

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA CÔNG NGHỆ THÔNG TIN 1**



**BÁO CÁO BÀI TẬP LỚN**  
**HỌC PHẦN: IOT VÀ ỨNG DỤNG**

**Đề tài:** Xe tự hành tránh chướng ngại vật với vi điều khiển ESP32 và cảm biến siêu âm HC-SR04

**Giảng viên hướng dẫn :** TS. Lê Văn Vịnh

**Nhóm báo cáo:** Nhóm 14

<b>STT</b>	<b>Họ và tên</b>	<b>Mã sinh viên</b>
1	Đặng Quốc Khánh	B22DCCN444
2	Vũ Trọng Khôi	B22DCCN468
3	Phạm Ngọc Long	B22DCCN504
4	Đỗ Hải Nam	B22DCCN552

HÀ NỘI – 2025

# MỤC LỤC

<b>MỤC LỤC .....</b>	<b>i</b>
<b>DANH MỤC TỪ VIẾT TẮT .....</b>	<b>iii</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>iv</b>
<b>DANH MỤC BẢNG BIỂU .....</b>	<b>v</b>
<b>MỞ ĐẦU .....</b>	<b>1</b>
<b>CHƯƠNG 1. BÀI TOÁN PHÁT TRIỂN XE TỰ HÀNH TRÁNH CHƯỚNG NGẠI VẬT .....</b>	<b>2</b>
1.1. Mô tả bài toán.....	2
1.2. Định hướng thực hiện.....	2
1.3. Phân công công việc.....	3
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT ĐỂ PHÁT TRIỂN XE TỰ HÀNH TRÁNH CHƯỚNG NGẠI VẬT .....</b>	<b>4</b>
2.1. Thiết kế hệ thống.....	4
2.1.1. Lớp thiết bị .....	4
2.1.2. Lớp trao đổi thông tin.....	4
2.1.3. Lớp ứng dụng .....	4
2.2. Vi điều khiển ESP32 .....	5
2.3. Mạch điều khiển động cơ L298N.....	6
2.4. Cảm biến siêu âm HC-SR04 .....	7
2.5. Nguyên lý hoạt động .....	8
2.5.1. Các thành phần chính của hệ thống .....	8
2.5.2. Quy trình hoạt động .....	8
2.5.3. Giao tiếp giữa các module.....	10
2.6. Sơ đồ lắp ráp.....	10
2.6.1. Sơ đồ nối dây.....	10
2.6.2. Sơ đồ kết nối chân .....	11
2.7. Công nghệ lập trình cho vi điều khiển .....	11
2.8. Giao thức kết nối BLE (Bluetooth Low Energy).....	12

2.9. Xây dựng ứng dụng điều khiển.....	12
<b>CHƯƠNG 3. CÀI ĐẶT THỰC NGHIỆM VÀ KẾT QUẢ.....</b>	<b>13</b>
3.1. Kết quả lắp ráp phần cứng .....	13
3.2. Kết quả xây dựng ứng dụng điều khiển xe .....	13
3.3. Nhận xét kết quả hoạt động thực tế.....	14
3.3.1. Kết quả hoạt động .....	14
3.3.2. Ưu điểm.....	14
3.3.3. Nhược điểm.....	15
<b>CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>16</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>17</b>

## DANH MỤC TỪ VIẾT TẮT

Từ viết tắt (hoặc từ tiếng Anh)	Giải thích
BLE	Bluetooth Low Energy

## DANH MỤC HÌNH ẢNH

Hình 2.1: Sơ đồ chân của vi điều khiển ESP32 .....	6
Hình 2.2: Sơ đồ chân của L298N.....	7
Hình 2.3: Sơ đồ chân của HC-SR04 .....	8
Hình 2.4: Quy trình hoạt động và thuật toán của xe tự hành .....	9
Hình 2.5: Sơ đồ nối dây cho mô hình xe tự hành .....	10
Hình 3.1: Hình ảnh thực tế của xe.....	13
Hình 3.2: Ứng dụng điều khiển xe .....	14

## **DANH MỤC BẢNG BIỂU**

Bảng 1.1: Phân công công việc cho các thành viên..... 3

Bảng 2.1: Sơ đồ kết nối chân ..... 11

## MỞ ĐẦU

Xe tự hành đang dần trở thành một xu hướng mới, quan trọng và được ứng dụng rộng rãi trong công nghệ hiện đại, nhất là trong lĩnh vực ô tô và tự động hóa nhằm hỗ trợ tối đa các công việc con người cần phải làm. Những ứng dụng tiêu biểu của xe tự hành có thể kể đến như: robot lau nhà, các loại xe tự lái như ô tô cá nhân, taxi tự lái, xe giao hàng tự động,... Tuy nhiên, việc lắp ráp và lập trình để xe tự hành có khả năng vận hành một cách chính xác, ổn định, đảm bảo an toàn cũng như sự tiện lợi, thoải mái cho người sử dụng còn gặp rất nhiều khó khăn, thách thức cần phải khắc phục, đặc biệt với các loại xe tự lái khi gặp tình huống nguy hiểm khi tham gia giao thông. Từ đó, có thể thấy việc cần tối ưu thuật toán điều khiển về độ chính xác và độ an, song song với việc tối ưu chi phí sản xuất và vận hành toàn là vô cùng quan trọng, nên được quan tâm và nghiên cứu nhiều hơn.

