

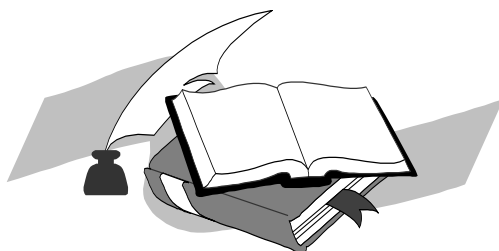
**TRƯỜNG ĐẠI HỌC TÂY NGUYÊN**  
**KHOA KHTN&CN**



**Tiểu luận**

**MÔN: LẬP TRÌNH NHÚNG NÂNG CAO**

**ĐỀ TÀI: ROBOT HÚT BỤI THÔNG MINH**



**Giảng viên hướng dẫn : Ths.Mai Cường Thọ**

**Nhóm Sinh Viên Thực Hiện: Vừ A Cu**

**Cù Thế Truyền**

**Mã Quốc Hùng**

**Phạm Nguyễn Thiên Ân**

**Nguyễn Đức Việt**

**TÂY NGUYÊN**  
**12/2022**

## LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cảm ơn Ban giám hiệu trường Đại Học Tây Nguyên đã tạo điều kiện để sinh viên chúng em có một môi trường học tập thoải mái về cơ sở hạ tầng cũng như cơ sở vật chất. Dưới sự truyền đạt kiến thức và giúp đỡ tận tình của Giảng Viên là hành trang quý báu cho sự nhận thức và hiểu biết thêm của em ngày hôm nay. Em xin ghi nhận nơi đây lòng biết ơn chân thành với tất cả các Thầy Cô Giảng Viên và đặc biệt là cô **ThS.Mai Cường Thọ**, người đã tận tình hướng dẫn em hoàn thành bài tiểu luận này.

Em xin chân thành cảm ơn Lãnh đạo trường Đại Học Tây Nguyên, Khoa Tự Nhiên & Công Nghệ và bộ môn **Lập Trình Nhúng Nâng Cao** đã tạo điều kiện cho em hoàn thành tốt công việc nghiên cứu của mình. Giúp em được mở mang tri thức về môn học này, một môn học hết sức quan trọng, đóng vai trò trong chuyên ngành Công Nghệ Thông Tin. Qua đó em có thể nhận thức một cách đầy đủ và toàn diện về kiến thức và những nội dung liên quan.

Trong quá trình thực hiện bài Tiểu Luận này, em nhận thấy mình đã cố gắng hết sức nhưng **Lập Trình Nhúng Nâng Cao** là môn học rộng lớn và bao quát. Vậy nên, với những kiến thức và thời gian, trong quá trình tìm hiểu sẽ không tránh khỏi nhiều thiếu sót, mong cô cùng toàn thể các thầy cô trong khoa tận tình góp ý và bổ sung để bài luận được hoàn thiện hơn.

**Em xin chân thành cảm ơn!**

*Đắk Lắk, ngày ... tháng ... năm 2022*

NHÓM SINH VIÊN

*(ký, họ tên)*

*Nhóm sinh viên đã ký*

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>1</b>
<b>DANH MỤC CÁC KÝ HIỆU VIẾT TẮT.....</b>	<b>3</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>4</b>
<b>PHẦN I. ĐẶT VẤN ĐỀ.....</b>	<b>5</b>
1. Lý do nghiên cứu.....	5
2. Mục tiêu nghiên cứu .....	5
3. Đối tượng nghiên cứu .....	5
4. Phương pháp nghiên cứu .....	5
5. Dự kiến sản phẩm .....	5
<b>PHẦN II. TỔNG QUAN .....</b>	<b>6</b>
1. Giới thiệu về Arduino Uno R3.....	6
2. L298N-Dual Full Bridge Driver.....	7
3. Bluetooth HC-05.....	9
4. Mit App Inventor .....	10
5. Cảm biến siêu âm HC-SR04 .....	11
6. Giới thiệu chung về phần mềm mô phỏng Proteus .....	13
7. Giới thiệu về Arduino IDE và ngôn ngữ lập trình Arduino.....	13
<b>Phần III. THIẾT KẾ, LẬP TRÌNH VÀ LẮP ĐẶT XE HÚT BỤI THÔNG MINH .....</b>	<b>15</b>
1. Thiết kế mạch trên phần mềm mô phỏng Proteus .....	15
2. Lập trình Code cho Arduino Uno .....	16
3. Lập trình ứng dụng Android trên Mit App Inventor .....	21
4. Lắp ráp mô hình xe hút bụi trên thực tế .....	24
<b>PHẦN IV. KẾT QUẢ .....</b>	<b>26</b>
1. Kết quả trên phần mềm mô phỏng Proteus .....	26
2. Kết quả trên mô hình thực tế.....	26
3. Kết luận.....	26
<b>BẢNG PHÂN CHIA CÔNG VIỆC.....</b>	<b>28</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>29</b>
<b>PHỤ LỤC .....</b>	<b>30</b>
<b>Ý KIẾN CỦA NGƯỜI HƯỚNG DẪN .....</b>	<b>35</b>

## **DANH MỤC CÁC KÝ HIỆU VIẾT TẮT**

<b>Chữ viết tắt</b>	<b>Nguyên nghĩa</b>
GND	Ground
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
Arduino IDE	Arduino Integrated Development Environment
BL	Bluetooth
L298N	L298N-Dual Full Bridge Driver
V <sub>ss</sub>	Virtual Switching System
VCC	Voltage Common Collector

## **DANH MỤC HÌNH ẢNH**

Hình 1.1. Board Arduino Uno R3 .....	6
Hình 1.2. Module L298N .....	7
Hình 1.3. Module Bluetooth HC-05 .....	9
Hình 1.4. Hình ảnh Mit App Inventor.....	10
Hình 1.5. Giao diện khởi động phần mềm Proteus.....	10
Hình 1.6. Giao diện phần mềm Arduino IDE .....	11
Hình 1.7. Mô hình hoạt động của HC-SR04.....	12
Hình 1.8. Giao diện khởi động phần mềm Proteus.....	13
Hình 1.9. Giao diện phần mềm Arduino IDE .....	14
Hình 2.1. Mô hình mạch trên phần mềm Proteus .....	15
Hình 2.2. Giao diện Android trên Mit App.....	22
Hình 2.3. Mã code của ứng dụng Arduino.....	23
Hình 2.4. QR của ứng dụng Android .....	23
Hình 2.5. Sơ đồ mạch lắp ráp trên thực tế .....	24
Hình 2.6. Sơ đồ mạch lắp ráp trên thực tế .....	25
Hình 2.7. Mô hình xe hoàn thiện.....	25
Hình 2.8. Mô hình sản phẩm robot hút bụi thông minh .....	26

## **PHẦN I. ĐẶT VẤN ĐỀ**

### **1. Lý do nghiên cứu**

Hiện nay điện thoại thông minh ngày càng phổ biến, hệ điều hành Android được xây dựng và phát triển liên tục với các chia sẻ về mã nguồn mở, việc sử dụng Smartphone để điều khiển, giám sát thiết bị đang là một xu hướng. Bên cạnh đó, thế giới ngày càng phát triển, tự động hóa ngày càng được nâng cao, mọi công việc đều ngày càng được đưa cho máy móc xử lý. Vì vậy, em quyết định thực hiện đề tài: “**Robot Hút Bụi Thông Minh**” để thiết kế và xây dựng ra con robot tự động làm sạch bụi bẩn trong nhà.

### **2. Mục tiêu nghiên cứu**

Nghiên cứu và thiết kế xe hút bụi điều khiển bằng bluetooth và tự động chạy.

### **3. Đối tượng nghiên cứu**

Sau khi tìm hiểu thông tin về đề tài, cùng với những hiểu biết sẵn có và tìm kiếm thông tin liên quan, em xác định các đối tượng và phạm vi cần nghiên cứu là:

- Module Arduino Uno R3: Cấu tạo, cách thức hoạt động, ứng dụng vào thực tiễn đề tài.
- Nghiên cứu Module Bluetooth HC-05: Các thông số kỹ thuật, nguyên lý hoạt động của module HC-05, ứng vào thực tiễn trong đề tài.
- Phần mềm Mit App Inventor: Thiết kế ứng dụng “motor” chạy trên hệ điều hành Android bằng Mip App Inventor.
- L298 – Dual Full Bridge Driver: Module điều khiển động cơ bánh xe.

