

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO
HỆ NHÚNG (IT4210)

ĐỀ TÀI:
Xe điều khiển cải tiến

Nhóm sinh viên thực hiện: **102**

- | | |
|----------------------------|-----------------|
| 1. Nguyễn Duy Khai | 20183771 |
| 2. Nguyễn Thị Trang | 20183843 |

Giảng viên hướng dẫn: **TS. Đỗ Công Thuận**

Hà Nội, tháng 1 năm 2022

MỤC LỤC

MỤC LỤC.....	2
LỜI NÓI ĐẦU	3
I. Giới thiệu về đề tài.....	5
1. Đặt vấn đề	5
2. Mô tả tổng quan đề tài	6
3. Giới thiệu về công nghệ không dây bluetooth.....	6
II. Phân tích hệ thống	8
1. Sơ đồ khối.....	8
2. Vai trò từng khối.....	8
III. Thiết kế hệ thống.....	10
1. Các linh kiện được sử dụng	10
2. Sơ đồ thiết kế mạch điện	14
3. Quy ước dữ liệu điều khiển trao đổi qua bluetooth	15
IV. Tiến hành xây dựng dự án	17
1. Xây dựng ứng dụng điều khiển trên điện thoại (Android)	17
2. Xây dựng thiết bị nhúng (xe điều khiển) và lập trình firmwre.....	20
V. Kết quả thu được	28
VI. Tổng kết	31
PHỤ LỤC.....	33
1. Phân công nhiệm vụ.....	33
2. Tài liệu tham khảo	33
3. Mã nguồn (source code) và các resource của dự án.....	34

LỜI NÓI ĐẦU

. Trong xã hội hiện đại và đặc biệt là những năm gần đây, hẳn ai cũng có thể thấy rõ được sự phát triển của thời đại gắn bó chặt chẽ ra sao với khoa học – kỹ thuật, công nghệ cùng sự bùng nổ của các hoạt động kinh doanh, thương mại toàn cầu. Cách riêng, nói về khoa học – kỹ thuật và công nghệ thì đa số mọi người sẽ nghĩ ngay đến một ngành nghề, một khái niệm mang tên “Công nghệ thông tin” như cách mà mạng xã hội gần đây hay có những câu đùa vui kiểu “IT (Công nghệ thông tin) là vua của các nghề”. Câu nói vui nhưng phần nào phản ánh được vai trò của ngành với thời đại.

Một định hướng mà có thể nói là vừa thú vị, vừa thiết thực trong công nghệ là nghiên cứu, xây dựng và phát triển các hệ thống nhúng (Embedded System) thông minh và IoT (Internet of Things – Internet vạn vật) – những từ khóa không thể thiếu khi nói đến “công nghệ của tương lai” hay “Cách mạng 4.0”. Nói về các hệ thống nhúng, nhất định vẫn còn nhiều người mơ hồ với khái niệm này mà họ không nhận ra được rằng: Những gì mà họ sử dụng, tiếp xúc, bắt gặp trong cuộc sống hằng ngày đa số đều có các hệ thống nhúng. Còn việc khái niệm về hệ thống nhúng là gì, chúng hoạt động ra sao thì ở nội dung này chúng ta tạm bỏ qua một bên, tạm chấp nhận rằng, những người theo dõi nội dung báo cáo này là những người đã có kiến thức nền tảng về công nghệ, về hệ thống nhúng, nếu có người đọc nào thắc mắc về nội dung này, xin vui lòng tìm kiếm thêm thông tin trên Internet – đó sẽ là những nguồn kiến thức có lẽ là sẽ hữu ích cho mọi người trong xã hội ngày càng hiện đại. Nói đến nhúng, đơn giản, trong ô-tô, ti-vi, tủ lạnh, xe bus,... đều có các hệ thống nhúng đã được xây dựng và vận hành bên trong, các hệ thống này giúp cho các thiết bị hoạt động hiệu quả và dễ dàng điều khiển theo ý đồ người sử dụng.

