

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**  
**KHOA ĐÀO TẠO QUỐC TẾ**



**HCMUTE**

**BÁO CÁO VỀ TIẾN TRÌNH PHÁT TRIỂN**  
**ĐỀ TÀI:**  
**DỰ ÁN XE DÒ LINE VÀ TRÁNH VẬT CẢN**

Thành phố Hồ Chí Minh, Tháng 12 năm 2024

## MỤC LỤC

<b>LỜI GIỚI THIỆU .....</b>	<b>2</b>
<b>PHẦN 1: PHƯƠNG PHÁP THỰC HIỆN .....</b>	<b>3</b>
<b>1. Lập kế hoạch dự án.....</b>	<b>3</b>
<b>2. Thiết kế hệ thống.....</b>	<b>6</b>
<b>3. Thiết kế mạch điện:.....</b>	<b>14</b>
<b>4. Lập trình và thuật toán điều khiển.....</b>	<b>15</b>
<b>PHẦN 2: TIẾN TRÌNH PHÁT TRIỂN DỰ ÁN .....</b>	<b>17</b>
<b>1. Khởi động dự án.....</b>	<b>17</b>
<b>2. Thiết kế và phát triển.....</b>	<b>17</b>
<b>a. Phần cứng.....</b>	<b>17</b>
<b>b. Phần mềm.....</b>	<b>24</b>
<b>KẾT LUẬN .....</b>	<b>40</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>42</b>

## **LỜI GIỚI THIỆU**

Dự án "Xe dò line và tránh vật cản" là một mô hình sáng tạo phát triển nhằm tạo ra một hệ thống tự động hoá có khả năng di chuyển trên các đường kẻ (line) và tự động phát hiện, tránh các vật cản trong môi trường di chuyển. Áp dụng từ nhiều ngành nhiều lĩnh vực như Công nghệ thông tin, Kỹ thuật cơ điện tử, Tự động hoá để tạo nên được sản phẩm có thể ứng dụng trong giao thông thông minh, vận hành logistics,... Dự án này không chỉ mang tính thử nghiệm mà còn hướng tới việc cải thiện hiệu quả công việc góp phần xây dựng thành phố thông minh trong tương lai.

Trong bối cảnh kỷ nguyên của cách mạng công nghiệp 4.0, một trong những ứng dụng đáng chú ý của công nghệ tự động hóa chính là các phương tiện giao thông thông minh, đặc biệt là các xe tự động có khả năng nhận diện và điều khiển mà không cần sự can thiệp của con người. Việc phát triển một chiếc xe có thể dò line và tránh vật cản là một bước tiến quan trọng trong quá trình ứng dụng công nghệ này vào thực tế, mở ra nhiều cơ hội trong việc xây dựng các hệ thống vận tải thông minh, an toàn hơn và hiệu quả hơn.

Mục tiêu chính của dự án là thiết kế và phát triển một chiếc xe tự động có khả năng di chuyển chính xác theo các đường kẻ trên mặt đất và đồng thời nhận diện các vật cản trong môi trường xung quanh để tránh va chạm. Để đạt được mục tiêu này, dự án sử dụng các cảm biến siêu âm và cảm biến quang học để thu thập dữ liệu về vị trí của xe và các vật cản, đồng thời sử dụng thuật toán điều khiển để xử lý dữ liệu và điều chỉnh hướng di chuyển của xe.

Báo cáo này sẽ trình bày chi tiết về quá trình phát triển dự án, bao gồm các bước thiết kế và triển khai phần cứng, lập trình thuật toán điều khiển, cũng như thử nghiệm và hiệu chỉnh hệ thống trong các điều kiện thực tế. Bên cạnh đó, báo cáo cũng sẽ đề cập đến các vấn đề gặp phải trong suốt quá trình thực hiện và những giải pháp được đưa ra để khắc phục các vấn đề đó.

# PHẦN 1: PHƯƠNG PHÁP THỰC HIỆN

## 1. Lập kế hoạch dự án

**Mục tiêu của dự án:** Xe dò line và tránh vật cản có khả năng di chuyển chính xác theo đường kẻ (line) và đồng thời phát hiện các vật cản trong môi trường xung quanh để tránh va chạm. Cụ thể dự án hướng đến hai mục tiêu tương đương với hai chức năng của xe :

- Thiết kế hệ thống dò line: Sử dụng các cảm biến quang học giúp xe nhận diện các đường kẻ được vẽ trên mặt đất và tự động di chuyển theo các đường kẻ đó.
- Thiết kế hệ thống phát hiện vật cản: Sử dụng cảm biến siêu âm để phát hiện vật cản trên đường đi của xe, từ đó đưa ra phương án để tránh và vượt qua.

Thông qua dự án này, chúng tôi mong muốn tạo ra một sản phẩm có khả năng ứng dụng trong các hệ thống giao thông thông minh, tự động hóa trong vận tải hoặc các hệ thống di chuyển tự động trong các môi trường không có người điều khiển.

**Phạm vi dự án:** Vì đây là dự án thử nghiệm nên tập trung vào phát triển một mô hình xe tự động nhỏ, có thể di chuyển theo đường kẻ và tránh vật cản trong môi trường thử nghiệm. Xe sẽ sử dụng các cảm biến quang học, siêu âm để phát hiện line và tránh vật cản. Các tiêu chí như độ chính xác của việc dò line, thời gian phản ứng khi phát hiện vật cản, và khả năng di chuyển linh hoạt sẽ được kiểm tra và đánh giá trong suốt quá trình thực hiện.

**Lộ trình thực hiện:** Được chia ra thành 5 giai đoạn nhỏ

**Giai đoạn 1:** Nghiên cứu và lập kế hoạch (1 tuần)

Do 4 thành viên là Nguyễn Phú Hữu, Nguyễn Phúc Lộc, Nguyễn Vũ Lâm, Trần Phạm Minh Khoa đảm nhiệm vai trò trong giai đoạn 1:

- Xác định thành phần phần cứng gồm những bộ phận nào? Từ đó đưa ra những linh kiện phù hợp.
  - Đây là bước thảo luận do Trần Phạm Minh Khoa đưa ra ý kiến cá nhân để 3 thành viên còn lại tham khảo và phản biện lại.
- Phân tích và lựa chọn thuật toán sao cho phù hợp cả phần cứng của xe.
  - Phân tích và lựa chọn một thuật toán phù hợp với phần cứng của xe là một điều vô cùng quan trọng quyết định xe có hoạt động ổn định hay không?
  - Do Nguyễn Phú Hữu sẽ trực tiếp lập trình và đưa ra phương án tối ưu cho xe. Sau khi đã thống nhất giữa các thành viên.
- Lập kế hoạch chi tiết và phân bổ công việc cho từng thành viên trong nhóm.
  - Nguyễn Phúc Lộc, Nguyễn Vũ Lâm sẽ là người lên kế hoạch cho cả nhóm phân công vai trò cho mỗi thành viên. Mỗi người sẽ được khai thác điểm mạnh riêng của bản thân trong dự án này.

## **Giai đoạn 2: Thiết kế và phát triển phần cứng ( 1 tuần )**

Sau khi hoàn thành giai đoạn 1 xong, đã có sự thống nhất về mặt phần cứng là lúc thực hiện giai đoạn 2.

- Mua sắm các linh kiện như kế hoạch đã đề ra (cảm biến, động cơ, vi điều khiển, khung xe..).
  - Vũ Đức Hoàng Luân và Đỗ Huỳnh Hữu Thắng có nhiệm vụ đi mua linh kiện điện tử theo yêu cầu.
- Thiết kế sơ đồ kết nối giữa các linh kiện cho xe.
  - Thiết kế sơ đồ kết nối đòi hỏi phải am hiểu về nối dây cơ bản giữa các linh kiện điện tử với nhau và phải rõ ràng khi đó khai báo trong lập trình cũng sẽ dễ dàng hơn.
  - Nguyễn Huy Khang, Võ Quang Liêm cũng đã tìm hiểu qua nên cũng dễ dàng làm hơn trong việc này.

- Lắp ráp hoàn chỉnh.
  - Nguyễn Huy Khang và Võ Quang Liêm đã vẽ ra sơ đồ kết nối. Lúc này, Trần Nhất Trung sẽ lắp ráp theo sơ đồ thành một mô hình hoàn chỉnh.

### **Giai đoạn 3: Lập trình và phát triển phần mềm ( 2 tuần )**

Đây là một giai đoạn đòi hỏi sự tập trung và là giai đoạn quyết định xem xe có hành động theo đúng yêu cầu đã đề ra hay chưa. Đưa ra gợi ý hành vi của xe, Huỳnh Ngọc Quốc Khang sẽ có những ý kiến mang tính xây dựng phù hợp từ đó Nguyễn Phú Hữu và Đào Văn Sơn sẽ là người viết ý tưởng đó vào xe.

- Lập trình cho vi điều khiển để xử lý tín hiệu từ cảm biến và điều khiển động cơ.
- Xây dựng thuật toán dò line do Nguyễn Phú Hữu đảm nhận.
- Xây dựng thuật toán tránh vật cản do Đào Văn Sơn đảm nhận.
- Tích hợp kiểm tra hoạt động của cả phần cứng và phần mềm.
  - Arduino IDE là phần mềm được chọn để thực hiện lập trình trên đó.
  - Proteus đơn giản để sử dụng để mô phỏng mô hình.

### **Giai đoạn 4: Thử nghiệm và điều chỉnh**

Giai đoạn này Nguyễn Phú Hữu và Đào Văn Sơn có trách nhiệm hướng dẫn, đồng hành cùng 3 thành viên là Nguyễn Văn Quý, Phan Anh Trí và Nguyễn Đình Vũ về cách thức hoạt động của xe dò line và tránh vật cản như thế nào?

- Hướng dẫn cách sửa và điều chỉnh sao cho phù hợp trong môi trường thực tế.
- Phân tích và điều chỉnh hệ thống để tối ưu hoá hiệu suất.

### **Giai đoạn 5: Đánh giá và báo cáo kết quả**

Hoàn thành 4 giai đoạn trên cũng là lúc đánh giá và viết báo cáo kết quả. 3 thành viên Huỳnh Ngọc Thiên Phương, Lê Ngọc Phương Uyên và Phan Thị Như Ý đảm nhiệm vai trò này.