### **4. Phương pháp nghiên cứu**

- Phương pháp nghiên cứu tài liệu: Nghiên cứu tài liệu trên Internet.
- Phương pháp thử và sai: Viết code và chạy code.

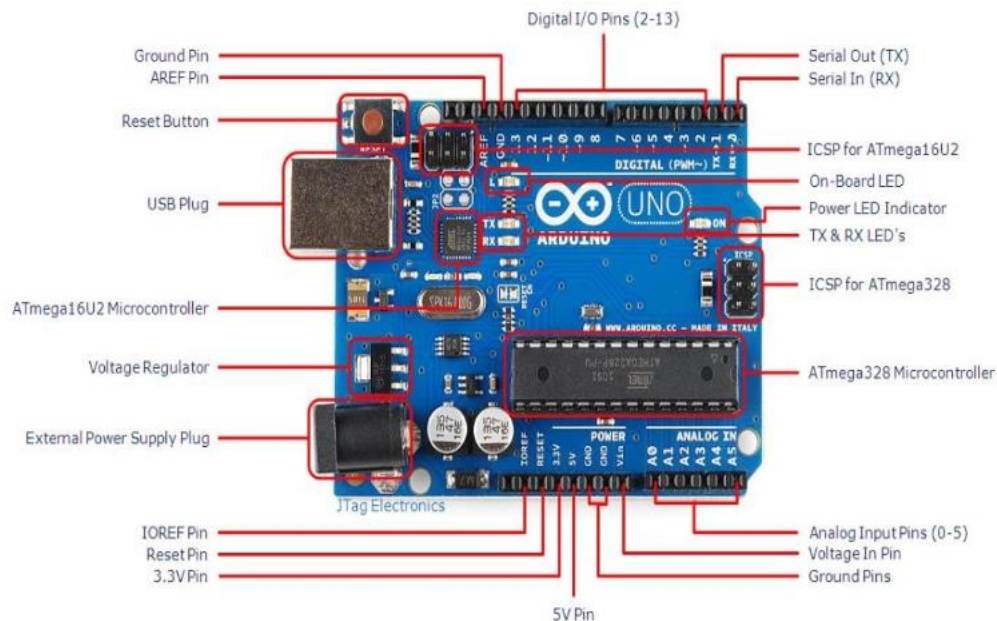
### **5. Dự kiến sản phẩm**

Robot xe hút bụi điều khiển bằng bluetooth và có chế độ tự động chạy bằng cảm biến vật cản.

## PHẦN II. TỔNG QUAN

### 1. Giới thiệu về Arduino Uno R3

Arduino Uno là một bảng mạch vi điều khiển nguồn mở dựa trên vi điều khiển Microchip ATmega328 được phát triển bởi Arduino.cc. Bảng mạch được trang bị các bộ chân Input/Output Digital và Analog có thể giao tiếp với các bảng mạch mở rộng khác nhau. Dựa trên nền tảng mở do Arduino.cc cung cấp các bạn dễ dàng xây dựng cho mình một dự án nhanh nhất (lập trình Robot, xe tự hành, điều khiển bật tắt LED...).



**Hình 1.1 Board Arduino Uno R3**

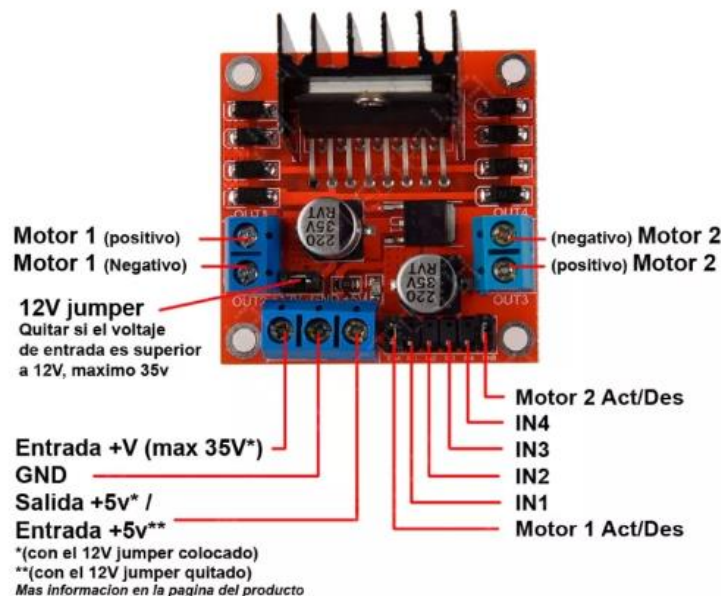
Cấu tạo chính của Arduino Uno bao gồm các phần sau:

- Cổng USB: đây là loại cổng giao tiếp để ta upload code từ PC lên vi điều khiển. Đồng thời nó cũng là giao tiếp serial để truyền dữ liệu giữa vi điều khiển và máy tính.
- Jack nguồn: để chạy Arduino thì có thể lấy nguồn từ cổng USB ở trên, nhưng không phải lúc nào cũng có thể cắm với máy tính được. Lúc đó ta cần một nguồn từ 9V đến 12V.
- Có 14 chân vào/ra số đánh số thứ tự từ 0 đến 13, ngoài ra có một chân nối đất (GND) và một chân điện áp tham chiếu (AREF).

- Vi điều khiển AVR: đây là bộ xử lý trung tâm của toàn bộ mạch. Với mỗi mẫu Arduino khác nhau thì con chip là khác nhau. Ở con Arduino Uno này thì sử dụng ATmega328.
- Các thông số chi tiết của Arduino Uno:
  - + Vi xử lý: Atmega328
  - + Điện áp hoạt động: 5V
  - + Điện áp đầu vào: 7-12V
  - + Điện áp đầu vào (Giới hạn): 6-20V
  - + Chân vào/ra (I/O) số: 14 (6 chân có thể cho đầu ra PWM)
  - + Chân vào tương tự: 6
  - + Dòng điện trong mỗi chân I/O: 40mA
  - + Dòng điện chân nguồn 3.3V: 50mA
  - + Bộ nhớ trong: 32 KB (ATmega328) SRAM: 2 KB (ATmega328) EEPROM: 1 KB (ATmega328)
  - + Xung nhịp: 16MHz

## **2. L298N-Dual Full Bridge Driver**

Đây là Bộ điều khiển động cơ cầu kép phổ biến, thường được sử dụng để điều khiển tốc độ và hướng quay của động cơ. Nó cũng có thể được sử dụng cho các sản phẩm khác như với dây đèn LED, rơ le và ống dẫn điện, v.v. Nó là một trình điều khiển động cơ nhỏ mạnh mẽ với bộ tản nhiệt hạng nặng. Có khả năng cấp nguồn cho động cơ 5-35V với tối đa 2A.



**Hình 1.2. Module L298N**



Cấu tạo chính của L298N:

- Hai chân cấp nguồn trực tiếp đến động cơ là 12V power và 5V power. Ngoài ra cũng có thể cấp nguồn 9-12V ở cổng 12V.
- Power GND là GND của nguồn cấp cho động cơ.
- 2 Jump A enable và B enable.
- 4 chân Input: IN1, IN2, IN3, IN4.
- 4 chân Output: OUT1, OUT2, OUT3, OUT4.