Nhóm 102 bao gồm 2 sinh viên năm 4 đang theo học chương trình đào tạo Công nghệ thông tin – Kỹ thuật máy tính tại Trường Công nghệ thông tin và truyền thông – Đại học Bách Khoa Hà Nội may mắn trong quá trình học tập trên ghế nhà trường được có cơ hội tiếp cận với nội dung kể trên thông qua học phần Hệ Nhúng (IT4210) với những kiến thức, trải nghiệm vô cùng hữu ích và thực tế cho những người phát triển nghề nghiệp theo hướng khoa học – công nghệ. Để củng cố và thực hành một cách trực quan cho những gì thu được trong suốt quá trình học, nhóm lựa chọn xây dựng đề tài “Xe điều khiển cải tiến”, xe điều khiển này dự kiến sẽ được điều khiển thông qua kết nối bluetooth với một ứng dụng trên SmartPhone đi kèm những tính năng mở rộng khác nhau tùy thuộc vào khả năng phát triển của nhóm. Chi tiết về đề tài này chúng ta sẽ cùng nhau làm rõ trong nội dung báo cáo dưới đây.

Báo cáo được trình bày thành 6 phần chính:

- I. Giới thiệu đề tài
- II. Phân tích hệ thống
- III. Thiết kế hệ thống
- IV. Tiến hành xây dựng dự án
- V. Kết quả thu được
- VI. Tổng kết

Nhóm sinh viên thực hiện chúng em xin gửi lời cảm ơn chân thành đến thầy Đỗ Công Thuần – Tiến sĩ - Giảng viên bộ môn Kỹ thuật máy tính – người trực tiếp giảng dạy chúng em trong suốt quá trình học tập của học phần cũng như đưa ra những hướng dẫn quý giá trong quá trình nhóm lên ý tưởng, xây dựng và phát triển đề tài. Chúng em cũng xin gửi lời cảm ơn đến các thầy cô trong bộ môn và nhà trường đã tạo điều kiện, sắp xếp, hỗ trợ cho chúng em có cơ hội tiếp cận và trải nghiệm những nội dung quý giá này một cách tốt nhất trong bối cảnh ảnh hưởng của dịch bệnh. Một lần nữa chúng em xin chân thành cảm ơn!

Nguyễn Duy Khai - Nguyễn Thị Trang

I. Giới thiệu về đề tài

1. Đặt vấn đề

Như đã đề cập ở phần mở đầu, các thiết bị điều khiển và các hệ thống nhúng ngày một nhiều và trở nên phổ biến quanh cuộc sống thường ngày với nền văn minh hiện đại. Thực tế mỗi khía cạnh của hoạt động hằng ngày đều bị chi phối bởi một vài loại hệ thống điều khiển. Dễ dàng tìm thấy các thiết bị điều khiển cho các máy móc, công cụ, điều khiển máy tính, các hệ thống giao thông, hệ thống năng lượng robot,... rồi các ứng dụng vĩ mô hơn như công nghệ kỹ thuật liên quan đến vũ khí công nghệ cao, ngành hàng không vũ trụ, hay đơn giản là các ví dụ về các phương tiện đi lại, đồ gia dụng, dân dụng mỗi người sử dụng hằng ngày.

Một khía cạnh khác là trong sinh hoạt hàng ngày của con người, người ta dần có xu hướng đòi hỏi mọi thứ phải trở nên tiện lợi, mọi người ngày một bận rộn và họ có xu hướng muốn mọi thứ đều có thể điều khiển từ xa thay vì phải bỏ thời gian đi đến từng thiết bị, sờ mó, tương tác trực tiếp vào thiết bị thì mới điều khiển được. Việc điều khiển từ xa sẽ giúp mọi người dễ dàng và thuận tiện hơn trong mọi thao tác, và những kết nối cho việc điều khiển này nên là không dây (Wireless) để giảm thiểu tối đa sự cồng kềnh, vướng mắc của không gian sử dụng. Cùng với đó là các chiếc điện thoại thông minh (SmartPhone) cũng được ưa chuộng vì sự tinh gọn, mọi người có thể sử dụng hầu hết các dịch vụ, chức năng của một chiếc máy tính mà không cần phải mang vác cồng kềnh.

