện các bước tiếp sau:

1. Arduino yêu cầu L298 điều khiển động cơ serve để xe rẽ trái.

* Ngược lại, nếu distanceLeft < 20 thì thực hiện bước sau:

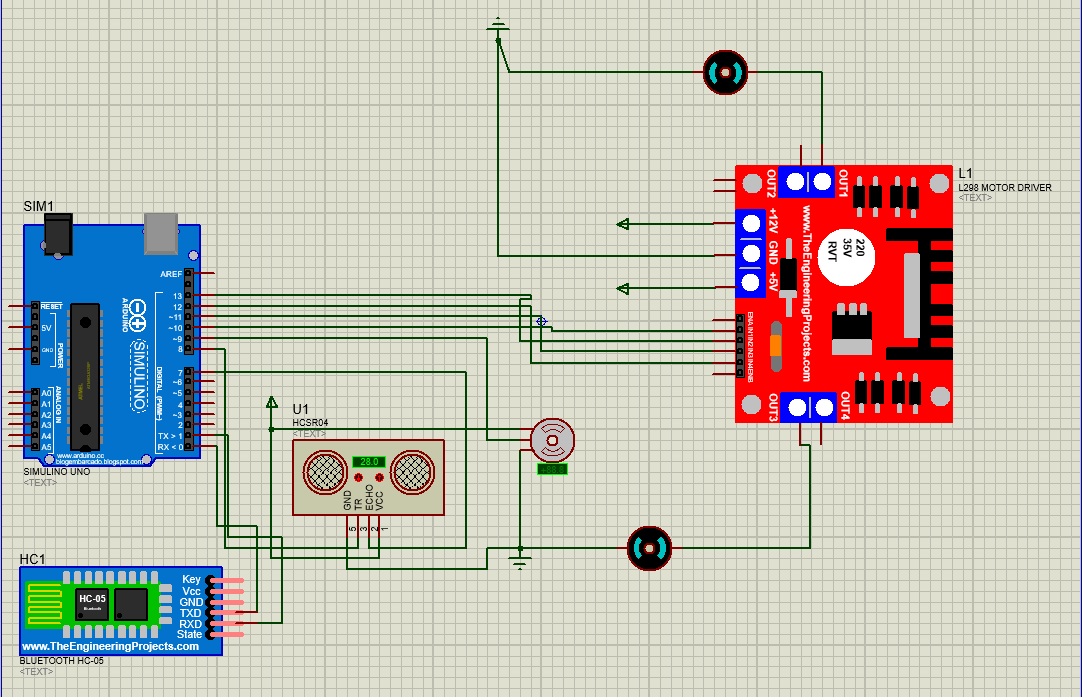
1. Arduino yêu cầu L298 điều chỉnh 2 động cơ CD lùi.

* Ngược lại nếu distance >20, thực hiện bước tiếp theo:

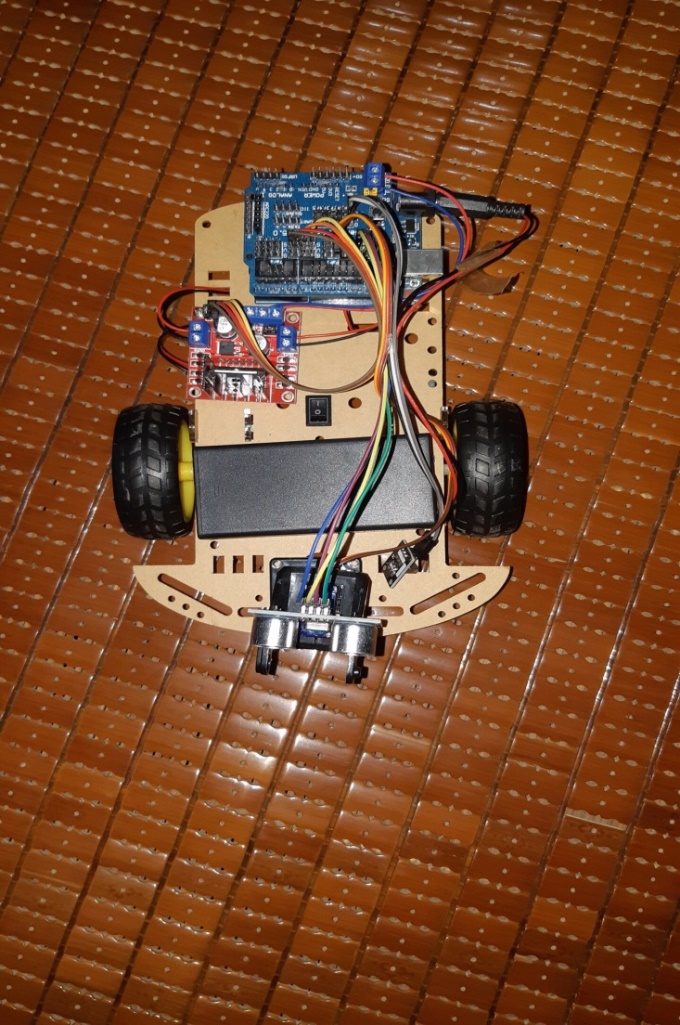
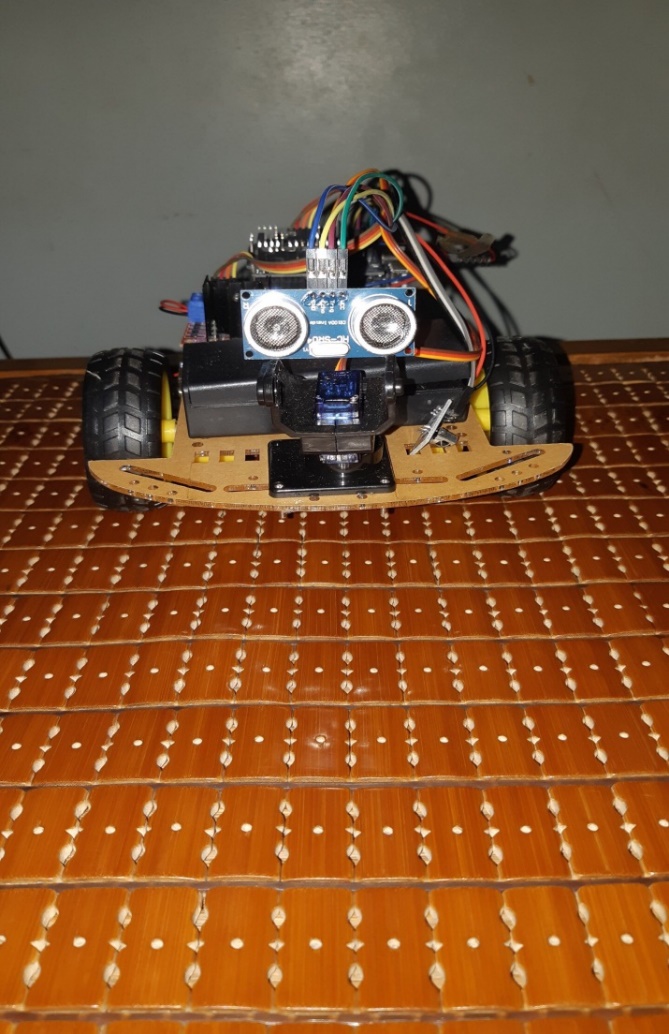
1. Arduino yêu cầu L298 điều chỉnh 2 động cơ CD tiến.

