HỌC VIỆN KỸ THUẬT MẬT MÃ

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CÔNG NGHỆ PHẦN MỀM NHÚNG

XE TỰ HÀNH ĐIỀU KHIỂN BẰNG BLUETOOTH

Giảng viên hướng dẫn: **Ths. Lê Đức Thuận**

Sinh viên thực hiện:

Trần Gia Lương	*CT030433*
Phạm Trà My	*CT030453*
Phan Hoàng Sơn	*CT030442*

Hà Nội, 2021

MỤC LỤC

LỜI NÓI ĐẦU 1
CHƯƠNG 1. GIỚI THIỆU TỔNG QUÁT VỀ HỆ THỐNG XE TRÁNH VẬT CẢN 2
1.1. Khái quát hệ thống 2
1.2. Thực tiến áp dụng Arduino, FreeRTOS vào đề tài 2
1.3. Khái niệm, nghiệp vụ hệ thống 3
1.3.1. Tổng quan 3
1.3.2. Thiết kế hệ thống 3
CHƯƠNG 2. GIỚI THIỆU THIẾT BỊ 5
2.1. Arduino UNO R4 5
2.1.1. Một vài thông số của Arduino UNO R3 5
2.1.2. Các chân năng lượng 6
2.1.3. Bộ nhớ 7
2.1.4. Các cổng vào ra 7
2.2. Motor Driver L298N 8
2.3. DC Motor 9
2.4. Module HC-05 9
2.5. Module HC-SR04 11
2.6. Thư viện Arduino 12
2.7. Giới thiệu về Arduino IDE và ngôn ngữ lập trình 13
CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG 15
3.1. Phân tích kiến trúc 15
3.2. Phân tích xe tự hành điều khiển bằng bluetooth 16
3.2.1. Biểu đồ ca sử dụng của hệ thống 16
3.2.2. Đặc ta các ca sử dụng 17
3.2.3. Phân tích các ca sử dụng 21
3.3. Phân tích phần mềm điều khiển 25
3.3.1. Biểu đô ca sử dụng của phần mềm điều khiển 25

3.3.2. Đặc tả ca sử dụng.....25
3.3.3. Phân tích các ca sử dụng.....28
CHƯƠNG 4. THIẾT KẾ ÁP DỤNG HỆ ĐIỀU HÀNH THỜI GIAN THỰC VÀO HỆ THỐNG.....31
4.1. Thiết kế phần mềm.....31
4.1.1. Tổng quan về thiết kế phần mềm.....31
4.1.2. Xây dựng các tác vụ trong hệ thống.....32
4.1.3. Phân tích các tác vụ.....33
4.2. Thiết kế phần cứng.....35
4.3. Thiết kế thiết bị điều khiển.....37
PHỤ LỤC.....38
CODE Arduino của xe tự hành điều khiển bằng bluetooth:.....38
TÀI LIỆU THAM KHẢO.....44

DANH MỤC HÌNH VẼ

Hình 2.1: Arduino UNO R3 5
Hình 2.2: Motor Driver L298N 9
Hình 2.3: DC Motor 9
Hình 2.4: Module HC-05 11
Hình 2.5: Module HC-SR04 12
Hình 2.6: Các thiết bị trên phần mềm Proteus 13
Hình 2.7: Màn hình Arduino IDE 14
Hình 3.1: Kiến trúc đơn giản của hệ thống 15
Hình 3.2: Biểu đồ ca sử dụng tông quan của hệ thống 16
Hình 3.3: Biểu đồ tuần tự ca sử dụng tiến lên 21
Hình 3.4: Biểu đồ tuần tự ca sử dụng lùi lại 22
Hình 3.5: Biểu đồ tuần tự ca sử dụng lùi lại 22
Hình 3.6: Biểu đồ tuần tự ca sử dụng sang trái 23
Hình 3.7: Biểu đồ tuần tự ca sử dụng dừng lại 23
Hình 3.8: Biểu đồ tuần tự ca sử dụng tăng giảm tốc độ 24
Hình 3.9; Biểu đồ tuần tự ca sử dụng tránh vật cản 24
Hình 3.10: Biểu đồ ca sử dụng của phần mềm điều khiển 25
Hình 3.11: Biểu đồ tuần tử ca sử dụng gửi tín hiệu tiến lên 28
Hình 3.12: Biểu đồ tuần tự ca sử dụng gửi tín hiệu lùi lại 29
Hình 3.13: Biểu đồ tuần tự ca sử dụng gửi tín hiệu sang phả 29
Hình 3.14: Biểu đồ tuần tự ca sử dụng gửi tín hiệu sang trái 30
Hình 3.15: Biểu đồ tuần tự ca sử dụng thay đổi tốc độ 30
Hình 4.1: Hệ điều hành FreeRTOS 32
Hình 4.2: Tổng quan các tác vụ trong hệ thống 32
Hình 4.3: Tác vụ tín hiệu từ bluetooth 33
Hình 4.4: Biểu đồ tuần tự tác vụ nhận tín hiệu vật cản 33

Hình 4.5: Biểu đồ tuần tự tác vụ điều khiển động cơ 34
Hình 4.6: Biểu đồ tuần tự chi tiết lệnh 34
Hình 4.7: Thiết kế mạch của hệ thống 35
Hình 4.8: Code mô phỏng cách cấu hình chân Serial 35
Hình 4.9: Cách nối module-hc05 36
Hình 4.10: Cách nối HC-SR04 36
Hình 4.11: Cách mắc module L298N 36
Hình 4.12: Cách mắc motor vào L298N 37
Hình 4.13: Thiết bị điều khiển 37

DANH MỤC BẢNG

Bảng 2.1: Thông số Arduino UNO R3 6
Bảng 2.2: Chi tiết L298N 8
Bảng 2.3: Chi tiết HC-05 10
Bảng 2.4: Chi tiết chân HC-SR04 12
Bảng 3.1: Đặc tả ca sử dụng tiến lên 17
Bảng 3.2: Đặc tả ca sử dụng lùi lại 18
Bảng 3.3: Đặc tả ca sử dụng dẽ sang phải 19
Bảng 3.4: Đặc tả ca sử dụng rẽ sang trái 19
Bảng 3.5: Ca sử dụng dừng lại 20
Bảng 3.6: Đặc tả ca sử dụng tăng giảm tốc độ 20
Bảng 3.7: Ca sử dụng tránh vật cản 21
Bảng 3.8: Ca sử dụng gửi tín hiệu tiến lên 26
Bảng 3.9: Ca sử dụng gửi tín hiệu lùi lại 26
Bảng 3.10: Ca sử dụng gửi tín hiệu sang phải 27
Bảng 3.11: Ca sử dụng gửi tín hiệu sang trái 27
Bảng 3.12: Ca sử dụng tăng giảm tốc độ 28

LỜI NÓI ĐẦU

Ngày nay khoa học công nghệ ngày càng phát triển, vi điều khiển AVR và vi điều khiển PIC ngày càng thông dụng và hoàn thiện hơn , nhưng có thể nói sự xuất hiện của Arduino vào năm 2005 tại Italia đã mở ra một hướng đi mới cho vi điều khiển. Sự xuất hiện của Arduino đã hỗ trợ cho con người rất nhiều trong lập trình và thiết kế, nhất là đối với những người bắt đầu tìm tòi về vi điều khiển mà không có quá nhiều kiến thức, hiểu biết sâu sắc về vật lý và điện tử . Phần cứng của thiết bị đã được tích hợp nhiều chức năng cơ bản và là mã nguồn mở, hệ thư viện rất phong phú và được chia sẻ miễn phí. Chính vì những lý do như vậy nên Arduino hiện đang dần phổ biến và được phát triển ngày càng mạnh mẽ trên toàn thế giới.

Trên cơ sở kiến thức đã học trong môn học : Tin học đại cương , Điện tử tương tự và số… cùng với những hiểu biết về các thiết bị điện tử, chúng em đã quyết định thực hiện đề tài: thiết kế hệ thống tưới cây tự động sử dụng hệ điều hành FreeRTOS với mục đích để tìm hiểu thêm về Arduino, làm quen với các thiết bị điện tử và nâng cao hiểu biết khi cho các bộ điều khiển chạy trên hệ điều hành FreeRTOS. Do kiến thức còn hạn hẹp nên chắc chắn không tránh khỏi những thiếu sót , hạn chế vì thế chúng em rất mong có được sự góp ý và nhắc nhờ từ thầy giáo để có thể hoàn thiện đề tài của mình

CHƯƠNG 1. GIỚI THIỆU TỔNG QUÁT VỀ HỆ THỐNG XE TRÁNH VẬT CẢN

1.1. Khái quát hệ thống

Trong thời đại công nghiệp hóa hiện đại hóa đất nước, việc chú trọng phát triển các ngành công nghiệp cũng như các ngành dịch vụ đang dần dần được coi trọng. Và như một nhu cầu thiết yếu, việc phát triển ra các hệ thống nhúng để đưa vào hỗ trợ trong công nghiệp, dịch vụ sẽ đem lại khoản lợi nhuận khổng lồ khi mà với cùng bằng một sức lao động người sử dụng các hệ thống nhúng tạo ra được nhiều giá trị sức lao động hơn.

Tài liệu này được tạo ra với mục đích phân tích, thiết kế ra một hệ thống nhúng có tên là *Xe tự hành điều khiển bằng bluetooth* . Xe này sẽ làm tiền đề để có thể phát triển thành các sản phẩm sử dụng trong công nghiệp, ví dụ như xe chuyển hàng giữa các kho bãi với nhau ở trong một công xưởng. Có thể thấy hiện nay, việc luân chuyển hàng hóa giữa các khu vực sản xuất trong một công xưởng vẫn cần một nhân công điều khiển xe, khi ta áp dụng công nghệ xe tự hành điều khiển bằng bluetooth thì có thể giảm được chi phí nhân công điều khiển xe, hơn nữa là một người có thể điều khiển nhiều xe làm nhiệm vụ, từ đó gia tăng lợi nhuận cho công ty sản xuất.

1.2. Thực tiến áp dụng Arduino, FreeRTOS vào đề tài

Ứng dụng của Arduino: Sử dụng 3 cảm biến khoảng cách HC-SR04 kết nối với Arduino và app điều khiển bằng bluetooth để tạo ra xe vận hành tránh vật cản. Cảm biến HC-SR04 sẽ đo khoảng cách tới những vật cản xung quanh, nếu khoảng cách đó đạt đến mức tối thiểu đã đặt ra thì Arduino sẽ báo lên màn hình cho người điều khiển. Mọi thao tác điều khiển đều dễ dàng thực hiện qua 1 thiết bị di động.

FreeRTOS là một hệ điều hành nhúng thời gian thực mã nguồn mở ra đời từ năm 2003, đến nay nó được phát triển rất mạnh mẽ và nhận được nhiều sự ủng hộ của các lập trình cho các hệ nhúng. FreeRTOS có tính khả chuyển, có thể sử dụng miễn phí hoặc dùng cho mục đích thương mại. Nó có nhiều ưu điểm nổi bật so với các hệ điều hành nhúng thời gian thực khác như có kích thước rất nhỏ gọn

nên rất phù hợp với các hệ nhúng thời gian thực nhỏ; được viết bằng ngôn ngữ C nên có độ phù hợp cao với các nền phần cứng khác nhau. Một điểm mạnh nữa trong hệ điều hành nhúng thời gian thực FreeRTOS là nó cho phép lập lịch đa tác vụ. Đa tác vụ trong hệ thống nhúng thời gian thực là khái niệm giống với đa tác vụ trong các hệ thống để bàn ở chỗ nó mô tả nhiều luồng thực thi bằng cách sử dụng cùng một bộ xử lý đơn. Giúp chúng ta quản lý các sự kiện ngắt của xe vận hành điều khiển.

1.3. Khái niệm, nghiệp vụ hệ thống

1.3.1. Tổng quan

Việc đặt ra bài toán, nhiệm vụ của hệ thống là rất cần thiết để đảm bảo thông tin về các vật tư thiết bị, bộ xử lý, …

- Thiết bị điều khiển là điện thoại di động có kết nối bluetooth
- Phạm vi áp dụng của hệ thống: Vận chuyển hàng hóa trong các khu công nghiệp
- Khoảng cách dừng lại khi gặp vật cản được thiết lập từ trước.

1.3.2. Thiết kế hệ thống

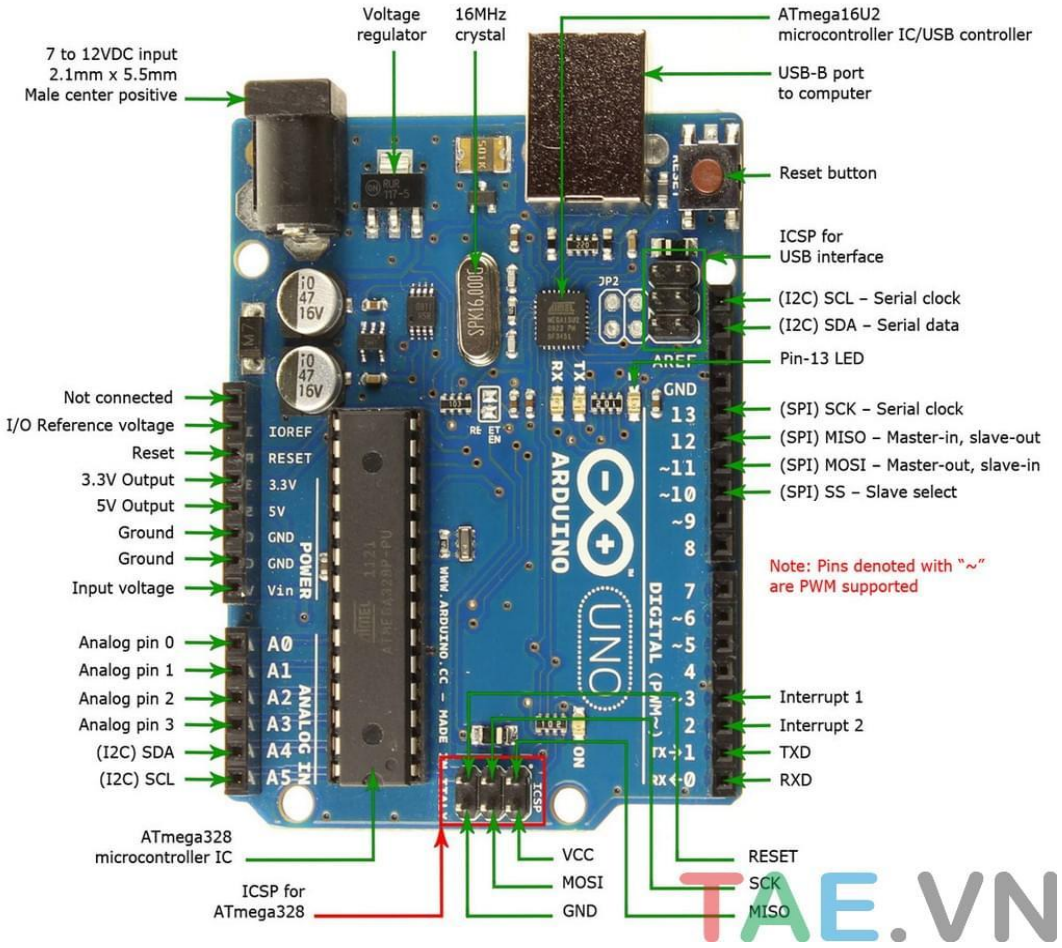
- Bộ điều khiển: Thiết bị điều khiển ở đây được chọn là điện thoại di động chạy hệ điều hành android, thiết bị sẽ được cài đặt một phầm mềm android có giao diện hỗ trợ các chức năng điều khiển xe. Sau đó thiết bị sẽ được kết nối và có thể điều kiển xe thông qua giao thức bluetooth.
- Xe vận hành: Được chia thành khối nhận lệnh, khối xử lý trung tâm và khối vận hành
- Khối nhận lệnh: nhận tín hiệu điều khiển được gửi từ Thiết bị điều khiển sang, có rất nhiều Module hỗ trợ nhận tín hiệu bluetooth, ở đây lựa chọn sử dụng Module HC-05. Module này sẽ nhận tín hiệu và truyền tín hiệu sang Arduino bằng giao thức UART.
- Khối xử lý trung tâm: đóng vai trò như là một bộ não của xe, nó sẽ nhận mọi tín hiện từ ngoài gửi vào và sẽ đi phân tích xử lý dữ liệu đấy. Kết quả của việc phân tích đó chính là đưa ra những quyết định điều kiển xe, ví dụ như là tiến lên hay lùi lại.