Trong quá trình học tập học phần hệ nhúng, nhận thấy bối cảnh và những nhu cầu kể trên, nhóm đặt ra mục tiêu phải xây dựng được sản phẩm với các tiêu chí:

- Ứng dụng được kiến thức về xây dựng các hệ thống nhúng đơn giản.
- Tận dụng được sự tiện lợi của công nghệ không dây với kết nối từ xa trong điều khiển.
- Phải là một sản phẩm được xem là “thông minh”. Mặc dù khái niệm này có vẻ còn mơ hồ.
- Ứng dụng được sự tiện lợi của SmartPhone (các ứng dụng Moblie) để đưa vào đề tài.

Nắm bắt những vấn đề và xác định được các tiêu chí, cùng với việc gần đây Vin Group để lại nhiều cảm xúc cho những người quan tâm đến công nghệ khi mà họ cho ra đời dòng xe điện VinFast như một đại diện của Việt Nam tạo nên tiếng vang thực sự lớn trên toàn thế giới. Một trong những điểm nổi trội của dòng xe này bên cạnh việc sử dụng năng lượng điện thay cho động cơ đốt trong là sự “thông minh nhân tạo” của hệ thống trợ lý hỗ trợ tài xế với khả năng lắng nghe yêu cầu của người lái thông qua giọng nói và thực hiện những “lệnh” này, trò

chuyện với người lái, hỗ trợ việc điều khiển xe cũng như nhu cầu tìm kiếm, truy cập thông tin và giải trí trong suốt quá trình lái xe. Cùng lấy cảm hứng từ đây, nhóm đi đến quyết định lựa chọn đề tài cho mình là “xe điều khiển cải tiến”, một xe “robot” minh họa được việc điều khiển thông qua kết nối không dây bluetooth với một điện thoại Android, các thao tác điều khiển được thực hiện bằng giao diện ứng dụng và bằng tiếng nói với một số câu đơn giản để ra lệnh cho xe chạy theo ý đồ người điều khiển.

2. Mô tả tổng quan đề tài

Đề tài sẽ bao gồm một xe điều khiển, xe này sẽ có động cơ tạo ra chuyển động cho các bánh xe để có thể di chuyển trong thế giới thực. Phần xử lý cho việc điều khiển động cơ của xe nhóm lựa chọn sử dụng Arduino Uno – chi tiết về phần này sẽ được trình bày trong nội dung tương ứng phía sau. Để điều khiển được xe, nhóm sẽ tiến hành xây dựng một ứng dụng Android cài đặt trên một chiếc điện thoại, điện thoại này sẽ kết nối với xe thông qua kết nối bluetooth. Người dùng sẽ thao tác trên màn hình với các nút bấm tương ứng hoặc sử dụng tiếng nói (phần tiếng nói cũng sẽ được ghi nhận trong ứng dụng Android) để gửi đi yêu cầu cho xe phải tiến (chạy về phía trước), lùi (chạy ngược về phía sau), rẽ trái và rẽ phải cũng như dừng khi được yêu cầu hoặc người dùng dùng thao tác điều khiển.

Vậy, tổng kết lại hệ thống sẽ có 3 thành phần chính:

- Xe điều khiển (phần thân xe và các động cơ)
- Kết nối bluetooth
- Ứng dụng điều khiển trên điện thoại

Và xe đi kèm các hoạt động vận hành: tiến, lùi, rẽ trái, rẽ phải và dừng lại. Chi tiết về đề tài sẽ tiếp tục được làm rõ.

