

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ

-----o0o-----



ĐỒ ÁN MÔN HỌC

THIẾT KẾ HỆ THỐNG CẢM BIẾN HỖ TRỢ NGƯỜI LÁI

GVHD: Võ Tấn Thông

SVTH: Vũ Quốc Việt

MSSV: 2015052

TP. HỒ CHÍ MINH, THÁNG 5 NĂM 2023

LỜI CẢM ƠN

Em xin bày tỏ lòng biết ơn đến thầy Võ Tấn Thông trên cương vị là người hướng dẫn chính của đề tài đã tận tình giúp đỡ trong suốt quá trình thực hiện đồ án.

Em cũng xin bày tỏ lòng biết ơn đến các thầy cô trong trường Đại học Bách Khoa Thành phố Hồ Chí Minh đã tận tình dạy dỗ và truyền thụ những kinh nghiệm quý báu trong suốt thời gian qua

Cuối cùng, em xin chân thành cảm ơn sự đóng góp ý kiến của tất cả các bạn sinh viên trong suốt quá trình thực hiện đồ án.

TÓM TẮT ĐỒ ÁN

Ngày nay, cảm biến là thiết bị được ứng dụng rất phổ biến trong việc chế tạo xe hơi. Các chức năng hỗ trợ người lái đang ngày càng được các công ty xe hơi phát triển và triển khai rộng rãi trên các dòng xe trên mọi phân khúc.

Đồ án này trình bày về hệ thống cảnh báo cản trước cho xe hơi sử dụng cảm biến sóng siêu âm HC SR04, mô hình xe tự lắp và kit Arduino.

Khi có xuất hiện vật cản phía trước xe ở trong phạm vi 26cm, xe sẽ tự động dừng lại. Người dùng cần nhấn nút cảnh báo trên phần mềm điều khiển xe để có thể điều khiển xe.

Tuy nhiên, nếu vẫn có vật cản trong phạm vi 26cm, xe sẽ không cho người dùng điều khiển đi thẳng, chỉ cho phép đi lùi hoặc xoay trái, xoay phải. Sau khi cảm biến xác định được vật cản đã ra ngoài phạm vi 26cm, xe sẽ tự động cho người dùng điều khiển xe chạy thẳng.

Do có sự trễ trong chương trình cũng như phụ thuộc vào ma sát của đường đi, khoảng cách cụ thể khi xe dừng lại sẽ cách khoảng từ 3 đến 10cm thay vì 26cm như trong cài đặt.

MỤC LỤC

1. GIỚI THIỆU	1
1.1 Tổng quan	1
1.2 Nhiệm vụ đề tài	1
2. LÝ THUYẾT	1
3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG	6
4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM	11
5. KẾT QUẢ THỰC HIỆN.....	23
6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	24
6.1 Kết luận.....	24
6.2 Hướng phát triển	25
7. TÀI LIỆU THAM KHẢO.....	25

1. GIỚI THIỆU

1.1 Tổng quan

Cảm biến cản trước trên xe hơi là một trong những phần quan trọng trong hệ thống an toàn của xe hơi. Các cảm biến này được sử dụng để phát hiện các vật cản phía trước, từ đó giúp tài xế có thể đưa ra quyết định an toàn và tránh tai nạn giao thông.

Các cảm biến cản trước trên xe hơi thường được đặt ở vị trí ở phía trước của xe, trên nắp capô hoặc ở khu vực bên dưới cản trước. Các cảm biến này thường sử dụng các công nghệ như siêu âm hoặc hồng ngoại để phát hiện vật cản phía trước và đưa ra cảnh báo cho tài xế thông qua hệ thống âm thanh hoặc đồ họa trên màn hình hiển thị.

Sử dụng cảm biến cản trước trên xe hơi giúp tài xế có thể đưa ra quyết định an toàn và tránh va chạm với các vật cản phía trước, giảm thiểu nguy cơ tai nạn giao thông. Các cảm biến này cũng có thể được kết hợp với các tính năng khác như hệ thống phanh tự động để tăng cường tính an toàn trên đường.

1.2 Nhiệm vụ đề tài

- *Nội dung 1:* Tìm hiểu nguyên lý, lý thuyết về hệ thống cảnh báo va chạm trên cản trước xe hơi
- *Nội dung 2:* Tìm hiểu về cảm biến siêu âm HC-SR04 và vi xử lý Arduino.
- *Nội dung 3:* Thiết kế xe lắp ráp và viết giải thuật cho đề tài.

2. LÝ THUYẾT

- **Tìm hiểu nguyên lý về hệ thống cảnh báo va chạm trên cản trước xe hơi.**

Hệ thống cảnh báo va chạm trên cản trước xe hơi sử dụng các công nghệ như siêu âm hoặc hồng ngoại để phát hiện vật cản phía trước và đo khoảng cách. Các cảm biến này sẽ tạo ra các sóng siêu âm hoặc tín hiệu hồng ngoại và sau đó đo thời gian mà sóng này phản xạ lại từ vật cản và quay trở lại cảm biến. Khi khoảng cách được tính toán, hệ thống sẽ đưa ra quyết định an toàn và cảnh báo tài xế nếu cần thiết.

Hệ thống cảnh báo va chạm trên cản trước xe hơi có thể được kết hợp với các tính năng an toàn khác như hệ thống phanh tự động để tăng cường tính an toàn trên đường. Việc sử dụng hệ thống cảnh báo va chạm trên cản trước xe hơi giúp giảm thiểu nguy cơ tai nạn giao thông và tăng cường tính an toàn cho tài xế và hành khách trên xe.

- **Tìm hiểu về cảm biến siêu âm HC-SR04 và vi xử lý Arduino.**

- *Vi xử lý Arduino.*

Arduino là một nền tảng phát triển phần cứng mã nguồn mở được thiết kế để dễ dàng lập trình và sử dụng. Arduino cung cấp cho người dùng một bộ công cụ phần mềm miễn phí để lập trình và một bộ vi điều khiển phổ biến để kết nối với các thiết bị ngoại vi. Vi điều khiển này chạy trên một kiến trúc RISC 8 bit hoặc 32 bit và được lập trình bằng ngôn ngữ lập trình C/C++ thông qua môi trường phát triển tích hợp (IDE).

Arduino có nhiều phiên bản khác nhau, bao gồm các phiên bản cơ bản với nhiều loại vi điều khiển khác nhau, từ các vi điều khiển đơn giản như ATmega328 cho đến các vi điều khiển mạnh hơn như ARM Cortex-M3. Arduino cũng có thể được mở rộng thông qua các board phụ trợ để kết nối với các cảm biến, mạch điều khiển và các thiết bị ngoại vi khác.

Các ứng dụng của Arduino rất đa dạng, từ các dự án điều khiển đơn giản như bật tắt đèn, điều khiển tưới cây tự động cho đến các dự án phức tạp hơn như thiết bị đo lường và giám sát môi trường, điều khiển robot và xe tự hành, và nhiều ứng dụng khác.