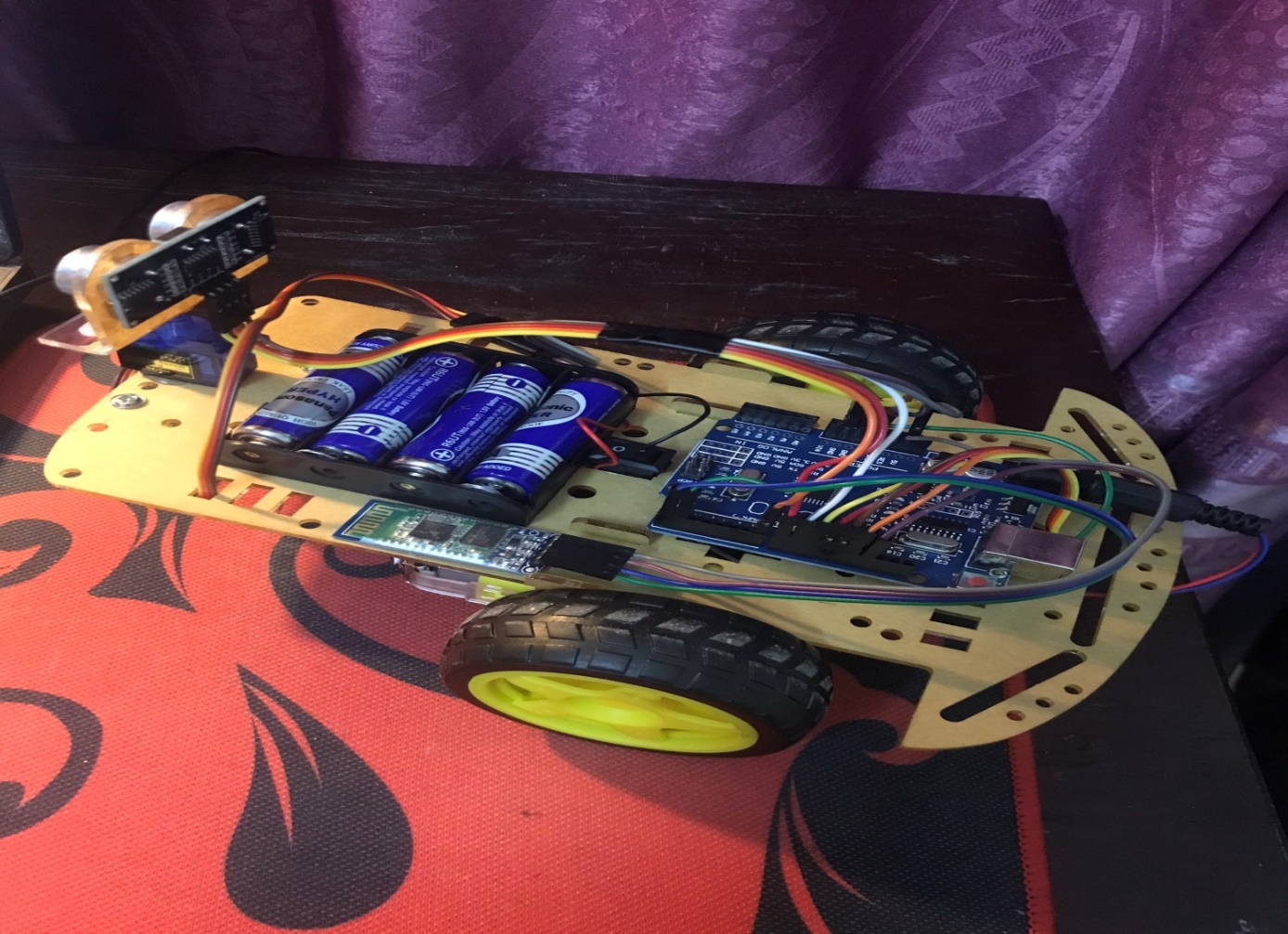
## Thiết kế mạch và mô hình thực tế

**Mạch trên proteus**



**Mạch thực tế**





**Hình Mạch thực tế**

## Chương trình Code

#include "Servo.h"

#define trig 8 //chân trig srf05

#define echo 7 //chân echo srf05

#define inA1 10 //in1 module L298n nối với arduino

#define inA2 13 //in2 module L298n nối với arduino

#define inB1 11 //in3 module L298n nối với arduino

#define inB2 12 //in4 module L298n nối với arduino

Servo myServo;

char command;

void setup() {

pinMode(inA1, OUTPUT);

pinMode(inA2, OUTPUT);

pinMode(inB1, OUTPUT);

pinMode(inB2, OUTPUT);

pinMode(trig,OUTPUT);

pinMode(echo,INPUT);

Serial.begin(9600);

myServo.attach(9);

}

void loop() {

if (Serial.available()>0) {

command=Serial.read();

Serial.println(command);

if(command=='F'){

int a=dokhoangcach(90);

if(a<=25){

huongxe(inA1, inA2, inB1, inB2,0);

Serial.println("khoang cach duoi 25cm");

}

else{

huongxe(inA1, inA2, inB1, inB2,1);

Serial.println("di thang");

}

}

if(command=='B'){

huongxe(inA1, inA2, inB1, inB2,2);

Serial.println("di lui");

}

if(command=='L'){

huongxe(inA1, inA2, inB1, inB2,4);

Serial.println("sang trai");

}

if(command=='R'){

huongxe(inA1, inA2, inB1, inB2,3);

Serial.println("sang phai");

}

if(command=='S'){

huongxe(inA1, inA2, inB1, inB2,0);

Serial.println("dung");

}

if(command=='A'){

Serial.println("di chuyen auto");

dichuyen(inA1, inA2, inB1, inB2, 25, 1000);

while(Serial.available()==0){

dichuyen(inA1, inA2, inB1, inB2, 25, 1000);

}

}

}

}

void dichuyen(int inR1, int inR2, int inL1, int inL2, int gioihan, int thoigiantre)

{

huongxe(inR1,inR2,inL1,inL2,1);

int khoangcach=dokhoangcach(90);

int khoangcachtrai;

int khoangcachphai;

if (khoangcach > gioihan||khoangcach==0)

{

huongxe(inR1,inR2,inL1,inL2,1);

Serial.println("");

delay(10);

}

if (khoangcach <= gioihan)

{

huongxe(inR1,inR2,inL1,inL2,2);

Serial.println("xe di lui 300ms");

delay(300);

//xe đứng yên

huongxe(inR1,inR2,inL1,inL2,0);

Serial.println("xe dung yen");

//quay cb sang trái và đo khoảng cách

khoangcachtrai = dokhoangcach(180);