- Khối vận hành: bao gồm các động cơ và Module Motor Driver L298N. Module này sẽ nhận tín hiệu đầu vào từ arduino, và sẽ điều khiển tốc độ nhanh, chậm của Motor thông qua cách điều khiển cường dộ dòng điện chạy qua động cơ bằng các Trasistor được thiết kế sẵn bên trong Module và vì vậy, Module Motor Driver L298N có thể điều khiển dòng điện lên tới 39V.

CHƯƠNG 2. GIỚI THIỆU THIẾT BỊ

2.1. Arduino UNO R4

Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lí những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lí tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD, …



Hình 2.1: Arduino UNO R3

2.1.1. Một vài thông số của Arduino UNO R3

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz

Dòng tiêu thụ	khoảng 30mA
Điện áp vào khuyên dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	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 (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

Bảng 2.1: Thông số Arduino UNO R3

2.1.2. Các chân năng lượng

GND (Ground): cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.

- 5V: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
- 3.3V: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
- Vin (Voltage Input): để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
- IOREF: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
- RESET: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

2.1.3. Bộ nhớ

Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

- 32KB bộ nhớ Flash: những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader
- 2KB cho SRAM (Static Random Access Memory): giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
- 1KB cho EEPROM (Electrically Eraseble Programmable Read Only Memory): giống như một chiếc ổ cứng mini

2.1.4. Các cổng vào ra

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở không được kết nối).

Một số chân digital có các chức năng đặc biệt:

- 2 chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết
- Chân PWM (~): 3, 5, 6, 9, 10, và 11: cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 → 28-1 tương ứng với 0V → 5V) bằng hàm analogWrite(). Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
- Chân giao tiếp SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.

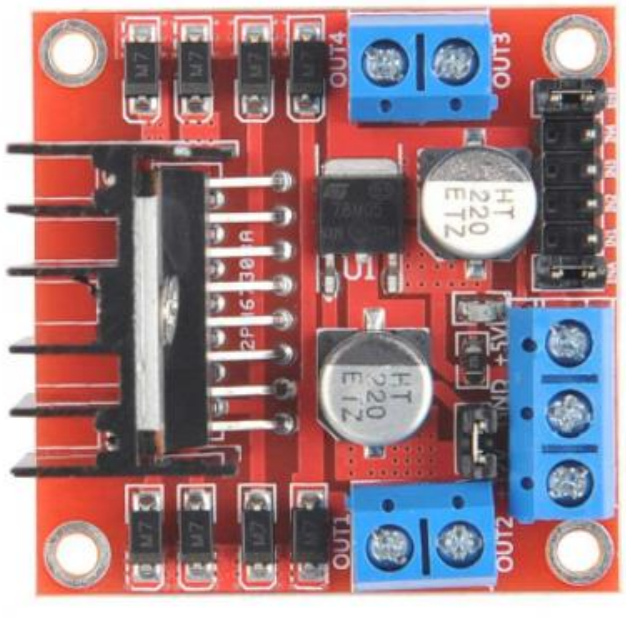
- LED 13: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.
- Arduino UNO có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 210-1) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân AREF trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V → 2.5V với độ phân giải vẫn là 10bit.
- Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

2.2. Motor Driver L298N

Motor Driver L298N là một module dùng để điều khiển động cơ dùng ở trong arduino, nó có thể điều khiển tối đa lên đến bốn động cơ riêng biệt, hoặc hai động cơ riêng biệt theo hai hướng quay khác nhau.

IN1 & IN2	Chân đầu vào cho motor 1
IN3 & IN4	Chân đầu vào cho motor 2
ENA	Cho phép các chân liên quan đến motor 1 hoạt động
ENB	Cho phép các chân liên quan đến motor 2 hoạt động
OUT1 & OUT2	Chân đầu ra của motor 1
OUT3 & OUT4	Chân đầu ra của motor 2
12V	Chân nguồn 12V
5V	Chân nguồn hỗ trợ mạch bên trong L298N
GND	Chân nối đất

Bảng 2.2: Chi tiết L298N



Hình 2.2: Motor Driver L298N

2.3. DC Motor

DC Motor là động cơ chính giúp xe chuyển động. DC Motor thường có hai đầu dẫn, một cho cực dương, một cho cực âm. Để giúp Motor quay, chúng ta nối hai dầu dẫn vào PIN. Nếu đổi hai đầu dẫn với nhau, Motor xẽ quay theo chiều ngược lại.

Hình 2.3: DC Motor

2.4. Module HC-05

Module HC-05 là một module bluetooth sử dụng giao thức Serial Port, nó được thiết kế để thiết lập kết nối nối tiếp không dây. Module này dựa trên IC

Bluetooth chip đơn BC417, tuân thủ tiêu chuẩn Bluetooth v2.0 và hỗ trợ cho cả giao diện UART và USB.

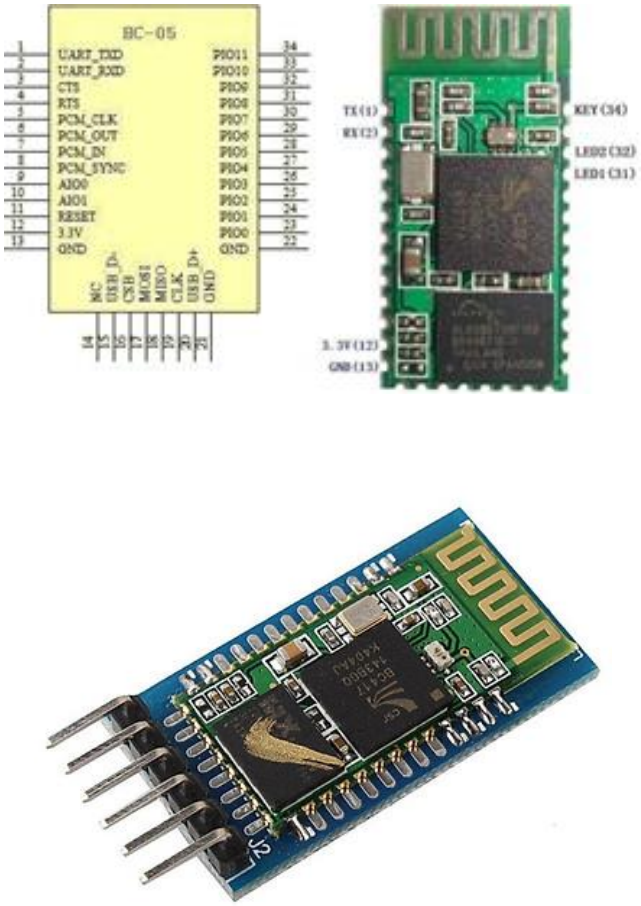
Đặc điểm kỹ thuật:

- Chuẩn Bluetooth: V2.0+EDR.
- Điện áp hoạt động: 3.3-4VDC, 30mA
- Kích thước: 28mm x 15mm x 2.35mm
- Tần số: 2.4G
- Tốc độ: 2.1Mbs(Max) / 160kbps
- Tốc độ baud mặc định: 9600, 8bit dữ liệu, 1bit Stop. Hỗ trợ tốc độ baud: 9600, 19200, 38400, 57600, 115200, 230400, 460800.
- Nhiệt độ làm việc: -20 ~ 75^{o}C
- Độ nhạy: -80dBm
- Module có hai chế độ làm việc

Chi tiết các chân của Module HC-05:

Chân	Chức năng
EN	Cho phép Module hoạt động khi chân này được bỏ trống hoặc được kết nối với chân 3.3V
+5V	Chân cấp nguồn
GND	Chân nối đất
TX	Chân Output của giao tiếp UART
RX	Chân Input của giao tiếp UART
STATE	Chân báo trạng thái. Pin ở mức thấp khi Module không được kết nối với thiết bị nào, ngược lại chân ở mức cao khi có thiết bị kết nối

Bảng 2.3: Chi tiết HC-05



Hình 2.4: Module HC-05

2.5. Module HC-SR04

Cảm biến khoảng cách HC-SR04 là cảm biến dùng để xác định khoảng cách trong phạm vi nhỏ bằng cách phát sóng siêu âm. Cảm biến có độ chính xác khá cao và độ ổn định trong quá trình sử dụng, đồng thời dễ dàng kết nối với các bảng mạch như là Arduino.

Chi tiết chân của Module HC-SR04:

Chân	Chi tiết
Vcc	Chân nguồn 5V
Trig	Chân nhận tín hiệu phát sóng
Echo	Chân xuất tín hiệu xung HIGH, chiều rộng của xung bằng với thời gian sóng siêu âm được phát từ cảm biến

	và quay về

Bảng 2.4: Chi tiết chân HC-SR04

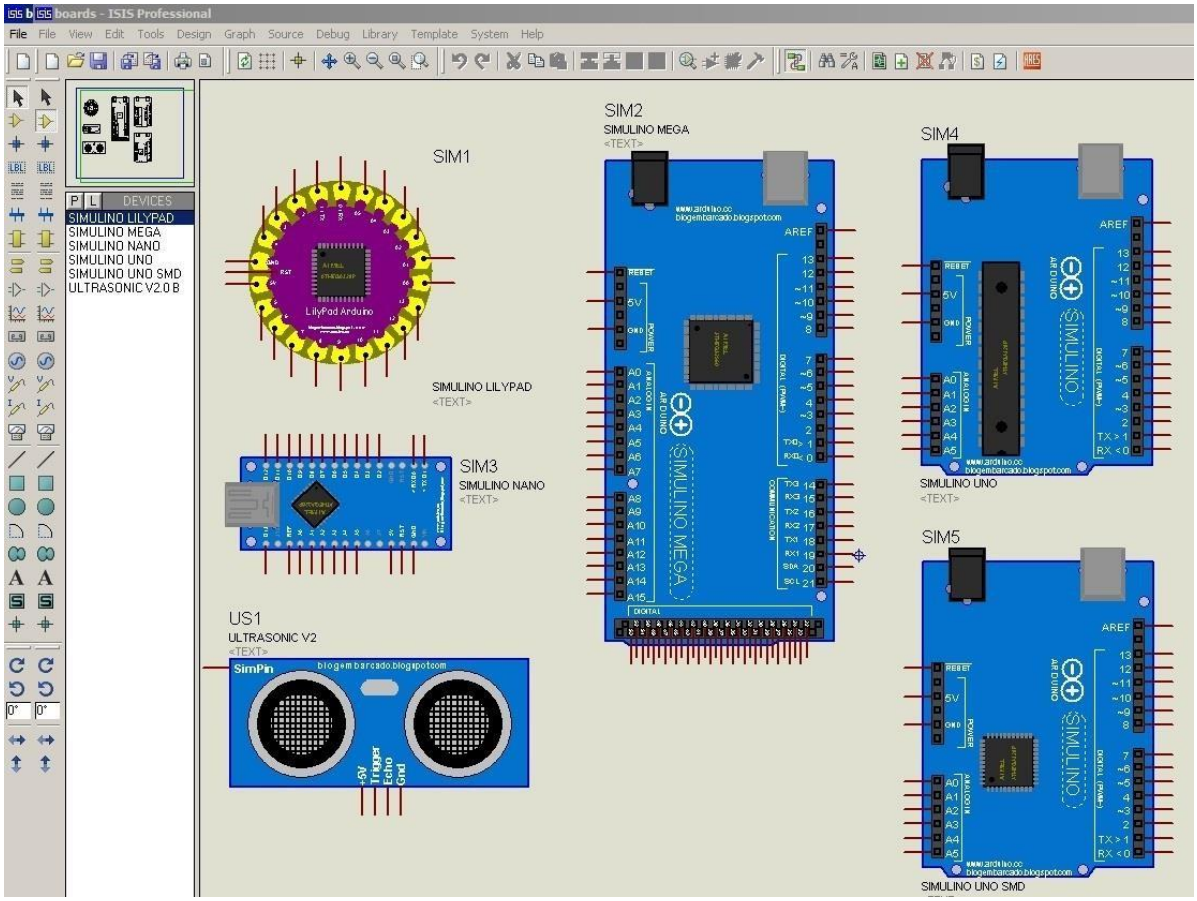
Hình 2.5: Module HC-SR04

2.6. Thư viện Arduino

Thư viện Arduino là một bổ sung rất hay cho phần mềm mô phỏng Proteus nó giúp cho việc mô phỏng Arduino được thuận tiện và dễ dàng hơn thay vì chỉ mô phỏng được chip ATmega328(nhân của Arduino), thư viện này được phát triển bởi các kĩ sư Cesar Osaka, Daniel Cezar, Roberto Bauer và được đăng tải trên blog tiếng Bồ Đào Nha.