Các thông số của L298N:

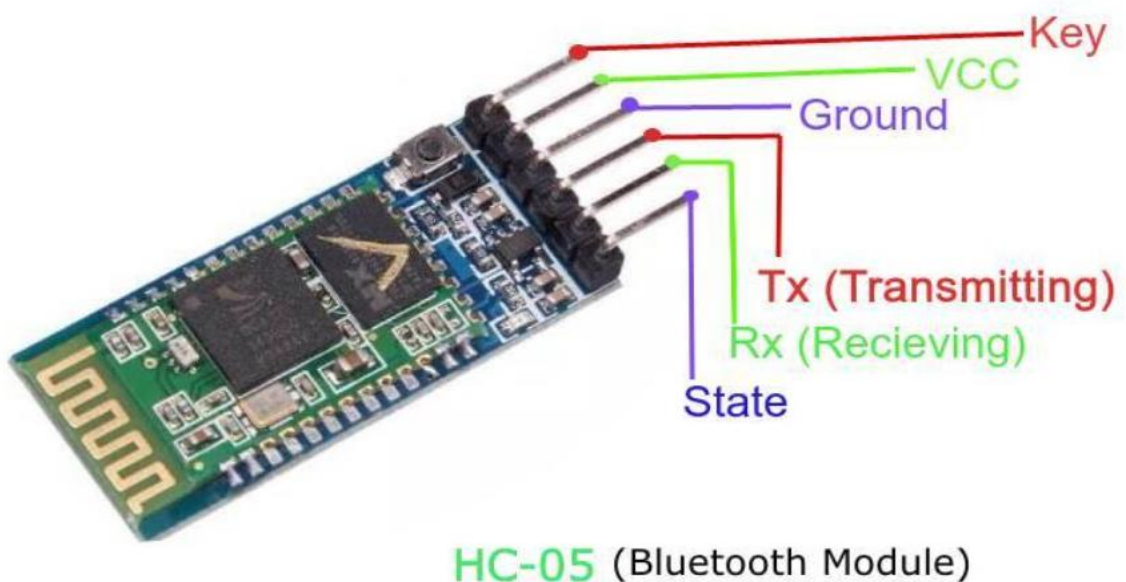
- Điện áp ở đầu vào: 5-30V.
- Điện áp tín hiệu điều khiển: 5-7V.
- Điện áp logic: LOW (0.3-1.5), HIGH (2.3-V<sub>ss</sub>).
- Dòng điện tối đa cho từng động cơ là 2A.
- Dòng yêu cầu của tín hiệu điều khiển từ 0 – 36 mA.
- Công suất: 1 cầu tương đương 25W.
- Kích thước: 43x43x27mm.

### 3. Bluetooth HC-05

Module thu phát Bluetooth HC05 dùng để thiết lập kết nối Serial giữa 2 thiết bị bằng sóng bluetooth. Điểm đặc biệt của module bluetooth HC05 là module có thể hoạt động được ở 2 chế độ: MASTER hoặc SLAVE. Trong khi đó, bluetooth module HC06 chỉ hoạt động ở chế độ 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.



**Hình 1.3. Module Bluetooth HC-05**

#### **Các thông số của Bluetooth HC-05:**

- Điện áp hoạt động: 3.3-5V.
- Mức điện áp chân giao tiếp: 3.3-5V.
- Dòng điện khi hoạt động: nhận kết nối khoảng 30mA, hoạt động bình thường khoảng 8mA.
- Cấu hình hỗ trợ: Cổng nối tiếp Bluetooth (Master/Slave).
- Giao thức Bluetooth: Bluetooth v2.0 + EDR
- Tần số: Băng tần ISM 2.4 GHz.
- Tốc độ: 1 Mbps/1 Mbps.
- Tính năng bảo mật: xác thực và mã hóa.
- Kích thước: 15.2x35.7x5.6mm.

#### 4. Mit App Inventor

App Inventor là công cụ lập trình dành cho mọi người, kể cả trẻ em. Được công bố dưới dạng phần mềm tự do (free software), App Inventor trở thành hiện tượng chưa từng có trong lĩnh vực lập trình cho thiết bị di động. App Inventor thực chất là một ứng dụng web, chạy bởi trình duyệt trên máy tính cá nhân. Tuy nhiên, người dùng vẫn phải cài đặt một phần mềm Java mang tên App Inventor Extras, có nhiệm vụ điều khiển điện thoại Android (kết nối với máy tính thông qua cổng USB). Nhờ vậy, người dùng có thể nhanh chóng chuyển ứng dụng từ máy tính cá nhân qua điện thoại Android để chạy thử.



Hình 1.4: Hình ảnh MIT App Inventor

#### 5. Servo SG90 180 độ

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kỳ từ 0o - 180o. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô hình), có loại thì sở hữu một momen lực bá đạo (vài chục Newton/m), hoặc có loại thì khỏe và không sắc chắc chắn,...



Hình 1.5. Động cơ Servo SG90

Công dụng chính của động cơ servo là đạt được góc quay chính xác trong khoảng từ 90 độ đến 180 độ. Việc điều khiển này có thể ứng dụng để lái robot, di chuyển các tay máy lên xuống, quay một cảm biến để quét khắp phòng...

Cấu tạo của động cơ Servo SG90:

- 1 động cơ quay.
- 3 dây nối: dây màu đen (GND) dùng để nối với GND của mạch, dây màu đỏ (Vcc) dùng để nối với chân cấp điện 5V của nguồn, dây màu cam (PWM) dùng để nối vào chân xung PWM của nguồn để điều khiển quay động cơ.

Các thông số của động cơ Servo SG90:

- Khối lượng : 9g
- Kích thước: 22.2x11.8.32 mm
- Momen xoắn: 1.8kg/cm
- Tốc độ hoạt động: 60 độ trong 0.1 giây
- Điện áp hoạt động: 4.8V(~5V)
- Nhiệt độ hoạt động: 0 °C – 55 °C
- Kết nối dây màu đỏ với 5V, dây màu nâu với mass, dây màu cam với chân phát xung của vi điều khiển. Ở chân xung cấp một xung từ 1ms-2ms theo để điều khiển góc quay theo ý muốn.

## **5. Cảm biến siêu âm HC-SR04**

Cảm biến siêu âm là một thiết bị cảm biến hoạt động dựa trên sóng siêu âm. Cũng giống như các loại cảm biến áp suất hay cảm biến nhiệt độ, cảm biến siêu âm được dùng chủ yếu là để đo khoảng cách hoặc vận tốc. Ngoài ra thì còn được sử dụng trong các ứng dụng như làm sạch bằng sóng siêu âm hoặc dùng trong siêu âm y khoa (siêu âm chuẩn đoán hình ảnh).



Hình 1.6. Cảm biến siêu âm HC-SR04

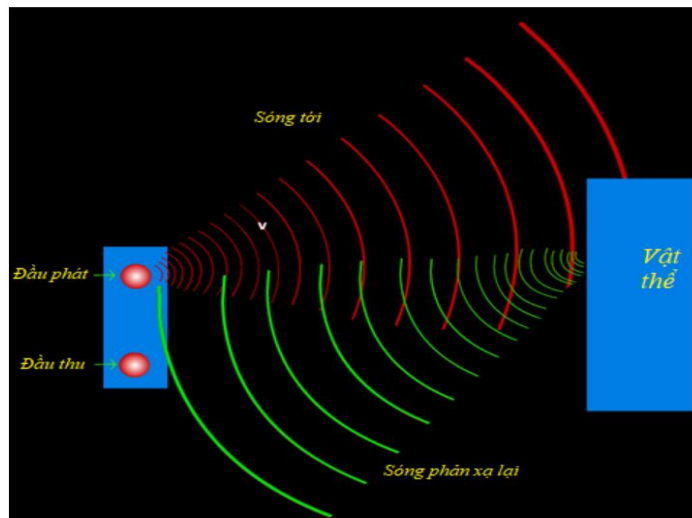
Cấu tạo chính của HC-SR04:

- 2 đầu kính: 1 đầu phát tín hiệu và 1 đầu nhận tín hiệu.
- 4 chân kết nối: chân Vss dùng để nối với nguồn 5V của mạch, chân GND dùng để nối với GND của mạch, chân Trig dùng để gửi tín hiệu đến mạch, chân Echo dùng để nhận tín hiệu từ mạch.