- Đánh giá kết quả thử nghiệm, hiệu suất của hệ thống.
- Viết báo cáo tổng kết và đề xuất các cải tiến cho hệ thống.

#### **Tài nguyên cần thiết:**

- Nhân lực: Các thành viên trong nhóm sẽ chia đều trách nhiệm cho nhau như thiết kế phần cứng, lập trình viết phần mềm và kiểm thử.
- Vật tư: Là các linh kiện cần thiết để xây dựng hệ thống:
- Công cụ viết phần mềm: Arduino IDE, phần mềm mô phỏng TinkerCAD.
- Trang thiết bị thử nghiệm: Bản đồ và vật cản.

## **2. Thiết kế hệ thống**

Dựa vào sự thống nhất trong giai đoạn 1, nhóm đã đưa ra quyết định chọn những linh kiện như sau:

#### **Chọn cảm biến và phần cứng:**

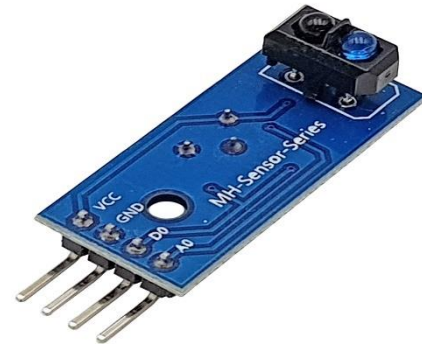
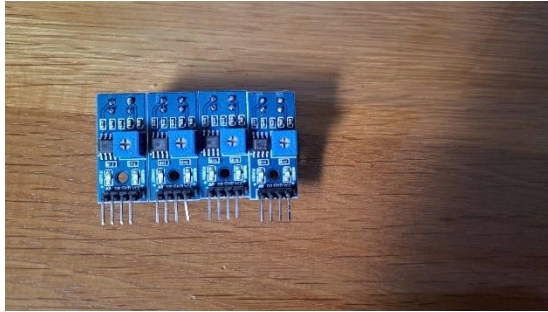
##### **➤ Module Cảm Biến Dò Line Đơn TCRT5000 (3 + 1):**

Các cảm biến dò line TCRT5000 sẽ được sử dụng để nhận diện đường kẻ (line) trên mặt đất. Cảm biến này hoạt động dựa trên nguyên lý phản xạ ánh sáng: khi có sự thay đổi màu sắc (đường kẻ sáng hoặc tối), cảm biến sẽ phát hiện và truyền tín hiệu về vi điều khiển để điều khiển hướng di chuyển của xe.

#### **Thông số kỹ thuật:**

- Nguồn cung cấp: 5VDC.
- Mạch sử dụng chip so sánh LM393.
- Dòng điện tiêu thụ: <10mA.
- Dải nhiệt độ hoạt động: 0oC ~ 50oC.
- Ngõ giao tiếp: 4 dây VCC, GND, DO, AO.
- Mức tín hiệu ngõ ra: TTL.
- Kích thước: 3.2 x 1.4mm.

### Hình ảnh thực tế:



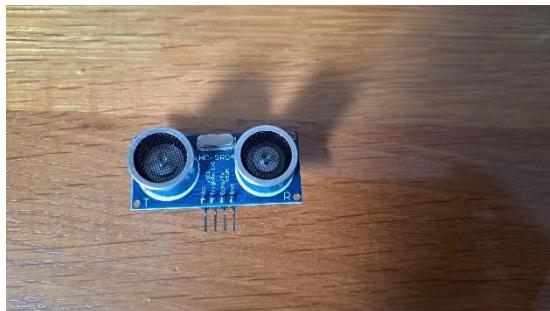
#### ➤ **Cảm Biến Siêu Âm HC-SR04 (1 cái):**

Cảm biến siêu âm HC-SR04 sẽ được sử dụng để phát hiện vật cản phía trước xe. Cảm biến này hoạt động bằng cách phát sóng siêu âm và đo khoảng cách từ xe đến vật cản bằng cách tính toán thời gian sóng siêu âm phản hồi.

#### **Thông số kỹ 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

### Hình ảnh thực tế:



#### ➤ **Động Cơ DC Giảm Tốc Vàng 2 Trục 1:48 (2 cái) + Bánh Xe:**



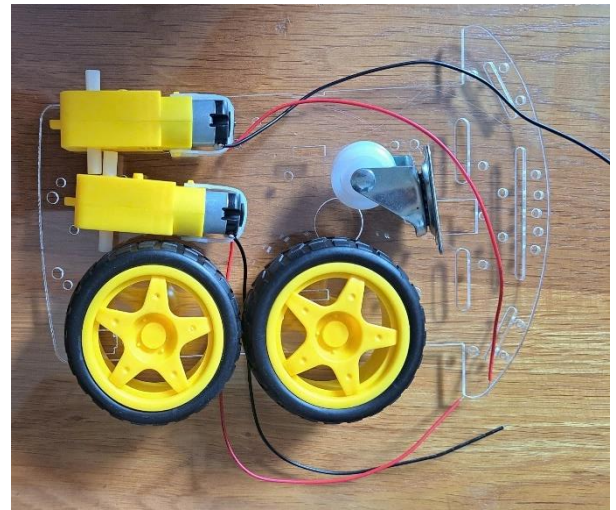
Hai động cơ DC giảm tốc sẽ giúp xe di chuyển. Mỗi động cơ gắn với một bánh xe, đảm bảo rằng xe có thể di chuyển theo các hướng và thay đổi hướng khi cần thiết.

Tỉ lệ giảm tốc 1:48 giúp động cơ cung cấp mô-men xoắn lớn hơn, phù hợp cho xe di chuyển chính xác theo đường kẻ và tránh vật cản.

#### **Thông số kĩ thuật:**

- Điện áp hoạt động: 3-9VDC
- Dòng điện tiêu thụ: 110140mA
- Tỉ số truyền: 1:120
- Số vòng/1phút:
  - 50 vòng/ 1 phút tại 3VDC.
  - 83 vòng/ 1 phút tại 5VDC.
- Moment: 1.0KG.CM
- Chất liệu: Nhựa, cao su, mút
- Đường kính: 65mm

#### **Hình ảnh thực tế:**



#### **➤ Mạch Điều Khiển Động Cơ DC L293D**

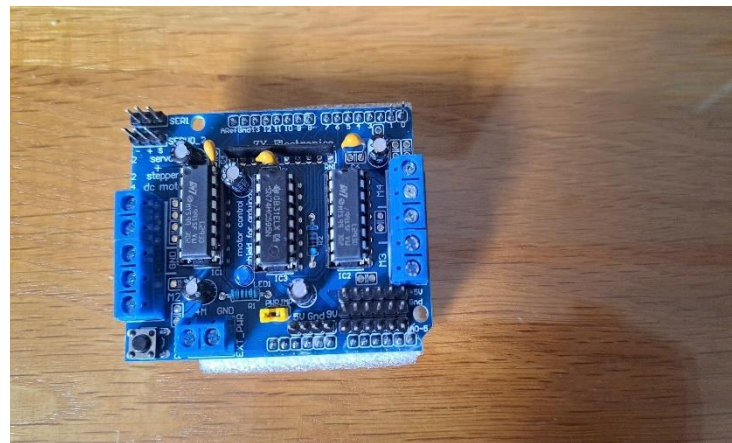
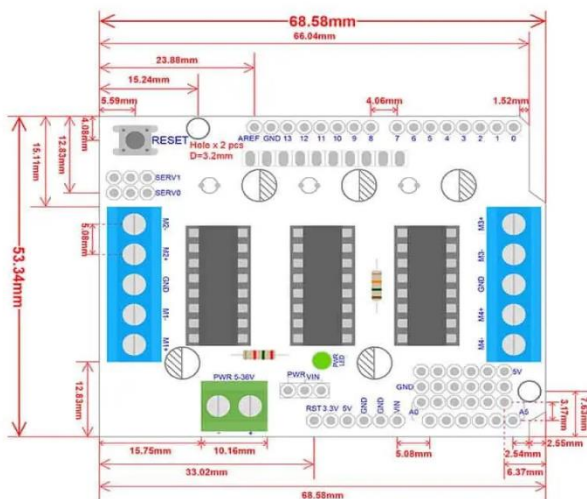
**(1 cái):**

Mạch L293D là mạch điều khiển động cơ giúp điều khiển hai động cơ DC. Mạch này cho phép điều khiển động cơ quay tới, quay lui và dừng lại thông qua các tín hiệu từ vi điều khiển.

### Thông số kĩ thuật:

- Nguồn hoạt động: 5VDC từ Arduino.
- Nguồn cấp cho động cơ: 4.8~35VDC
- Có Jumper để thiết lập nguồn cấp cho động cơ nối với chân Vin để cấp nguồn cho Arduino tuy nhiên nếu nguồn cấp cho động cơ > 9VDC vui lòng không sử dụng và cấp nguồn riêng (áp dụng cho các mạch Arduino TQ, không áp dụng cho Vietduino).
- IC driver động cơ: 2 x L293D
- Điều khiển được 4 động cơ DC hoặc 2 động cơ bước công suất nhỏ ( < 600mA, 4.5~36VDC)
- Điều khiển được 2 động cơ RC Servo.
- Tích hợp ngõ ra cho các chân Analog của Arduino.
- Tích hợp nút nhấn Reset.