Thư viện bao gồm các linh kiện sau:

- Arduino Uno (Phiên bản chip ATmega328 chân DIP)
- Arduino Uno (Phiên bản chip ATmega328 chân SMD)
- Arduino Mega
- Arduino Lilypad
- Arduino Nano
- Cảm biến siêu âm Ultrasonic V2



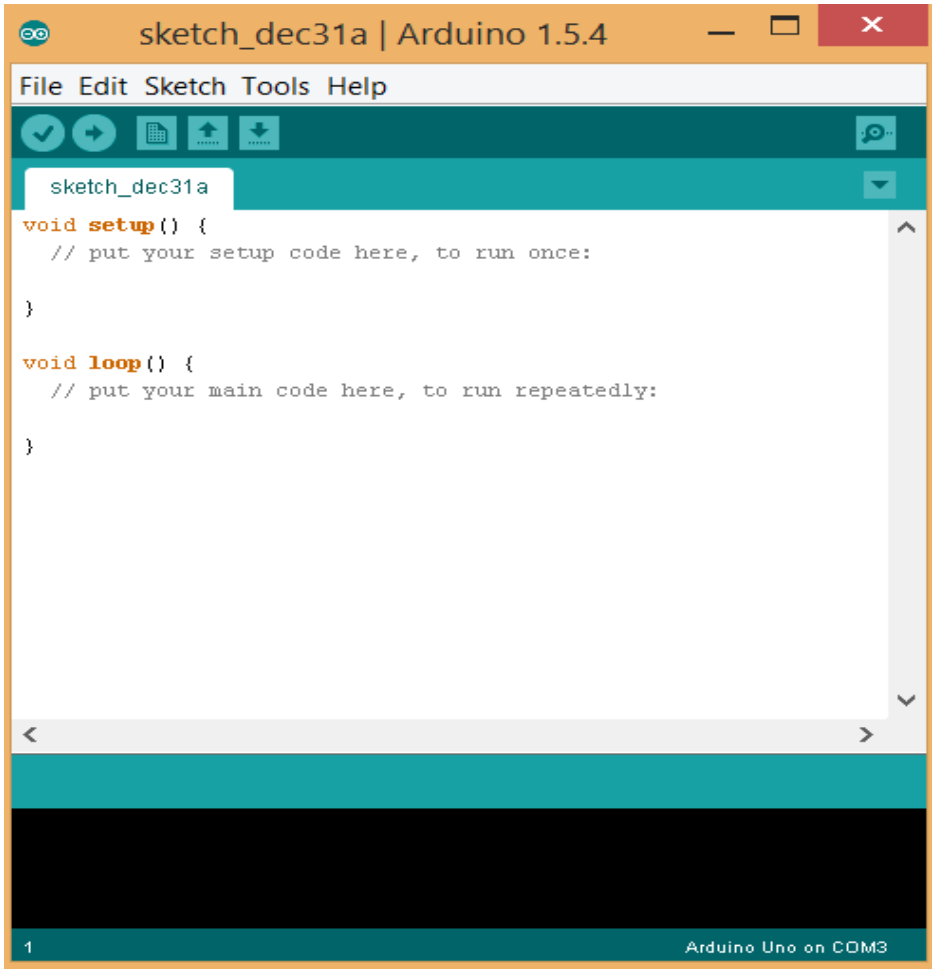
Hình 2.6: Các thiết bị trên phần mềm Proteus

2.7. Giới thiệu về Arduino IDE và ngôn ngữ lập trình

Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho Arduino, tuy nhiên sức mạnh thực sự của Arduino nằm ở phần mềm. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng

C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.

Arduino IDE là phần mềm dùng để lập trình cho Arduino. Môi trường lập trình Arduino IDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh OSX và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.



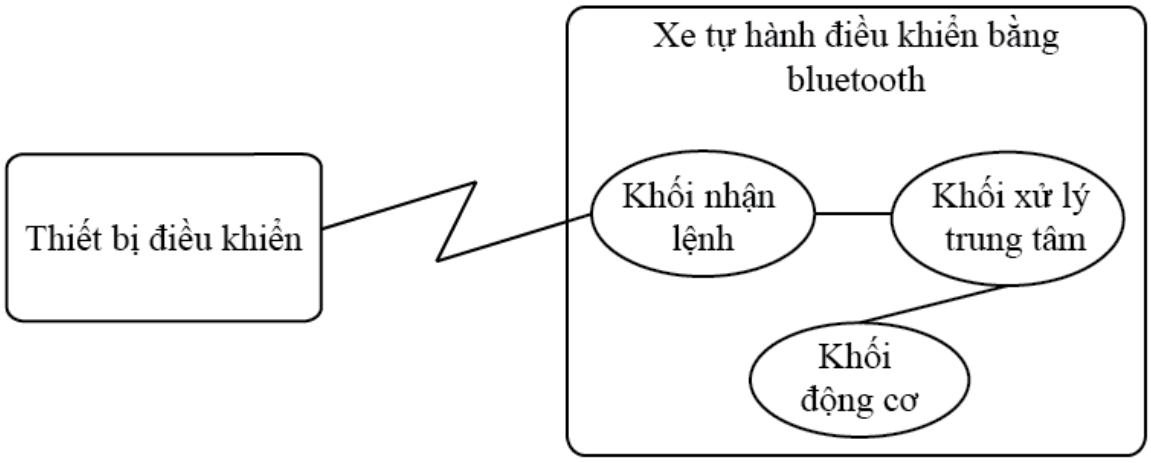
Hình 2.7: Màn hình Arduino IDE

Ngôn ngữ lập trình có thể được mở rộng thông qua các thư viện C++. Và do ngôn ngữ lập trình này dựa trên nền tảng ngôn ngữ C của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR vào chương trình nếu muốn.

CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG

3.1. Phân tích kiến trúc

Kiến trúc đơn giản của *Xe tự hành điều khiển bằng bluetooth* được chia làm hai phần. Phần thứ nhất là thiết bị điều khiển, ở đây có thể dùng điện thoại di động có kết nối bluetooth làm thiết bị điều khiển. Phần thứ hai là xe được điều khiển, xe được điều khiển sẽ được chia tiếp thành 3 khối lần lượt là khối nhận lệnh, khối điều khiển và khối thực thi.



Hình 3.1: Kiến trúc đơn giản của hệ thống

Thiết bị điều khiển ở đây được chọn là điện thoại di động chạy hệ điều hành android, thiết bị sẽ được cài đặt một phầm mềm android có giao diện hỗ trợ các chức năng điều khiển xe. Sau đó thiết bị sẽ được kết nối và có thể điều kiển xe thông qua giao thức bluetooth.

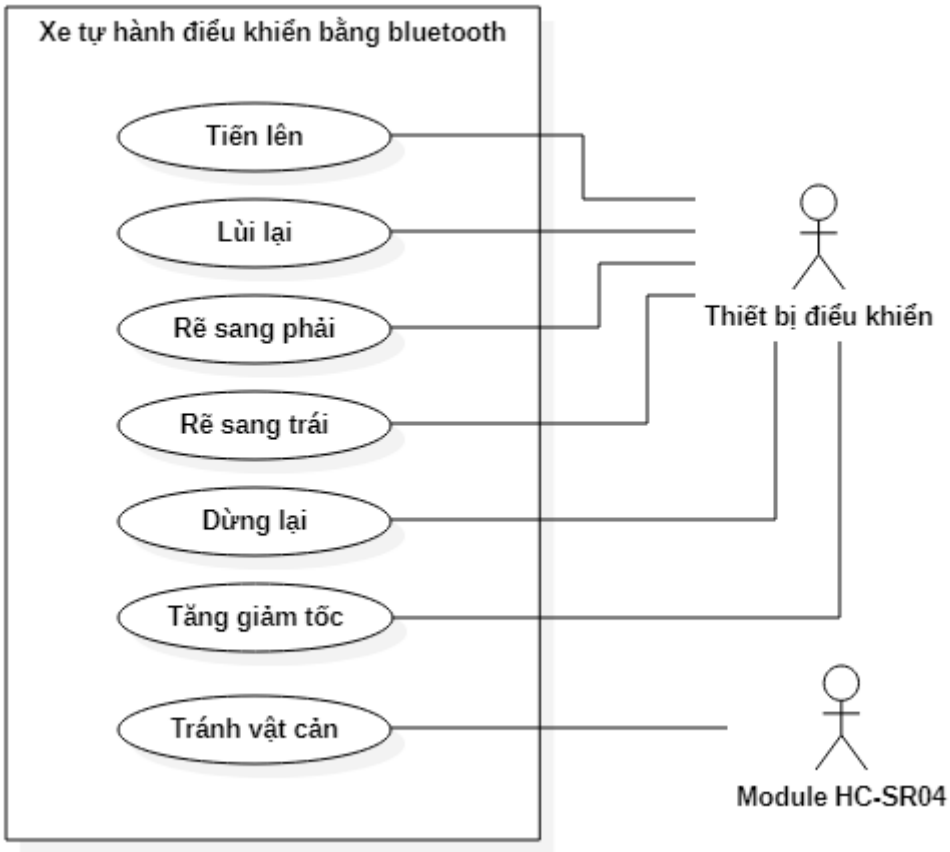
Ở đầu bên kia là xe tự hành điều khiển bằng bluetooth. Xe sẽ có một khối chuyên nhận tín hiệu điều khiển được gửi từ Thiết bị điều khiển sang, có rất nhiều Module hỗ trợ nhận tín hiệu bluetooth và nhóm đã lựa chọn sử dụng Module HC-05. Module này sẽ nhận tín hiệu và truyền tín hiệu sang Arduino bằng giao thức UART.

Khối điều khiển của xe tự hành điều khiển bằng bluetooth đóng vai trò như là một bộ não của xe, nó sẽ nhận mọi tín hiện từ ngoài ngửi vào và sẽ đi phân tích xử lý dữ liệu đấy. Kết quả của việc phân tích đó chính là đưa ra những quyết định điều kiển xe, ví dụ như là tiến lên hay lùi lại.

Cuối cùng ở đây là khối vận hành, khối vận hành bao gồm các động cơ và Module Motor Driver L298N. Do động cơ ở đây sử dụng hiệu điện thế và cường độ dòng điện lớn hơn nhiều so với khối điều khiển, vì vậy cần phải dùng một thiết bị điều khiển trung gian đó là Module Motor Driver L298N. Module này sẽ nhận tín hiệu đầu vào từ arduino, và sẽ điều khiển tốc độ nhanh, chậm của Motor thông qua cách điều khiển cường dộ dòng điện chạy qua động cơ bằng các Trasistor được thiết kế sẵn bên trong Module và vì vậy, Module Motor Driver L298N có thể điều khiển dòng điện lên tới 39V.

3.2. Phân tích xe tự hành điều khiển bằng bluetooth

3.2.1. Biểu đồ ca sử dụng của hệ thống



Hình 3.2: Biểu đồ ca sử dụng tông quan của hệ thống

Về mặt tổng quan, hệ thống xe tự hành điều khiển bằng bluetooth có bao gồm hai Actor đó chính là Thiết bị điều khiển và Module HC-SR04, kèm theo đó hệ thống có tổng cộng bảy ca sử dụng đó chính là Tiến lên, Lùi lại, Rẽ sang phải,

Rẽ sang trái, Tăng giảm tốc độ, Dừng lại và tránh vật cản, đó là bảy chứng năng cơ bản để tạo thành một hệ thống xe tự hành điều khiển bằng bluethooth.

Đối với Actor là Thiết bị điều khiển, nó được phép thực hiện sáu chức năng ứng với bốn ca sử dụng đó là Tiến lên, Lùi lại, Rẽ sang phải, Rẽ sang trái, Dừng lại, Tăng giảm tốc độ. Với sáu ca sử dụng này, Thiết bị điều khiển có thể tùy ý điều khiển cách thức hoạt động của xe để đạt được mục đích ví dụ như là cho xe di chuyển từ A sang B.

Đối với Actor là Module HC-SR04, nó được phép thực hiện một chức năng ứng với ca sử dụng Tránh vật cản, ca sử dụng này được tạo ra với mục đích là để tránh tai nạn không đáng có xảy ra khi xe gặp chướng ngại vật, HC-SR04 sẽ truyền thông tin khoảng cách của xe so với HC-SR04 về bộ xử lý trung tâm, từ đó bộ xử lý trung tâm sẽ đưa ra quyết định dừng xe.

3.2.2. Đặc ta các ca sử dụng

3.2.2.1. Ca sử dụng tiến lên

Use Case	Tiến lên
Actor	Thiết bị điều khiển
Brief Description	Điều khiển cho xe tiến lên
Pre-condition	Thiết bị đã được kết nối Bluetooth
Basic Flows	1. Người điều khiển gửi tín hiệu tiến lên ở trên thiết bị điều khiển. 2. Module HC-05 nhận thông tin điều khiển tiến lên và chuyển tín hiệu điều khiển cho arduino. 3. Arduino nhận tín hiệu điều khiển tiến và xuất tín hiệu khởi động động cơ hai bên ra các chân cần thiết, và tốc độ của 2 bên động cơ bằng nhau. 4. Động cơ được khởi động và xe bắt đầu tiến lên.
Alternative Flows	

Bảng 3.1: Đặc tả ca sử dụng tiến lên

3.2.2.2. Ca sử dụng lùi lại

Use Case	Lùi lại
Actor	Thiết bị điều khiển
Brief Description	Điều khiển cho xe lùi lại
Pre-condition	Thiết bị đã được kết nối Bluetooth
Basic Flows	1. Người điều khiển ra tín hiệu lùi lại ở trên thiết bị điều khiển. 2. Module HC-05 nhận thông tin điều lùi lại phải và chuyển tín hiệu điều khiển cho arduino. 3. Arduino nhận tín hiệu điều khiển lùi lại và xuất tín hiệu khởi động động cơ hai bên lùi lại. 4. Động cơ được khởi động và xe đi lùi lại.
Alternative Flows	

Bảng 3.2: Đặc tả ca sử dụng lùi lại

3.2.2.3. Ca sử dụng rẽ sang phải

Use Case	Rẽ sang phải
Actor	Thiết bị điều khiển
Brief Description	Điều khiển cho xe rẽ sang phải
Pre-condition	Thiết bị đã được kết nối Bluetooth
Basic Flows	1. Người điều khiển ra tín hiệu rẽ sang phải ở trên thiết bị điều khiển. 2. Module HC-05 nhận thông tin điều khiển rẽ sang phải và chuyển tín hiệu điều khiển cho arduino. 3. Arduino nhận tín hiệu khoảng cách các vật thể bên phải 4. Nếu không có vật thể bên phải.Arduino nhận tín hiệu điều khiển rẽ sang phải và xuất tín hiệu

	khởi động động cơ bên phải chạy chậm hơn động cơ bên trái. 5. Động cơ được khởi động.
Alternative Flows	4.1. Nếu khoảng cách vật thể bên phải nhỏ hơn 25cm thì Arduino sẽ dừng động cơ.