Xuất phát từ nhu cầu đó, nhóm báo cáo thực hiện xây dựng một mô hình xe tự hành có khả năng bám đường, tự động phát hiện và né vật cản và có thể điều khiển xe theo ý muốn của người sử dụng. Trong báo cáo này, nhóm đề xuất một mô hình xe tự hành dựa trên cảm biến siêu âm HC-SR04 để phát hiện vật cản, sử dụng vi điều khiển ESP32 để vận hành, xây dựng app thông qua kết nối Bluetooth để có thể điều khiển xe theo ý thích. Bài báo cáo gồm những nội dung sau:

**Chương 1:** Trình bày tổng quan về bài toán xây dựng xe tự hành, hướng tiếp cận của nhóm và phân công công việc cho từng thành viên.

**Chương 2:** Trình bày cơ sở lý thuyết và cách thực hành lắp ráp cũng như thuật toán, công nghệ sử dụng cho mô hình xe tự hành.

**Chương 3:** Trình bày kết quả lắp ráp thực nghiệm.

**Chương 4:** Kết luận và đưa ra hướng phát triển trong tương lai

# CHƯƠNG 1. BÀI TOÁN PHÁT TRIỂN XE TỰ HÀNH TRÁNH CHƯỚNG NGẠI VẬT

## 1.1. Mô tả bài toán

Xe tự hành là phương tiện có khả năng nhận biết môi trường xung quanh, lập kế hoạch hành vi/ quỹ đạo, và điều khiển thực thi (vô lăng, ga, phanh...) để di chuyển mà không cần sự can thiệp trực tiếp của người lái [1]. Trong báo cáo này, nhóm thực hiện xây dựng mô hình xe tự hành bao gồm các chức năng sau:

- + Tự động tránh vật cản: Sử dụng cảm biến để quét và nhận diện liên tục các vật cản theo hướng đi, đánh giá khoảng trống phía trước và độ rộng so với thân xe và biên an toàn. Khi gặp phải vật cản, xe sẽ tự động đánh lái sang hướng có khoảng trống gần nhất, tránh vật cản sau đó đổi về hướng cũ và tiếp tục chạy thẳng.
- + Điều khiển từ xa thông qua Bluetooth: Tạo ứng dụng điều khiển ô tô, sử dụng các nút điều hướng để có thể đi thẳng, lùi, rẽ trái, rẽ phải, dừng lại hoặc chọn chế độ tự động lái. Nếu trong quá trình điều khiển mà gặp vật cản, xe sẽ tự dừng và phát cảnh báo về ứng dụng

## 1.2. Định hướng thực hiện

Để xe hoạt động, nhóm báo cáo chuẩn bị thân xe và 4 bánh xe. Gắn 4 động cơ DC 9V vào 4 bánh xe, giúp bánh xe có thể quay và xe có thể di chuyển. Để điều khiển hướng cho bánh xe, nhóm sử dụng L298N, đây là mạch điều khiển 2 kênh, điều khiển tốc độ và hướng quay của 4 động cơ DC gắn với bánh xe.

Để xe có thể tự hành, nhóm sử dụng ESP32 làm vi điều khiển trung tâm. ESP32 sẽ lấy dữ liệu từ cảm biến, xử lý dữ liệu và yêu cầu các bộ phận khác hoạt động. Ngoài ra, ESP32 còn giúp nhận lệnh điều khiển từ người dùng thông qua Bluetooth. Cảm biến được sử dụng là cảm biến siêu âm HC-SR04, giúp đo khoảng cách tới vật cản, nhằm giúp xe phát hiện vật cản. Để xe có thể lựa chọn hướng di chuyển để né vật cản, sử dụng một Servo G90, giúp cảm biến có thể quay sang 2 bên nhằm đo đạc và nhận diện vùng không có vật cản để di chuyển.



Để cả hệ thống hoạt động, cần cấp nguồn. Nhóm sử dụng 2 pin lithium 18650 mắc nối tiếp để cung cấp điện áp đầu vào cho mạch bởi vì loại pin này dễ tìm kiếm, có thể sạc lại và cung cấp dòng ổn định.

Để xe có thể hoạt động, nhóm lựa chọn MicroPython để xử lý dữ liệu nhận vào từ cảm biến và vi xử lý thông qua mô hình điện toán biên (xử lý thông tin tại chỗ nhằm giảm thiểu độ trễ khi xử lý dữ liệu, giúp xe có thể phản ứng nhanh hơn, tránh va chạm tốt hơn với những vật cản bất ngờ). Bên cạnh đó, MicroPython còn hỗ trợ giao thức BLE để giao tiếp giữa ứng dụng điều khiển và xe để thực hiện tính năng điều khiển xe

### 1.3. Phân công công việc

Dưới đây là bảng phân công công việc theo các nội dung cần làm của nhóm báo cáo:

STT	Tên thành viên	Công việc đảm nhận
1	Vũ Trọng Khôi	Tìm hiểu, thiết kế và triển khai phần cứng Xây dựng chức năng kết nối BLE
2	Phạm Ngọc Long	Xây dựng chế độ tự động tránh vật cản cho xe Thuyết trình và demo xe tự hành
3	Đặng Quốc Khánh	Xây dựng logic chế độ điều khiển thủ công cho xe Viết báo cáo và làm slide cho bài thuyết trình
4	Đỗ Hải Nam	Xây dựng ứng dụng điều khiển xe

**Bảng 1.1: Phân công công việc cho các thành viên**

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT ĐỂ PHÁT TRIỂN XE TỰ HÀNH TRÁNH CHƯỚNG NGẠI VẬT

### 2.1. Thiết kế hệ thống

#### 2.1.1. Lớp thiết bị

Lớp này bao gồm các thiết bị phần cứng vật lý như vi điều khiển, các cảm biến, động cơ, mạch điều khiển. Trong báo cáo này, xe tự hành sử dụng các thiết bị phần cứng sau:

- + Bộ điều khiển trung tâm là vi điều khiển ESP32
- + Cảm biến là cảm biến siêu âm HC – SR04: dùng để đo khoảng cách giữa xe và các vật cản xung quanh
- + Động cơ, mạch điều khiển:
  - Servo G90: dùng để quay cảm biến
  - Các động cơ DC 9V: giúp bánh xe quay và di chuyển
  - L298N: mạch điều khiển động cơ giúp điều khiển tốc độ và hướng quay của các động cơ DC 9V

#### 2.1.2. Lớp trao đổi thông tin

Lớp này có trách nhiệm kết nối, truyền và nhận dữ liệu giữa vi điều khiển với các động cơ, cảm biến khác và các thiết bị điều khiển bên ngoài. Có 2 loại giao tiếp trao đổi thông tin:

- + Giao tiếp giữa các chân cảm biến, vi xử lý và động cơ
- + Giao tiếp với ứng dụng thông qua giao thức BLE

#### 2.1.3. Lớp ứng dụng

Lớp này chịu trách nhiệm cho các tính năng và logic của xe, bao gồm:

- + Tự động tránh vật cản
  - Sử dụng thuật toán để tránh vật cản và điều hướng nhờ MicroPython
- + Điều khiển từ xa

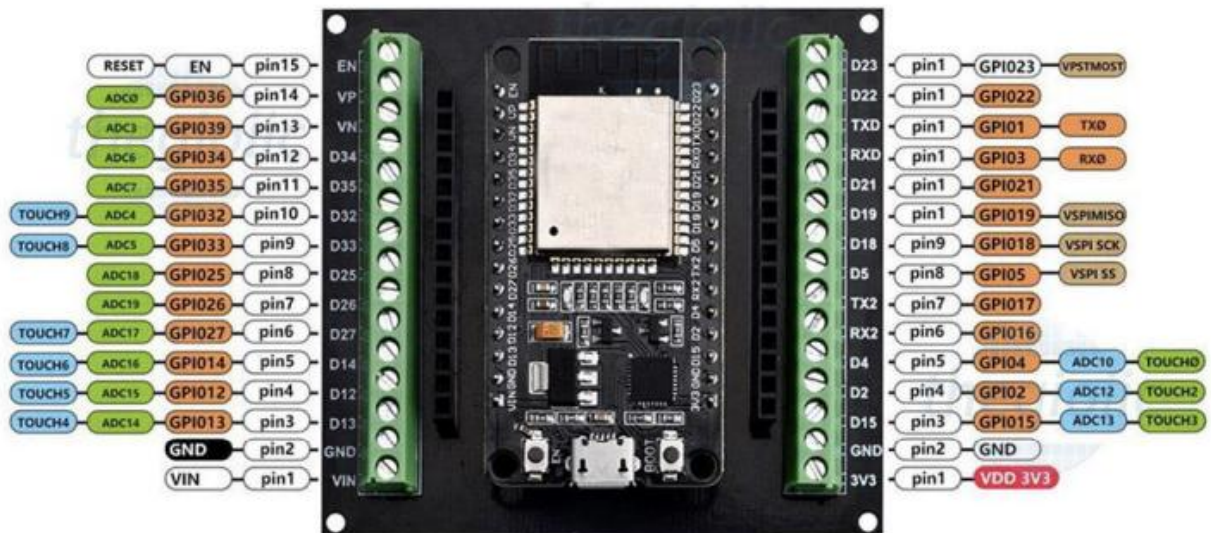
- Nhận lệnh điều hướng từ ứng dụng điều khiển thông qua giao thức BLE
- Xe tự động dừng, không thể tiến lên dù có sự điều khiển khi phía trước có vật cản
- Có thể chuyển đổi giữa 2 chế độ tự động và thủ công
- + Ứng dụng điều khiển
  - Xây dựng ứng dụng có giao diện điều hướng và chuyển chế độ đơn giản thông qua Flutter

## 2.2. Vi điều khiển ESP32

Là một dòng vi điều khiển có khả năng kết nối Wi-Fi và Bluetooth tích hợp, được sản xuất bởi Espressif Systems. Với vi xử lý có tốc độ cao và bộ nhớ lớn, ESP32 có thể xử lý các tác vụ phức tạp, được thiết kế để tiêu thụ ít năng lượng, điều này làm cho nó phù hợp cho các ứng dụng cho xe điều khiển tự hành hay di động.

### Thông số kỹ thuật [2]:

- + Bộ vi xử lý: Dual-core Tensilica Xtensa LX6, tốc độ lên đến 240 MHz.
- + Bộ nhớ:
  - SRAM: 520 KB
  - Flash ngoài: Thường từ 4 MB trở lên
- + Kết nối không dây:
  - Wi-Fi: 2.4 GHz, chuẩn 802.11 b/g/n
  - Bluetooth: BLE 4.2 hoặc 5.0 (tùy phiên bản) 6 April 18, 2025
- + Giao tiếp ngoại vi: GPIO, ADC, DAC, I2C, I2S, SPI, UART, PWM Ethernet, CAN bus (một số model)
- + Tiêu thụ năng lượng thấp: Có các chế độ Deep Sleep, Light Sleep để tiết kiệm năng lượng.