Arduino được phát triển với mục đích đơn giản hóa việc phát triển các ứng dụng IoT (Internet of Things) và điều khiển tự động. Arduino có thể được sử dụng để giảm thời gian và chi phí phát triển sản phẩm, giúp các nhà phát triển tập trung vào ứng dụng thực tế hơn là việc phát triển phần cứng.

- *Cảm biến siêu âm HC-SR04*

Cảm biến siêu âm HC-SR04 là một loại cảm biến không tiếp xúc được sử dụng để đo khoảng cách từ đối tượng đến cảm biến. Cảm biến này hoạt động bằng cách phát ra một tín hiệu siêu âm và đo thời gian mà tín hiệu này phản xạ từ đối tượng và trở về cảm biến.

Cảm biến HC-SR04 được thiết kế với hai bộ phận chính: phần phát sóng siêu âm và phần thu sóng siêu âm. Phần phát sóng siêu âm phát ra tín hiệu siêu âm ở tần số 40kHz, trong khi phần thu sóng siêu âm sẽ nhận tín hiệu siêu âm này và tính toán khoảng cách từ đối tượng đến cảm biến.

Cảm biến HC-SR04 thường được sử dụng trong các ứng dụng liên quan đến đo khoảng cách, như đo khoảng cách để tránh va chạm trong xe hơi, đo khoảng cách để hướng dẫn điều khiển robot, hay đo độ cao của nước trong bể chứa.

Để sử dụng cảm biến HC-SR04, người dùng có thể kết nối cảm biến với vi điều khiển như Arduino và sử dụng mã lệnh để đọc giá trị đo được từ cảm biến. Các giá trị đọc được từ cảm biến này sẽ được chuyển đổi thành đơn vị đo lường thực tế, ví dụ như đo khoảng cách bằng centimet hoặc inch.

- **Thiết kế xe lắp ráp.**

- *Xe lắp ráp*

Khung xe robot 3 bánh được thiết kế phù hợp với các yêu cầu chế tạo robot dò đường, robot dò thám, robot tránh vật cản, có thể lắp ghép với cánh tay robot để điều khiển. Là sản phẩm để làm robot mô hình, đặc biệt phù hợp với Arduino.

Xe được thiết kế rất đẹp, chắc chắn có thể hoạt động linh hoạt, các bánh xe được lắp ghép đối xứng, giữ cân bằng và chính xác trong di chuyển. Trọn bộ khung xe robot 3 bánh bao gồm đầy đủ các chi tiết được lắp ghép gọn gàng.

- + Bánh xe

Có 2 bánh cố định được gắn chặt với động cơ làm nhiệm vụ truyền lực cho xe chạy: được làm bằng nhựa, lốp xe làm bằng cao su mềm, có độ bám đường tốt, chắc chắn. Kích thước đường kính bánh xe 65mm.

1 bánh xe dẫn động giúp xe chạy đa hướng. Vật liệu từ nhựa và thép, có vòng bi xoay.

- + Động cơ

2 Động cơ có hộp giảm tốc, có thể lập trình bấm xung để điều chỉnh tốc độ của động cơ. Sử dụng nguồn nuôi có mức điện áp hoạt động 3 – 6V.

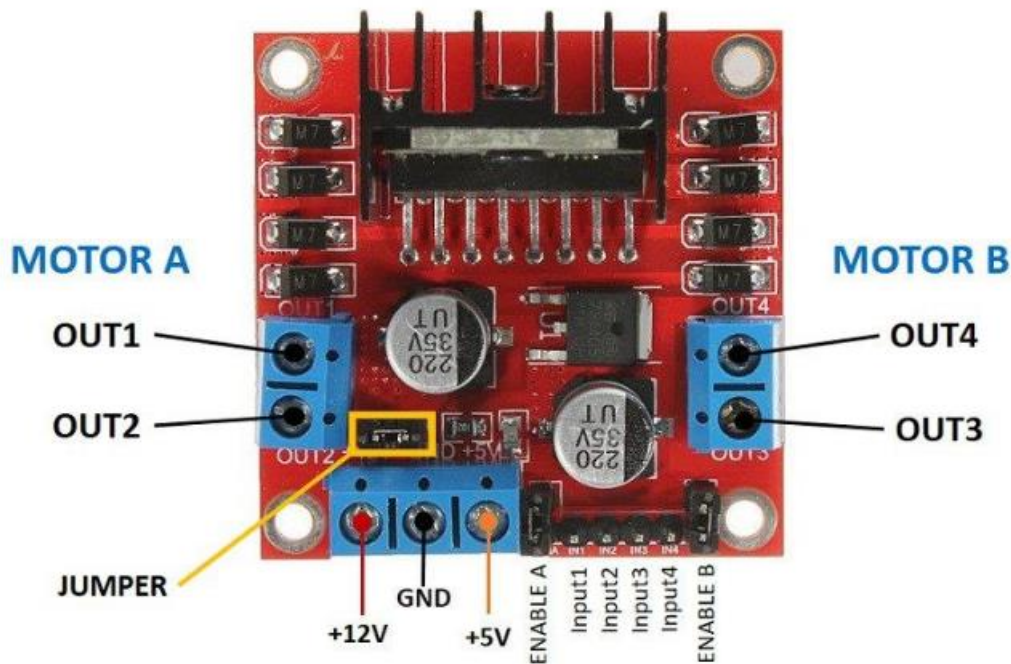
- + Khung xe

Được làm bằng nhựa có nhiều lỗ sẵn để bắt ốc vào những vị trí cần thiết để gắn board mạch Arduino, đế pin, công tắc, động cơ, bánh trước để gắn các board mạch gắn cảm biến. Kích thước của khung là 220×150mm.

- *Module điều khiển xe L298*

Module điều khiển động cơ L298 là bộ điều khiển sử dụng mạch cầu H để dễ dàng điều khiển chiều quay và tốc độ của tối đa 2 động cơ DC.

Module này dễ sử dụng, chi phí thấp, dễ lắp đặt. Module L298N được dùng để chế tạo xe điều khiển, xe dò đường, các thiết bị sử dụng động cơ DC hoặc động cơ bước ... sử dụng kết hợp với vi điều khiển.



- Module thu phát Bluetooth HC-05

Module Bluetooth thu phát HC-05 dùng để thiết lập kết nối Serial giữa 2 thiết bị bằng sóng bluetooth. Điểm đặc biệt của module bluetooth HC-05 là module có thể hoạt động được ở 2 chế độ: MASTER hoặc SLAVE.

+ Ở chế độ SLAVE: bạn cần thiết lập kết nối từ smartphone, laptop, usb bluetooth để dò tìm module sau đó pair với mã PIN là 1234. Sau khi pair thành công, bạn đã có 1 cổng serial từ xa hoạt động ở baud rate 9600.