Bảng 3.3: Đặc tả ca sử dụng dẽ sang phải

3.2.2.4. Ca sử dụng rẽ sang trái

Use Case	Rẽ sang trái
Actor	Thiết bị điều khiển
Brief Description	Điều khiển cho xe rẽ sang trái
Pre-condition	Thiết bị đã được kết nối Bluetooth
Basic Flows	1. Người điều khiển ra tín hiệu rẽ sang trái ở trên thiết bị điều khiển. 2. Module HC-05 nhận thông tin điều khiển rẽ sang trái và chuyển tín hiệu điều khiển cho arduino. 3. Arduino nhận tín hiệu khoảng cách các vật thể bên trái 4. Nếu không có vật thể bên trái. Arduino nhận tín hiệu điều khiển rẽ sang trái và xuất tín hiệu khởi động động cơ bên trái chạy chậm hơn động cơ bên phải. 5. Động cơ được khởi động.
Alternative Flows	4.1. Nếu khoảng cách vật thể bên trái nhỏ hơn 25cm thì Arduino sẽ dừng động cơ.

Bảng 3.4: Đặc tả ca sử dụng rẽ sang trái

3.2.2.5. Ca sử dụng dừng lại

Use Case	Dừng lại

Actor	Thiết bị điều khiển
Brief Description	Điều khiển cho xe rẽ sang trái
Pre-condition	Thiết bị đã được kết nối Bluetooth
Basic Flows	1. Người điều khiển ra tín hiệu dừng ở trên thiết bị điều khiển. 2. Module HC-05 nhận thông tin điều khiển dừng lại và chuyển tín hiệu điều khiển cho arduino. 3. Động cơ dừng lại.
Alternative Flows	

Bảng 3.5: Ca sử dụng dừng lại

3.2.2.6. Ca sử dụng tăng giảm tốc độ

Use Case	Tăng giảm tốc độ
Actor	Thiết bị điều khiển
Brief Description	Điều khiển cho xe tăng giảm tốc độ
Pre-condition	Thiết bị đã được kết nối Bluetooth
Basic Flows	1. Người điều khiển ra tín hiệu tăng giảm tốc độ ở trên thiết bị điều khiển. 2. Module HC-05 nhận thông tin điều khiển tăng giảm tốc độ và chuyển tín hiệu điều khiển cho arduino. 3. Arduino gửi tín hiệu tốc độ đến cho L298N để điều khiển tốc độ Motor.
Alternative Flows	

Bảng 3.6: Đặc tả ca sử dụng tăng giảm tốc độ

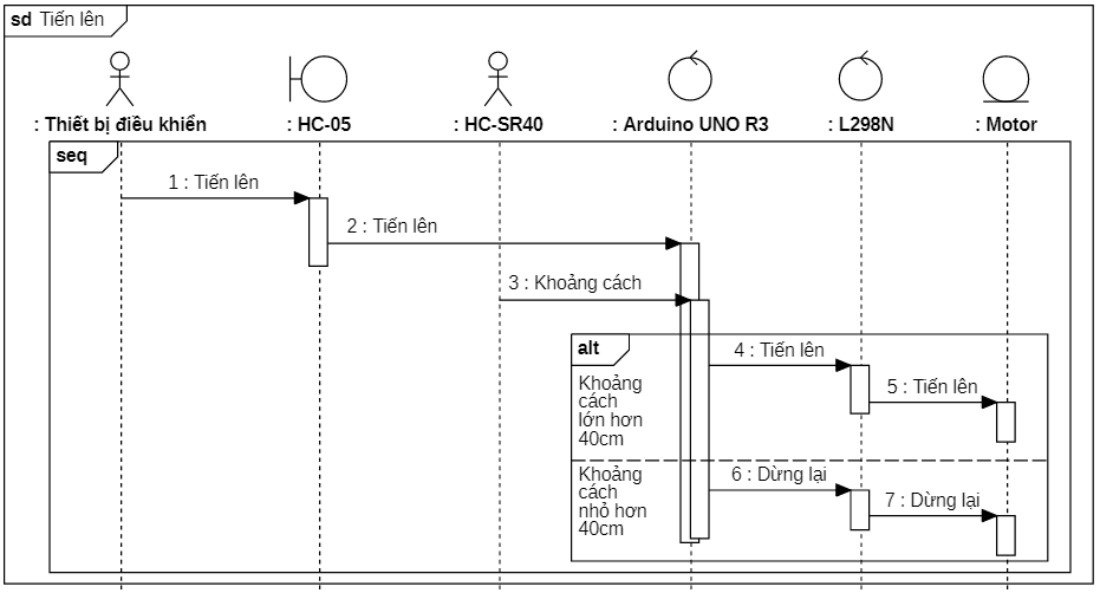
3.2.2.7. Ca sử dụng tránh vật cản

Use Case	Tránh vật cản
Actor	Module HC-SR04
Brief Description	Cảm biến đo khoản cách để tránh vật cản
Pre-condition	
Basic Flows	1. Cảm biến đo khoảng cách từ xe cho tới vật cản ở phía trước. 2. Cảm biến gửi thông tin khoảng cách vật cản cho arduino. 3. Nếu khoảng cách giữa xe và vật cản nhỏ hơn 5cm, xe sẽ không thể tiến lên phía trước, Thiết bị điều khiển cần điều khiển xe tránh sang hướng khác
Alternative Flows	3.1. Nếu khoảng cách giữa xe và vật cản lơn hơn 40cm, xe có thể tiến lên tiếp.

Bảng 3.7: Ca sử dụng tránh vật cản

3.2.3. Phân tích các ca sử dụng

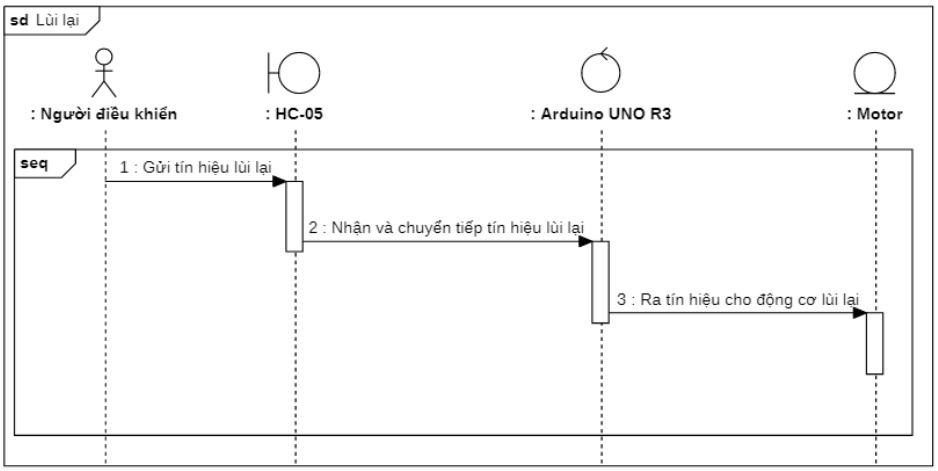
3.2.3.1. Ca sử dụng tiến lên



Hình 3.3: Biểu đồ tuần tự ca sử dụng tiến lên

Thiết bị điều khiển ra tín hiệu tiến lên ở trên thiết bị điều khiển, Module HC-05 nhận thông tin điều khiển tiến lên và chuyển tín hiệu điều khiển cho arduino.Arduino nhận tín hiệu điều khiển tiến và xuất tín hiệu khởi động động cơ hai bên ra các chân cần thiết, và tốc độ của 2 bên động cơ bằng nhau.Động cơ được khởi động và xe bắt đầu tiến lên

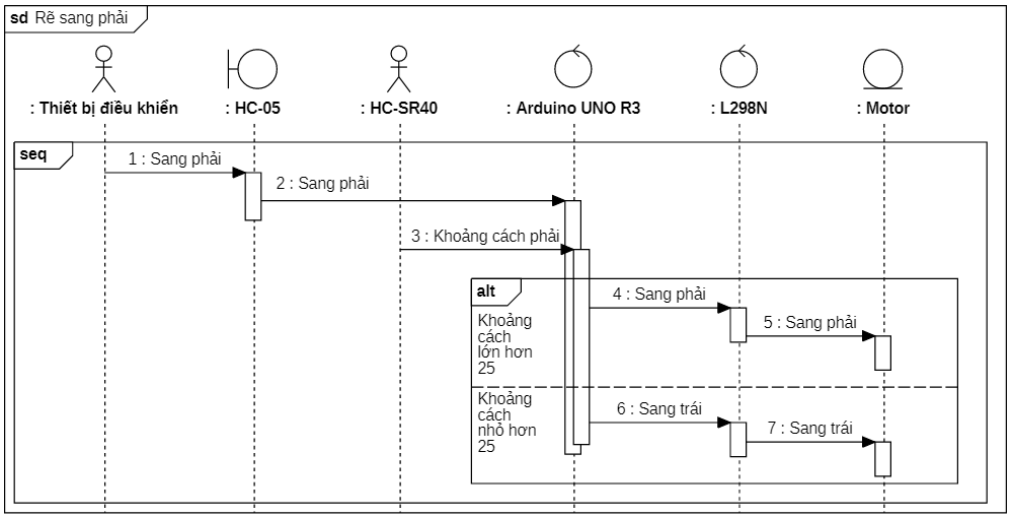
3.2.3.2. Ca sử dụng lùi lại



Hình 3.4: Biểu đồ tuần tự ca sử dụng lùi lại

Thiết bị điều khiển ra tín hiệu lùi lại ở trên thiết bị điều khiển.Module HC-05 nhận thông tin điều lùi lại phải và chuyển tín hiệu điều khiển cho arduino.Arduino nhận tín hiệu điều khiển lùi lại và xuất tín hiệu khởi động động cơ hai bên lùi lại.Động cơ được khởi động và xe đi lùi lại.

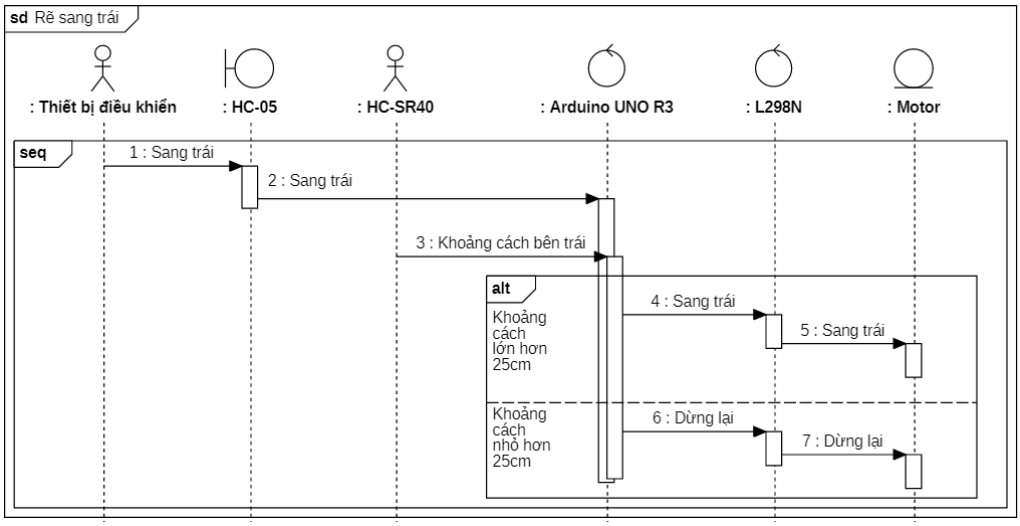
3.2.3.3. Ca sử dụng sang phải



Hình 3.5: Biểu đồ tuần tự ca sử dụng lùi lại

Thiết bị điều khiển ra tín hiệu rẽ sang phải ở trên thiết bị điều khiển.Module HC-05 nhận thông tin điều khiển rẽ sang phải và chuyển tín hiệu điều khiển cho Arduino. Arduino nhận tín hiệu điều khiển rẽ sang phải và tín hiệu vật cản từ HC-SR04 bên phải, nếu bên phải không có vật cản thì Arduino xuất tín hiệu khởi động động cơ bên phải chạy chậm hơn động cơ bên trái. Động cơ được khởi động

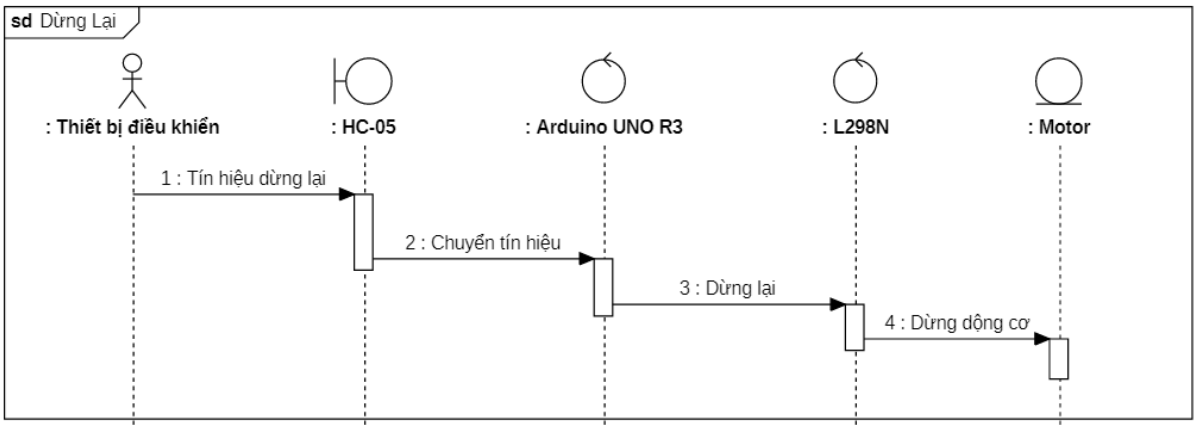
3.2.3.4. Ca sử dụng sang trái



Hình 3.6: Biểu đồ tuần tự ca sử dụng sang trái

Thiết bị điều khiển ra tín hiệu rẽ sang trái ở trên thiết bị điều khiển. Module HC-05 nhận thông tin điều khiển rẽ sang trái và chuyển tín hiệu điều khiển cho arduino. Arduino nhận tín hiệu điều khiển rẽ sang trái và xuất tín hiệu khởi động động cơ bên trái chạy chậm hơn động cơ bên phải. Động cơ được khởi động.