### Hình ảnh thực tế:



➤ **Mạch Arduino Uno R3 (1 cái):**

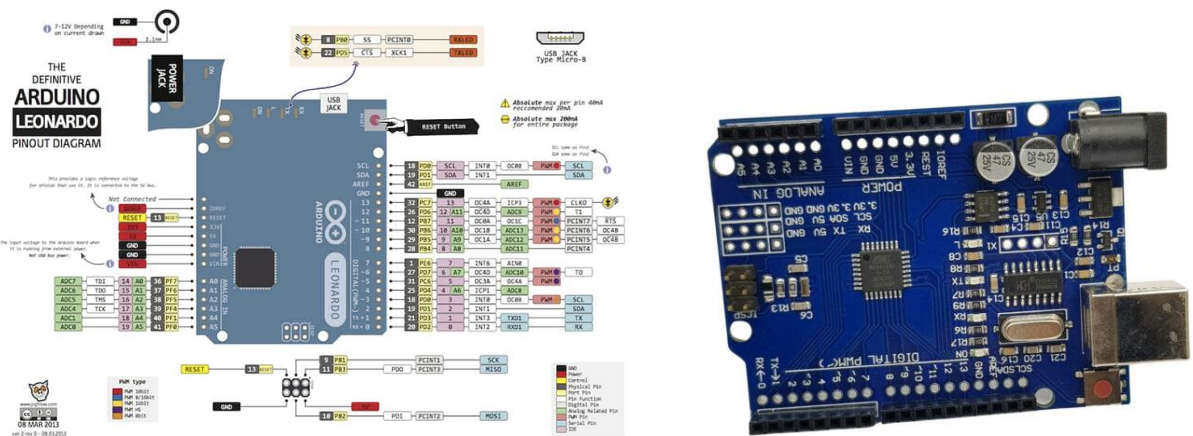
Arduino Uno R3 là vi điều khiển trung tâm của hệ thống, sẽ xử lý tín hiệu từ cảm biến dò line và cảm biến siêu âm, đồng thời điều khiển mạch L293D để điều khiển động cơ DC. Arduino sẽ chạy các chương trình lập trình sẵn để thực hiện các hành động của xe (dò line, phát hiện và tránh vật cản).

**Thông số kỹ thuật:**

Chip điều khiển chính	ATmega328P
Điện áp hoạt động	5V. Tốt nhất bạn cấp nguồn 5V cho Arduino từ cổng USB. Nếu dùng nguồn ngoài( cắm từ giắc DC): Khuyến dùng 7-9V để mạch hoạt động tốt. Khi điện áp lên tới 12V IC ỏn áp rất nóng dễ hư hỏng mạch.
Số chân Analog	6
Số chân Digital	14 ( 6 chân PWM)
Dòng ra trên chân digital	Max 40 mA
Dòng ra trên chân 5V	500 mA
Dòng ra trên chân 3.3V	50 mA
Dung lượng bộ nhớ Flash	32 KB (ATmega328P)
SRAM	2 KB (ATmega328P)

EEPROM	1 KB (ATmega328P)
Tốc độ	16 MHz
Trọng lượng	25 gram
Chân PWM (~)	3, 5, 6, 9, 10, và 11

**Hình ảnh thực tế:**



### ➤ Động cơ servo SG90 180 độ (1 cái):

Động cơ servo SG90 là một thiết bị cơ điện tử nhỏ gọn được thiết kế để kiểm soát chuyển động với độ chính xác cao trong không gian hạn chế.

#### Thông số kỹ thuật:

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

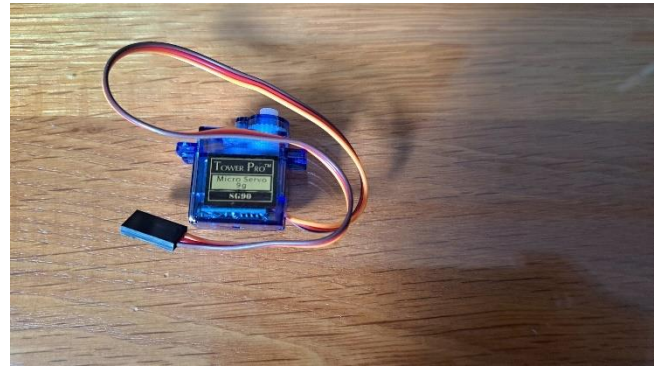
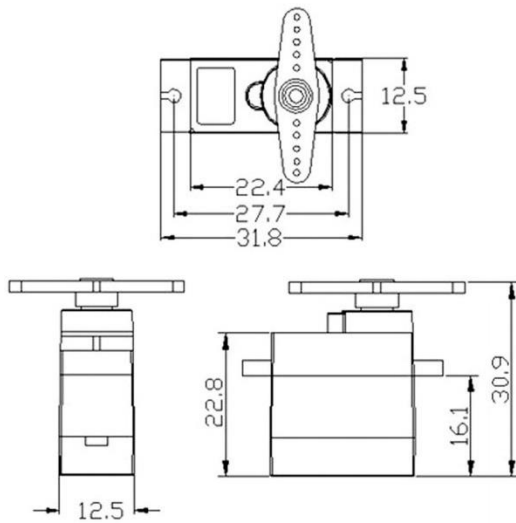
#### Phương pháp điều khiển PWM:

- Độ rộng xung 0.5ms ~ 2.5ms tương ứng 0-180 độ
- Tần số 50Hz, chu kỳ 20ms

### Sơ đồ dây:

- Đỏ: Dương nguồn
- Nâu: Âm nguồn
- Cam: Tín hiệu

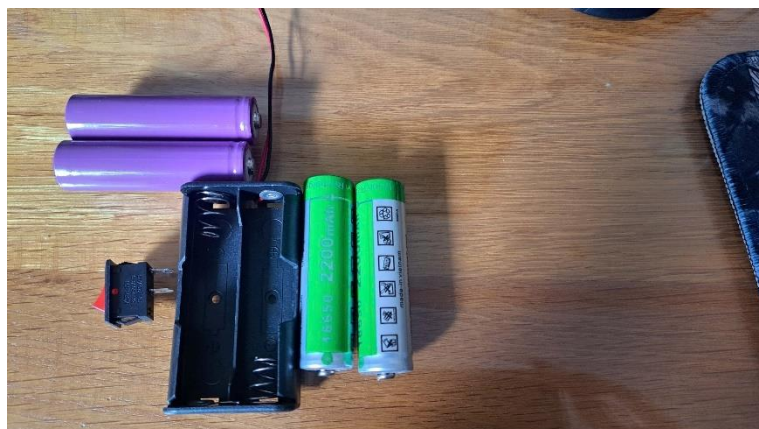
### Hình ảnh thực tế:



### ➤ Hộp Pin 18650 2 cell (1 cái) và Pin Cell 18650 2200mAh (2 + 2):

Hộp pin và pin 18650 cung cấp năng lượng cho xe. Với dung lượng 2200mAh, loại pin này đảm bảo xe có đủ năng lượng để hoạt động trong thời gian dài mà không cần sạc lại quá thường xuyên.

### Hình ảnh thực tế:





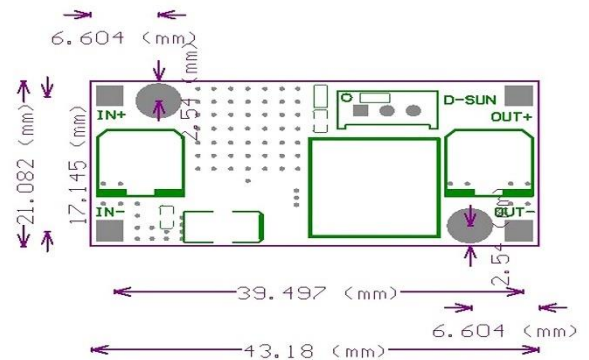
➤ **Mạch Giảm Áp DC LM2596 3A (1 cái):**

Mạch LM2596 sẽ giúp giảm áp từ nguồn điện 7.4V (pin 18650) xuống mức điện áp ổn định 5V để cấp nguồn cho Arduino và các cảm biến.

**Thông số kỹ thuật:**

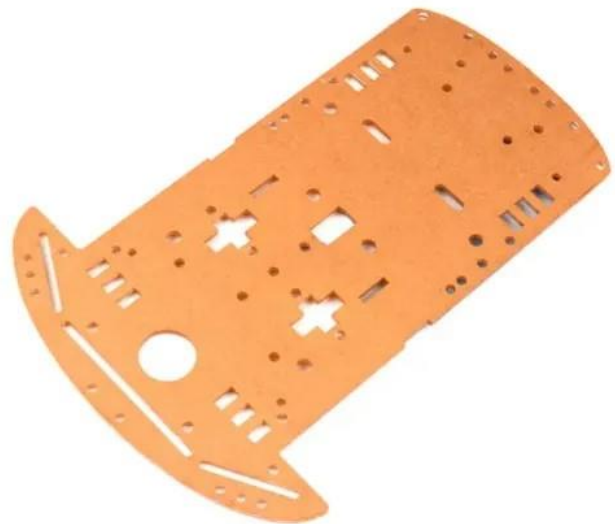
- Điện áp đầu vào: Từ 3V đến 30V.
- Điện áp đầu ra: Điều chỉnh được trong khoảng 1.5V đến 30V.
- Dòng đáp ứng tối đa là 3A.
- Hiệu suất: 92%
- Công suất: 15W
- Kích thước: 45 (dài) \* 20 (rộng) \* 14 (cao) mm

**Hình ảnh thực tế:**



➤ **Khung xe bằng mica:**

Khung xe sẽ được chế tạo từ mica để giữ các linh kiện lại với nhau một cách chắc chắn và dễ dàng gắn các bộ phận vào.



Công việc thiết kế và đi dây trên bản vẽ do Nguyễn Huy Khang và Võ Quang Liêm có trách nhiệm vẽ sơ đồ kết nối giữa các linh kiện với nhau trong giai đoạn 2. Mạch điện của hệ thống gồm các thành phần chính như sau:

- 14

- **Cảm biến siêu âm HC-SR04** sẽ được kết nối với các chân Analog từ A0-A5 trên mạch điều khiển L293D giống như cảm biến TCRT5000 gắn trực tiếp trên Arduino, giúp đo khoảng cách từ xe đến vật cản phía trước. Nếu phát hiện vật cản, Arduino sẽ gửi lệnh dừng hoặc điều chỉnh hướng di chuyển của xe.
- **Mạch điều khiển L293D** sẽ được kết nối trực tiếp với các chân điều khiển trên Arduino để điều khiển các động cơ DC. Mạch này giúp điều khiển động cơ quay tới, quay lui và dừng lại tùy theo tín hiệu từ Arduino.
- **Nguồn điện** sẽ được cung cấp từ 2 pin 18650 thông qua mạch giảm áp LM2596 để cung cấp điện áp ổn định 5V cho Arduino và các linh kiện khác.