+ Ở chế độ MASTER: module sẽ tự động dò tìm thiết bị bluetooth khác (1 module bluetooth HC-06, usb bluetooth, bluetooth của laptop...) và tiến hành pair chủ động mà không cần thiết lập gì từ máy tính hoặc smartphone.

Module Bluetooth thu phát HC-05 được thiết kế nhỏ gọn ra chân tín hiệu giao tiếp cơ bản và nút bấm để vào chế độ AT COMMAND, mạch được thiết kế để có thể cấp nguồn và giao tiếp qua 3.3VDC hoặc 5VDC, thích hợp cho nhiều ứng dụng khác nhau: Robot Bluetooth, điều khiển thiết bị qua Bluetooth,....



- *PIN 18650 3.7V*

Pin 18650 thuộc 1 nhóm trong các loại Pin Lithium-Ion, chúng có kích thước 18mm x 65 mm (đường kính x chiều dài) hình trụ

Pin 18650 được sử dụng phổ biến trong các thiết bị cầm tay như: Pin sạc dự phòng, máy khoan, máy cắt, quạt cầm tay, đèn pin, pin laptop. Chúng có ưu điểm nhỏ gọn, thời gian sử dụng lâu dài, có thể ghép thành điện áp cao, và quan trọng nhất là an toàn với người dùng



3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

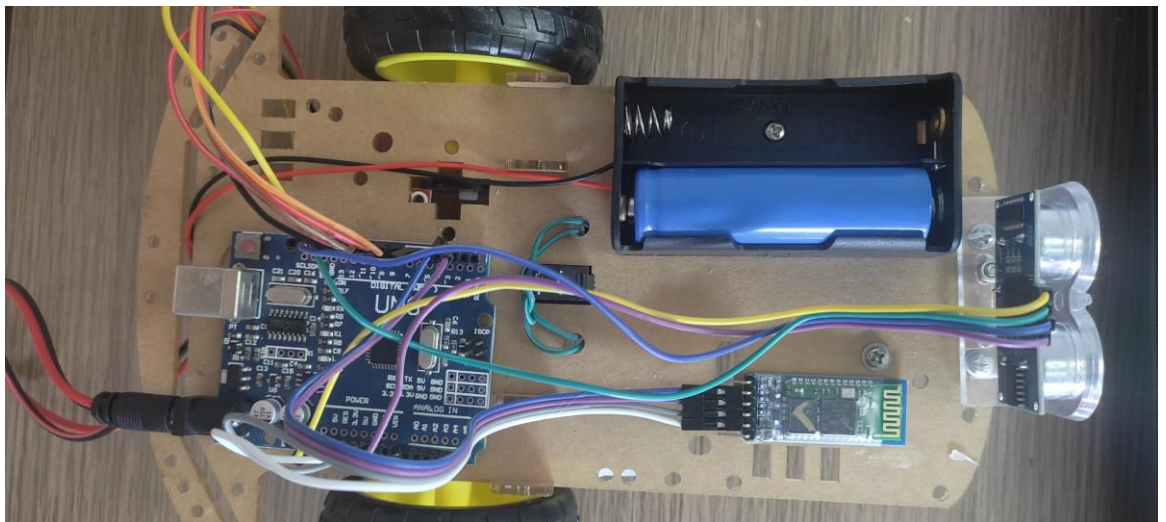
- **Yêu cầu thiết kế**

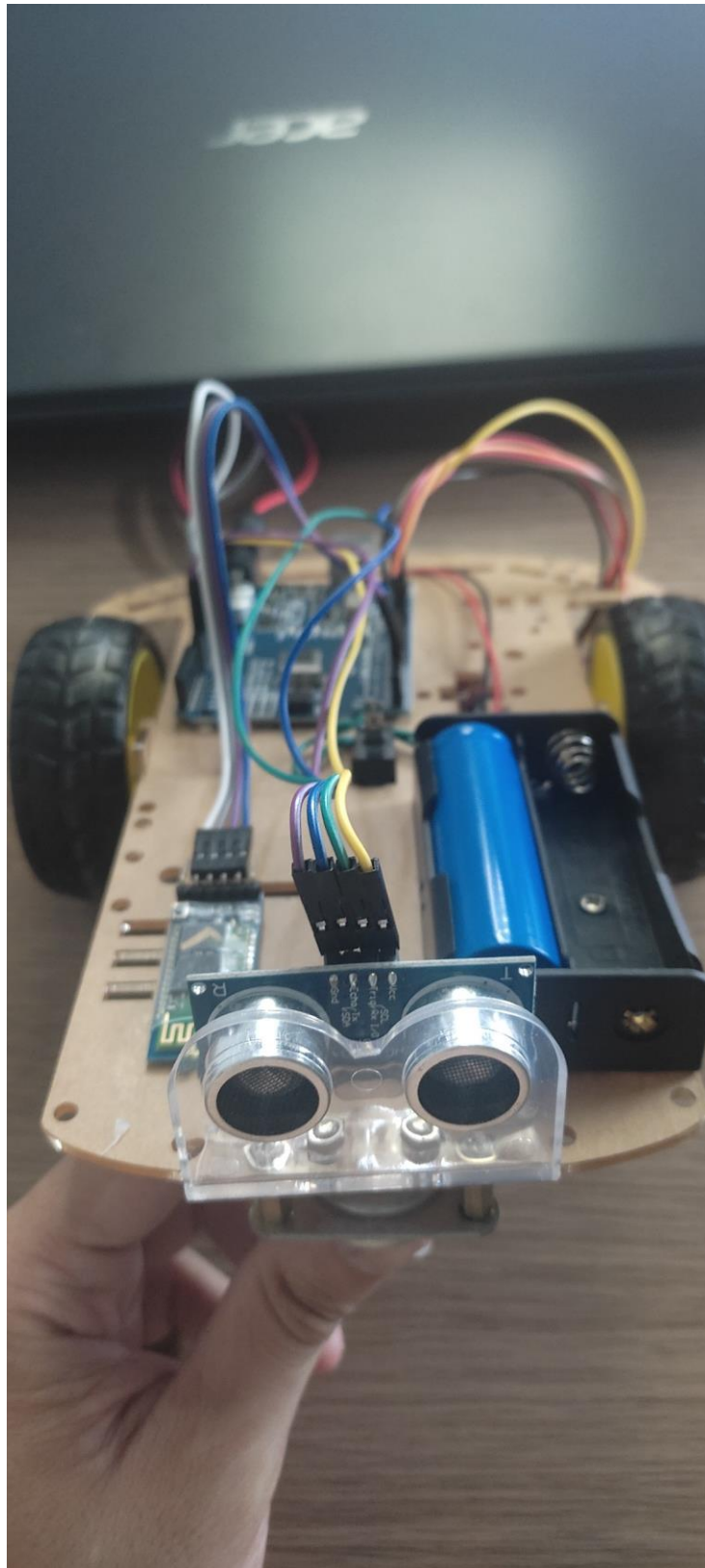
Các yêu cầu thiết kế cần đảm bảo:

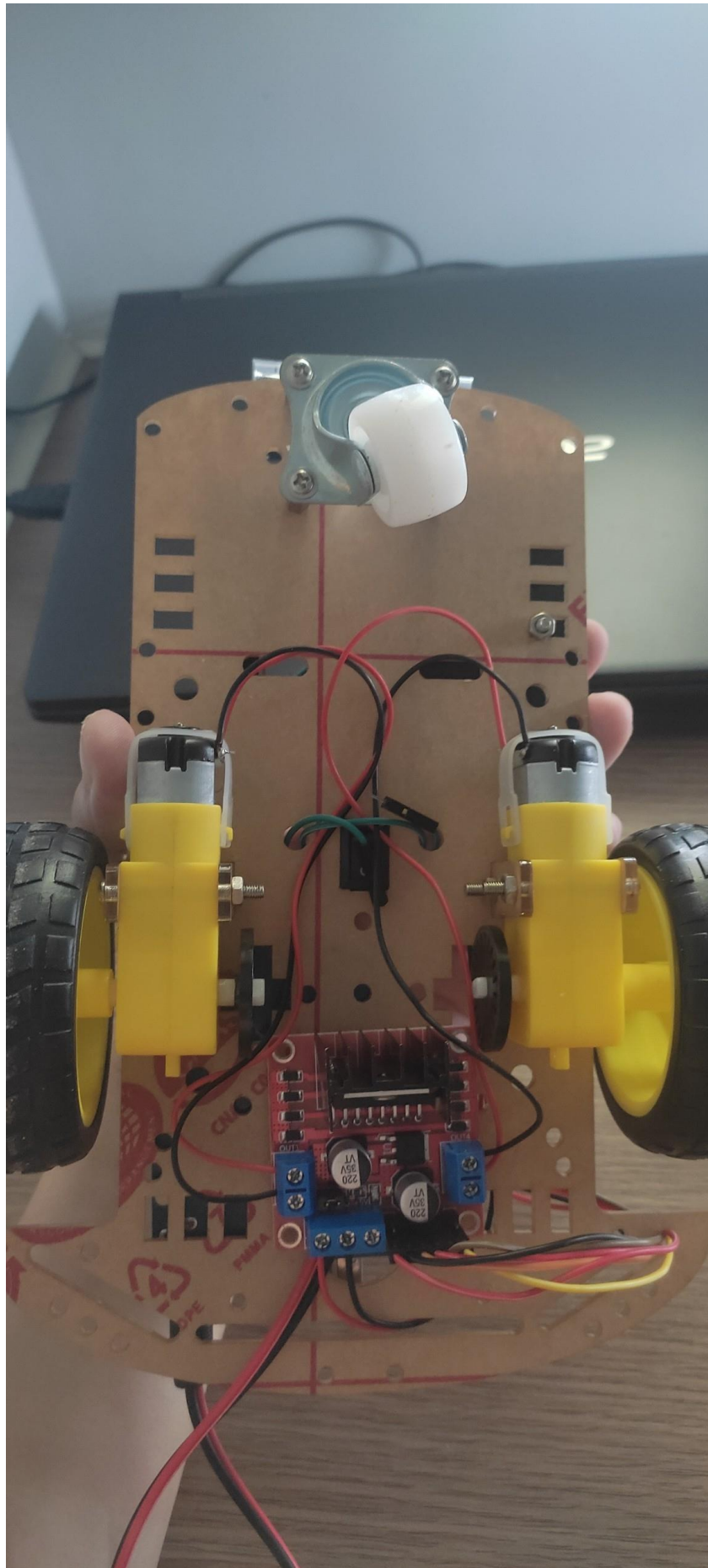
- Lắp ráp xe mô hình cần đảm bảo các bộ phận nằm đúng vị trí, xe có thể đứng vững và đảm bảo trọng tâm xe cần phải nằm ở giữa.
- Cần lắp pin để đảm bảo xe được cung cấp đủ năng lượng để hoạt động, đảm bảo việc xe không quá chậm hoặc quá nhanh, tránh gây sai sót khi đọc dữ liệu từ cảm biến.
- Cảm biến cần đặt ở đầu xe, được cố định và có thể dễ dàng đọc được khoảng cách chính xác giữa xe và vật cản trước mặt.
- Cần thiết kế dây nối giữa các linh kiện được gọn gàng, tránh ảnh hưởng đến quá trình di chuyển của xe và đảm bảo độ thẩm mỹ

- **Phân tích thiết kế**

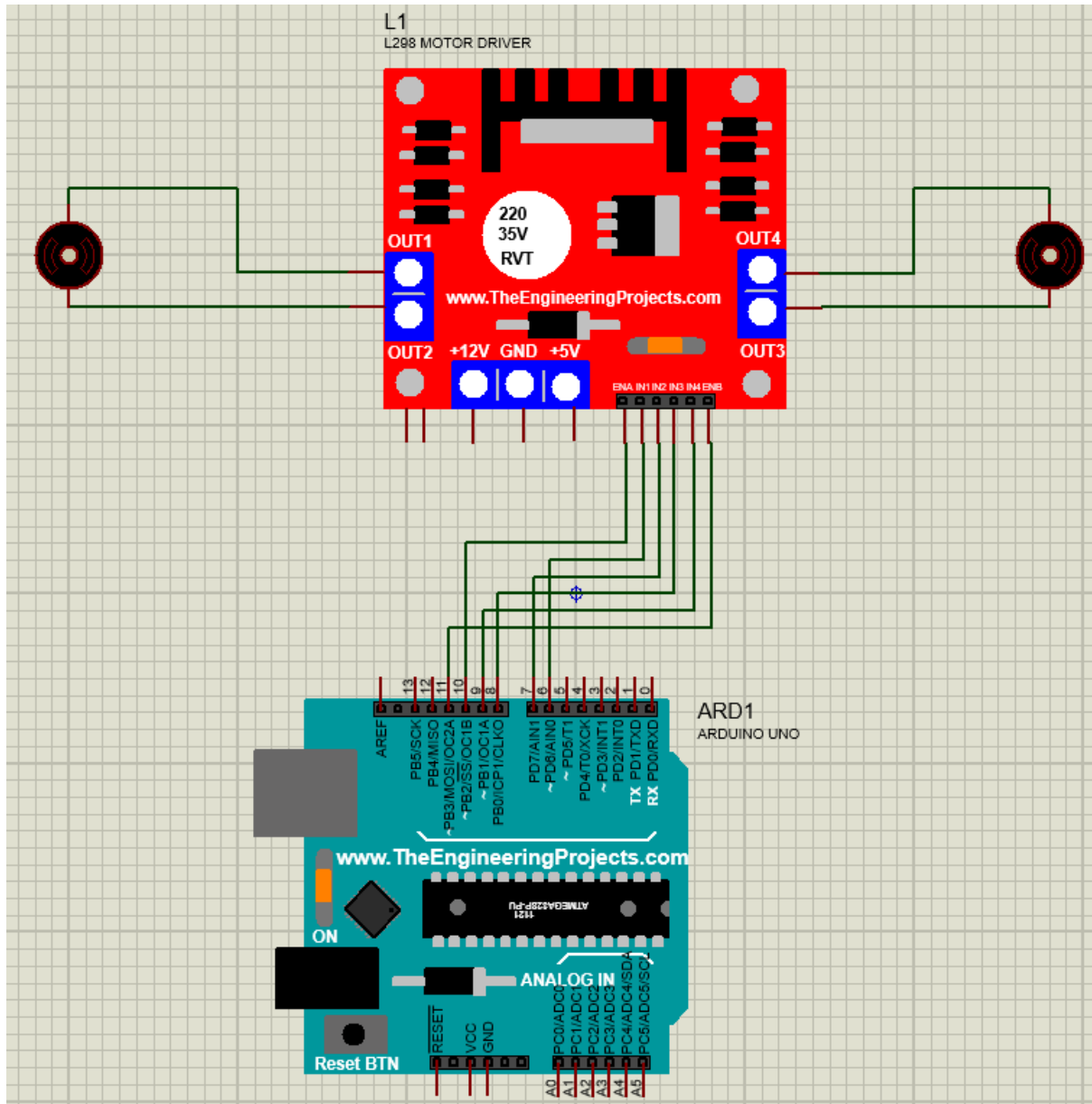
- Phân tích rõ cách thức dẫn đến phương pháp thiết kế từ yêu cầu đã đặt ra
- Từ những yêu cầu đã nêu ra ở trên, em đã thực hiện lắp ráp xe như sau:



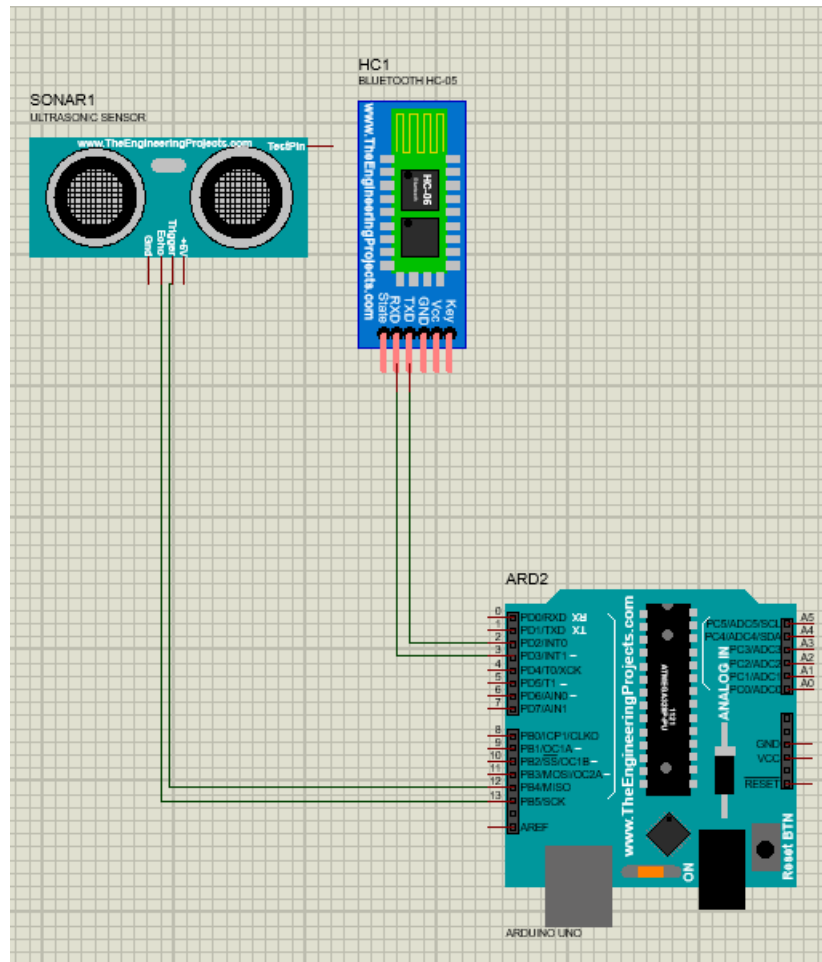




- Em thi công xe với mặt trên đặt cảm biến siêu âm HC-SR04, 2 pin 18650 3.7V, module bluetooth HC-05 và mạch Arduino. Mặt sau thực hiện việc lắp bánh xe cũng như module điều khiển xe L298. Việc đặt L298 ở dưới nhằm mục đích giải phóng không gian phía mặt trên xe. Việc đặt 2 viên pin 18650 ở đầu xe giúp cho xe có độ cân bằng, tránh tình trạng xe “bốc đầu” khi di chuyển.
- Mặt dưới của xe:



- Mặt trên của xe:



4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

- **Yêu cầu đặt ra cho phần mềm**

- Các yêu cầu đặt ra khi thực hiện đồ án về hệ thống dừng khẩn cấp khi có vật cản của xe điều khiển bluetooth bao gồm:

- + Độ trễ khi thao tác điều khiển xe trên điện thoại cho đến lúc xe chạy không được vượt quá 0.5s.

- + Thông tin về khoảng cách giữa xe và vật cản phải được liên tục cập nhật, tối thiểu 2 lần trong 1s để tránh việc xe không phanh kịp khi có vật cản.

- + Cần phải giảm thiểu việc bỏ lỡ tín hiệu điều khiển do mạch Arduino đang trong quá trình phát xung và đọc thông tin khoảng cách giữa và vật cản. Nếu bỏ lỡ tín hiệu điều khiển được gửi từ điện thoại xuống, xe có thể trong tình trạng đi liên tục hoặc đứng im không theo yêu cầu của người điều khiển.

- + Khi thực hiện việc bấm xung trên 2 chân ENA và ENB trên L298, cần lựa chọn giá trị bấm xung phù hợp để đảm bảo xe có thể di chuyển với tốc độ không quá nhanh hoặc quá chậm để đảm bảo việc đọc thông tin khoảng cách và dừng xe được thực hiện trước khi va chạm.

- + Người lập trình cũng cần chọn giá trị bấm xung phù hợp cho mỗi 1 động cơ. Mỗi 1 động cơ có thể sẽ cung cấp 1 tốc độ khác nhau khi bấm cùng 1 giá trị xung, do đó người dùng cần thử nghiệm nhiều lần để tìm ra giá trị phù hợp.

- + Khi xe dừng do có vật cản trước mặt, chương trình cần yêu cầu người dùng có thao tác xác nhận để có thể điều khiển xe sau đó.