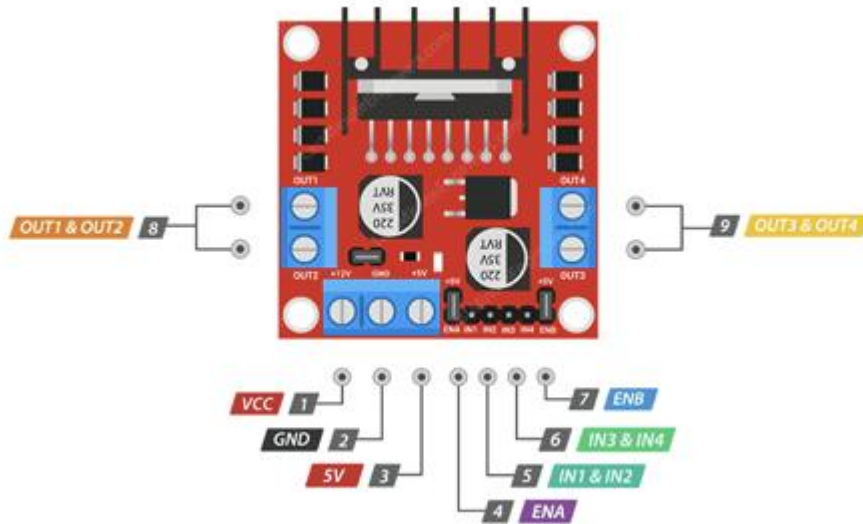
Hình 2.1: Sơ đồ chân của vi điều khiển ESP32

### 2.3. Mạch điều khiển động cơ L298N

L298N một cầu H kép, có thể điều khiển hai động cơ DC độc lập hoặc một động cơ DC hai chiều (quay thuận và quay ngược) một cách độc lập, có khả năng chịu dòng lớn, cho phép điều khiển động cơ DC có công suất cao. Nó có thể được điều khiển bằng tín hiệu điện từ các vi điều khiển như Arduino, Raspberry Pi hoặc bất kỳ vi điều khiển nào khác có thể tạo ra tín hiệu điện. L298N thường đi kèm với bảo vệ nhiệt độ và quá dòng, giúp bảo vệ mạch khi hoạt động ở điều kiện tải cao. Có thể được sử dụng trong nhiều ứng dụng khác nhau, từ robot di động đến hệ thống tự động hóa và điều khiển động cơ

### Thông số kỹ thuật [3]:

- + Điện áp cấp cho động cơ (VS): Từ 5V đến 46V.
- + Điện áp logic (VSS): 5V.
- + Dòng điện tối đa: 2A cho mỗi kênh.
- + Công suất tiêu thụ: 25W.
- + Bảo vệ quá nhiệt và chống ngắn mạch.



Hình 2.2: Sơ đồ chân của L298N

## 2.4. Cảm biến siêu âm HC-SR04

HC-SR04 là một cảm biến siêu âm phổ biến, được sử dụng để đo khoảng cách đến vật thể bằng cách phát và nhận sóng siêu âm. Nó hoạt động dựa trên nguyên lý phát xung siêu âm từ đầu phát (Trigger), sau đó nhận sóng phản xạ từ vật thể qua đầu thu (Echo). Bằng cách đo thời gian giữa phát và nhận sóng, cảm biến tính toán được khoảng cách đến vật thể.

### Thông số kỹ thuật [4]:

- + Điện áp hoạt động: 5V DC
- + Dòng tiêu thụ: < 2mA
- + Tần số siêu âm: 40kHz
- + Khoảng cách đo: 2 cm – 450 cm
- + Độ chính xác:  $\pm 3$  mm
- + Góc phát sóng: 15 độ về 2 bên
- + Tín hiệu đầu ra: mức cao: 5V, mức thấp: 0V
- + Thời gian phản hồi: < 100 ms
- + Nhiệt độ hoạt động: -15°C đến +70°C