Thông số kỹ thuật của HC-SR04:

- Điện áp hoạt động: 5VDC
- Dòng tiêu thụ: 10~40mA
- Tín hiệu giao tiếp: TTL
- Chân tín hiệu: Echo, Trigger.
- Góc quét: <15 độ
- Tần số phát sóng: 40Khz
- Khoảng cách đo được: 2~450cm (khoảng cách xa nhất đạt được ở điều kiện lý tưởng với không gian trống và bề mặt vật thể bằng phẳng, trong điều kiện bình thường cảm biến cho kết quả chính xác nhất ở khoảng cách <100cm).
- Sai số: 0.3cm (khoảng cách càng gần, bề mặt vật thể càng phẳng sai số càng nhỏ).
- Kích thước: 43mm x 20mm x 17mm

Hoạt động của cảm biến siêu âm: Đầu tiên, đầu cảm biến sẽ phát ra 1 chùm sóng siêu âm xuống bề mặt cần đo khoảng cách. Khi sóng siêu âm gặp bề mặt vật cản sẽ phản xạ ngược lại. Khi đó cảm biến sẽ thu lại các chùm sóng siêu âm này. Dựa vào thời gian phản xạ và vận tốc của sóng, cảm biến sẽ tính ra được khoảng cách từ cảm biến xuống bề mặt vật cản.



Hình 1.7. Mô hình hoạt động của HC-SR04



## 6. Giới thiệu chung về phần mềm mô phỏng Proteus

Phần mềm Proteus là phần mềm cho phép mô phỏng hoạt động của mạch điện tử bao gồm phần thiết kế mạch và viết chương trình điều khiển cho các họ vi điều khiển như MCS-51, PIC, AVR, ... Proteus là phần mềm mô phỏng mạch điện tử của Labcenter Electronics, mô phỏng cho hầu hết các linh kiện điện tử thông dụng, đặc biệt hỗ trợ cho các MCU như PIC, 8051, AVR, Motorola.

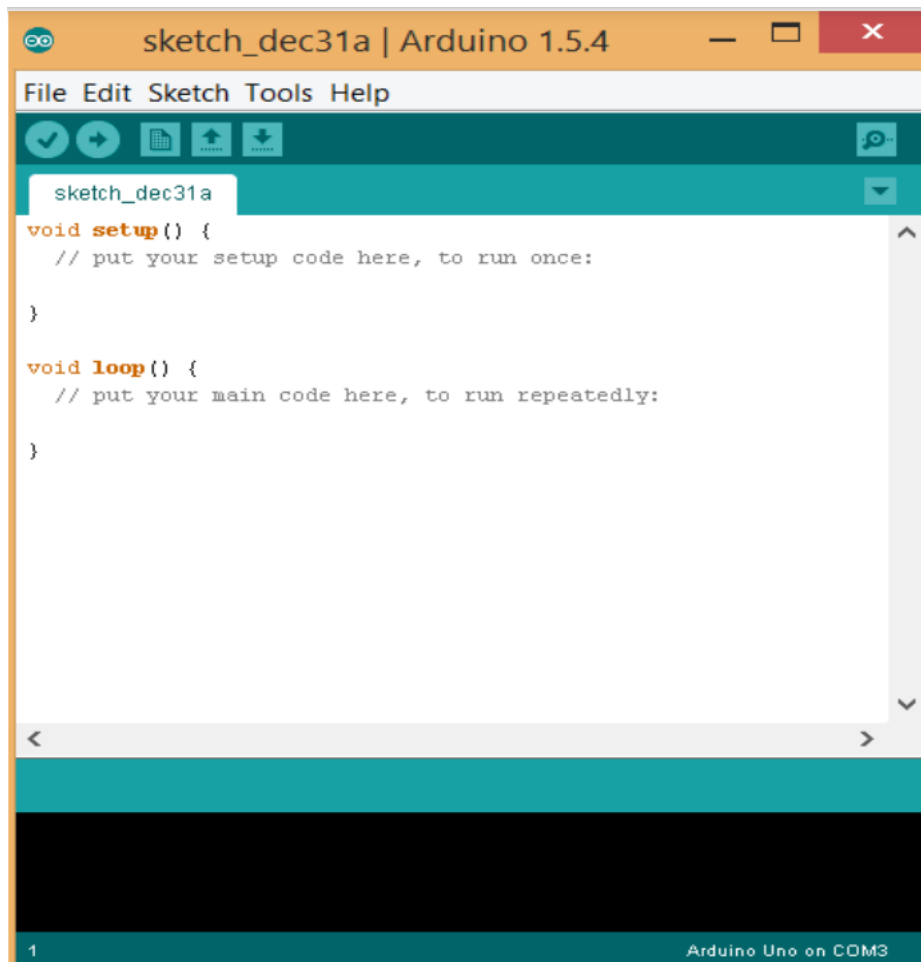
Phần mềm bao gồm 2 chương trình: ISIS cho phép mô phỏng mạch và ARES dùng để vẽ mạch in. Proteus là công cụ mô phỏng cho các loại vi điều khiển khá tốt, nó hỗ trợ các dòng vi điều khiển PIC, 8051, PIC, dsPIC, AVR, HC11,...các giao tiếp I2C, SPI, CAN, USB, Ethernet...ngoài ra còn mô phỏng các mạch số, mạch tương tự một cách hiệu quả.



**Hình 1.8. Giao diện khởi động phần mềm Proteus**

## 7. Giới thiệu về Arduino IDE và ngôn ngữ lập trình Arduino

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.



**Hình 1.9. Giao diện phần mềm Arduino IDE**

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.

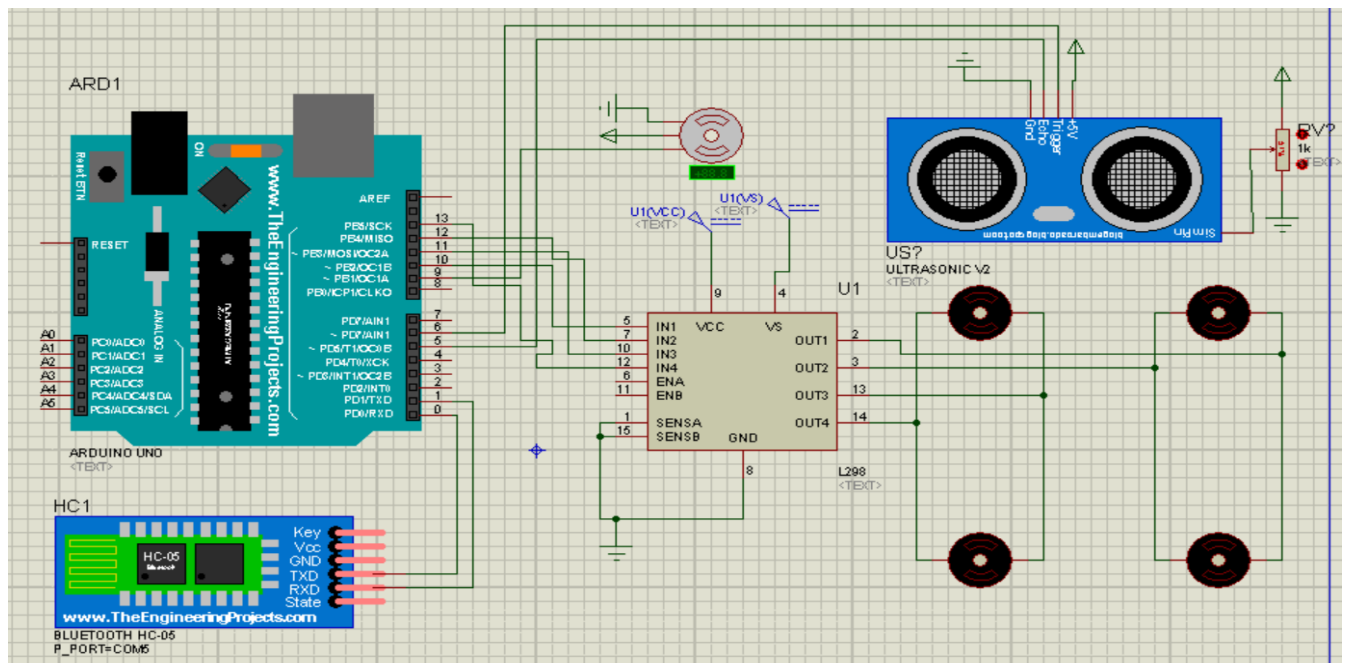
Ngôn ngữ lập trình có thể được mở rộng thông qua các thư viện C++. Và do ngôn ngữ lập trình này dựa trên nền tảng ngôn ngữ C của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR vào chương trình nếu muốn.