3. Giới thiệu về công nghệ không dây bluetooth

Kết nối giữa xe và ứng dụng điều khiển trên điện thoại mà nhóm lựa chọn là kết nối không dây bluetooth. Vậy, bluetooth là công nghệ kết nối không dây nhằm trao đổi dữ liệu ở cự ly gần giữa các thiết bị điện tử. Nhờ công nghệ này, các dữ liệu sẽ truyền tải thông qua sự kết nối giữa các thiết bị di động và cố định mà không cần dùng tới hệ thống dây vật lý trước đây. Trên thực tế, trong điều kiện hoàn hảo để kết nối thì bluetooth 5.0 có khoảng cách hoạt động hiệu quả lên đến 40-60m.

Bluetooth sử dụng sóng radio tần số 2.4GHz. Mặc dù đó là tần số chung với công nghệ wi-fi nhưng nó lại không hề có sự xung đột do bước sóng và phạm vi sử dụng của bluetooth ngắn hơn nhiều. Muốn tích hợp bluetooth phải theo chuẩn nhất định, thông qua chuẩn này các thiết bị có thể nhận ra và tương tác với nhau khi được kết nối bluetooth. Công nghệ này sẽ có một số ưu điểm, nhược điểm đáng chú ý như sau:

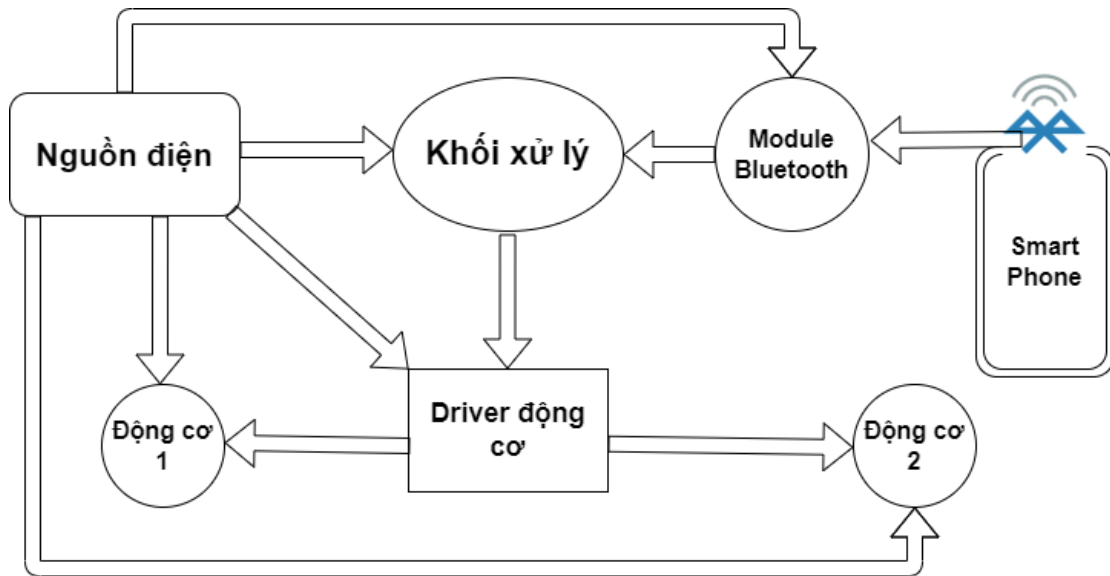
- Về ưu điểm:
 - Không cần dây cáp
 - Không ảnh hưởng sức khỏe con người
 - Bảo mật cao
 - Giá thành ngày một giảm
 - Tiêu thụ năng lượng ngày một giảm
 - Không gây nhiễu
 - Khả năng tương thích cao giữa các thiết bị
- Nhược điểm:
 - Tốc độ thấp
 - Tốc độ có thể bị yếu khi có nhiều vật cản
 - Khoảng cách kết nối còn ngắn so với công nghệ mạng không dây khác
 - Thời gian thiết lập lâu

Nhưng nhìn chung, với yêu cầu ở mức chưa quá phức tạp như đề tài mà nhóm đã đưa ra thì kết nối bluetooth là đảm bảo để xây dựng hệ thống.