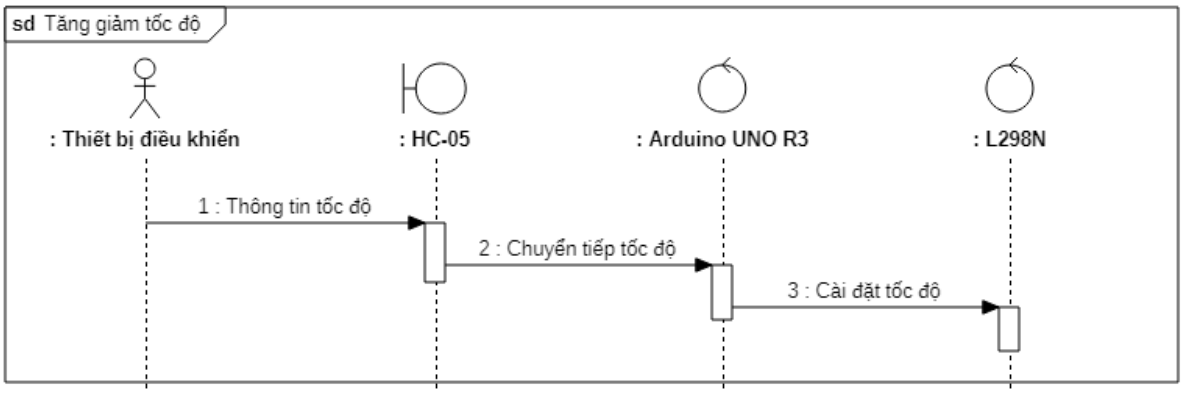
3.2.3.5. Ca sử dụng dừng lại



Hình 3.7: Biểu đồ tuần tự ca sử dụng dừng lại

Khi thiết bị người dùng gửi tín hiệu dừng lại thông qua bluetooth, HC-05 sẽ nhận được tín hiệu và gửi tiếp tín hiệu đến Arduino UNO R3, Arduino sẽ nhận tín hiệu, kèm theo là gửi tín hiệu điều khiển ra các chân để dừng động cơ lại.

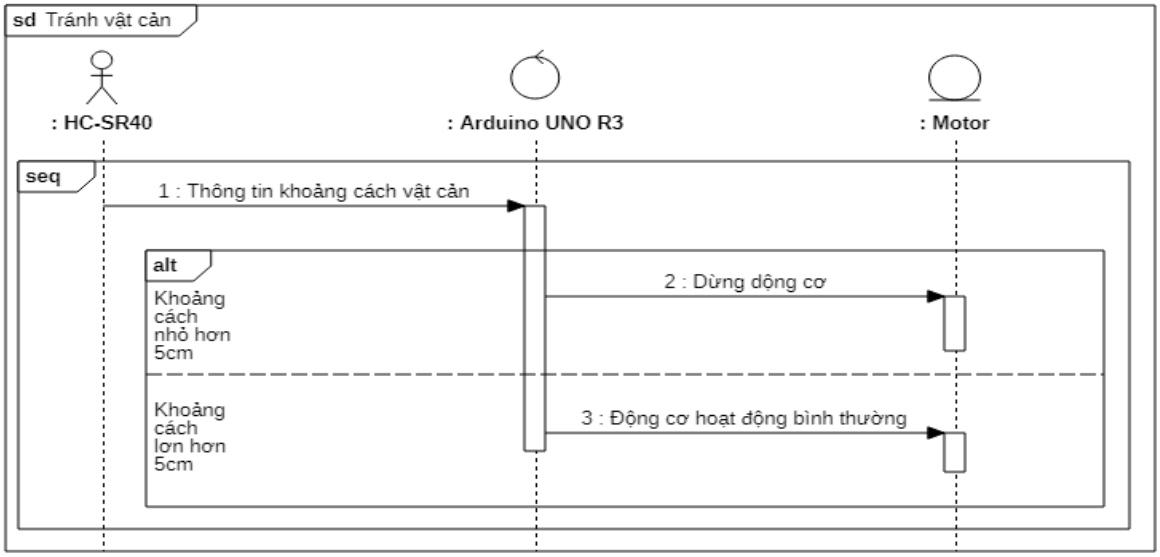
3.2.3.6. Ca sử dụng tăng giảm tốc độ



Hình 3.8: Biểu đồ tuần tự ca sử dụng tăng giảm tốc độ

Người điều khiển kéo chọn tốc độ ở trên màn hình điều khiển, Thiết bị sẽ bắt sự kiện và gửi tín hiệu thông qua bluetooth, Arduino nhận và chỉnh lại tốc độ xe.

3.2.3.7. Ca sử dụng tránh vật cản



Hình 3.9; Biểu đồ tuần tự ca sử dụng tránh vật cản

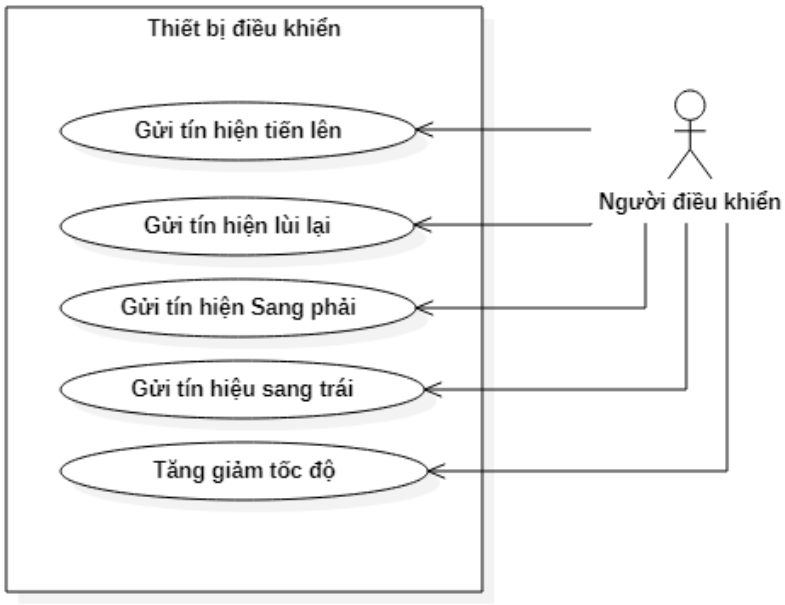
Cảm biến đo khoảng cách từ xe cho tới vật cản ở phía trước. Cảm biến gửi thông tin khoảng cách vật cản cho arduino. Nếu khoảng cách giữa xe và vật cản

nhỏ hơn 5cm, xe sẽ không thể tiến lên phía trước, thiết bị điều khiển cần điều khiển xe tránh sang hướng khác

3.3. Phân tích phần mềm điều khiển

3.3.1. Biểu đô ca sử dụng của phần mềm điều khiển

Phầm mềm điều khiển được thiết kế để chạy trên nền tảng hệ điều hành android, nó sẽ có các chức năng bao gồm kết nối bluetooth tới thiết bị , gửi tín hiệu điều khiển tiến lên, điều khiển lùi lại, điều khiển rẽ sang trái và điều khiển rẽ sang phải.



Hình 3.10: Biểu đồ ca sử dụng của phần mềm điều khiển

3.3.2. Đặc tả ca sử dụng

3.3.2.1. Ca sử dụng gửi tín hiệu tiến lên

Use Case	Gửi tín hiện tiến lên
Actor	Người điều khiển
Brief Description	Thiết bị điều khiển gửi tín hiện điều khiển tiến lên
Pre-condition	Thiết bị đã kết nối Bluetooth

Basic Flows	1. Người dùng ấn nút tiến lên ở trên màn hình. 2. Thiết bị bắt sự kiện ACTION_DOWN. 3. Thiết bị gửi tín hiệu tiến lên bằng blueooth. 4. Người dùng thả nút tiến lên. 5. Thiết bị bắt sự kiện ACTION_UP. 6. Thiết bị gửi tín hiệu dừng bằng bluetooth.
Alternative Flows	

Bảng 3.8: Ca sử dụng gửi tín hiệu tiến lên

3.3.2.2. Ca sử dụng gửi tín hiệu lùi lại

Use Case	Gửi tín hiện lùi xuống
Actor	Người điều khiển
Brief Description	Thiết bị điều khiển gửi tín hiện điều khiển lùi xuống
Pre-condition	Thiết bị đã kết nối Bluetooth
Basic Flows	1. Người dùng ấn và giữ nút lùi xuống ở trên màn hình. 2. Thiết bị bắt sự kiện ACTION_DOWN. 3. Thiết bị gửi tín hiệu lùi lại thông qua bluetooth. 4. Người dùng thả nút ấn ra. 5. Thiết bị bắt sự kiện ACTION_UP. 6. Thiết bị gửi tín hiệu dừng xe lại.
Alternative Flows	

Bảng 3.9: Ca sử dụng gửi tín hiệu lùi lại

3.3.2.3. Ca sử dụng gửi tín hiệu sang phải

Use Case	Gửi tín hiệu sang phải
Actor	Người điều khiển

Brief Description	Thiết bị điều khiển gửi tín hiện điều khiển sang phải
Pre-condition	Thiết bị đã kết nối Bluetooth
Basic Flows	1. Người dùng ấn và giữ nút sang phải ở trên màn hình. 2. Thiết bị bắt sự kiện ACTION_DOWN. 3. Thiết bị gửi tín hiệu sang phải bằng bluetooth. 4. Người dùng thả nút ấn ra. 5. Thiết bị bắt sự kiện ACTION_UP. 6. Thiết bị gửi tín hiệu dừng xe lại.
Alternative Flows	

Bảng 3.10: Ca sử dụng gửi tín hiệu sang phải

3.3.2.4. Ca sử dụng gửi tín hiệu sang trái

Use Case	Gửi tín hiệu sang trái
Actor	Người điều khiển
Brief Description	Thiết bị điều khiển gửi tín hiện điều khiển sang trái
Pre-condition	Thiết bị đã kết nối Bluetooth
Basic Flows	1. Người dùng ấn và giữ nút sang trái ở trên màn hình. 2. Thiết bị bắt sự kiện ACTION_DOWN. 3. Thiết bị gửi tín hiệu sang trái bằng bluetooth. 4. Người dùng thả nút ấn ra. 5. Thiết bị bắt sự kiện ACTION_UP. 6. Thiết bị gửi tín hiệu dừng lại bằng bluetooth.
Alternative Flows	

Bảng 3.11: Ca sử dụng gửi tín hiệu sang trái

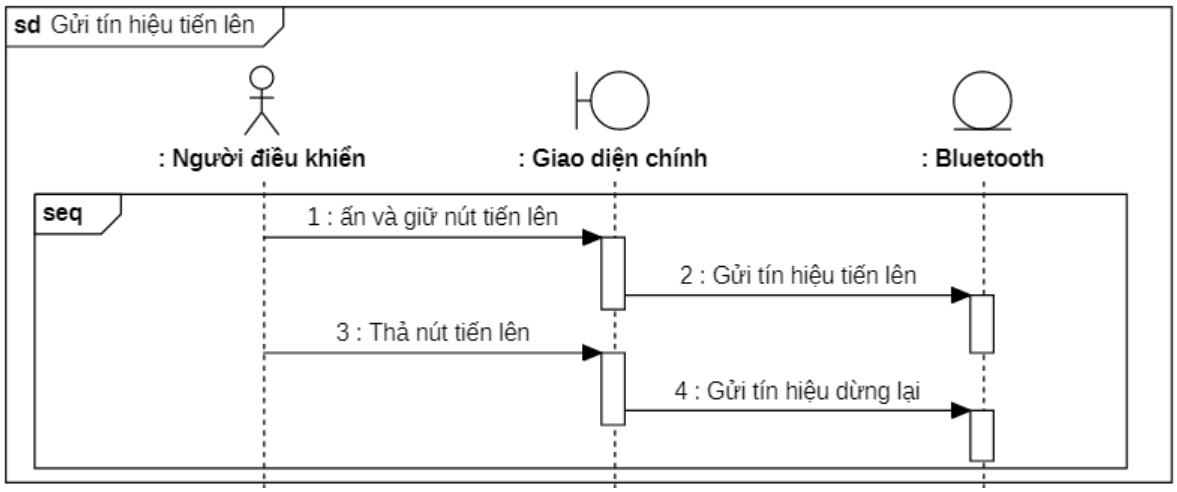
3.3.2.5. Ca sử dụng tăng giảm tốc độ

Use Case	Tăng giảm tốc độ
Actor	Người điều khiển
Brief Description	Thiết bị điều khiển gửi tín hiện điều tăng giảm tốc độ
Pre-condition	Thiết bị đã kết nối Bluetooth
Basic Flows	1. Người dùng chọn tốc độ nhờ thanh trượt trên màn hình 2. Thiết bị bắt sự kiện onChange. 3. Thiết bị gửi tín hiệu tốc độ bằng bluetooth.
Alternative Flows	

Bảng 3.12: Ca sử dụng tăng giảm tốc độ

3.3.3. Phân tích các ca sử dụng

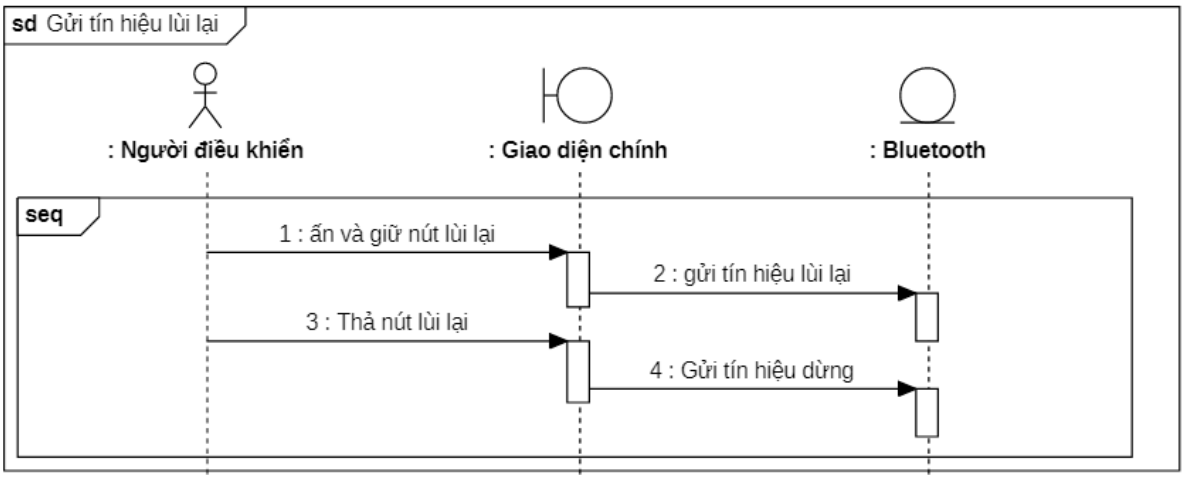
3.3.3.1. Ca sử dụng gửi tín hiệu tiến lên



Hình 3.11: Biểu đồ tuần tử ca sử dụng gửi tín hiệu tiến lên

Người dùng ấn nút tiến lên ở trên màn hình. Thiết bị gửi thông tin tiến lên thông qua bluetooth.