- + Sau khi đã có thao tác xác nhận việc xe dừng do có vật cản, chương trình sẽ chỉ cho người dùng điều khiển xe đi lùi hoặc xoay phải, xoay trái. Xe chỉ có thể đi thẳng khi cảm biến siêu âm HC-SR04 xác nhận được việc xe đã ra khỏi khoảng cách nguy hiểm với vật cản. Trong chương trình, em đặt khoảng cách xe cần thiết lập lại với vật cản là 27cm.

- **Phân tích**

- Khi thực hiện phần mềm cho đề tài này, các bước cần làm để xe có thể có hoạt động đúng yêu cầu bao gồm các hàm sau:

- + *Hàm nhận tín hiệu từ chương trình điều khiển:* Khi người điều khiển ấn lệnh điều khiển trên chương trình, kit Arduino sẽ đọc dữ liệu được gửi tới

module Bluetooth HC-05, nhận về kí tự mà người dùng muốn xe chạy theo hướng nào từ đó gọi hàm điều khiển xe tương ứng.

+ *Hàm điều khiển xe*: Sau khi nhận được tín hiệu từ phần mềm trên điện thoại, dựa vào tín hiệu đó Arduino sẽ gọi các hàm điều khiển xe để có thể điều khiển cho động cơ chạy theo ý người sử dụng. Trong chương trình, em quy định các kí tự tương ứng với cách xe chạy như sau:

Ký tự	Điều khiển xe
S	Dừng
F	Đi thẳng
B	Đi lùi
L	Quay trái
R	Quay phải
H	Lùi trái
J	Lùi phải
I	Thẳng phải
G	Thẳng trái
X	Bật cảnh báo khẩn cấp
x	Tắt cảnh báo khẩn cấp

+ *Hàm đọc khoảng cách từ cảm biến siêu âm HC-SR04*: Trong hàm này ta cần lập trình để HC-SR04 phát ra sóng siêu âm từ chân Trig và thu lại được sóng từ chân Echo. Trong Arduino đã cung cấp sẵn câu lệnh pulseIn nhằm có thể giúp được độ dài của xung HIGH trên chân Echo, từ đó có thể đo được khoảng cách giữa xe và vật cản.

+ *Hàm khử nhiễu khi đọc khoảng cách từ cảm biến siêu âm HC-SR04*: Việc xảy ra nhiễu dẫn đến khoảng cách giữa xe không chính xác là thường xuyên do đó ta cần 1 hàm giúp tránh được ảnh hưởng của nhiễu trong môi trường cũng như của thiết bị. Trong phạm vi đồ án 1, em dùng phương pháp lọc nhiễu trung bình động. Phương pháp này dựa trên việc lấy trung bình các giá trị liên tiếp nhau, từ đó có thể tránh việc nhiễu ảnh hưởng tới quá trình điều khiển xe. Em chọn việc lấy trung bình của 3 giá trị liên tiếp nhau. Việc lấy trung bình của quá nhiều số cũng có thể làm ảnh hưởng tới thời gian thực của việc chạy xe.

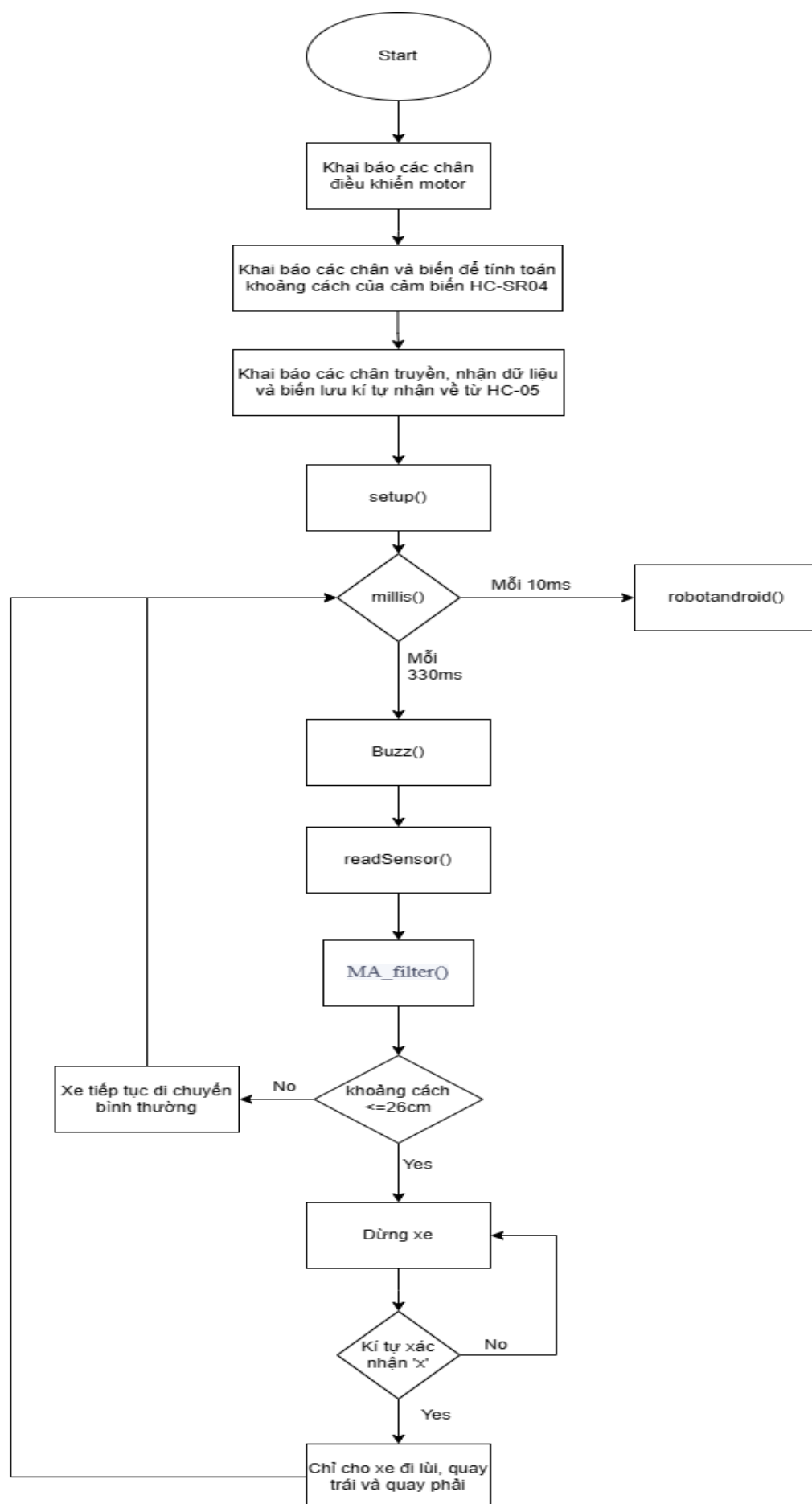
+ *Hàm cảnh báo khoảng cách*: Khi Arduino nhận được thông tin khoảng cách sau khi đi qua hàm khử nhiễu, dữ liệu khoảng cách sẽ được đưa vào hàm có tên gọi Warning. Trong hàm này, dữ liệu khoảng cách sẽ được kiểm tra xem có nằm trong vùng nguy hiểm không. Nếu khoảng cách bé hơn 26cm, chương trình sẽ lập tức dừng xe. Ngược lại nếu xe nằm ngoài vùng nguy hiểm, xe sẽ tiếp tục di chuyển theo ý người điều khiển.