II. Phân tích hệ thống

1. Sơ đồ khối

Với những yêu cầu đã đặt ra khi lên ý tưởng về đề tài, chúng ta có thể hình dung toàn cảnh hệ thống



Nhìn chung, hệ thống sẽ bao gồm 6 khối chính: nguồn điện, khối xử lý, Driver động cơ, khối động cơ với 2 động cơ được sử dụng tương ứng (đánh số 1 và 2), module bluetooth và phần ứng dụng trên điện thoại di động SmartPhone.

2. Vai trò từng khối

- **Nguồn điện:** Khối này có vẻ không tham gia vào các hoạt động logic và trao đổi dữ liệu phục vụ cho các ý đồ điều khiển trong hệ thống nhưng cũng không điều khi nói “đây là trái tim của hệ thống nhúng mà chúng ta đang xây dựng” khi mà hệ thống này chịu trách nhiệm cung cấp năng lượng điện đến tất cả các khối còn lại (trừ thành phần điện thoại di động) để chúng có thể vận hành.
- **Khối xử lý:** Đây có thể nói là đầu não, là phần xử lý trung tâm cho hoạt động của thiết bị nhúng. Khối xử lý này sẽ nhận tài nguyên, nhận các lệnh điều khiển từ mã nguồn của người lập trình để từ đó thực hiện tương ứng việc nhận dữ liệu từ module bluetooth, kiểm tra, xử lý logic để phát đi các yêu cầu điều khiển đến phần driver động cơ đang kết nối với nó và từ driver này, các động cơ sẽ chuyển động theo ý đồ đã được lập trình. Dự kiến phần này sẽ được đảm nhiệm bởi board mạch Arduino Uno.

- **Driver động cơ:** nhận nhiệm vụ kết nối phần động cơ với khối xử lý, đây có thể xem như một thành phần mở rộng của khối xử lý. Thành phần này sẽ tương tác trực tiếp đến phần động cơ, cụ thể là 2 động cơ sẽ được sử dụng trong dự án, phần driver này hỗ trợ việc ghép nối cũng như tạo nên sự thuận tiện, dễ dàng hơn cho người lập trình khi lập trình trên khối xử lý mà muốn thể hiện các yêu cầu điều khiển đến các động cơ đã được kết nối đến driver này. Linh kiện dự kiến được lựa chọn cho phần này là Driver Shield L2393D.

- **Phần động cơ (động cơ 1 và động cơ 2):** Đây là thành phần trực tiếp tạo ra chuyển động cho bánh xe khi mà dự án nhóm lựa chọn là xe điều khiển, nghĩa là nó có khả năng di chuyển. Phần động cơ này sẽ thực hiện các chuyển động quay tương ứng theo yêu cầu mà khối xử lý đã trao đổi với phần Driver. Động cơ mà nhóm sử dụng dự kiến là serve DC với 2 động cơ tương ứng với 2 bánh xe. Xe sẽ có 3 bánh để di chuyển, 2 bánh xe phụ trách lăn bánh và điều khiển, bánh xe còn lại sẽ tạo thăng bằng cho xe và tự động lăn bánh theo quán tính mà 2 bánh xe điều khiển chạy phía trước đã tạo ra (bánh xe phụ này có tính cơ động có thể lăn tự do và xoay 360 độ).

- **Module bluetooth:** Phần này nhận nhiệm vụ thu, phát tín hiệu bluetooth và được lắp đặt trên hệ thống nhúng của xe, từ đó nó tạo cho xe một kết nối đến điện thoại di động (SmartPhone này đang cùng ghép nối bluetooth với module). Qua đó, điện thoại và xe có thể trao đổi dữ liệu, dữ liệu trao đổi này thường sẽ do điện thoại gửi đi, module bluetooth nhận dữ liệu tương ứng và gửi vào cho khối xử lý kiểm tra và thao tác. Dự kiến đề tài sử dụng module thu phát bluetooth HC-05.