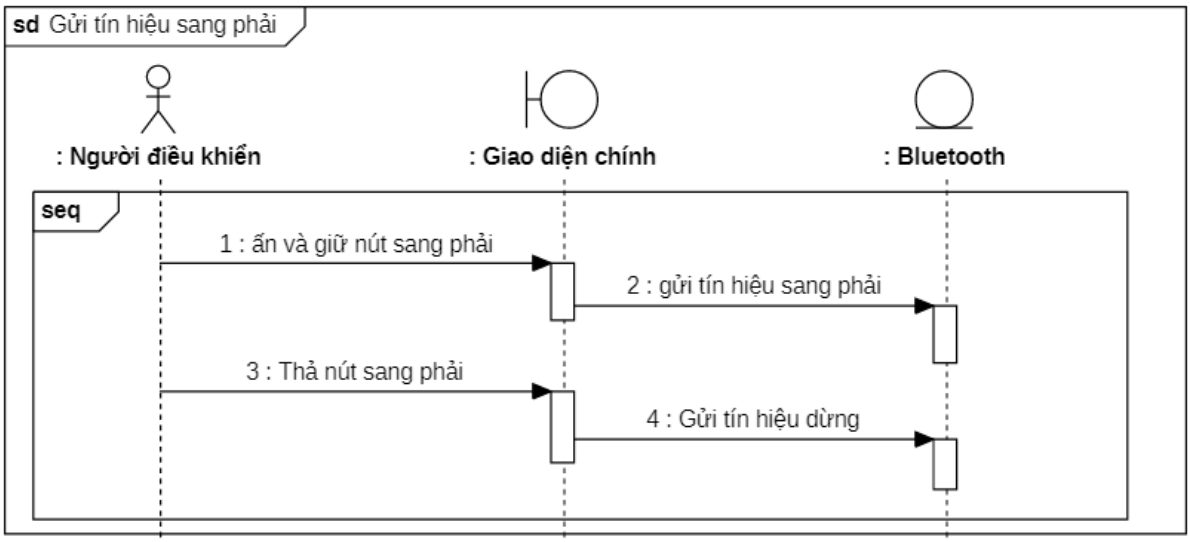
3.3.3.2. Ca sử dụng gửi tín hiệu lùi lại



Hình 3.12: Biểu đồ tuần tự ca sử dụng gửi tín hiệu lùi lại

Người dùng ấn nút lùi xuống ở trên màn hình. Thiết bị gửi thông tin lùi xuống thông qua bluetooth.

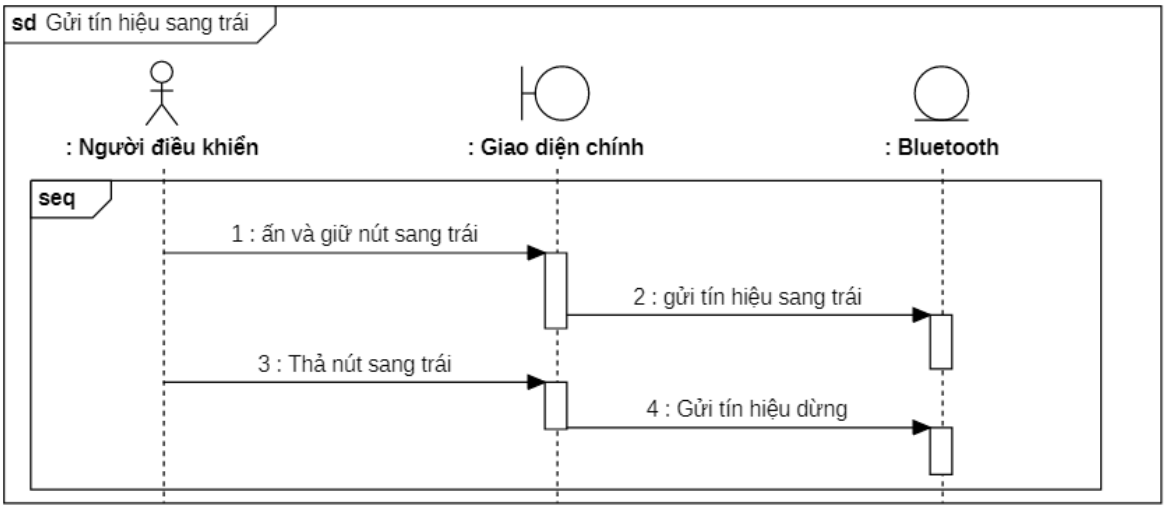
3.3.3.3. Ca sử dụng gửi tín hiệu sang phải



Hình 3.13: Biểu đồ tuần tự ca sử dụng gửi tín hiệu sang phả

Người dùng ấn nút sang phải ở trên màn hình. Thiết bị gửi thông tin sang phải thông qua bluetooth.

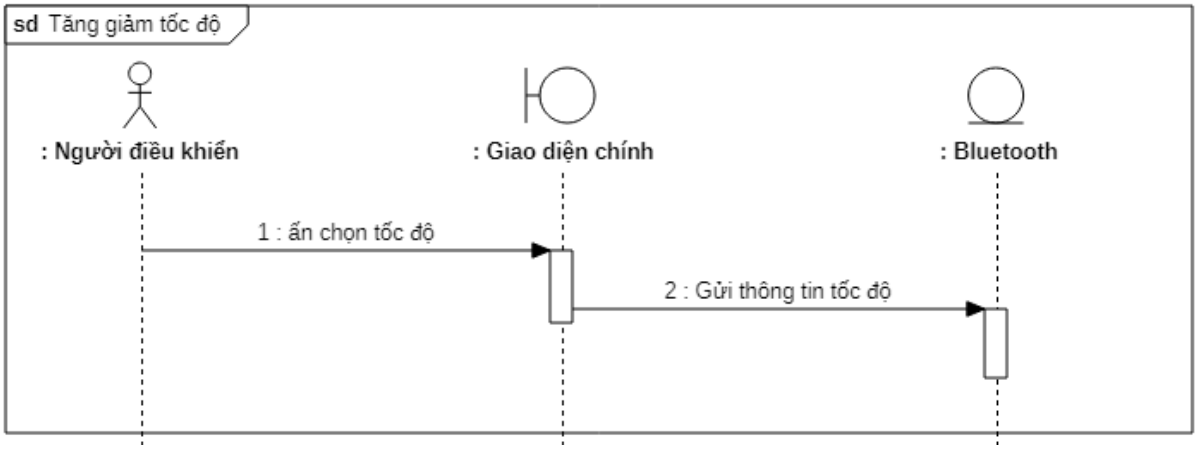
3.3.3.4. Ca sử dụng gửi tín hiệu sang trái



Hình 3.14: Biểu đồ tuần tự ca sử dụng gửi tín hiệu sang trái

Người dùng ấn nút sang trái ở trên màn hình. Thiết bị gửi thông tin sang trái thông qua bluetooth.

3.3.3.5. Ca sử dụng thay dổi tốc độ



Hình 3.15: Biểu đồ tuần tự ca sử dụng thay đổi tốc độ

Người dùng ấn chọn tốc độ dựa vào thanh tốc độ hiển thị ở trên màn hình, thiết bị sẽ bắt sự kiện ấn nút và gửi thông tin tốc độ thông qua bluetooth.

CHƯƠNG 4. THIẾT KẾ ÁP DỤNG HỆ ĐIỀU HÀNH THỜI GIAN THỰC VÀO HỆ THỐNG

4.1. Thiết kế phần mềm

4.1.1. Tổng quan về thiết kế phần mềm

Đối với bất cứ hệ thống nào, phần mềm hệ thống có ý nghĩa như là một bộ não, chi phối mọi hành động được đưa ra. Ngoài yếu tố phần cứng, sự thành công của một hệ thống nhúng còn được quyết định bởi sự hoàn hảo của phần mềm mà nó được cài đặt. Ở đây khi cấu trúc linh kiện phần cứng là như nhau đối với mỗi loại xe, vậy điểm tạo nên sự khác biệt giữa các xe đó nằm ở phần mềm mỗi loại xe sử dụng.

Và tất nhiên, vấn đề thời gian thực là vấn đề hầu hết các hệ thống nhúng cần đáp ứng, nhất là đối với hệ thống xe tự hành nói chung. Sẽ thật là nguy hiểm khi mà một xe tự hành gặp vật cản nhưng tác vụ quản lý hệ thống phanh lại không đáp ứng được trong một khoảng thời gian an toàn.

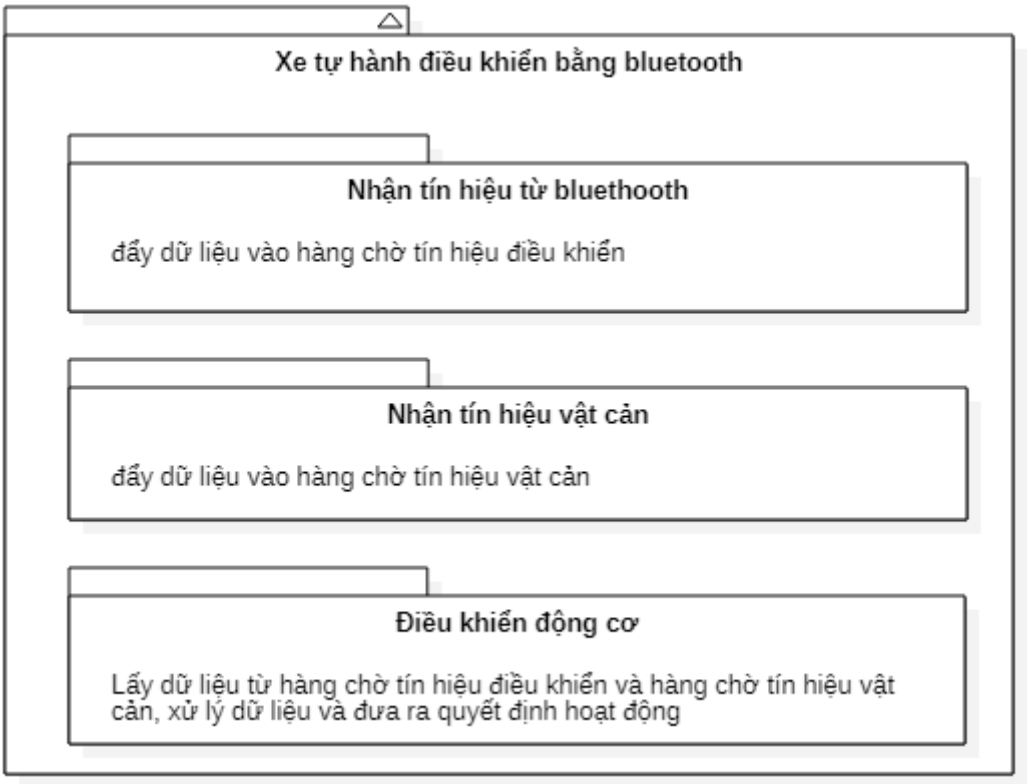
Để đáp ứng vấn đề thời gian thực trong hệ thống nhúng, tốc độ thực thi và độ đơn giản của mã lệnh được đặt lên hàng đầu, do đó ngôn ngữ phổ biến nhất được sử dụng để tạo lên các hệ thống nhúng là C/C++, lý do bởi vì C/C++ là ngôn ngữ lập trình gần với hợp ngữ và ngôn ngữ máy, nên tốc độ thực thi nhanh hơn nhiều so với một số loại ngôn ngữ khác.

Một phần quan trọng nữa trong hệ thống nhúng đó chính là hệ điều hành thời gian thực. Các hệ thống nhúng không thể sử dụng được các hệ điều hành lớn và công kềnh như là Window, thay vào đó nó sử dụng các hệ điều hành chuyên biệt khác để điều phối cấp phát tài nguyên hệ thống. Hệ điều hành FreeRTOS là một trong các hệ điều hành chuyên biệt đó, nó được viết và thực thi dựa trên ngôn ngữ C và nó giúp ra tạo, quản lý các tác vụ trong hệ thống, kèm theo đó là những thư viện hỗ trợ liên quan giúp ta xây dựng hệ thống nhúng một cách đơn giản và hiệu quả.

Hình 4.1: Hệ điều hành FreeRTOS

4.1.2. Xây dựng các tác vụ trong hệ thống

Các tác vụ trong hệ thống xe tự hành điều khiển bằng bluethooth được chia làm ba tác vụ chính là nhận tín hiệu đều khiển từ bluethooth, nhận tín hiệu vật cản và điều khiển động cơ hoạt động.