**Sơ đồ mạch điện** (được vẽ trên phần mềm TinkerCAD) sẽ mô tả cách kết nối các linh kiện, đảm bảo rằng tất cả các phần cứng có thể hoạt động ổn định và hiệu quả khi thực hiện nhiệm vụ dò line và tránh vật cản.

#### **Đảm bảo hoạt động ổn định:**

Để đảm bảo rằng hệ thống hoạt động ổn định và hiệu quả, các linh kiện sẽ được kết nối chắc chắn và sử dụng các mạch điện bảo vệ (ví dụ như cầu chỉnh lưu để bảo vệ nguồn, mạch giới hạn dòng để bảo vệ động cơ). Ngoài ra, phần mềm điều khiển sẽ được tối ưu hóa để xử lý các tín hiệu từ cảm biến nhanh chóng và chính xác, giúp xe có thể thực hiện các thao tác điều khiển mượt mà và tránh va chạm một cách hiệu quả.

**SƠ ĐỒ NỐI KẾT NỐI MẠCH** ta sẽ nói bên phần 2!

## **4. Lập trình và thuật toán điều khiển**

### **Lập trình vi điều khiển:**

Trong dự án này Arduino Uno R3 là trung tâm xử lý tín hiệu và điều khiển hoạt động của xe. Các tín hiệu từ cảm biến dò line và cảm biến siêu âm sẽ được Arduino tiếp nhận và xử lý thông qua các chương trình lập trình được viết trong môi trường Arduino IDE.



Sau khi Trần Nhất Trung hoàn thành công việc lắp ráp, phần còn lại là phần mềm do Nguyễn Phú Hữu và Đào Văn Sơn phụ trách:

**1. Kết nối cảm biến và động cơ:**

- Các cảm biến TCRT5000 và HC-SR04 được kết nối vào các chân Analog trên mạch điều khiển L293D gắn trực tiếp với Arduino. Arduino sẽ nhận các tín hiệu từ cảm biến và chuyển sang các lệnh điều khiển động cơ.
- Các động cơ DC được điều khiển thông qua mạch L293D, với các chân điều khiển từ Arduino để điều chỉnh chuyển động của xe (quay trái, phải, tiến, lùi).

**2. Xử lý tín hiệu cảm biến:**

- Tín hiệu từ cảm biến TCRT5000 (dò line) và cảm biến HC-SR04 (siêu âm) sẽ được đọc và xử lý qua các hàm trong chương trình. Các tín hiệu này sẽ được chuyển đổi thành dữ liệu để điều khiển động cơ.
- Chương trình sử dụng hàm **digitalRead()** để đọc tín hiệu từ cảm biến TCRT5000 và thư viện **<NewPing.h>** sẽ xử lý đọc tín hiệu từ cảm biến HC-SR04.

**3. Điều khiển động cơ:**

- Để điều khiển động cơ, Arduino sử dụng các chân PWM (Pulse Width Modulation) để điều chỉnh tốc độ và hướng quay của động cơ. Các hàm được cho trong thư viện **<AFMotor.h>** được sử dụng để gửi tín hiệu điều khiển cho mạch L293D.

**XÂY DỰNG THUẬT TOÁN** là một phần quan trọng ta sẽ nói tiếp bên phần 2!

## **PHẦN 2: TIỀN TRÌNH PHÁT TRIỂN DỰ ÁN**

### **1. Khởi động dự án**

Giai đoạn khởi động dự án luôn là bước quan trọng nhất là nền tảng để thúc đẩy trong quá trình phát triển. Chúng tôi đã đặt ra những mục tiêu rất rõ ràng: phát triển một hệ thống xe tự hành có khả năng nhận diện vạch kẻ, điều khiển theo vạch đi được xác định, và tránh được các vật cản xuất hiện bất ngờ trong suốt quá trình di chuyển. Mục tiêu này không chỉ yêu cầu sự chính xác trong thiết kế và lập trình mà còn đòi hỏi khả năng thích ứng trong môi trường thực tế, nơi các yếu tố ngoại cảnh có thể thay đổi bất ngờ.

Ngay khi mục tiêu đã được xác định rõ là chuẩn bị và chọn linh kiện cần thiết cho dự án. Chúng tôi đã họp lại nói chuyện với nhau để phân công công việc cho từng thành viên mỗi người có một thế mạnh sẽ làm phần mình giỏi nhất. Dự án được phân chia thành các giai đoạn rõ ràng, mỗi giai đoạn có một mục tiêu cụ thể, thời gian thực hiện, và nguồn lực đảm bảo. Việc chuẩn bị kỹ lưỡng này giúp đảm bảo chất lượng kịp tiến độ cho dự án. Các thành viên trong nhóm hiểu rõ nhiệm vụ, cung cấp một lộ trình phù hợp hợp lý cũng giúp các thành viên không bị khủng hoảng về tâm lý khi làm.

### **2. Thiết kế và phát triển**

Đây là giai đoạn quyết định phần lớn đến sự thành công của dự án, bao gồm 2 phần:

- Phần cứng
- Phần mềm

#### **a. Phần cứng**

**Kiến trúc hệ thống:** Xác định cấu trúc tổng thể của hệ thống, bao gồm các thành phần linh kiện với nhau, kết nối giữa các bộ phận, và cách thức hoạt động của chúng.

**Thiết kế hệ thống:** Sơ đồ kết nối giữa các linh kiện phải chính xác, dễ nhìn khi đó ai nhìn vào cũng có thể hiểu, chú thích rõ ràng cụ thể.

**Lựa chọn linh kiện:** Cũng được cả nhóm thống nhất phải phù hợp các tiêu chí:

- **Tính tương thích:** Các linh kiện cần phải tương thích với nhau và hệ thống phần mềm mà nhóm đang phát triển.
- **Chi phí:** Ngân sách của dự án đóng vai trò quan trọng trong việc lựa chọn linh kiện. Do là sinh viên nên những linh kiện được chọn đều được chọn kĩ càng nhưng vẫn đảm bảo về chất lượng, hợp lý trong dự án.
- **Hiệu suất:** Các linh kiện phải có hiệu suất cao, đáp ứng các yêu cầu kỹ thuật của hệ thống, đặc biệt trong xử lý thông tin nhanh và chính xác.
- **Độ bền và độ tin cậy:** Các linh kiện được mua trong những cửa hàng có tiếng như IC ĐÂY RỒI, Nshop,... đảm bảo độ bền cao và khả năng hoạt động ổn định trong thời gian dài, đặc biệt trong môi trường hoạt động khắc nghiệt.

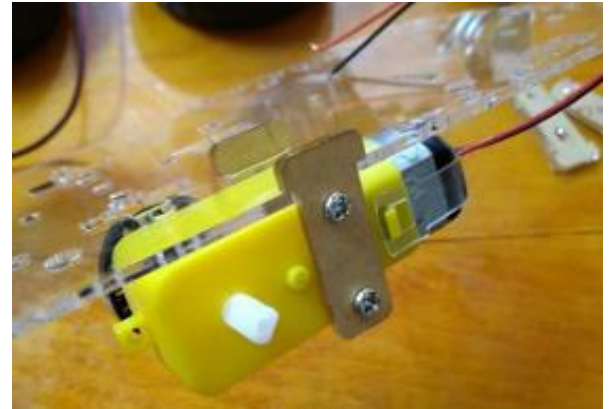
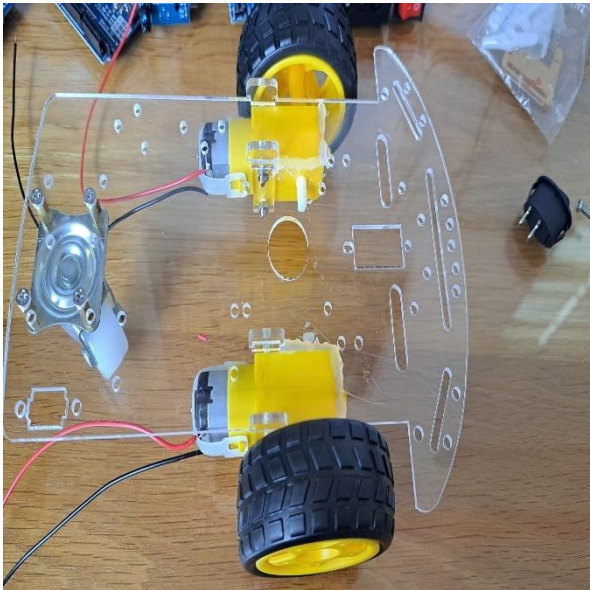
**Hướng dẫn sơ đồ nối mạch:** Được cung cấp bởi Nguyễn Huy Khang, Võ Quang Liêm và quá trình lắp ráp của Trần Nhất Trung.

**Bước 1:** Lắp ráp 2 Động Cơ DC Giảm Tốc Vàng 2 Trục 1:48 + Bánh xe vào khung xe.

- Hàn dây vào động cơ.



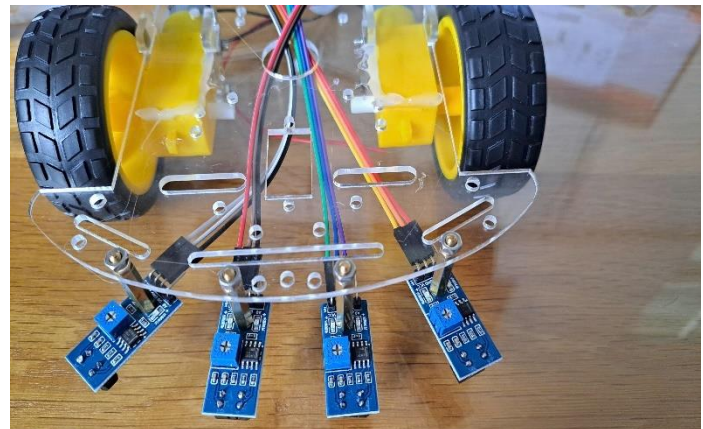
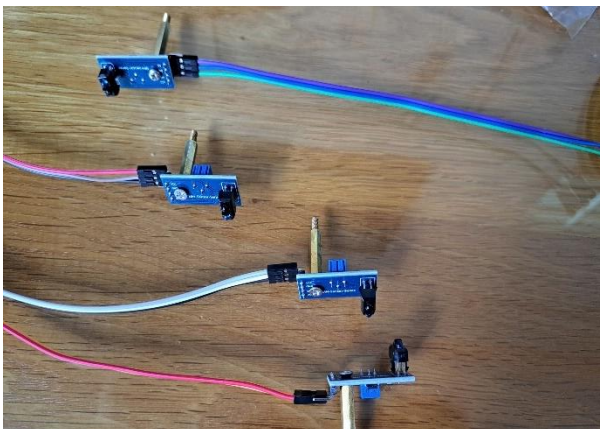
- Lắp động cơ vào khung xe, sử dụng thanh nẹp và ốc vít.