- **Phần SmartPhone:** Đây là thành phần đại diện cho hoạt động của ứng dụng trên điện thoại di động mà người dùng có thể thao tác. Ứng dụng di động này sẽ yêu cầu kích hoạt bluetooth trên máy, từ đó ghép nối với module bluetooth trên thiết bị nhúng để tạo ra một đường ống kết nối (Connection Pipe) phục vụ cho việc trao đổi dữ liệu. Từ đó người lập trình có thể triển khai được việc đặt các thao tác điều khiển mà người dùng tương tác trên ứng dụng sẽ tương ứng với một gói dữ liệu tương ứng gửi qua bluetooth để thiết bị nhúng nhận và xử lý.

III. Thiết kế hệ thống

Sau khi đã hoàn thiện được ý tưởng và phân tích hệ thống thành các khối, nhóm đã có cơ sở để lập kế hoạch cụ thể cho việc xây dựng và phát triển sản phẩm. Căn cứ vào các khối đã phân tích ở trên, nhóm tiến hành lên danh sách linh kiện cần sử dụng tương ứng cũng như tìm hiểu để đưa ra được sơ đồ ghép nối các linh kiện phần cứng này sao cho hợp lý chuẩn bị cho các pha phát triển sau đó.

1. Các linh kiện được sử dụng

- **Bo mạch vi điều khiển Arduino Uno R3:**



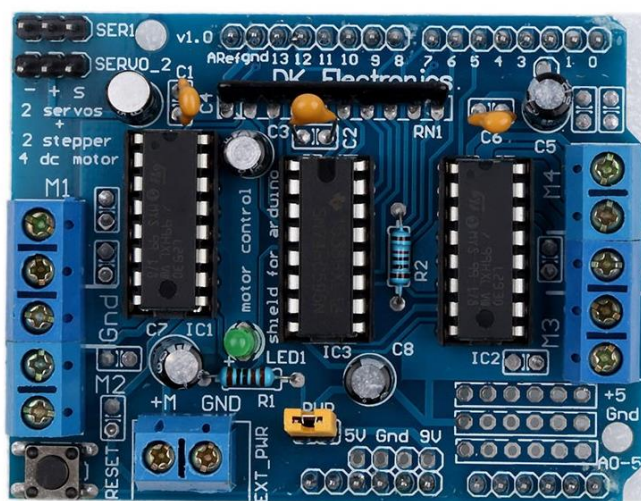
Chắc hẳn những người học lập trình hiện nay khi bắt đầu tiếp cận đến các hệ thống nhúng, cố gắng thử tìm hiểu và lập trình các dự án đơn giản để xây dựng kiến thức nền tảng không mấy ai còn xa lạ với khái niệm “Arduino”. Arduino có thể nói là một nền tảng mã nguồn mở được sử dụng một số dự án liên quan đến lập trình và điện tử. Đi liền với khái niệm này, Arduino mà nhóm đề cập đến ở đây là một linh kiện, một board mạch vi điều khiển (như hình minh họa) phục vụ cho nhu cầu lập trình với nền tảng mã nguồn mở kể trên. Arduino Uno R3 được phát triển dựa trên vi điều khiển Microchip ATmega328P với bảng mạch được trang bị các chân đầu vào, đầu ra Digital và Analog có khả năng giao tiếp với các bảng mạch mở rộng cũng như các thiết bị ngoại vi khác nhau. Nhờ đó, Arduino là sự lựa chọn tốt cho dự án mà nhóm đang hướng đến phát triển cho việc tương tác dữ liệu qua bluetooth và điều khiển động cơ quay thông thường.

Thông số kỹ thuật của Arduino Uno mà nhóm sử dụng như sau (Tham khảo tại: arduino.vn):

Chip điều khiển:	ATmega328P
Điện áp hoạt động:	5V

Điện áp đầu vào(khuyên dùng):	7-12V
Điện áp đầu vào (giới hạn):	6-20V
Số chân Digital:	14 (of which 6 provide PWM output)
Số chân PWM Digital:	6
Số chân Analog:	6
Dòng điện DC trên mỗi chân I/O:	20 mA
Dòng điện DC trên chân 3.3V:	50 mA
Flash Memory:	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM:	2 KB (ATmega328P)
EEPROM:	1 KB (ATmega328P)
Tốc độ thạch anh:	16 MHz
LED_BUILTIN:	13
Chiều dài:	68.6 mm
Chiều rộng:	53.4 mm
Cân nặng:	25 g