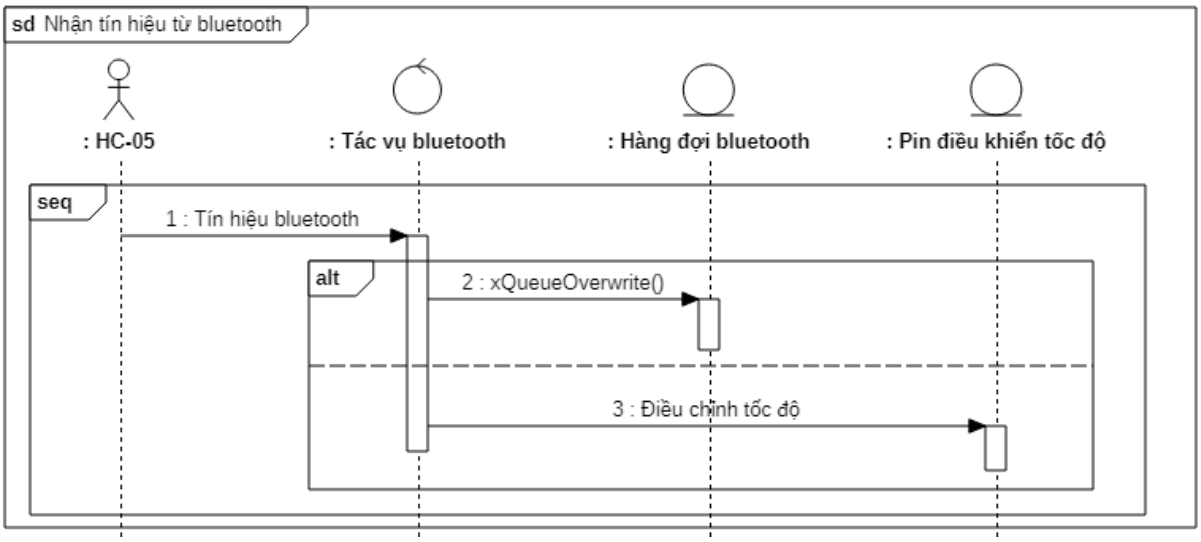


Hình 4.2: Tổng quan các tác vụ trong hệ thống

Đối với tác vụ nhận tín hiệu từ bluetooth, nhóm sử dụng thư viện build-in của hệ điều hành FreeRTOS là queue.h để xây dựng lên hàng đợi. Khi nhận được một message của Module HC-05 gửi về, nó sẽ đẩy tín hiệu này vào trong hàng

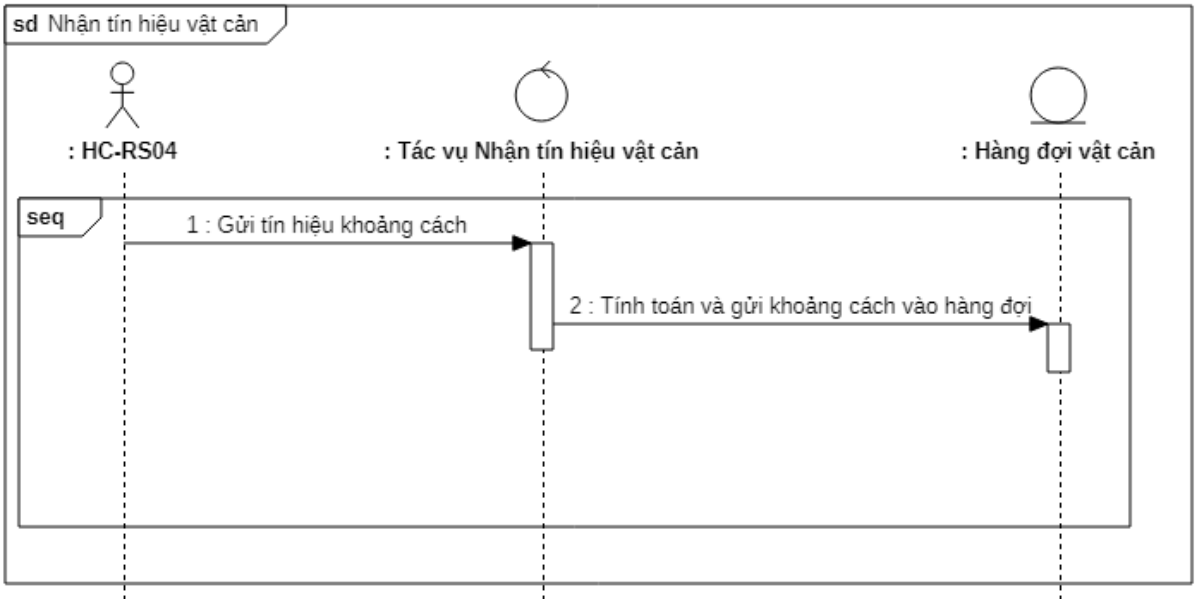
chờ để cho tác vụ điều khiển động cơ đọc và thực thi lệnh, trong trường hợp đó là tín hiệu tăng giảm tốc độ, thì nó sẽ cài đặt tốc độ cho xe bằng API analogWrite()

4.1.3. Phân tích các tác vụ



Hình 4.3: Tác vụ tín hiệu từ bluetooth

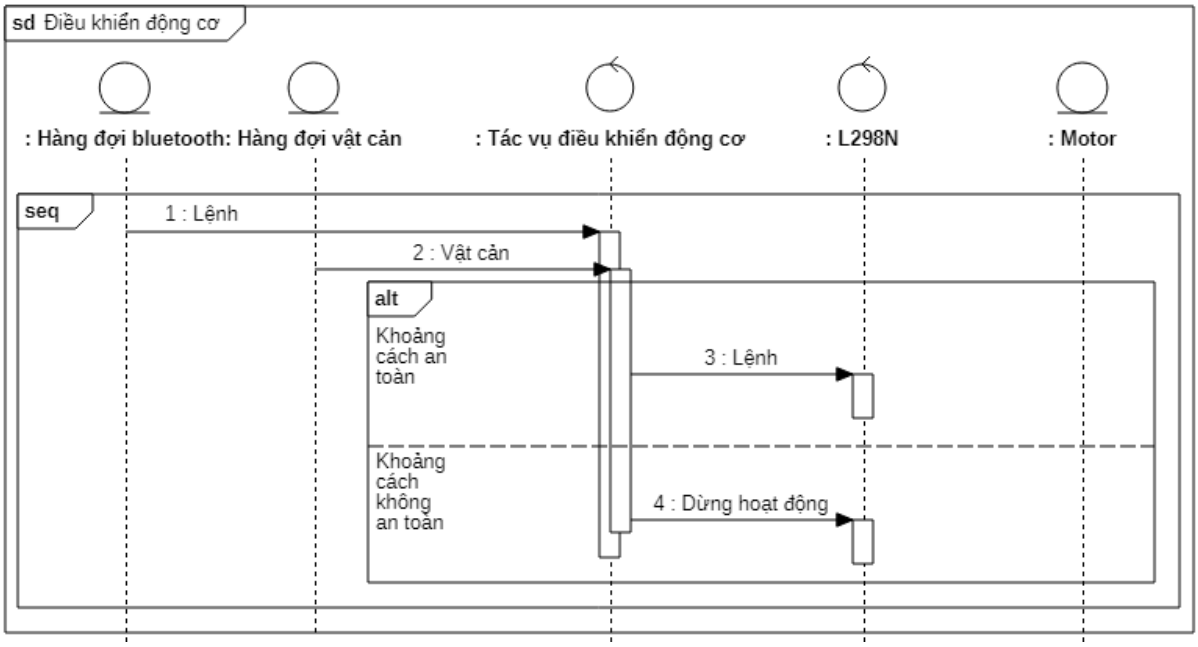
Đối với tách vụ nhận tí hiệu vật cản, Module HC-SR04 sẽ liên tục gửi thông tin khoảng cách của vật cản vào hàng đợi, từ đó tác vụ điều khiển động cơ lấy thông tin tín hiệu ra, phân tích xem có nên cho xe di chuyển tiếp hay không.



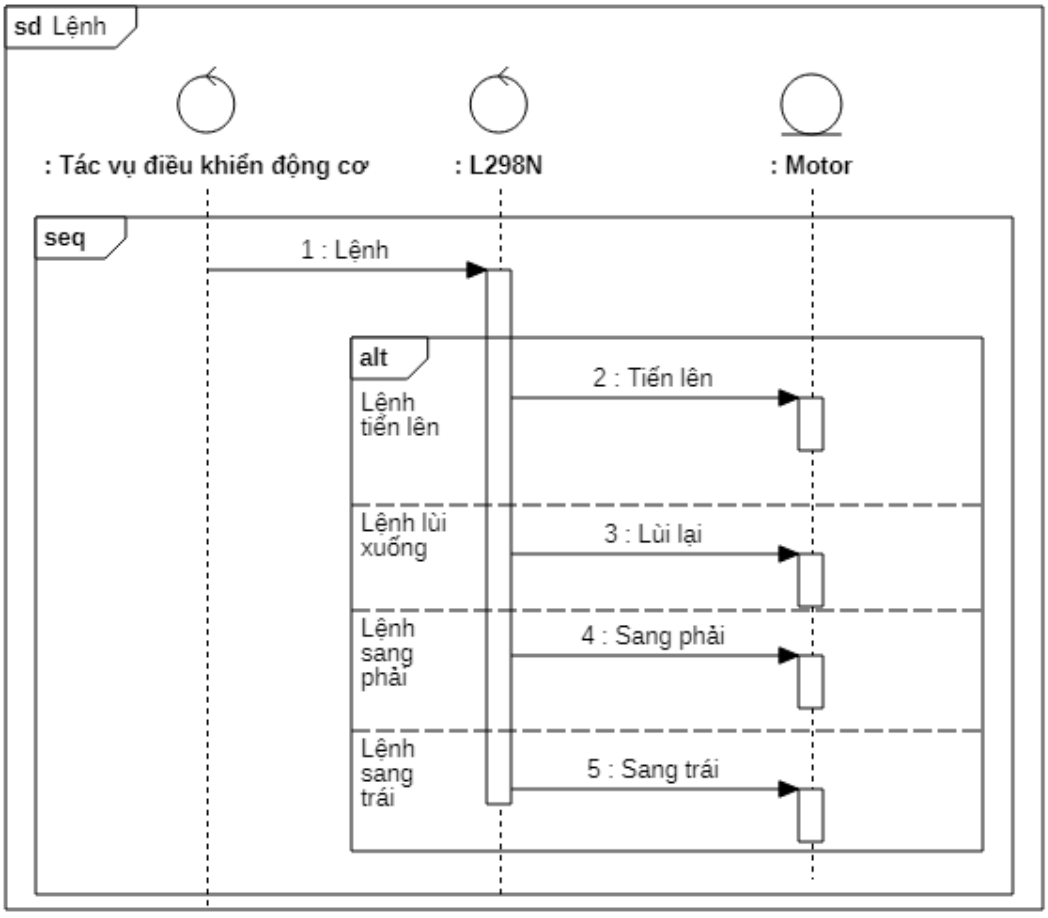
Hình 4.4: Biểu đồ tuần tự tác vụ nhận tín hiệu vật cản

Cuối cùng là tác vụ điều khiển động cơ, tác vụ này sẽ có nghiệm vụ lấy dữ liệu từ hàng đợi bluetooth và hàng đợi vật cản. Đầu tiên, tác vụ xe kiểm tra

xem xe có bị vật cản không, nếu không thì sẽ cho động cơ hoạt động theo lệnh được đẩy vào trong hàng đợi bluetooth.

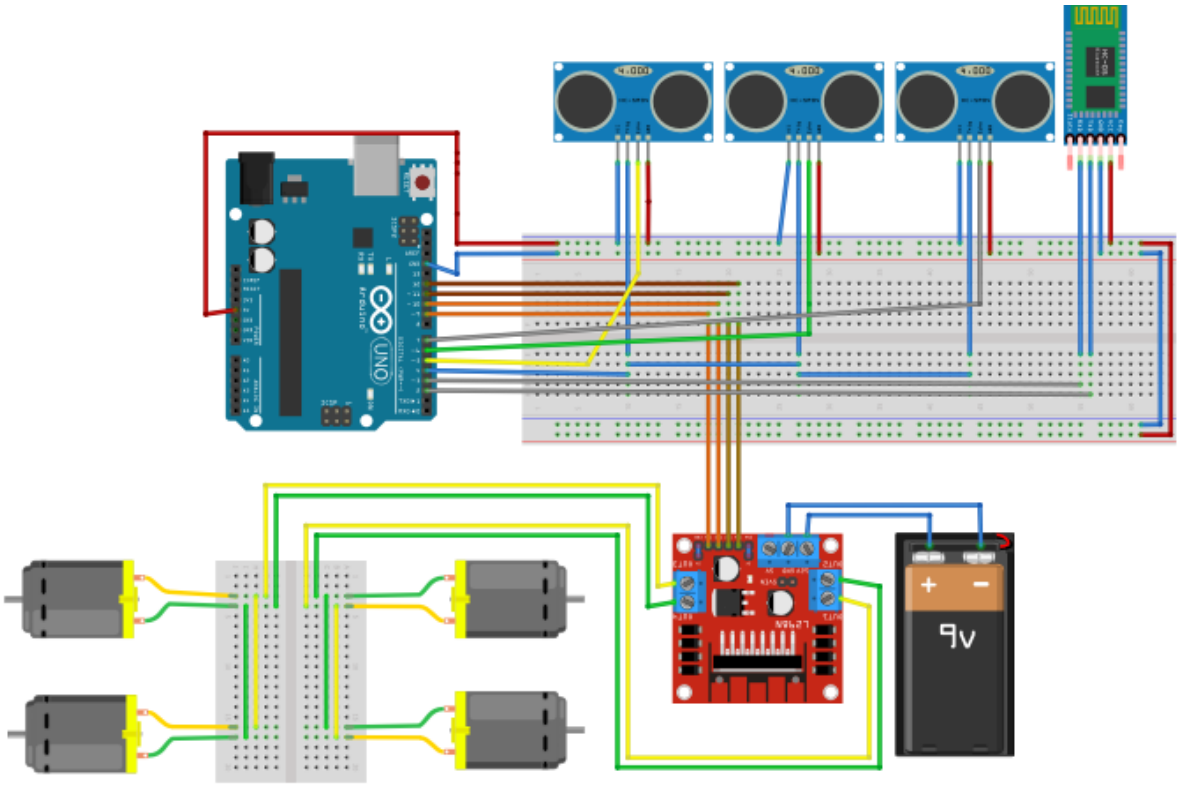


Hình 4.5: Biểu đồ tuần tự tác vụ điều khiển động cơ



Hình 4.6: Biểu đồ tuần tự chi tiết lệnh

4.2. Thiết kế phần cứng



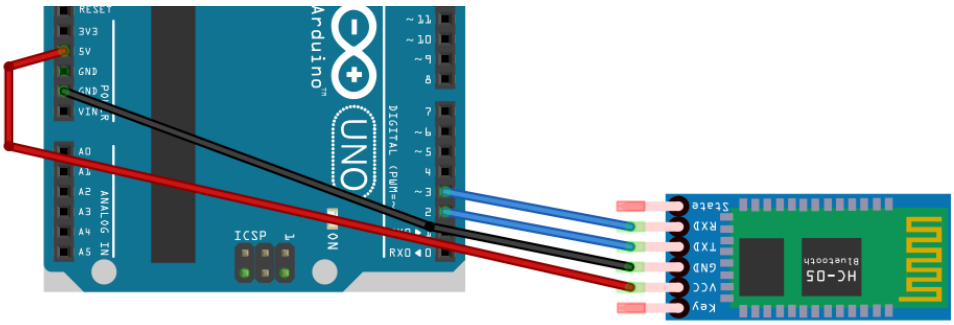
Hình 4.7: Thiết kế mạch của hệ thống

Đối với Module bluetooth HC-05, do khuyến nghị không nên sử dụng hai chân Serial là chân 1 và chân 2 ở trên Arduino, vì dễ dẫn đến xung đột khi nạp chương trình, nên nhóm thiến hành config và sử dụng chân 2,3 lần lượt làm chân TXD và RXD để trao đổi thông tin bluetooth. Arduino có thư viện build-in là SoftwareSerial.h để hỗ trợ chuyển đổi hai chân digital 2 và 3 thành chân Serial.

```
#include <SoftwareSerial.h>
#define TX_PIN 3;
#define RX_PIN 2;
SoftwareSerial bluetooth(TX_PIN, RX_PIN);
```