## Phần III. THIẾT KẾ, LẬP TRÌNH VÀ LẮP ĐẶT XE HÚT BỤI THÔNG MINH

### 1. Thiết kế mạch trên phần mềm mô phỏng Proteus

Thiết kế mạch trên phần mềm mô phỏng Proteus sử dụng các thành phần chính sau :

- Mạch điều khiển Arduino Uno R3
- Mạch điều khiển động cơ L298
- Module Bluetooth HC-05
- POT-HG
- Ultrasonic V2.0
- MotorPWMServo
- Motor
- Ground
- DC



Hình 2.1. Mô hình mạch trên phần mềm Proteus

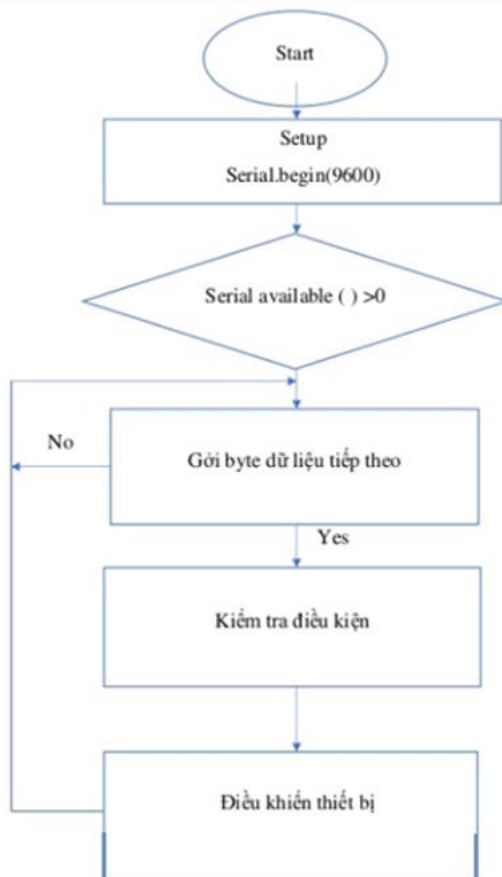


Kết nối các thành phần lại với nhau như sau:

- Chân RXD(0),TXD(1) của Arduino nối lần lượt với chân TXD,RXD của Bluetooth HC-05.
- Chân 5,6 của Arduino nối lần lượt với các chân Echo,Trig của HC-SR04.
- Chân OUT2(3),OUT4(14) của L298 nối với cực dương của motor, chân OUT1(2), OUT3(13) của L298 nối với cực âm của motor.
- Chân VCC(9), VS(4) của L298 nối với 2 DC có điện áp lần lượt là 12V, 5V.
- Chân SENA(1),SENSB(15),GND(8) của L298 nối với 1 Ground.
- Chân 10,11,12,13 của Arduino nối lần lượt với chân IN1(5), IN3(7),IN2(10), IN4(12).
- Một chân ngoài của MotorPWMServo nối vào chân số 9 của Arduino, chân giữa nối với Power, chân còn lại nối với Ground.

## **2. Lập trình Code cho Arduino Uno**

Lưu đồ giải thuật của code Arduino:



Mã code Arduino:

- Khai báo thư viện và các biến sử dụng:

```
#include <Servo.h>
#include <SoftwareSerial.h>
Servo myservo;

int pos = 0;
const int trig = 6;
const int echo = 5;
int tien1 = 10;
int tien2 = 11;
int lui1 = 12;
int lui2 = 13;
int dongcoservo = 9;
int gioihan = 25;//khoảng cách nhận biết vật
int i;
int x = 0;
unsigned long thoigian;
int khoangcach;
int khoangcachtrai, khoangcachphai;
int maxspeed=30;
char bluetooth;
```

- Các hàm con cần trong hệ thống:

```
void tudong();
void dokhoangcach();
void dithang(int duongdi);
void disangtrai();
void disangphai();
void dilui();
void resetdongco();
void quaycbsangphai();
void quaycbsangtrai();
```

- Hàm setup ta thiết đặt các giá trị của chân Arduino thành OUTPUT để xuất dữ liệu ra và mở cổng kết nối Bluetooth cùng với reset trạng thái đầu của xe hút bụi.

```
void setup() {  
    myservo.attach(9);  
    pinMode(trig, OUTPUT);  
    pinMode(echo, INPUT);  
    pinMode(tien1, OUTPUT);  
    pinMode(tien2, OUTPUT);  
    pinMode(lui1, OUTPUT);  
    pinMode(lui2, OUTPUT);  
    digitalWrite(tien1, 0);  
    digitalWrite(tien2, 0);  
    digitalWrite(lui1, 0);  
    digitalWrite(lui2, 0);  
    myservo.write(90);  
    delay(500);  
    Serial.begin(9600);  
}
```

- Hàm loop chứa chương trình kiểm tra kết nối với thiết bị và câu lệnh switch case để điều khiển xe:

- + Nếu số byte (ký tự) đọc từ cổng kết nối lớn hơn 0 thì ta gán biến 'bluetooth' bằng dữ liệu đọc từ cổng kết nối.

```
if (Serial.available() > 0) {  
    bluetooth = Serial.read();  
}
```

- + Sau đó dùng lệnh switch case để kiểm tra dữ liệu biến 'bluetooth' nhận được rồi thiết lập các giá trị để điều khiển xe: nếu biến 'bluetooth'="A" thì xe tự động lái, nếu biến 'bluetooth'="S" thì xe dừng, nếu 'bluetooth'="F" thì xe đi, nếu 'bluetooth'="B" thì xe lùi, nếu 'bluetooth'="R" thì xe quay phải, nếu 'bluetooth'="L" thì xe quay trái.

```
switch (bluetooth) {  
    case 'A':  
        tudong();  
        break;  
    case 'S':  
        digitalWrite(tien1, 0);  
        digitalWrite(lui1, 0);  
        digitalWrite(tien2, 0);  
        digitalWrite(lui2, 0);  
        break;  
    case 'F':  
        // ...  
}
```

```
        digitalWrite(tien1, 1);
        digitalWrite(tien2, 1);
        break;
    case 'B':
        digitalWrite(lui1, 1);
        digitalWrite(lui2, 1);
        break;
    case 'L':
        digitalWrite(tien1, 1);
        //digitalWrite(lui1, 1);
        break;
    case 'R':
        digitalWrite(tien2, 1);
        //digitalWrite(lui2, 1);
        break;
}
```