**Bước 2:** Lấy 3 dây cắm test board Cái-Cái 20 cm kết nối vào 3 chân của Cảm Biến Dò Line Đơn TCRT5000 (VCC, GND, D0). Sử dụng trụ đồng 20mm+ ốc để cố định cảm biến vào khung xe.

**\*CHÚ THÍCH:**

- + **VCC** là chân kết nối với chân cấp nguồn 5V trên mạch điều khiển L293D.
- + **GND** là chân nối đất với GND của nguồn.
- + **D0** là chân đầu ra số, cung cấp ngưỡng (HIGH/LOW) dựa trên ngưỡng ánh sáng mà cảm biến nhận được.



**Bước 3:** Lắp giá đỡ cho cảm biến siêu âm HC-SR04 và giá đỡ cho Servo SG90

- Cắm 4 dây cắm test board Cái-Cái 20 cm vào 4 chân của cảm biến siêu âm HC-SR04(VCC, Trig, Echo, GND).

**\*CHÚ THÍCH:**

+ **VCC** là chân kết nối với chân cấp nguồn 5V trên mạch điều khiển L293D

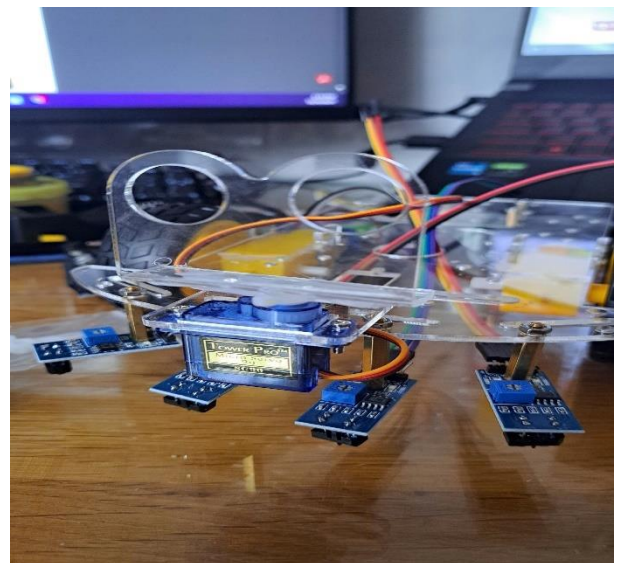
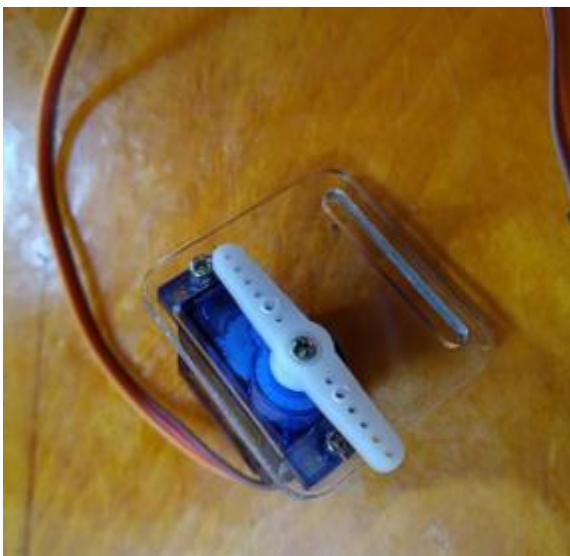
+ **Trig** là chân tín hiệu kích hoạt. Sau khi nhận tín hiệu này, cảm biến sẽ phát ra xung siêu âm từ bộ phát (ultrasonic transmitter).

+ **Echo** là chân tín hiệu phản hồi. Sau khi phát sóng siêu âm, chân Echo sẽ giữ mức **cao** (High) trong khoảng thời gian sóng âm di chuyển tới vật cản và phản xạ về bộ nhận (ultrasonic receiver).

Thời gian chân Echo giữ mức cao tỉ lệ thuận với khoảng cách đo được: Khoảng cách(cm)= (Thời gian xung cao( $\mu$ s) x 0.034)/2

Trong đó, 0,034cm/ $\mu$ s là tốc độ âm thanh(340m/s), và chia 2 do sóng đi và về.

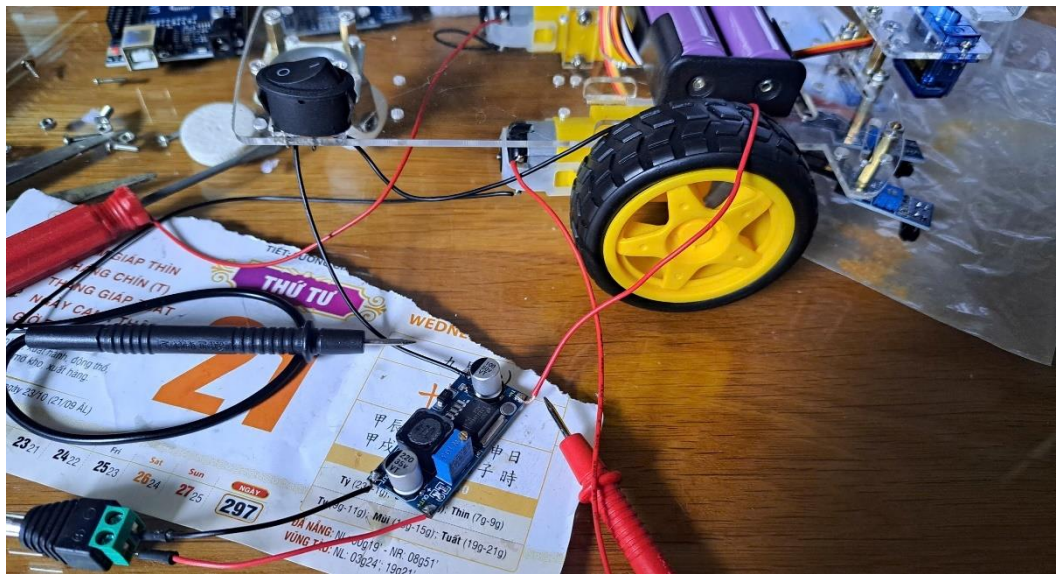
+ **GND** là chân nối đất với GND của nguồn.



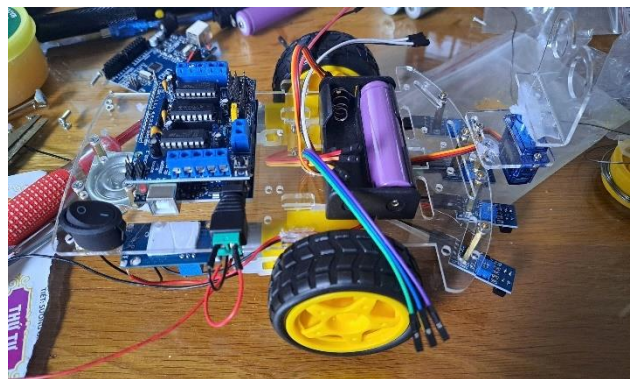


**Bước 4:** Hàn dây kết nối với Mạch Giảm Áp DC LM2596 3A và kết nối hộp pin 18650 2 cell và công tắc.

- Trên Mạch Giảm Áp DC LM2596 3A có 2 bên IN và OUT. Trong đó IN là điện áp đầu vào nối từ nguồn đi vào, OUT là điện áp đầu ra nhận được sau khi điều chỉnh. Ta sẽ nối dây đỏ của hộp pin với IN+, dây đen của hộp pin được nối vào 1 chân của công tắc và chân còn lại của công tắc sẽ nối vào IN-. OUT+ nối với đầu cắm + DC, OUT- nối với đầu cắm – DC.



**Bước 5:** Lắp Mạch điều khiển L293D lên trên mạch Arduino Uno R3 sau đó cố định bằng keo dán trên khung xe.



**Bước 6:** Nối dây cảm biến vào Mạch điều khiển L293D.

- Nối 3 cảm biến dò line:

<b>SHIELD L293D</b>	<b>NỐI</b>	<b>CẢM BIẾN DÒ LINE</b> (sử dụng: VCC, GND, D0)
A5	-	D0 (SensorLeft)
A4	-	D0 (SensorMiddle)
A3	-	D0 (SensorRight)
+5V	-	VCC
GND	-	GND

- Nối dây cảm biến siêu âm HC-SR04:

<b>SHIELD L293D</b>	<b>NỐI</b>	<b>CẢM BIẾN SIÊU ÂM</b> <b>HC-SR04</b>
+5V	-	VCC
A1	-	Trig
A0	-	Echo
GND	-	GND

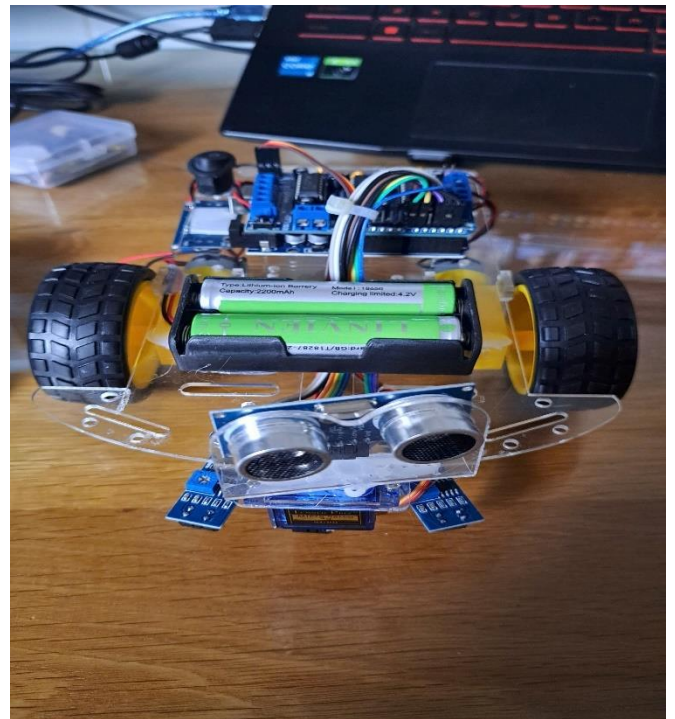
- Nối dây cho Servo SG90 quay cảm biến siêu âm:

<b>Shield L293D</b> (nối với Servo 2)	<b>Nối</b>	<b>Động cơ SG90</b>
S	-	PWM (dây cam)
+	-	+5V (dây đỏ)
-	-	GND (dây nâu)

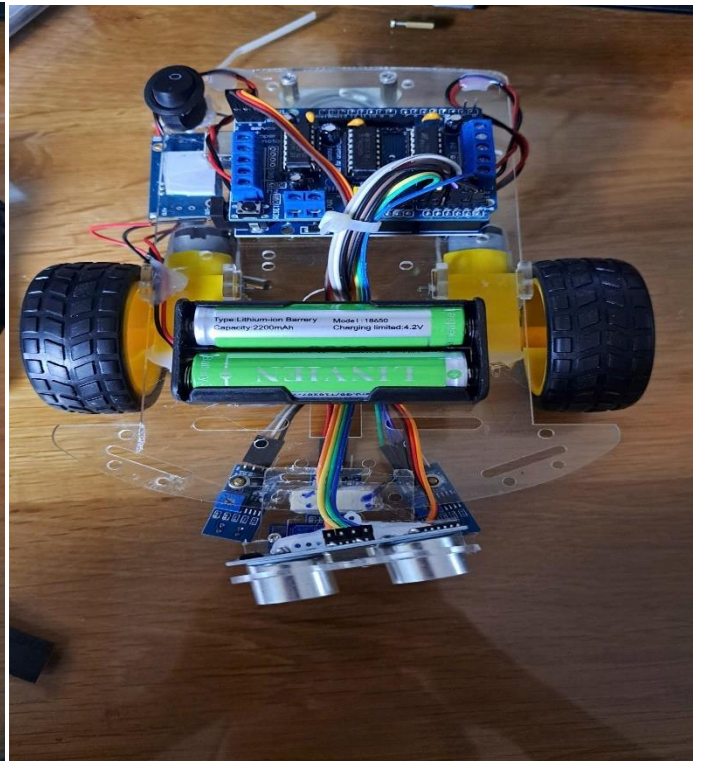
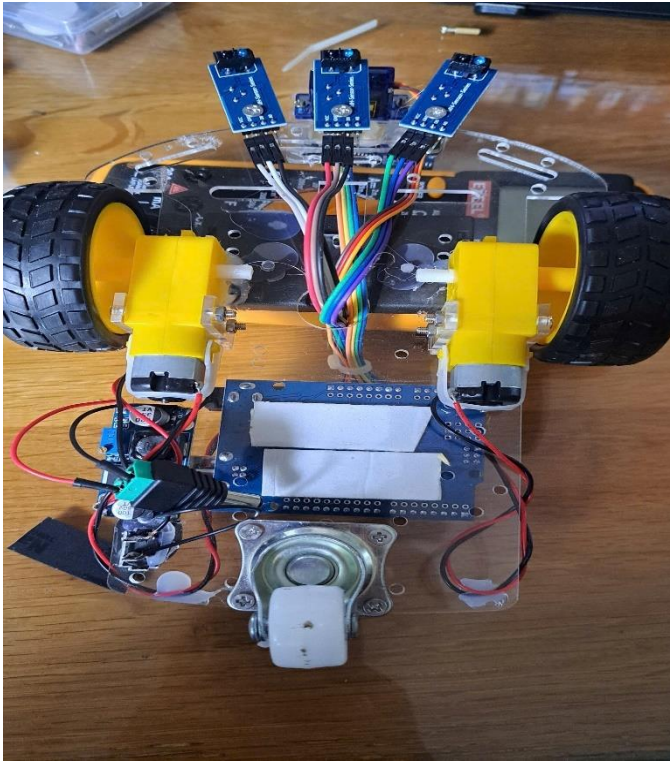
- Nối dây cho động cơ DC:

SHIELD L293D	NỐI	ĐỘNG CƠ DC (BÁNH XE)
M2	-	Động cơ DC(phải)
M3	-	Động cơ DC(trái)
<b>Chú ý:</b> Nếu lắp dây mà động cơ chạy ngược (ví dụ: nhấn tiến, động cơ quay lùi, Chúng ta đảo lại 2 dây động cơ cắm vào L293D )		

**LẮP PIN, XE SÁNG ĐÈN VẬY LÀ ĐÃ HOÀN THÀNH PHẦN CỨNG!!**







## b. Phần mềm

- Chúng ta sẽ chia xe ra làm 2 chức năng tương đương với việc dò line và tránh vật cản. Dò line do Nguyễn Phú Hữu thực hiện và tránh vật cản do Đào Văn Sơn thực hiện.

**DÒ LINE:** Xe tự động có khả năng di chuyển dọc theo một đường vạch được định sẵn, thường là một đường màu đen hoặc trắng trên nền có màu tương phản. Đây là một ứng dụng cơ bản trong lĩnh vực xe tự hành và được sử dụng rộng rãi trong giáo dục và nghiên cứu để tìm hiểu về cảm biến, điều khiển và lập trình.

### ➤ Nguyên lý hoạt động

Xe dò line hoạt động dựa trên cảm biến ánh sáng (thường là cảm biến hồng ngoại) để phát hiện sự khác biệt giữa vạch và nền. Dưới đây là các bước hoạt động cơ bản:

#### 1. Phát hiện vạch đường:

- Xe sử dụng các cảm biến hồng ngoại (IR sensors) để nhận biết cường độ ánh sáng phản xạ từ bề mặt.
- Khi cảm biến nhận thấy sự thay đổi ánh sáng (ví dụ: ánh sáng phản xạ từ vạch màu đen thấp hơn so với nền trắng), nó sẽ xác định được vị trí của đường.

## 2. Điều chỉnh hướng di chuyển:

- Dựa trên dữ liệu từ cảm biến, bộ điều khiển của xe (thường là vi điều khiển như Arduino) sẽ quyết định cách di chuyển.
- Nếu cảm biến phát hiện xe lệch khỏi vạch, tín hiệu điều khiển sẽ được gửi đến động cơ để điều chỉnh hướng trở lại đúng đường.

## 3. Di chuyển liên tục:

- Xe sẽ liên tục dò vạch và điều chỉnh hướng để di chuyển theo đường.

### Vì sao lại là 3 cảm biến mà không phải 4 hoặc 5?

- Vì đây là mô hình xe dò line cơ bản thử nghiệm, với 3 cảm biến sẽ giúp tiết kiệm được số trường hợp khi lập trình giao tiếp chủ yếu sẽ là cảm biến giữa (**SensorMiddle**) và 2 cảm biến còn lại khi phát hiện tín hiệu vạch sẽ điều chỉnh cho xe cân bằng và ổn định.

### Mục đích của code:

#### Phát hiện đường vạch:

- Sử dụng ba cảm biến hồng ngoại (IR) gắn ở dưới xe để nhận biết đường (line) trên bề mặt.
- Cảm biến trái (**SensorLeft**), cảm biến giữa (**SensorMiddle**), và cảm biến phải (**SensorRight**) được đặt để phát hiện vạch đường.

#### Điều khiển động cơ:

- Tùy thuộc vào tín hiệu cảm biến, xe sẽ tiến thẳng, cua trái, cua phải hoặc điều chỉnh nhẹ để giữ xe trên đường.

#### Xử lý trường hợp ngoại lệ:

- Nếu xe không phát hiện vạch đường trong thời gian dài, nó sẽ dừng để tránh chạy lạc hướng.

### Giải thích code:

#### 1. Khởi tạo động cơ:

```
AF_DCMotor motor1(2); // Động cơ phải
```

```
AF_DCMotor motor2(3); // Động cơ trái
```

Thư viện **AFMotor** đơn giản hóa việc điều khiển động cơ, chỉ cần khai báo kênh và tốc độ (thay vì phải trực tiếp điều khiển các tín hiệu PWM hay chân H-Bridge).

#### 2. Định nghĩa cảm biến:

```
const int SensorLeft = A5; // cảm biến trái kết nối với chân A5
```

```
const int SensorMiddle = A4; // cảm biến giữa kết nối với chân A4
```

```
const int SensorRight = A3; // cảm biến phải kết nối với chân A3
```

#### 3. Vòng lặp chính **loop()**: Vòng lặp này liên tục kiểm tra tín hiệu từ cảm biến và đưa ra hành động phù hợp.

- Đọc tín hiệu cảm biến

```
SL = digitalRead(SensorLeft);
```

```
SM = digitalRead(SensorMiddle);
```

```
SR = digitalRead(SensorRight);
```

High (1): Cảm biến phát hiện ánh sáng phản xạ (vạch trắng).

Low (0): không phát hiện ánh sáng phản xạ (nền đen).

- Hiện thị trạng thái cảm biến

```
Serial.print("SL: "); Serial.print(SL);
```

```
Serial.print(" | SM: "); Serial.print(SM);
```

```
Serial.print(" | SR: "); Serial.println(SR);
```

Đề gửi trạng thái lên Serial Monitor để theo dõi.