Hình 2.3: Sơ đồ chân của HC-SR04

## 2.5. Nguyên lý hoạt động

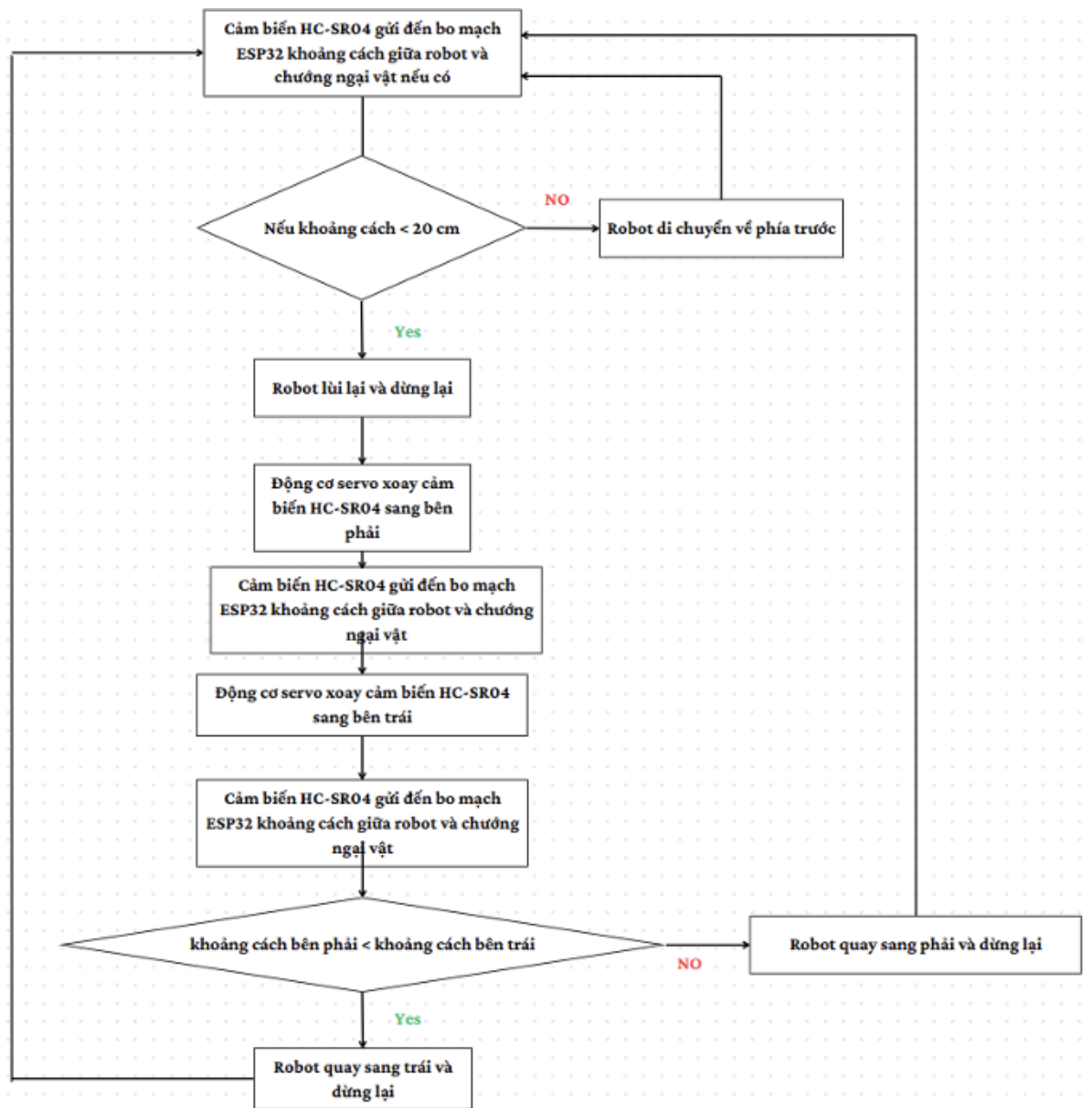
### 2.5.1. Các thành phần chính của hệ thống

- + ESP32: Là bộ điều khiển trung tâm, xử lý dữ liệu từ cảm biến siêu âm và nhận lệnh điều khiển từ người dùng qua kết nối Bluetooth.
- + L298N: Mạch điều khiển 2 kênh, điều khiển tốc độ và hướng quay của 4 động cơ DC gắn với bánh xe.
- + HC-SR04: Cảm biến siêu âm để đo khoảng cách phía trước xe, giúp phát hiện vật cản.
- + Servo SG90: Dùng để quay cảm biến HC-SR04 sang trái/phải, mở rộng vùng quét môi trường phía trước.
- + 4 động cơ DC 9V: Gắn vào 4 bánh xe, giúp xe có khả năng di chuyển tiến, lùi, rẽ trái/phải linh hoạt.
- + Nguồn cấp: Dùng pin 12V hoặc bộ 4 pin AA kết hợp với mạch hạ áp 9V và 5V để cấp nguồn ổn định cho toàn hệ thống (Sử dụng 2 pin lithium 18650 nối tiếp, cung cấp điện áp đầu vào cho mạch).

### 2.5.2. Quy trình hoạt động

- + ESP32 gửi tín hiệu đến servo để quay cảm biến HC-SR04 theo các hướng khác nhau (trái, phải, giữa).

- + Tại mỗi vị trí, ESP32 kích hoạt HC-SR04 phát xung siêu âm và đo thời gian phản hồi để tính khoảng cách đến vật cản.
- + Dựa vào dữ liệu khoảng cách thu được, ESP32 quyết định hướng di chuyển của xe:
  - Nếu không có vật cản phía trước, xe tiếp tục di chuyển thẳng.
  - Nếu phát hiện vật cản, ESP32 so sánh khoảng cách bên trái và bên phải để chọn hướng tránh vật cản.
- + ESP32 điều khiển L298N để thay đổi hướng và tốc độ của các động cơ DC, giúp xe di chuyển theo hướng đã chọn.



Hình 2.4: Quy trình hoạt động và thuật toán của xe tự hành



### 2.5.3. Giao tiếp giữa các module

ESP32 giao tiếp với các module khác như sau:

+ **Với L298N:**

- ESP32 sử dụng các chân GPIO để gửi tín hiệu điều khiển đến các chân IN1, IN2, IN3, IN4 của L298N, điều khiển hướng quay của động cơ.
- Tín hiệu PWM từ ESP32 được gửi đến các chân ENA và ENB của L298N để điều chỉnh tốc độ quay của động cơ.