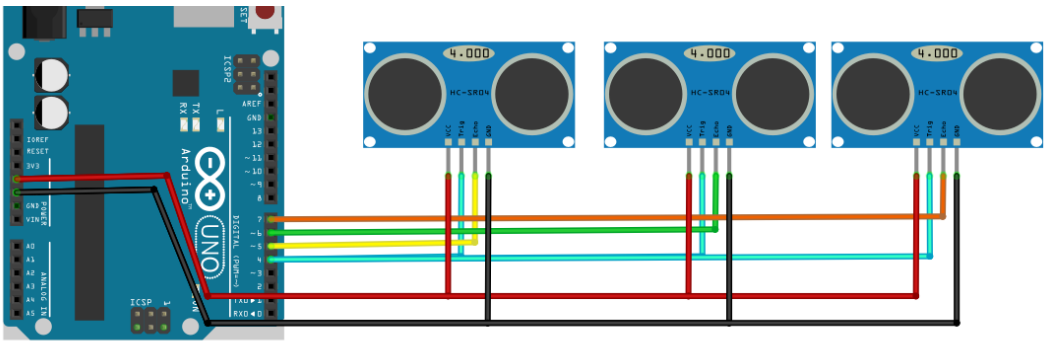
Hình 4.8: Code mô phỏng cách cấu hình chân Serial

Ta lần lượt nối chân 3 của Arduino với chân RXD của Module HC-05, nối chân 2 của Arduino với chân TXD của Module HC-05, tiếp theo tiến hành nối Vcc và GND của Module HC-05 vào Arduino.



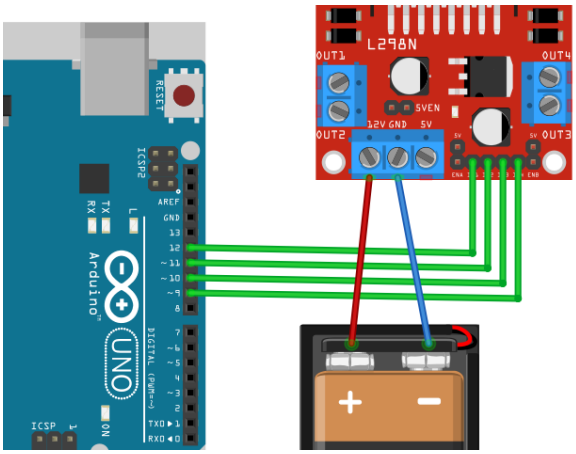
Hình 4.9: Cách nối module-hc05

Đối với Module HC-SR04, ở đây đã sử dụng 4 module HC-SR04, nối chân trig của cả 3 module chung vào pin 4 của Arduino, nối Vcc và GND cho cả 3 module, và lần lượt nối chân echo của từng module vào chân 5, 6 và 7.



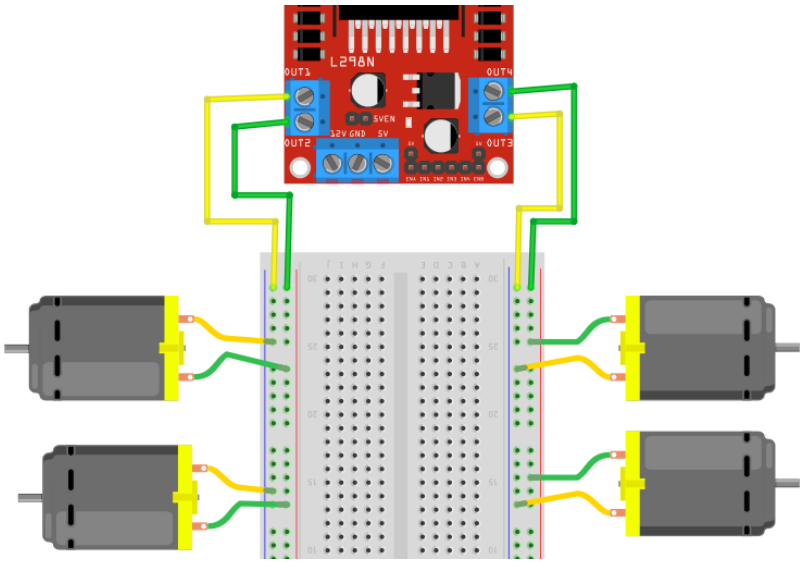
Hình 4.10: Cách nối HC-SR04

Đối với Module L298D, mắc lần lượt bốn chân của Module L298D là IN1, IN2, IN3, IN4 vào bốn chân 9, 10, 11, 12 của arduino, mắc cổng 12V của L298D vào cực dương của pin, mắc chân GND vào cực âm của pin.



Hình 4.11: Cách mắc module L298N

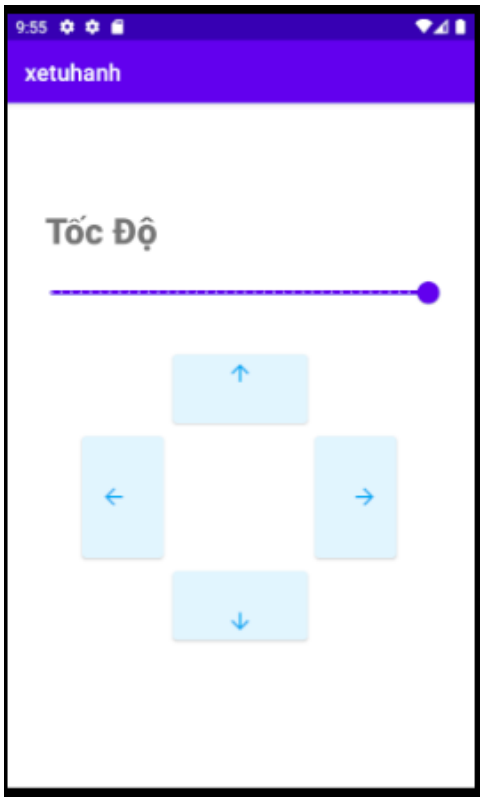
Cuối cùng, mắc lần lượt các động cơ vào Module L298D theo sơ đồ sau:



Hình 4.12: Cách mắc motor vào L298N

4.3. Thiết kế thiết bị điều khiển

Thiết bị điều khiển được dùng là điện thoại chạy hệ điều hành android, được cài phần mềm chuyên biệt để điều khiển xe. Màn hình chứa bao gồm bốn nút là nút điều khiển lên, xuống, sang phải, sang trái.



Hình 4.13: Thiết bị điều khiển

PHỤ LỤC

CODE Arduino của xe tự hành điều khiển bằng bluetooth:

```
#include<Arduino_FreeRTOS.h>
#include<task.h>
#include<queue.h>
#include<SoftwareSerial.h>

QueueHandle_t frontQueue;
QueueHandle_t rightQueue;
QueueHandle_t leftQueue;
QueueHandle_t commandQueue;

SoftwareSerial bluetooth(2,3);

int trig=4;
int echoRight=5;
int echoLeft=7;
int echoFront=6;

int speedPin = 9;

int rightmotor1=8;
int rightmotor2=10;
int leftmotor1=11;
int leftmotor2=12;

void setup(){
  //=========== active pin and serial ==========
  Serial.begin(9600);
  bluetooth.begin(9600);
  pinMode(trig, OUTPUT);
  pinMode(echoRight, INPUT);
  pinMode(echoFront, INPUT);
  pinMode(echoLeft, INPUT);
  pinMode(speedPin, OUTPUT);
  pinMode(rightmotor1, OUTPUT);
  pinMode(rightmotor2, OUTPUT);
  pinMode(leftmotor1, OUTPUT);
```

```
  pinMode(leftmotor2, OUTPUT);
  //=========================================

  analogWrite(speedPin,250);

  //================ create queue ================
  frontQueue = xQueueCreate(1,sizeof(int));
  rightQueue = xQueueCreate(1,sizeof(int));
  leftQueue = xQueueCreate(1,sizeof(int));
  commandQueue = xQueueCreate(1,sizeof(int));
  //=========================================

  //============== create task ==================
  xTaskCreate(hdsr04,"khoangcach", 200,NULL,1,NULL);
  xTaskCreate(start,"chay", 200,NULL,1,NULL);
  xTaskCreate(command,"cmd",200,NULL,1,NULL);
  vTaskStartScheduler();
  //=============================================
}

void loop(){}

void start(void *param){
  int leftspace;
  int rightspace;
  int frontspace;
  while(1){

    //===== get command from queue =========
    int cmd;
    xQueuePeek(commandQueue,&cmd,(TickType_t)0);

    //======================================

    //======== L398N Controller ============
    switch(cmd){
      case '1':
        xQueuePeek(frontQueue, &frontspace, ( TickType_t )0);
        if(frontspace>40){
          front();
        }else stp();
```

```
        break;
      case '2':
        back();
        break;
      case '3':
        xQueuePeek(leftQueue, &leftspace, ( TickType_t )0);
        if(leftspace>25){
          left();
        }else stp();
        break;
      case '4':
        xQueuePeek(rightQueue, &rightspace, ( TickType_t )0);
        if(rightspace>25){
          right();
        }else stp();
        break;
       case '5':
        stp();
        break;
      default:
        break;
    }
    //====================================
    vTaskDelay(20/ portTICK_PERIOD_MS);
  }
}


void command(void *param){
  while(1){
    if(bluetooth.available()){
      int cmd;
      cmd = bluetooth.read();
      if(cmd< 60 && cmd >40){
        xQueueOverwrite(commandQueue,&cmd);
      }else if (cmd>=100 && cmd<=250){
        analogWrite(speedPin, cmd);
      }
    }
    vTaskDelay(40/ portTICK_PERIOD_MS);
  }
```

```
}

void hdsr04(void *param){
  unsigned long duration1;
  unsigned long duration2;
  unsigned long duration3;

  int distance1;
  int distance2;
  int distance3;
  while(1){
    //   right hc sr04

    digitalWrite(trig,LOW);
    vTaskDelay(2/portTICK_PERIOD_MS);
    digitalWrite(trig,HIGH);
    vTaskDelay(10/portTICK_PERIOD_MS);
    digitalWrite(trig,LOW);
    duration1 = pulseIn(echoRight,HIGH);
    distance1 = int(duration1/2/29.412);
    if(distance1>0){
      xQueueOverwrite(rightQueue,&distance1);
    }
    vTaskDelay(20/portTICK_PERIOD_MS);

    // front hc-sr04

    digitalWrite(trig,LOW);
    vTaskDelay(2/portTICK_PERIOD_MS);
    digitalWrite(trig,HIGH);
    vTaskDelay(10/portTICK_PERIOD_MS);
    digitalWrite(trig,LOW);
    duration2 = pulseIn(echoFront,HIGH);
    distance2 = int(duration2/2/29.412);
    if(distance2>0){
      xQueueOverwrite(frontQueue,&distance2);
    }
    vTaskDelay(20/portTICK_PERIOD_MS);

    // left
    digitalWrite(trig,LOW);
```

```
    vTaskDelay(2/portTICK_PERIOD_MS);
    digitalWrite(trig,HIGH);
    vTaskDelay(10/portTICK_PERIOD_MS);
    digitalWrite(trig,LOW);
    duration3 = pulseIn(echoLeft,HIGH);
    distance3 = int(duration3/2/29.412);
    if(distance3>0){
      xQueueOverwrite(leftQueue,&distance3);
    }
    vTaskDelay(20/portTICK_PERIOD_MS);
  }
}

void back(){
  digitalWrite(leftmotor1, HIGH);
  digitalWrite(leftmotor2, LOW);
  digitalWrite(rightmotor1, HIGH);
  digitalWrite(rightmotor2, LOW);
}
void front(){
  digitalWrite(leftmotor1, LOW);
  digitalWrite(leftmotor2, HIGH);
  digitalWrite(rightmotor1, LOW);
  digitalWrite(rightmotor2, HIGH);
}
void right(){
  digitalWrite(leftmotor1, LOW);
  digitalWrite(leftmotor2, HIGH);
  digitalWrite(rightmotor1, HIGH);
  digitalWrite(rightmotor2, LOW);
}
void left(){
  digitalWrite(leftmotor1, HIGH);
  digitalWrite(leftmotor2, LOW);
  digitalWrite(rightmotor1, LOW);
  digitalWrite(rightmotor2, HIGH);
}

void stp(){
  digitalWrite(leftmotor1, LOW);
  digitalWrite(leftmotor2, LOW);
```

```
  digitalWrite(rightmotor1, LOW);
  digitalWrite(rightmotor2, LOW);
}
```

TÀI LIỆU THAM KHẢO

Nguyễn Ngọc Bình, *Công nghệ phần mềm nhúng*, Nhà xuất bản đại học Quốc gia Hà Nội, 2013.

Lê Thị Hồng Vân, *Giáo trình hệ điều hành nhúng thời gian thực*, Học viện Kỹ thuật Mật Mã.

http://arduino.vn/bai-viet/42-arduino-uno-r3-la-gi

http://arduino.vn/bai-viet/893-cach-dung-module-dieu-khien-dong-co-l298n-cau-h-de-dieu-khien-dong-co-dc

http://arduino.vn/bai-viet/953-huong-dan-thiet-lap-chi-tiet-cho-module-bluetooth

http://arduino.vn/bai-viet/233-su-dung-cam-bien-khoang-cach-hc-sr04

http://arduino.vn/bai-viet/639-du-xe-dieu-khien-tu-xa-qua-bluetooth

https://www.youtube.com/watch?v=ChRvQ1nh_Og

https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf

http://arduino.vn/bai-viet/953-huong-dan-thiet-lap-chi-tiet-cho-module-bluetooth