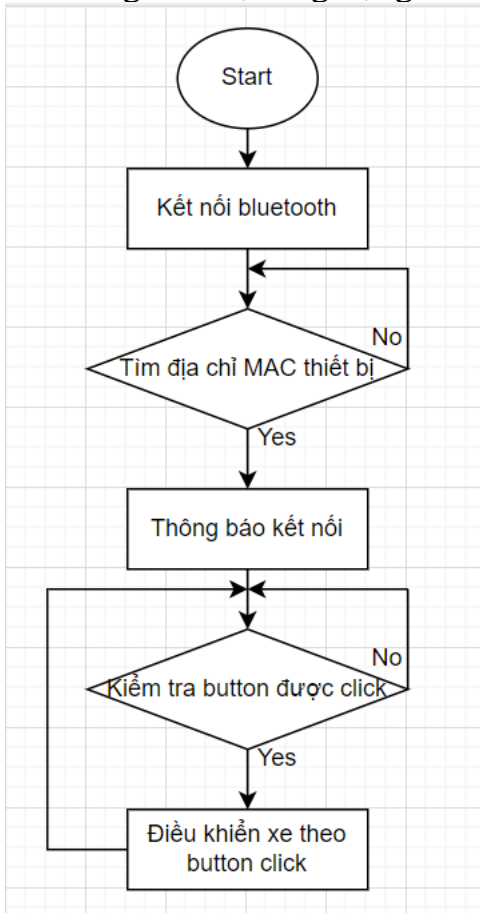
- Hàm con *tudong()*:
  - + Reset khoảng cách bằng 0.
  - + Đo khoảng cách bằng hàm *dokhoangcach()*. Nếu khoảng cách lớn hơn 30cm thì xe đi thẳng. Nếu khoảng cách nhỏ hơn 30 cm thì xe dừng và tiến hành đo khoảng cách 2 bên.
  - + Nếu khoảng cách trái = khoảng cách phải và nhỏ hơn 30 thì xe đi lùi.
  - + Nếu khoảng cách trái > khoảng cách phải và lớn hơn 30 thì xe đi sang trái.
  - + Nếu khoảng cách phải > khoảng cách trái và lớn hơn 30 thì xe đi sang phải.
  - + Lặp lại các bước trên n lần khi xe di chuyển.

```
void tudong(){
    khoangcach = 0;
    dokhoangcach();
    if (khoangcach > gioihan || khoangcach == 0)
    {
        dithang();
    }
    else
    {
        resetdongco();
        quaycbsangtrai();
        dokhoangcach();
        khoangcachtrai = khoangcach;
        quaycbsangphai();
    }
}
```

```
dokhoangcach();
khoangcachphai = khoangcach;
if (khoangcachphai < 30 && khoangcachtrai < 30) {
    dilui();
}
else
{
    if (khoangcachphai >= khoangcachtrai)
    {
        disangphai();
        delay(500);
    }
    if (khoangcachphai < khoangcachtrai)
    {
        disangtrai();
        delay(500);
    }
}
}
```

### 3. Lập trình ứng dụng Android trên Mit App Inventor

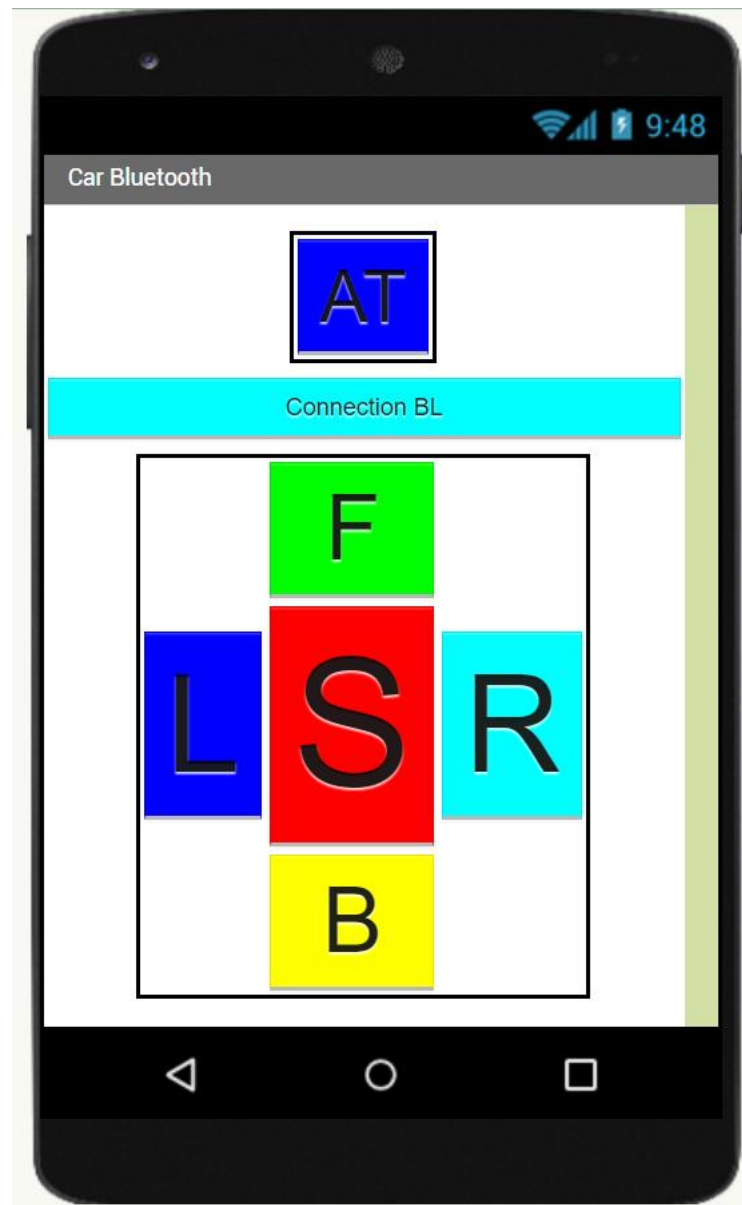
Lưu đồ giải thuật ứng dụng Android:



Thiết kế ứng dụng Android với giao diện đơn giản, dễ hiểu, dễ thao tác với các thành phần cơ bản sau:

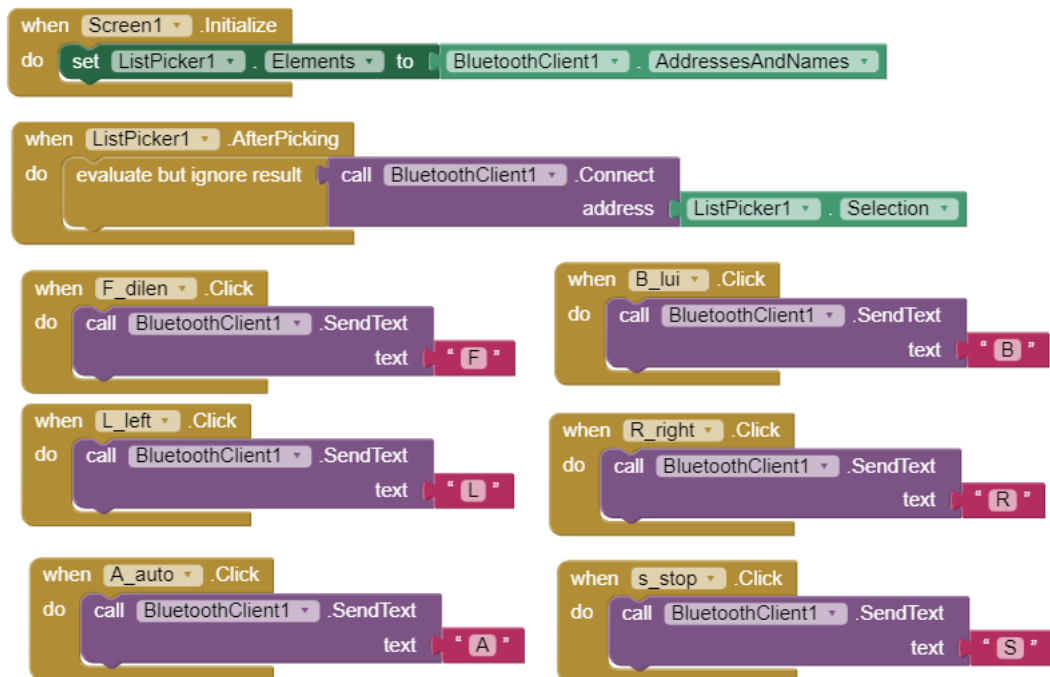
- + 1 listPicker: chứa danh sách các thiết bị kết nối.
- + 6 button: điều hướng xe di chuyển.
- + 1 BluetoothClient: Thiết lập 1 thiết bị bluetooth để kết nối với các thiết bị khác bằng bluetooth.