+ **Với HC-SR04:**

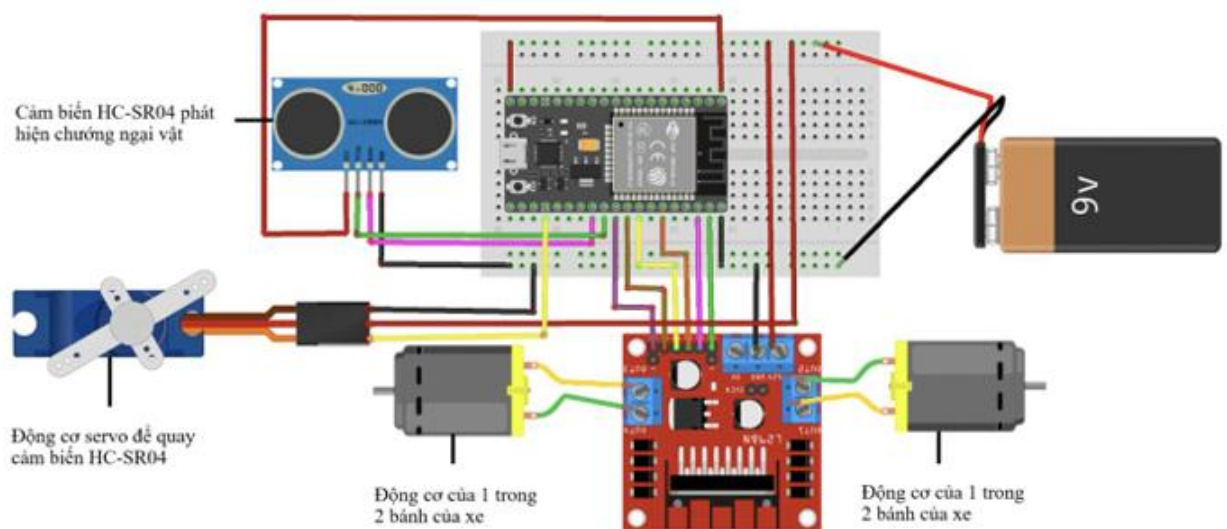
- ESP32 gửi xung ngắn (khoảng  $10\mu s$ ) đến chân TRIG của HC-SR04 để kích hoạt phát xung siêu âm.
- Sau đó, ESP32 đo thời gian xung phản hồi tại chân ECHO để tính khoảng cách đến vật cản.

+ **Với Servo SG90:**

- ESP32 gửi tín hiệu PWM đến chân điều khiển của servo để điều chỉnh góc quay, thay đổi hướng quét của cảm biến HC-SR04.

## 2.6. Sơ đồ lắp ráp

### 2.6.1. Sơ đồ nối dây



Hình 2.5: Sơ đồ nối dây cho mô hình xe tự hành



### 2.6.2. Sơ đồ kết nối chân

ESP32	L298N	Servo SG90	HC-SR04	Động cơ DC	Nguồn
D23	ENA				
D22	IN1				
D21	IN2				
D19	IN3				
D18	IN4				
D17			TRIG		
D16			ECHO		
D15		Orange			
D5	ENB				
GND	GND	Brown	GND		-
VIN (5V)	12V	Red			+
3V3			VCC		

**Bảng 2.1: Sơ đồ kết nối chân**

## 2.7. Công nghệ lập trình cho vi điều khiển

MicroPython là phiên bản nhẹ và được tối ưu hóa của ngôn ngữ lập trình Python 3 phổ biến., được thiết kế đặc biệt để chạy trên các thiết bị có tài nguyên rất hạn chế, chẳng hạn như vi điều khiển và hệ thống nhúng.

Với việc phát triển xe tự hành, nhóm sử dụng các thư viện như machine, PIN, PWM, time để điều khiển, thư viện bluetooth để kết nối Bluetooth...

Dù MicroPython mang lại sự thuận tiện về cú pháp và dễ đọc mã lệnh, tuy nhiên việc sử dụng Python trên ESP32 vẫn còn một số hạn chế như:

- + Thiếu thư viện hỗ trợ chuyên sâu cho điều khiển động cơ, cảm biến, servo so với Arduino C/C++.
  - + Quản lý thời gian thực và đồng bộ nhiều thiết bị trong môi trường
- MicroPython gặp nhiều thách thức, dễ gây xung đột hoặc treo chương trình.

- + Tài liệu và ví dụ thực tế bằng Python cho các dự án điều khiển xe còn hạn chế, khiến việc tra cứu và phát triển gặp khó khăn.

## 2.8. Giao thức kết nối BLE (Bluetooth Low Energy)

Bluetooth Low Energy (BLE) là một giao thức truyền thông không dây được thiết kế đặc biệt để tiêu thụ năng lượng cực thấp (Low Energy). Do đó, BLE được sử dụng nhiều trong các thiết bị IoT và di động, nơi cần truyền một lượng nhỏ dữ liệu không liên tục. Vì vậy, BLE phù hợp để làm cầu nối giữa ứng dụng và xe tự hành vì mô nó cần tốc độ phản hồi nhanh và tiết kiệm pin với nguồn pin lithium 18650.

- + Sử dụng thư viện bluetooth và lớp BLESimplePeripheral để biến ESP32 thành một thiết bị ngoại vi Bluetooth.
- + Cơ chế xử lý sự kiện (Event-driven): Hệ thống lắng nghe các sự kiện ngắt (IRQ) như `_IRQ_CENTRAL_CONNECT` (kết nối) hoặc `_IRQ_GATTS_WRITE` (nhận dữ liệu điều khiển từ điện thoại).