Trong vùng khoảng cách bé hơn 26cm, chương trình sẽ yêu cầu người dùng thao tác xác nhận nguy hiểm trên app điện thoại. Nếu không nhận được ký tự xác nhận, xe vẫn sẽ tiếp tục đứng im. Tuy nhiên nếu nhận được ký tự xác nhận, xe sẽ chỉ cho người dùng lùi xe, quay trái, quay phải cho đến khi cảm biến xác nhận được khoảng cách giữa xe và vật cản an toàn (lớn hơn 26cm)

+ *Millis()*: Millis() có nhiệm vụ trả về một số - là thời gian (tính theo mili giây) kể từ lúc mạch Arduino bắt đầu chương trình của bạn. Nó sẽ tràn số và quay số 0 (sau đó tiếp tục tăng) sau 50 ngày.

Trong chương trình, millis() có nhiệm vụ gọi hàm đọc cảm biến khoảng cách mỗi 330ms và hàm vận hành xe mỗi 10ms. Cần làm điều đó để tránh việc xe bị bỏ lỡ ký tự điều khiển khi trong trạng thái đọc khoảng cách.

+ *Lưu đồ giải thuật*



+ *Source code*

```
#include <SoftwareSerial.h>
int bluetoothTx = 2; // định nghĩa chân 2 là chân truyền tín hiệu
int bluetoothRx = 3; // định nghĩa chân 3 là chân nhận tín hiệu
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

/*-----*/

/*-----*/

//định nghĩa chân motor trái
#define enA 11
#define in1 6
#define in2 7

//định nghĩa chân motor phải
#define enB 10
#define in3 8
#define in4 9

/*-----*/
#define LENGTH 3
const int Trig = 12; // chân trig của HC-05
const int Echo = 13; // chân echo của HC-05
int f = 0; //biến phục vụ việc lưu các biến khoảng cách để đem vào bộ
lọc
int input[LENGTH] = {}; //hàm lưu các biến khoảng cách để đem vào bộ
lọc
int Distance; // biến lưu khoảng cách trước khi lọc
int output=1000; //biến lưu khoảng cách sau khi
lọc
/*-----*/

char blue='S';// là vùng nhớ để so sánh kí tự trả về từ app.
```

```
char sos='y'; //kí tự báo có vật cản
const int spd=130; // giá trị bấm xung( tay ga).
bool isForward = true; // biến trạng thái cho phép đi thẳng

/*-----*/
unsigned long previousMillis_firstBuzz = 0; // lưu trữ thời gian trước đó của
hàm firstBuzz
unsigned long previousMillis_robot_android = 0; // lưu trữ thời gian trước đó
của hàm robot_android

void setup() {
  bluetooth.begin(115200);
  bluetooth.print("$$$");
  delay(100);
  bluetooth.println("U,9600,N");
  bluetooth.begin(9600);
  Serial.begin(9600);

  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  pinMode(Trig, OUTPUT); // chân trig sẽ phát tín hiệu
  pinMode(Echo, INPUT); // chân echo sẽ nhận tín hiệu
}

/*****
*****/
/*****
*****/

int Buzz() {
  unsigned long Duration; // biến đo thời gian
  //digitalWrite(firstTrig, 0); // tắt chân trig
  //delayMicroseconds(2);
  digitalWrite(Trig, 1); // phát xung từ chân trig
```

```
delayMicroseconds(3); // xung có độ dài 4 microseconds
digitalWrite(Trig, 0); // tắt chân trig
Duration = pulseIn(Echo, HIGH);
Distance = int(Duration / 2 / 29.412);
return Distance;
}
```

```
int MA_filter(int input[]) { //lọc tín hiệu nhiễu
    int sum = 0;
    for (int i = 0; i <= LENGTH; i++) {
        sum = sum + input[i];
    }
    output = sum / LENGTH;
    return output;
}
```

```
void readSensor(){
    for (int i = 0; i <= LENGTH; i++) {
        Buzz();
        input[f] = Distance;
        f++;
        delay(4);
    }
    MA_filter(input);
    f = 0;
}
```

```
/*liên quan để điều khiển xe*/
// dieu khien motor trai
void motor_left(int spd, byte dir)
{
    if (dir == 0) // motor trái quay tới
    {
        digitalWrite(in2, LOW);
        digitalWrite(in1, HIGH);
    }
}
```

```
else if (dir == 1) // motor trái quay lùi
{
    digitalWrite(in2, HIGH);
    digitalWrite(in1, LOW);
}
else if (dir == 2) // motor trái đứng yên
{
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
}
analogWrite(enA, spd ); // băm xung cho motor trái
}

// dieu khien motor phai
void motor_right(int spd, byte dir)
{
    if (dir == 0) // motor phải quay tới
    {
        digitalWrite(in4, LOW);
        digitalWrite(in3, HIGH);
    }
    else if (dir == 1) // motor phải quay lùi
    {
        digitalWrite(in4, HIGH);
        digitalWrite(in3, LOW);
    }
    else if (dir == 2) // motor phải đứng lùi
    {
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);
    }
    analogWrite(enB, spd );
}

void forw () // chạy thang
{
```

```
//Serial.println("motor đi thang");
    motor_right(spd-1 , 0);
    motor_left( spd , 0);
}

void back () // chạy lùi
{
    //Serial.println("motor đi lùi");
    motor_right(spd-1 , 1);
    motor_left( spd , 1);
}

void right () // quay phải
{
    //Serial.println("motor của phải");
    motor_right(spd , 0);
    motor_left( spd , 2);
}

void left () // quay trái
{
    //Serial.println("motor của trái");
    motor_left( spd , 0);
    motor_right( spd , 2);
}

void leftforw () // quay trái chậm
{
    //Serial.println("motor của trái chậm");
    motor_left( spd-20 , 0);
    motor_right( spd , 0);
}

void rightforw () // quay phải chậm
{
    //Serial.println("motor của phải chậm ");
    motor_left( spd , 0);
    motor_right( spd-20 , 0);
}
```

```
}  
void rightback () // lui phải cham  
{  
    //Serial.println("motor lui phải cham ");  
    motor_left( spd , 1);  
    motor_right( spd-20 , 1);  
}  
void leftback () // lui trái cham  
{  
    //Serial.println("lui trái cham ");  
    motor_left( spd-20 , 1);  
    motor_right( spd , 1);  
}  
void tack()  
{  
    motor_left( spd , 2);  
    motor_right( spd , 2);  
}  
  
void robot_android(){  
    if ( bluetooth.available() )  
    {  
        blue =bluetooth.read();  
        Serial.println("blue là ");  
        Serial.println(blue);  
    }  
    if(isForward==true){  
        switch (blue){  
            case ('S'):  
                tack();  
                break;  
            case ('F'):  
                forw();  
                break;  
            case ('B'):  
                back();
```



```
break;
case ('L'):
left();
break;
case ('R'):
right();
break;
case ('H'):
leftback();
break;
case ('J'):
rightback();
break;
case ('I'):
rightforw();
break;
case ('G'):
leftforw();
break;
}
}
if (isForward==false){
switch (blue){
case('F'):
break;
case ('S'):
tack();
break;
case ('B'):
back();
break;
case ('H'):
leftback();
break;
case ('J'):
rightback();
```

```
        break;
    }
}

void warning() {
    if (/*firstDistance*/ output <= 26) {
        //Serial.println("SOS_1 1");
        if(isForward==true){
            while(sos!='x'){
                sos =bluetooth.read();
                tack();
            }
            sos='y';
        }
        isForward = false;
    }
    if(/*firstDistance*/ output >= 27){
        //Serial.println("SOS_1 0");
        isForward = true;
    }
}

void loop() {
    /**Xử lý tín hiệu trên cảm biến HC 05*/
    //delay(2);

    unsigned long currentMillis = millis(); // lấy thời gian hiện tại
    // thực hiện hàm robot_android với chu kì 200ms
    if (currentMillis - previousMillis_robot_android >= 10) {
        robot_android(); // gọi hàm điều khiển xe
        previousMillis_robot_android = currentMillis; // cập nhật lại thời gian trước
        đó
    }
    // thực hiện hàm firstBuzz với chu kì 100ms
    if (currentMillis - previousMillis_firstBuzz >= 330) {
```

```

//firstBuzz(); // gọi hàm đọc cảm biến
readSensor();
warning();
previousMillis_firstBuzz = currentMillis; // cập nhật lại thời gian trước đó
}

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