- Điều kiện để xe di chuyển

Đi thẳng nếu cảm biến giữa phát hiện line.

```

if (SM == HIGH) {
    forward(141, 141); // Tiến thẳng với tốc độ bằng nhau
}

```

Rẽ trái hoặc phải khi hai cảm biến phát hiện vạch.

```

else if ((SM == HIGH && SR == HIGH)) {
    turnLeft(); //cảm biến giữa và trái phát hiện vạch, xe rẽ phải.
}
else if ((SL == HIGH && SM == HIGH)) {
    turnRight(); //cảm biến giữa và phải phát hiện vạch, xe rẽ trái.
}

```

**Lý do sử dụng:** Hai cảm biến phát hiện vạch cho biết xe đang lệch khỏi trung tâm và cần điều chỉnh.

Điều chỉnh nhẹ nếu chỉ một cảm biến ngoài phát hiện vạch.

```

else if (SL == HIGH && SM == LOW) {
    adjustLeft();
}
else if (SR == HIGH && SM == LOW) {
    adjustRight();
}

```

**Mục đích:** Nếu chỉ cảm biến trái hoặc phải phát hiện vạch, xe cần điều chỉnh nhỏ để quay lại đường.

**Lý do sử dụng:** Đây là trường hợp thường xảy ra khi xe vừa lệch khỏi vạch đường.

Dừng khi không phát hiện vạch.

```
else {
    handleNoLine(currentMillis); }
```

**Mục đích:** Nếu không có cảm biến nào phát hiện vạch, xe xử lý trường hợp đặc biệt (dừng hoặc tiếp tục dò).

#### 4. Điều khiển động cơ:

- Hàm forward()

```
void forward(int speedLeft, int speedRight) {
    motor1.setSpeed(speedRight);
    motor1.run(FORWARD);
    motor2.setSpeed(speedLeft);
    motor2.run(FORWARD);
}
```

- Hàm stopMotors()

```
void stopMotors() {
    motor1.setSpeed(0); //Dừng động cơ phải
    motor1.run(RELEASE);
    motor2.setSpeed(0); //Dừng động cơ trái
    motor2.run(RELEASE);
}
```

- Hàm turnLeft() và turnRight()

<pre>void turnLeft() {     motor2.setSpeed(0);    // Dừng động     cơ trái     motor2.run(RELEASE);     motor1.setSpeed(198);  // Động cơ     phải quay nhanh hơn }</pre>	<pre>void turnRight() {     motor1.setSpeed(0); // Dừng động     cơ phải     motor1.run(RELEASE);     motor2.setSpeed(198); //Động cơ     trái quay nhanh hơn }</pre>
---	---

<pre> motor1.run(FORWARD); delay(520); // Giữ thời gian để robot quay đủ checkMiddleSensor(); // Kiểm tra lại tín hiệu cảm biến giữa } </pre>	<pre> motor2.run(FORWARD); delay(500); // Giữ thời gian để robot quay đủ checkMiddleSensor(); // Kiểm tra lại tín hiệu cảm biến giữa } </pre>
---	---

- Điều chỉnh nhẹ adjustLeft() và adjustRight()

<pre> void adjustLeft() {     motor2.setSpeed(205);//Tăng tốc động cơ trái     motor2.run(FORWARD);     motor1.setSpeed(105);//Giảm tốc động cơ phải     motor1.run(BACKWARD);     delay(210);     checkMiddleSensor();// Kiểm tra lại tín hiệu cảm biến giữa } </pre>	<pre> void adjustRight() {     motor1.setSpeed(205);//Tăng tốc động cơ phải     motor1.run(FORWARD);     motor2.setSpeed(106);//Giảm tốc động cơ phải     motor2.run(BACKWARD);     delay(210);     checkMiddleSensor();// Kiểm tra lại tín hiệu cảm biến giữa } </pre>
--	---

- Hàm kiểm tra cảm biến giữa xem còn nằm trên line không  
checkMiddleSensor().

```

void checkMiddleSensor() {
    SM = digitalRead(SensorMiddle);
    if(SM == HIGH) {
        forward(150,150);
    }
}

```

```

    }
    else {
        adjustDirection(); // Điều chỉnh lại hướng nếu không có tín hiệu
        giữa }
    }

```

- Hàm điều chỉnh hướng adjustDirection()

```

void adjustDirection() {
    motor1.setSpeed(130);
    motor1.run(FORWARD);
    motor2.setSpeed(130);
    motor2.run(FORWARD);
}

```

- Hàm handleNoLine()

**Mục đích:** Xử lý khi **không có tín hiệu từ cảm biến** (tức là cả ba cảm biến đều đọc mức **LOW**), robot không xác định được đường dẫn.

**Chức năng:**

- Nếu đã **mất tín hiệu quá lâu** (sensorTimeout), robot sẽ **dừng hoàn toàn** để tránh chạy sai đường.
- Nếu mới mất tín hiệu trong thời gian ngắn, robot sẽ dựa vào cảm biến **trái** và **phải** để:
  - Quẹo trái nếu cảm biến trái (**SL**) vẫn phát hiện tín hiệu.
  - Quẹo phải nếu cảm biến phải (**SR**) vẫn phát hiện tín hiệu.
  - Tiếp tục đi thẳng nếu cả hai cảm biến ngoài không nhận được tín hiệu rõ ràng.

```

void handleNoLine(unsigned long currentMillis) {

```

```

    // Nếu không có tín hiệu từ cảm biến, kiểm tra xem có nên dừng hay không

```

```

if (currentMillis - lastSensorTime > sensorTimeout) {
    stopMotors(); // Nếu quá 3 giây mà không có tín hiệu, dừng lại
}
else {
    // Nếu chưa có tín hiệu từ cảm biến giữa, vẫn di chuyển dựa trên cảm biến
    ngoài
    if (SL == HIGH) {
        goLeft(160, 160); // Quẹo trái nhẹ
    }
    else if (SR == HIGH) {
        goRight(161, 161); // Quẹo phải nhẹ
    }
    else {
        forward(150, 150); // Tiếp tục đi thẳng nếu cảm biến ngoài phát hiện line
    }
}
}

```

**TRÁNH VẬT CẢN:** được lập trình để tự động phát hiện và tránh các chướng ngại vật trong môi trường. Sử dụng các cảm biến siêu âm để đo khoảng cách đến các vật cản và đưa ra quyết định về hướng di chuyển nhằm tránh va chạm.

- **Nguyên lý hoạt động:** Xe tránh vật cản hoạt động dựa trên việc sử dụng **cảm biến siêu âm**, **servo**, và **động cơ** để nhận biết môi trường xung quanh, từ đó đưa ra các quyết định di chuyển phù hợp. Dưới đây là nguyên lý hoạt động chi tiết:

### 1. Khởi tạo:

- Servo được đưa về góc trung lập (90 độ).



- Cảm biến siêu âm đo khoảng cách ban đầu để kiểm tra môi trường xung quanh.

## **2. Di chuyển tiến:**

- Khi không phát hiện vật cản phía trước (khoảng cách lớn hơn 24 cm), xe di chuyển tiến thẳng.
- Cả hai động cơ quay theo chiều kim đồng hồ với tốc độ đặt trước.

## **3. Phát hiện vật cản:**

- Nếu cảm biến siêu âm phát hiện vật cản ở khoảng cách nhỏ hơn hoặc bằng 24 cm, xe dừng lại.

## **4. Xử lý tránh vật cản:**

- Xe lùi lại một khoảng ngắn để tạo không gian.
- Servo xoay cảm biến siêu âm qua phải và trái để đo khoảng cách ở cả hai hướng.
- So sánh khoảng cách đo được:
  - Nếu khoảng cách bên phải lớn hơn hoặc bằng bên trái: xe rẽ phải.
  - Ngược lại: xe rẽ trái.

## **5. Tiếp tục di chuyển:**

- Sau khi rẽ, xe tiếp tục đo khoảng cách phía trước. Nếu không có vật cản, xe lại di chuyển tiến.

## **6. Lặp lại:**

- Vòng lặp này tiếp diễn liên tục trong suốt quá trình hoạt động để đảm bảo xe luôn tránh được các vật cản.

## **Mục đích của code:**

### **Phát hiện vật cản phía trước:**

- Sử dụng cảm biến siêu âm để đo khoảng cách từ xe đến vật cản.
- Nếu phát hiện khoảng cách ngắn hơn giá trị giới hạn (24 cm), xe sẽ ngừng di chuyển để xử lý tình huống.

### **Tự động quyết định hướng tránh:**

- Dùng servo xoay cảm biến siêu âm sang trái và phải để quét khoảng cách.
- So sánh khoảng cách bên trái và phải để chọn hướng tối ưu (hướng nào có khoảng cách lớn hơn).

### **Điều khiển động cơ để thực hiện các hành động:**

- Di chuyển tiến, lùi, hoặc rẽ trái, phải tùy theo tình huống.
- Điều chỉnh tốc độ động cơ để đảm bảo xe hoạt động mượt mà và cân bằng.

### **Tái định vị và tiếp tục di chuyển:**

- Sau khi tránh được vật cản, xe quay trở lại chế độ di chuyển thẳng cho đến khi gặp vật cản tiếp theo.

### **Giải thích code:**

#### **1. Khai báo thư viện**

```
#include <AFMotor.h> // Thư viện điều khiển động cơ DC
#include <NewPing.h> // Thư viện điều khiển cảm biến siêu âm
#include <Servo.h> // Thư viện điều khiển servo
```

#### **2. Định nghĩa chân kết nối và thông số**

```
#define ECHO_PIN A0 // Chân Echo của cảm biến siêu âm
#define TRIG_PIN A1 // Chân Trigger của cảm biến siêu âm
#define MAX_DISTANCE 200 // Khoảng cách tối đa cần đo (cm)
#define MAX_SPEED 235 // Tốc độ tối đa của động cơ
#define MAX_SPEED_OFFSET 20 // Hệ số bù tốc độ để cân bằng giữa hai động cơ
```

#### **3. Khởi tạo các đối tượng**

```
AF_DCMotor motor1(2); // Động cơ 1 trên kênh 2
```

```

AF_DCMotor motor2(3);    // Động cơ 2 trên kênh 3
Servo myservo;           // Đối tượng servo
NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE); // Cảm biến
siêu âm
boolean goesForward = false; // Trạng thái di chuyển tiến
int distance = 100;       // Khoảng cách ban đầu
int speedSet = 0;         // Tốc độ hiện tại
char command;             // Lệnh nhận từ Bluetooth (không sử dụng trong đoạn
code này)
int dugme = 1;           // Trạng thái chế độ tránh vật cản

```

#### 4. Cài đặt ban đầu trong hàm setup()

```

void setup()
{
  myservo.attach(9);      // Gắn servo vào chân 9
  myservo.write(90);      // Đặt góc trung lập (90 độ)
  delay(1000);            // Chờ servo ổn định

  distance = readPing();  // Đọc khoảng cách ban đầu
  delay(100);            // Chờ giữa các lần đọc
  distance = readPing();
  delay(100);

  Serial.begin(9600);     // Khởi tạo giao tiếp Serial
}