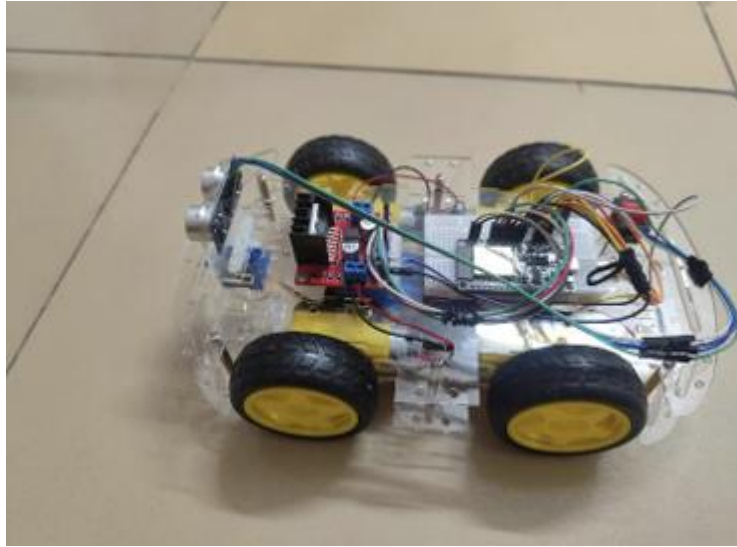
## 2.9. Xây dựng ứng dụng điều khiển

Flutter là một bộ công cụ giao diện người dùng, cho phép xây dựng các ứng dụng dành cho các thiết bị di động (Android, iOS, Web và máy tính để bàn) từ một codebase duy nhất. Mọi thứ trong Flutter đều là Widget (các khối xây dựng cơ bản của ứng dụng như Text, Button, Layout,...) và được tổ chức theo cấu trúc cây để người sử dụng dễ dàng theo dõi cấu hình và trạng thái hiện tại của nó. Flutter phù hợp để xây dựng ứng dụng điều khiển cho xe tự hành là vì:

- + Khả năng phát triển đa nền tảng: chỉ cần viết code một lần và ứng dụng có thể chạy trên cả hệ điều hành Android và iOS
- + Giao diện đẹp, linh hoạt, cấu trúc trực quan giúp dễ dàng cho việc xây dựng giao diện
- + Hỗ trợ tốt cho giao thức BLE mà nhóm báo cáo đang sử dụng
- + Tốc độ cao, hiệu suất tốt, giúp giảm độ trễ khi gửi lệnh điều khiển xe

## CHƯƠNG 3. CÀI ĐẶT THỰC NGHIỆM VÀ KẾT QUẢ

### 3.1. Kết quả lắp ráp phần cứng



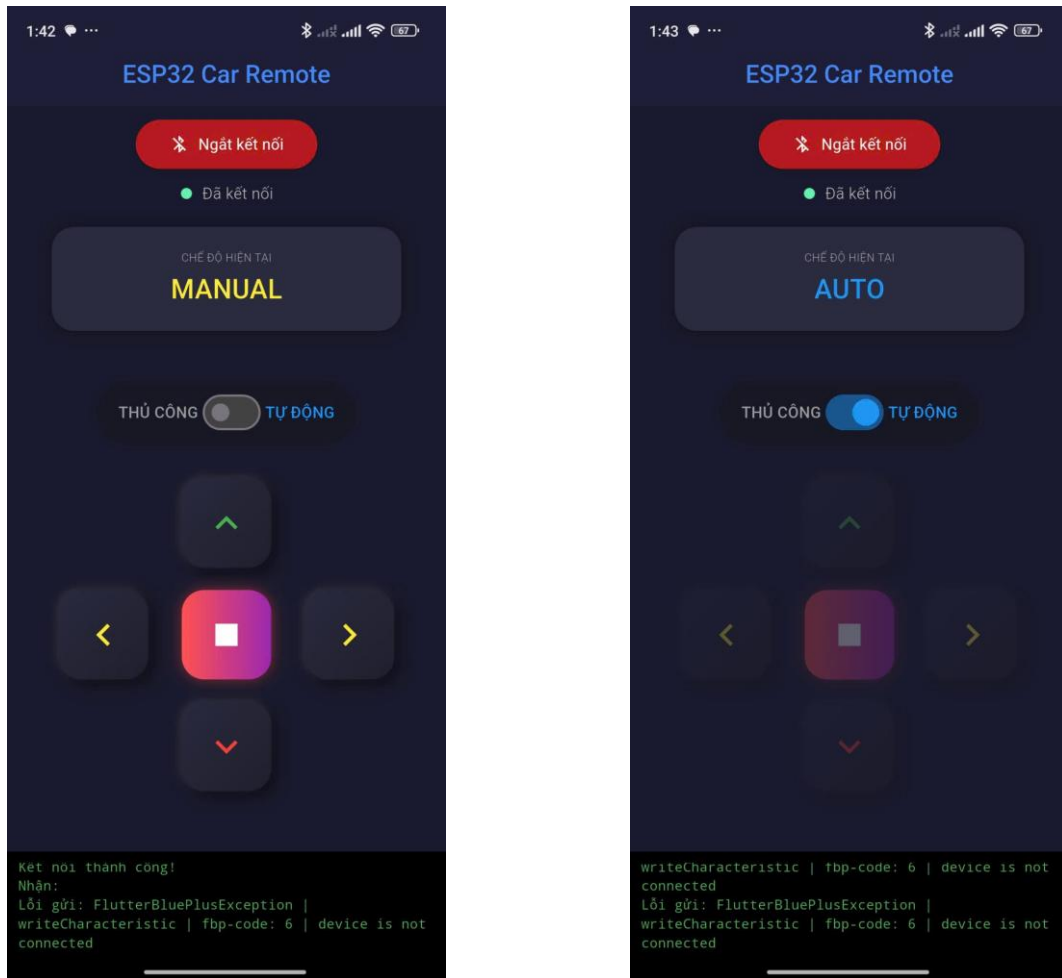
*Hình 3.1: Hình ảnh thực tế của xe*

### 3.2. Kết quả xây dựng ứng dụng điều khiển xe

Hướng dẫn sử dụng ứng dụng điều khiển xe:

- + Chế độ tự động:
  - Kết nối xe với Bluetooth, ở nút chọn chế độ, nhấn để nút hiện sang phía chữ “TỰ ĐỘNG”
  - Màn hình hiển thị “Chế độ hiện tại: AUTO” là thành công
- + Chế độ điều khiển:
  - Ở nút chọn chế độ, gạt sang hướng có chữ “THỦ CÔNG”
  - Màn hình hiện “Chế độ hiện tại: MANUAL” là thành công
  - Sử dụng các nút điều hướng để điều khiển tiến lên, lùi lại, sang trái, sang phải, nút ở chính giữa để xe dừng lại

Dưới đây là giao diện của ứng dụng điều khiển xe:



Hình 3.2: Ứng dụng điều khiển xe

### 3.3. Nhận xét kết quả hoạt động thực tế

#### 3.3.1. Kết quả hoạt động

- + Xe phản ứng nhanh với vật cản, có thể tránh các chướng ngại đơn giản hiệu quả.
- + Điều khiển thủ công bằng Bluetooth ổn định, tín hiệu truyền nhanh và chính xác.
- + Servo giúp mở rộng vùng quan sát giúp xe đưa ra quyết định hướng đi tốt hơn.
- + Hệ thống hoạt động tốt trong điều kiện ánh sáng bình thường và không có vật phản siêu âm yếu.
- + Xe có thể tự động dừng nếu cố tình điều khiển đâm vào chướng ngại vật

#### 3.3.2. Ưu điểm

- + Xe có thể hoạt động hoàn toàn tự động hoặc điều khiển từ xa bằng Bluetooth.
- + Thiết kế nhỏ gọn, chi phí thấp, dễ dàng thay đổi và nâng cấp.

- + Nguồn pin lithium 18650 tiện lợi, sạc lại được, cung cấp dòng ổn định.
- + Dễ dàng mở rộng chức năng như thêm cảm biến khác, kết nối Wi-Fi, điều khiển bằng giọng nói,...

### 3.3.3. Nhược điểm

- + Cảm biến siêu âm có phạm vi đo giới hạn, có thể gặp khó khăn khi phát hiện vật quá nhỏ, góc nghiêng, dễ bị ảnh hưởng bởi môi trường nhiều vật phản xạ hoặc hấp thụ sóng âm.
- + Servo quay còn chậm, ảnh hưởng tới tốc độ xử lý khi vật cần xuất hiện đột ngột.
- + Bluetooth chỉ hoạt động tốt ở khoảng cách ngắn trong bán kính 1m đến 2m, cần cải tiến nếu muốn điều khiển từ xa xa hơn.

## CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Dự án đã hoàn thành việc thiết kế và chế tạo mô hình xe tự hành có khả năng tránh vật cản một cách tự động và ổn định. Hệ thống sử dụng cảm biến siêu âm HC-SR04 kết hợp với servo quay để mở rộng vùng quét, giúp xe nhận diện và né tránh chướng ngại vật hiệu quả. Bộ điều khiển trung tâm ESP32 xử lý dữ liệu cảm biến và điều khiển động cơ thông qua mạch cầu H L298N. Mô hình hoạt động ổn định trong môi trường thử nghiệm, đáp ứng tốt các mục tiêu đặt ra về tự động hóa và khả năng di chuyển độc lập.

Để nâng cao hiệu quả và mở rộng chức năng của xe tự hành, nhóm đề xuất một số hướng phát triển sau:

- + Kết nối bằng Wi-Fi: Bổ sung khả năng giao tiếp qua Wi-Fi, cho phép điều khiển xe từ xa thông qua mạng nội bộ hoặc internet. Điều này giúp mở rộng phạm vi điều khiển và dễ dàng tích hợp với các nền tảng IoT hoặc ứng dụng web.
- + Mở rộng phạm vi kết nối Bluetooth: Tối ưu hóa kết nối Bluetooth bằng cách sử dụng các module hoặc anten hỗ trợ công suất cao hơn, nhằm đảm bảo tín hiệu ổn định và tăng khoảng cách điều khiển giữa thiết bị và xe.
- + Nâng cấp hệ thống cảm biến: Thay thế hoặc kết hợp thêm các cảm biến hiện đại hơn để:
  - Phát hiện vật cản ở khoảng cách xa hơn và chính xác hơn.
  - Nhận diện được cả những vật cản có kích thước nhỏ.
  - Tăng tốc độ quét môi trường xung quanh, giúp xe phản ứng nhanh hơn khi di chuyển ở tốc độ cao.
- + Ứng dụng trí tuệ nhân tạo (AI): Nghiên cứu áp dụng các thuật toán học máy để xe có khả năng tự học và thích nghi với môi trường vận hành, từ đó đưa ra quyết định điều hướng chính xác và tối ưu hơn trong thời gian thực. Những cải tiến này sẽ giúp xe tự hành trở nên thông minh hơn, hoạt động hiệu quả hơn và có thể ứng dụng trong nhiều lĩnh vực thực tế.

## TÀI LIỆU THAM KHẢO

- [1] Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixão, T., Mutz, F., Veronese, L., Oliveira-Santos, T. & De Souza, A. F. (2021). *Self-Driving Cars: A Survey. Expert Systems with Applications*, 165, 113816. doi:10.1016/j.eswa.2020.113816
- [2] <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>
- [3] [https://datasheet.lcsc.com/lcsc/2312251759\\_MSKSEMI-L298N-MS\\_C19632282.pdf](https://datasheet.lcsc.com/lcsc/2312251759_MSKSEMI-L298N-MS_C19632282.pdf)
- [4] <https://www.alldatasheet.net/datasheet-pdf/view/1132203/ETC2/HC-SR04.html>