```

5. KẾT QUẢ THỰC HIỆN

- Kết quả thực hiện**

Lần thực hiện	Khoảng cách ban đầu (cm)	Khoảng cách sau khi dừng (cm)
1	30	15
2	50	10
3	80	4
4	100	11

- Giải thích và phân tích về kết quả thu được**

Sau khi thực hiện 4 lần chạy thực tế của xe, ta có thể rút ra 2 điều chính như sau:

- Khoảng cách thực tế xe dừng lại hoàn toàn bé hơn 26cm - khoảng cách em cài đặt trong phần mềm để xe dừng lại khi phát hiện vật cản. Điều này có thể được giải thích bằng việc do xe có quán tính khi di chuyển cũng như có độ trễ khi Arduino thực hiện xử lý các tác vụ. Ước tính thời gian trễ khoảng 1s.
- Trong 4 lần đo trên, xuất hiện 1 lần mà khoảng cách dừng có sự khác biệt hoàn toàn so với 3 lần còn lại. Xe dừng lại hoàn toàn khi khoảng cách chỉ còn 4cm có thể được giải thích là khi phát hiện vật cản trong khoảng cách 26cm, chương trình Arduino đang thực hiện tác vụ nhận tín hiệu từ bluetooth. Tuy xác suất xảy ra chỉ khoảng 20% nhưng đây cũng là hạn chế lớn trong đồ án em thực hiện.

6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Sau khi thực hiện đề tài “thiết kế hệ thống cảm biến hỗ trợ người lái”, em rút ra được 1 số ưu điểm và khuyến điểm của đề tài như sau:

- **Ưu điểm:**

- Đây là đề tài mang tính thực tế cao, việc sinh viên được tận tay lắp ráp xe, lắp cảm biến cũng như chạy mô phỏng mang đến sự trực quan nhất cho sinh viên.
- Là 1 đề tài giúp cho sinh viên hiểu thêm về việc sử dụng cảm biến HC-SR04 nói riêng cũng như tất cả các loại cảm biến nói chung. Am hiểu cách dùng các loại cảm biến là bước đệm quan trọng để thực hiện các sản phẩm điện tử trong tương lai.
- Sinh viên nắm được cách tương tác giữa điện thoại và xe qua thiết bị bluetooth HC-05. Điều khiển từ qua bluetooth, wifi hay bất kì phương thức nào khác sẽ là tương lai của rất nhiều thiết bị điện tử trong thời gian sắp tới.
- Đồ án dành cho sinh viên có niềm đam mê với các công nghệ được áp dụng trên xe hơi, khi xu hướng thế giới đang là xe tự vận hành cũng như áp dụng rất nhiều tính năng cần thiết hỗ trợ vận hành. Sản phẩm tuy nhỏ gọn nhưng là dụng cụ mô phỏng hữu ích khi sinh viên muốn thực hiện các đề tài khác mô phỏng công nghệ trên xe hơi.
- Nắm được cách sử dụng Arduino cũng là 1 trong những ưu điểm của đề tài này. Arduino sẽ là nền tảng để em tiến tới sử dụng những kit có độ phức tạp cao hơn như STM32, Tiva,...

- **Khuyết điểm**

- Độ chính xác của xe chưa được cao. Thời gian xử lý khoảng cách vật cản và dừng xe lại còn chưa được ổn định.
- Thực tế xe hơi sẽ có vận tốc thay đổi liên tục, do đó xe cần thay đổi khoảng cách vật cản liên tục để phù hợp với thực tế. Đây là điều em chưa thực hiện được trên đề tài này.
- Để có thể áp dụng được trong thực tế, xe cần phải có thêm nhiều cảm biến lắp ở đầu xe và 2 bên xi nhan. Cảm biến cần dùng loại có độ nhạy cao hơn và sử dụng kit có độ xử lý cao cũng như làm được nhiều tác vụ cùng lúc.
- Xe lắp ráp không có tốc độ cao được như xe hơi thực tế do đó việc mô phỏng trên xe lắp ráp không được sát như thực tế.

6.2 Hướng phát triển

Sau đề tài này, việc phát triển cảm biến vật cản trước đầu xe hoàn toàn có thể được phát triển để hướng tới việc áp dụng trong thực tế chạy trên đường.

7. TÀI LIỆU THAM KHẢO

- [1] Linh kiện 69, *Làm xe Robot điều khiển bằng Bluetooth*, <https://youtu.be/mq-bmOwWMWc>
- [2] Shop linh kiện điện tử, *Combo tự làm xe 3 bánh bluetooth Arduino*, [Combo tự làm xe 3 bánh bluetooth Arduino - Nshop \(nshopvn.com\)](http://nshopvn.com)