```

#### 5. Vòng lặp chính loop()

```

void loop(){

```

```

long duration, distance;
while (dugme == 1) // Khi chế độ tránh vật cản được bật
{
    int distanceR = 0; // Khoảng cách bên phải
    int distanceL = 0; // Khoảng cách bên trái
    delay(40);        // Chờ giữa các lần đọc

    if (distance <= 24) // Nếu khoảng cách <= 24 cm
    {
        moveStop();      // Dừng xe
        delay(100);
        moveBackward();  // Lùi xe
        delay(300);
        moveStop();      // Dừng xe
        delay(200);

        distanceR = lookRight(); // Đo khoảng cách bên phải
        delay(200);
        distanceL = lookLeft();  // Đo khoảng cách bên trái
        delay(200);

        if (distanceR >= distanceL) // So sánh để chọn hướng
        {
            turnRight();    // rẽ phải
            moveStop();
        } else {
            turnLeft();     // rẽ trái

```

```

        moveStop();
    }
} else {
    forward();    // Tiếp tục di chuyển tiến
}

distance = readPing(); // Cập nhật khoảng cách
}
}

```

- **Hoạt động chính:**

1. Nếu phát hiện vật cản gần ( $\leq 24$  cm):
  - Dừng xe, lùi lại.
  - Quét sang phải và trái bằng cảm biến siêu âm.
  - Chọn hướng xa nhất để rẽ.
2. Nếu không có vật cản, tiếp tục di chuyển tiến.

## 6. Các hàm điều khiển di chuyển

### Tiến:

```

void forward() {
    motor1.setSpeed(200);
    motor1.run(FORWARD);
    motor2.setSpeed(200);
    motor2.run(FORWARD);
}

```

### Lùi:

```

void back() {

```

```

    motor1.setSpeed(255);
    motor1.run(BACKWARD);
    motor2.setSpeed(255);
    motor2.run(BACKWARD);
}

```

#### **Rẽ trái:**

```

void left() {
    motor1.setSpeed(255);
    motor1.run(FORWARD);
    motor2.setSpeed(0);
    motor2.run(RELEASE); // Dừng động cơ phải
}

```

#### **Rẽ phải:**

```

void right() {
    motor1.setSpeed(0);
    motor1.run(RELEASE); // Dừng động cơ trái
    motor2.setSpeed(255);
    motor2.run(FORWARD);
}

```

#### **Dừng:**

```

void moveStop() {
    motor1.run(RELEASE);
    motor2.run(RELEASE);
}

```

## **7. Quét và đọc khoảng cách**

**Đo khoảng cách:**

```
int readPing() {  
    delay(70);  
    int cm = sonar.ping_cm();  
    if (cm == 0) {  
        cm = 250; // Nếu không đo được, gán giá trị tối đa  
    }  
    return cm;  
}
```

- Đọc khoảng cách từ cảm biến siêu âm. Nếu không có tín hiệu trả về, gán khoảng cách lớn hơn giới hạn.

**Quét phải:**

```
int lookRight() {  
    myservo.write(50); // Quay servo sang phải  
    delay(200);  
    int distance = readPing(); // Đo khoảng cách  
    delay(100);  
    myservo.write(105); // Trả servo về trung lập  
    return distance;  
}
```

- Servo xoay phải, đọc khoảng cách, sau đó trở về vị trí trung lập.

**Quét trái:**

```
int lookLeft() {  
    myservo.write(160); // Quay servo sang trái  
    delay(200);  
    int distance = readPing(); // Đo khoảng cách  
    delay(100);
```

```
myservo.write(105); // Trả servo về trung lập  
return distance;  
}
```

- Servo xoay trái, đọc khoảng cách, sau đó trở về vị trí trung lập.

**LƯU Ý CHUNG:** Code được viết để phù hợp với dự án này nên khi sử dụng lên cấu hình xe khác sẽ phải thay đổi thông số cho phù hợp!

Link code tham khảo: <https://github.com/HuuzNx/Line-Following-and-Obstacle-Avoiding-Robot-Project.git>



## KẾT LUẬN

Dự án **Xe dò line và tránh vật cản** là một cột mốc quan trọng trong hành trình học tập và nghiên cứu của chúng tôi. Đây là cơ hội để áp dụng các kiến thức lý thuyết vào thực tiễn, từ việc lập trình, thiết kế hệ thống điều khiển đến giải quyết các vấn đề thực tế khi robot hoạt động. Quá trình thực hiện dự án không chỉ giúp nhóm hiểu rõ hơn về cách các cảm biến và động cơ hoạt động mà còn rèn luyện kỹ năng tư duy sáng tạo, làm việc nhóm và xử lý vấn đề.

Xe dò line là minh chứng rõ ràng cho khả năng điều hướng chính xác theo đường dẫn được thiết kế trước. Qua việc lập trình và sử dụng cảm biến quang học, chúng tôi đã hiểu rõ cách robot có thể nhận diện và bám theo vạch kẻ trên đường. Tính năng tránh vật cản cũng mang đến nhiều thử thách thú vị, khi yêu cầu xe phải đưa ra các quyết định kịp thời, như rẽ trái hoặc phải, dựa trên dữ liệu thu thập từ cảm biến siêu âm. Việc kết hợp cả hai chức năng này đã tạo ra một robot tự hành thông minh, có khả năng hoạt động hiệu quả trong môi trường phức tạp.

Trong quá trình thực hiện, nhóm đã gặp nhiều khó khăn như hiệu chỉnh cảm biến, cân bằng tốc độ động cơ, và xử lý các lỗi phát sinh trong lúc chạy thử nghiệm. Tuy nhiên, chính những thử thách này đã giúp nhóm học hỏi thêm nhiều điều và hiểu sâu hơn về công nghệ robot. Bằng cách làm việc nhóm, trao đổi ý kiến và không ngừng cải tiến, chúng tôi đã hoàn thiện sản phẩm và đạt được mục tiêu đề ra.

Dự án không chỉ mang lại một sản phẩm hoạt động ổn định mà còn mở ra nhiều hướng phát triển mới. Trong tương lai, nhóm có thể tiếp tục cải tiến hệ thống bằng cách tích hợp trí tuệ nhân tạo để xe tự học hỏi và đưa ra các quyết định thông minh hơn. Ngoài ra, các ứng dụng thực tế của dự án có thể được

mở rộng trong nhiều lĩnh vực như giao thông, sản xuất công nghiệp, hoặc robot phục vụ trong đời sống hàng ngày.

Dự án này là một bài học lớn cho cả nhóm, không chỉ về mặt kỹ thuật mà còn về tinh thần làm việc. Nó cho chúng tôi thấy rằng, khi có sự đam mê, nỗ lực và sáng tạo, mọi khó khăn đều có thể vượt qua. Thành công của dự án là động lực để cả nhóm tiếp tục cố gắng trong các nghiên cứu và ứng dụng công nghệ trong tương lai.

Nhìn lại toàn bộ quá trình, chúng tôi tự hào vì đã hoàn thành một sản phẩm vừa mang tính học thuật vừa có tính thực tế. Dự án không chỉ là kết quả của sự nỗ lực mà còn là minh chứng cho khả năng sáng tạo của chúng tôi. Đây sẽ là bước đệm để cả nhóm tự tin hơn trên con đường học tập và chinh phục các mục tiêu lớn hơn trong tương lai.

## **TÀI LIỆU THAM KHẢO**

- [1]. Arduino.vn
- [2]. Nshop (nshopvn.com)
- [3]. ICĐÂYRỒI (icdayroi.com)

## LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất tới Thầy Phan Văn Ca – giảng viên môn *Introduction to ECET* – người đã tận tình giảng dạy, định hướng và đồng hành cùng chúng em trong suốt hành trình học tập này. Từ những bài học thực tiễn đến những chia sẻ đầy tâm huyết, Thầy đã giúp chúng em không chỉ mở mang kiến thức mà còn rèn luyện tư duy logic, tinh thần làm việc nhóm và sự chủ động trước mọi thử thách.

Dự án “Xe dò line và tránh vật cản” chính là thành quả của cả quá trình học tập và làm việc nghiêm túc của nhóm chúng em. Trong suốt quá trình thực hiện, dù gặp không ít khó khăn, từ việc nghiên cứu lý thuyết đến áp dụng vào thực tiễn, chúng em luôn nhận được sự hỗ trợ kịp thời và những lời khuyên quý giá từ Thầy. Điều này không chỉ giúp chúng em hoàn thành dự án mà còn tích lũy thêm nhiều kinh nghiệm thực tế, làm tiền đề để phát triển các kỹ năng chuyên môn trong tương lai. Hơn thế nữa, dự án này không chỉ đánh dấu sự kết thúc của môn học mà còn mở ra một hành trình mới, nơi chúng em tự tin hơn trên con đường theo đuổi ước mơ và hoài bão của mình. Chúng em hy vọng sẽ tiếp tục vận dụng những bài học và giá trị mà Thầy truyền đạt để tạo nên những thay đổi tích cực, đóng góp cho xã hội và đất nước.

Một lần nữa, chúng em xin được gửi lời tri ân sâu sắc tới Thầy. Kính chúc Thầy dồi dào sức khỏe, luôn thành công trong sự nghiệp giáo dục và tiếp tục truyền cảm hứng cho nhiều thế hệ sinh viên mai sau.

Nhóm sinh viên thực hiện.

1.Nguyễn Phú Hữu 2.Nguyễn Phúc Lộc 3.Nguyễn Huy Khang 4.Trần Phạm Minh Khoa 5.Trần Nhất Trung 6.Võ Quang Liêm 7.Huỳnh Ngọc Quốc Khang 8.Đào Văn Sơn 9.Nguyễn Vũ Lâm 10.Đỗ Huỳnh Hữu Thắng 11.Huỳnh Ngọc Thiên Phương 12.Lê Ngọc Phương Uyên 13.Phan Thị Như

Ý 14. Phan Anh Trí 15. Nguyễn Văn Quý 16. Vũ Đức Hoàng Luân 17. Nguyễn Đình Vũ