Serial.println("quay cam bien sang trai");

//quay cb sang phải và đo khoảng cách

khoangcachphai = dokhoangcach(0);

Serial.println("quay cam bien sang phai");

if (khoangcachtrai>khoangcachphai)

{

if(khoangcachtrai>gioihan||khoangcachtrai==0){

//quay xe sang trái

huongxe(inR1,inR2,inL1,inL2,4);

Serial.println("quay xe sang trai");

delay (thoigiantre/2);

}

else{

//quay xe về phía sau

huongxe(inR1,inR2,inL1,inL2,3);

Serial.println("quay ve sau");

delay(thoigiantre);

}

}

else

{

if (khoangcachphai>gioihan||khoangcachphai==0)

{

//quay xe sang phải

huongxe(inR1,inR2,inL1,inL2,3);

Serial.println("quay xe sang phai");

delay (thoigiantre/2);

}

else{

//quay xe về sau

huongxe(inR1,inR2,inL1,inL2,4);

Serial.println("quay ve sau");

delay(thoigiantre);

}

}

}

}

int dokhoangcach(int goc)

{

myServo.write(goc);

unsigned long thoigian;

int khoangcach;

digitalWrite(trig,0);

delayMicroseconds(2);

digitalWrite(trig,1);

delayMicroseconds(10);

digitalWrite(trig,0);

thoigian = pulseIn(echo,HIGH);

khoangcach = int(thoigian/2/29.412);

Serial.println(khoangcach);

return khoangcach;

}

void huongxe(int inR1, int inR2, int inL1, int inL2, int trangthai)

{

switch(trangthai)

{

case 0:// không di chuyển

dieukhiendongco(inR1, inR2, 0);

dieukhiendongco(inL1, inL2, 0);

break;

case 1://đi thẳng

dieukhiendongco(inR1, inR2, 1);

dieukhiendongco(inL1, inL2, 1);

break;

case 2:// lùi lại

dieukhiendongco(inR1, inR2, 2);

dieukhiendongco(inL1, inL2, 2);

break;

case 3:// quay trái

dieukhiendongco(inR1, inR2, 1);

dieukhiendongco(inL1, inL2, 2);

break;

case 4:// quay phải

dieukhiendongco(inR1, inR2, 2);

dieukhiendongco(inL1, inL2, 1);

break;

}

}

void dieukhiendongco(int dongco1,int dongco2, int num)

{

switch (num)

{

case 0://2 bánh đứng yên

digitalWrite(dongco1,LOW);

digitalWrite(dongco2,LOW);

break;

case 1:// Quay bánh thứ nhất

digitalWrite(dongco1,HIGH);

digitalWrite(dongco2,LOW);

break;

case 2:// Quay bánh thứ 2

digitalWrite(dongco1,LOW);

digitalWrite(dongco2,HIGH);

break;

}

}

# KẾT LUẬN

##### Ưu điểm:

##### Xe được thiết kế đơn giản, giá thành thấp dễ dàng cho mọi người tìm hiểu nghiên cứu.

##### Về cơ bản đã tránh được các vật cản hoàn thành mục tiêu đặt ra.

##### Khuyết điểm:

##### Khi quét vật cản để chọn phương hướng nhiều khi vẫn gặp phải trục trặc khi chọn hướng rẽ.

##### Trong điều kiện cho phép chúng em chỉ thực hiện mô hình ở mức độ đơn giản nhưng trong quá trình thực hiện vẫn chưa được hoàn hảo, mạch thi công còn thiếu sót và hạn chế. Xe hoạt động thật sự chưa được ổn định.

##### Hướng phát triển:

##### Cải thiện xe mô hình lớn hơn, nhanh hơn, trang trí đẹp hơn và chạy ổn định hơn.

# TÀI LIỆU THAM KHẢO

* 1. Phạm Quang Huy, Lê Cảnh Trung. Lập trình điều khiển với arduino*. Khoa học và kỹ thuật*,2019.
  2. Phạm Quang Huy , Nguyễn Trọng Hiếu *,* Vi điều khiển và ứng dụng arduino dành cho người tự học , *Bách Khoa Hà Nội*.
  3. <http://arduino.vn/bai-viet/948-bam-xung-dieu-khien-toc-do-dong-co-bang-l298>
  4. [<http://arduino.vn/bai-viet/233-su-dung-cam-bien-khoang-cach-hc-sr04>](http://arduino.vn/result/1306-robot-tranh-vat-can)
  5. <http://arduino.vn/bai-viet/181-gioi-thieu-servo-sg90-va-cach-dieu-khien-bang-bien-tro>
  6. <http://tae.vn/mach-dieu-khien-dong-co-l298> : Website về mạch L298