Giao diện ứng dụng trên Mit App Inventor:



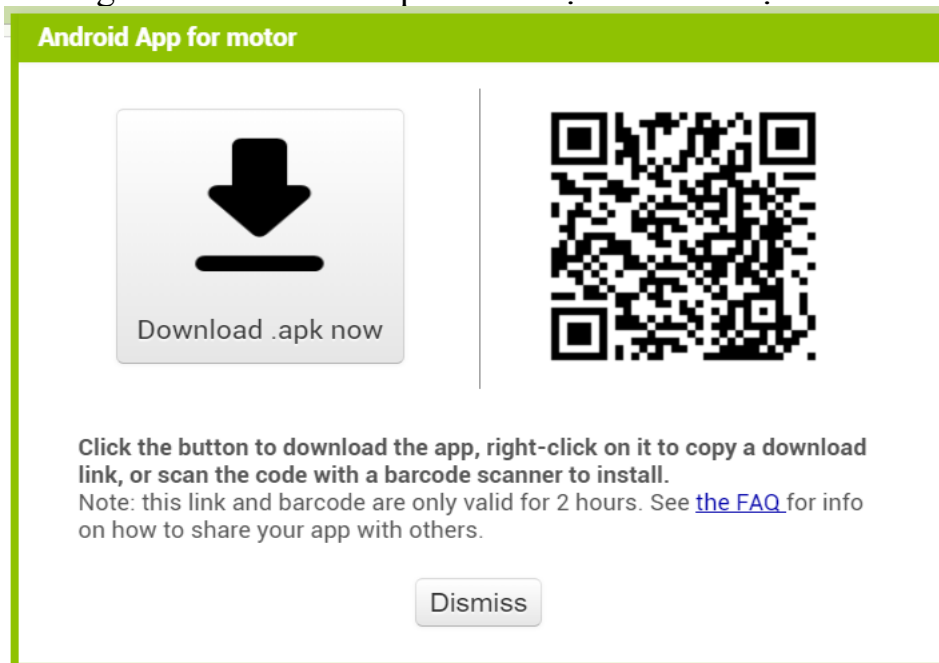
**Hình 2.2. Giao diện Android trên Mit App**

Thực hiện viết code trên Phần mềm Mit App Inventor bằng phương pháp kéo thả, ta được đoạn code sau:



**Hình 2.3. Mã code của ứng dụng Android**

Đóng gói chương trình thành file \*.apk để cài đặt trên thiết bị Android:



**Hình 2.4. QR của ứng dụng Android**



#### **4. Lắp ráp mô hình xe hút bụi trên thực tế**

Linh kiện để lắp ráp mô hình xe gồm:

- + 1 Mạch điều khiển Arduino Uno R3.
- + 1 Mạch điều khiển động cơ L298N.
- + 1 Module Bluetooth HC-05.
- + 5 Motor và 4 bánh xe.
- + 1 cánh quạt.
- + 1 Bảng bìa cứng để làm thân xe.
- + Dây kết nối.



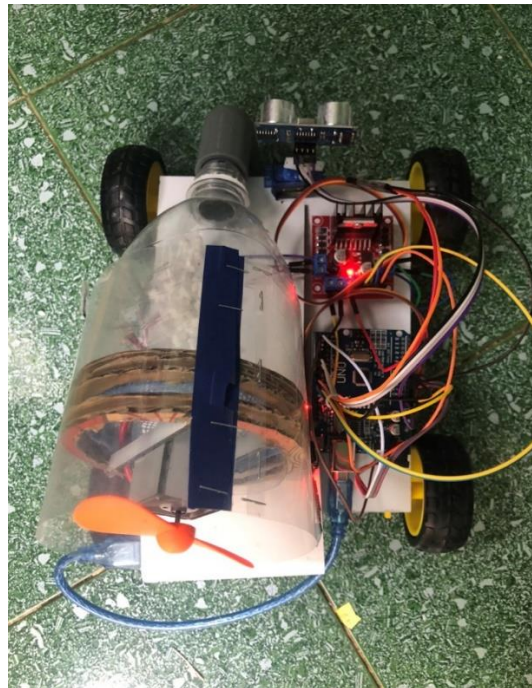
**Hình 2.5. Linh kiện lắp ráp robot hút bụi**

Sơ đồ mạch lắp ráp trên thực tế:

Arduino	Module L298	Servo	SRF-04	Motor robot
	12V			
VIN	5V			
GND	GND	Nâu		
GND			GND	
5V			VCC	
5V		Đỏ		
5			ECHO	
6			TRIG	
9		Cam		
10	IN 1			
11	IN 3			
12	IN 2			
13	IN 4			
	OUT 1			Motor trái
	OUT 2			
	OUT 3			Motor Phải
	OUT 4			

**Hình 2.6. Sơ đồ mạch lắp ráp trên thực tế**

Hoàn thiện lắp ráp mô hình xe trên thực tế:



**Hình 2.7. Mô hình xe hoàn thiện**



## PHẦN IV. KẾT QUẢ

### 1. Kết quả trên phần mềm mô phỏng Proteus

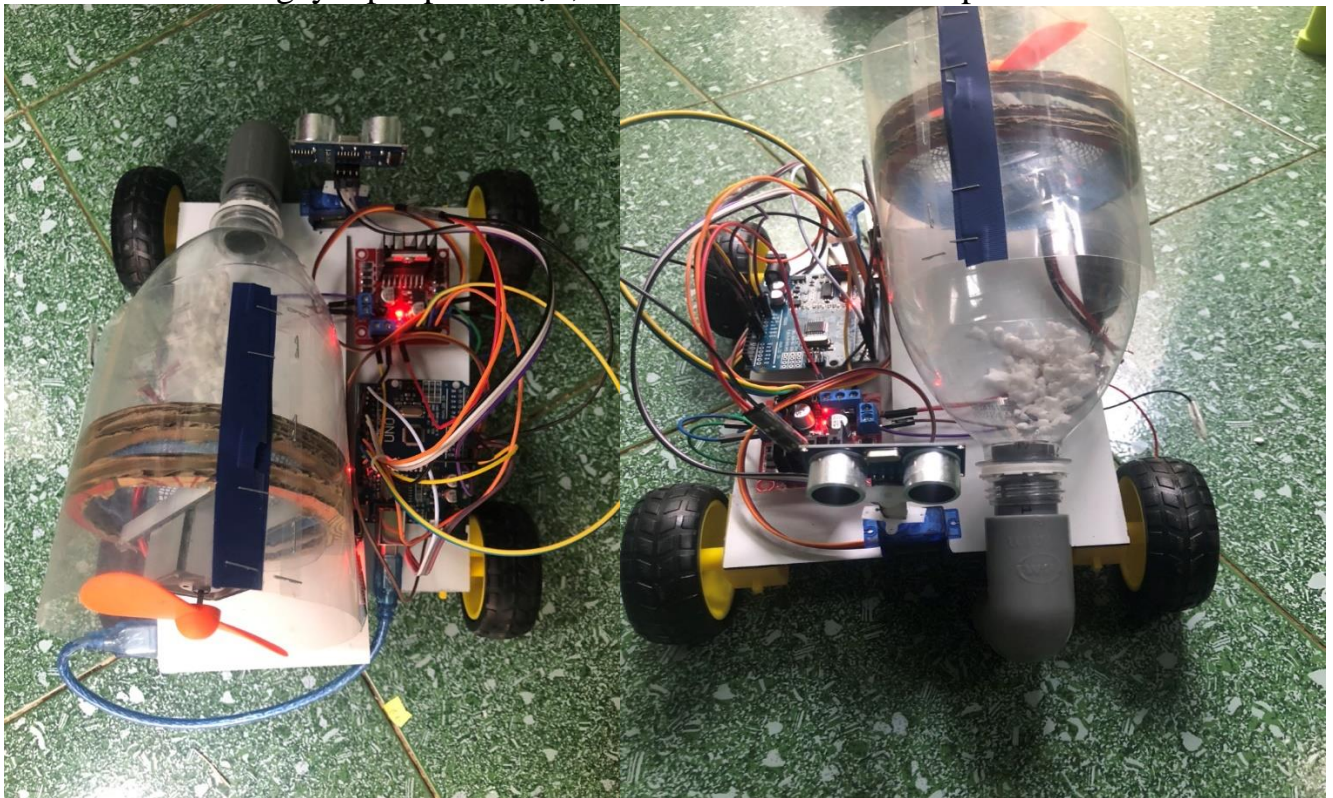
Sau một vài ngày tích cực làm việc nhóm thì nhóm đã hoàn thiện được “mô hình xe hút bụi” chạy trên phần mềm mô phỏng Proteus.

Link demo mô hình trên phần mềm mô phỏng Proteus:

[https://www.youtube.com/watch?v=pu\\_vjbDApjM](https://www.youtube.com/watch?v=pu_vjbDApjM)

### 2. Kết quả trên mô hình thực tế

Sau nhiều ngày lắp ráp tích cực, nhóm đã hoàn thành sản phẩm đề ra.



Hình 2.8. Mô hình sản phẩm robot hút bụi thông minh

Link demo: <https://www.youtube.com/watch?v=daw-oaBgcHM>

### 3. Kết luận

Ưu điểm:

- Tất cả các thành viên đã tham gia làm việc nhóm tích cực, hoàn thành tốt nhiệm vụ được giao.
- Nhóm đã bước đầu hoàn thiện được sản phẩm đáp ứng các chức năng đề ra trước đó.

- Sản phẩm đã hoạt động đúng như dự kiến.

Nhược điểm:

- Do các linh kiện đặt qua shoplonline nên chất lượng chưa đảm bảo, vì thế mà robot chưa thực sự hoạt động trơn tru.
- Do kỹ năng lập trình còn yếu, nên chưa thực sự tối ưu hóa được code Arduino.
- Sắp xếp các linh kiện trên robot còn chưa hợp lý, chưa lắp ráp bộ khung hoàn thiện thành mô hình xe hút bụi hoàn chỉnh.

## **BẢNG PHÂN CHIA CÔNG VIỆC**

Họ Tên	Công việc
Vừ A Cu	<ul style="list-style-type: none"><li>- Nhóm trưởng: phân chia công việc</li><li>- Tổng hợp, kiểm tra, hoàn thiện sản phẩm</li><li>- Viết Code Arduino</li><li>- Hoàn thiện báo cáo</li><li>- Hoàn thiện tiểu luận</li><li>- Hỗ trợ lắp ráp mạch</li></ul>
Phạm Nguyễn Thiên Ân	<ul style="list-style-type: none"><li>- Thiết kế bản mạch mô phỏng proteus</li><li>- Hỗ trợ lắp ráp mạch</li></ul>
Mã Quốc Hùng	<ul style="list-style-type: none"><li>- Thiết kế và lập trình ứng dụng Android trên Mit App Inventor</li><li>- Hỗ trợ lắp ráp mạch</li></ul>
Cù Thế Truyền	<ul style="list-style-type: none"><li>- Viết tiểu luận</li><li>- Hỗ trợ lắp ráp mạch</li></ul>
Nguyễn Đức Việt	<ul style="list-style-type: none"><li>- Viết báo cáo</li><li>- Hỗ trợ lắp ráp mạch</li></ul>

## **TÀI LIỆU THAM KHẢO**

- [1] Auto Ultrasonic car  
<https://create.arduino.cc/projecthub/ishaq-yang/auto-ultrasonic-car-a85c6f>
- [2] Arduino  
<http://arduino.vn/>
- [3] How to make Bluetooth car  
<https://circuitbest.com/how-to-make-bluetooth-controlled-car-using-arduino>
- [4]. Làm xe hút bụi thông minh  
<https://www.youtube.com/watch?v=S-E24CeIO1c&t=511s>
- [5]. Sách Learn Google Flutter Fast: 65 Example Apps tác giả Mark Clow

## PHỤ LỤC

```
#include <Servo.h>
#include <SoftwareSerial.h>
Servo myservo;

int pos = 0;
const int trig = 6;
const int echo = 5;
int tien1 = 10;
int tien2 = 11;
int lui1 = 12;
int lui2 = 13;
int dongcoservo = 9;
int gioihan = 25;//khoảng cách nhận biết vật
int i;
int x = 0;
unsigned long thoigian;
int khoangcach;
int khoangcachtrai, khoangcachphai;
int maxspeed=30;
char bluetooth;

void tudong();
void dokhoangcach();
void dithang(int duongdi);
void disangtrai();
void disangphai();
void dilui();
void resetdongco();
void quaycbsangphai();
void quaycbsangtrai();

void setup() {
    myservo.attach(9);
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
    pinMode(tien1, OUTPUT);
    pinMode(tien2, OUTPUT);
    pinMode(lui1, OUTPUT);
    pinMode(lui2, OUTPUT);
    digitalWrite(tien1, 0);
    digitalWrite(tien2, 0);
    digitalWrite(lui1, 0);
```

```
digitalWrite(lui1, 0);
myservo.write(90);
delay(500);
Serial.begin(9600);
}

void loop()
{
  if (Serial.available() > 0) { //kiem tra ket noi du lieu
    bluetooth = Serial.read(); //bien bluetooth nhan gia tri tu thiet bi bluetooth
    tra ve
    digitalWrite(tien1, 0);
    digitalWrite(lui1, 0);
    digitalWrite(tien2, 0);
    digitalWrite(lui2, 0);
    delay(10);
    switch (bluetooth) {
      case 'A':
        tudong();
        break;
      case 'S':
        digitalWrite(tien1, 0);
        digitalWrite(lui1, 0);
        digitalWrite(tien1, 0);
        digitalWrite(lui2, 0);
        break;
      case 'F':
        digitalWrite(tien1, 1);
        digitalWrite(tien2, 1);
        break;
      case 'B':
        digitalWrite(lui1, 1);
        digitalWrite(lui2, 1);
        break;
      case 'L':
        digitalWrite(tien1, 1);
        //digitalWrite(lui1, 1);
        break;
      case 'R':
        digitalWrite(tien2, 1);
        //digitalWrite(lui2, 1);
        break;
    }
  }
}
```



```
}

// tudong();
// }
void tudong(){
    khoangcach = 0;
    dokhoangcach();
    if (khoangcach > gioihan || khoangcach == 0)
    {
        dithang();
    }
    else
    {
        resetdongco();
        quaycbsangtrai();
        dokhoangcach();
        khoangcachtrai = khoangcach;
        quaycbsangphai();
        dokhoangcach();
        khoangcachphai = khoangcach;
        if (khoangcachphai < 30 && khoangcachtrai < 30) {
            dilui();
        }
        else
        {
            if (khoangcachphai >= khoangcachtrai)
            {
                disangphai();
                delay(500);
            }
            if (khoangcachphai < khoangcachtrai)
            {
                disangtrai();
                delay(500);
            }
        }
    }
}

void dithang()
{

    digitalWrite(tien1, 1);
    digitalWrite(tien2, 1);
}
```

```
}

void disangphai()
{
    resetdongco();
    digitalWrite(lui1, 1);
    delay(2000); //thời gian lùì
    digitalWrite(lui1, 0);

}

void disangtrai()
{
    resetdongco();
    digitalWrite(lui2, 1);
    delay(2000); //thời gian
    digitalWrite(lui2, 0);

}

void dilui()
{
    resetdongco();
    for (i = 0; i < 20; i++)
        digitalWrite(lui1, 1);
        digitalWrite(lui2, 1);
        delay(2000);

    digitalWrite(lui1, 0);
    digitalWrite(lui2, 0);
}

void resetdongco()
{
    digitalWrite(tien1, 0);
    digitalWrite(tien2, 0);
    digitalWrite(lui1, 0);
    digitalWrite(lui2, 0);
}

void dokhoangcach()
{
```

```
digitalWrite(trig, 0);
delayMicroseconds(2);
digitalWrite(trig, 1);
delayMicroseconds(10);
digitalWrite(trig, 0);
// Đo độ rộng xung 1 ở chân echo.
thoigian = pulseIn(echo, 1);

khoangcach = thoigian / 2 / 29.412;

}

void quaycbsangtrai()
{
    myservo.write(180);           // tell servo to go to position in variable 'pos'
    delay(1000);
    dokhoangcach();
    myservo.write(90);           // tell servo to go to position in variable 'pos'
}
void quaycbsangphai()
{
    myservo.write(0);            // tell servo to go to position in variable 'pos'
    delay(1000);
    dokhoangcach();
    myservo.write(90);           // tell servo to go to position in variable 'pos'
}
void resetservo()
{
    myservo.write(90);
}
```

## **Ý KIẾN CỦA NGƯỜI HƯỚNG DẪN**

**Nhận xét:** .....

.....

.....

.....

.....

.....

....., ngày... tháng... năm 2022

**GIÁO VIÊN HƯỚNG DẪN**  
(Ký và ghi rõ họ tên)