- **Board mở rộng Arduino Motor Shield L293D:**



Đây là một module mở rộng chuyên dụng cho các ứng dụng điều khiển động cơ với nhiều chức năng cũng như khá dễ dàng trong việc tiếp cận, vận hành và sử dụng. Module này có thư viện hỗ trợ cho việc lập trình với Arduino, cụ thể thì đây là một driver phổ biến cho điều khiển động cơ DC, sử dụng IC điều khiển động cơ DC L293D. Các chân được thiết kế tương thích với Arduino, do đó dễ dàng cắm trực tiếp vào vừa vặn với board mạch Arduino Uno và tiến hành lập trình.

Thông số kỹ thuật đáng chú ý:

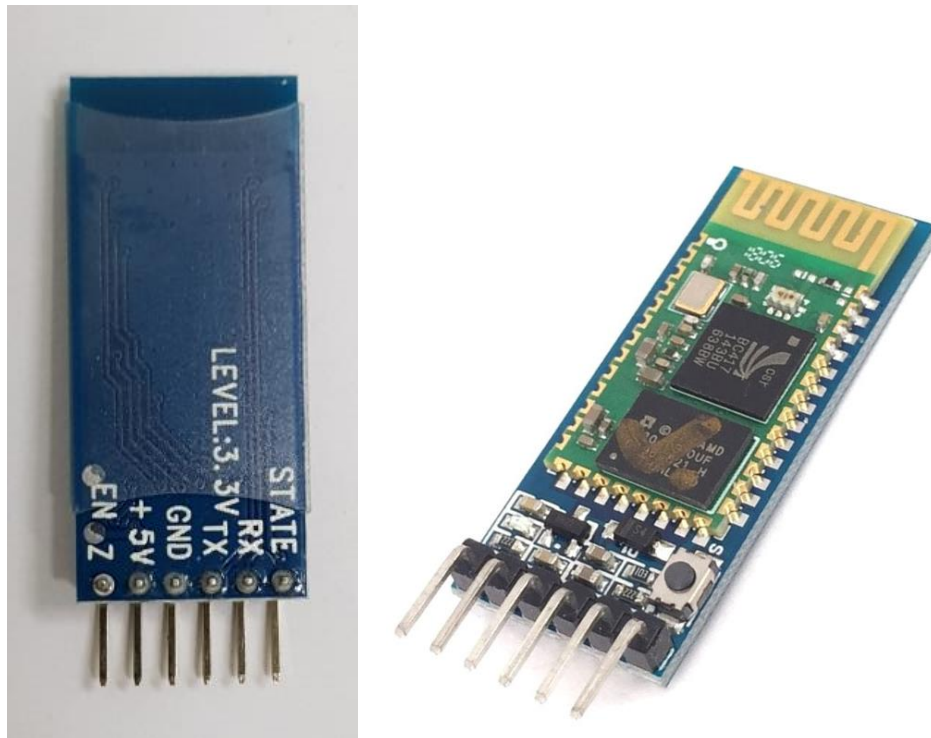
Nguồn hoạt động:	5VDC từ Arduino
Nguồn cấp cho động cơ:	4.8~35VDC
IC driver động cơ:	2 x L293D
IC khác:	1 x IC 74HC595
Chân điều khiển:	<ul style="list-style-type: none"> • 4 động cơ DC: M1, M2, M3, M4 • 2 động cơ bước công suất nhỏ (< 600mA, 4.5~36VDC) • 2 động cơ RC Servo
Tích hợp ngõ ra cho các chân Analog của Arduino:	Có
Tích hợp nút nhấn Reset:	Có

- **Động cơ giảm tốc DC:**



Đây đơn thuần là động cơ quay khá phổ biến phục vụ cho việc tạo ra chuyển động cho bánh xe. Với dự án này, nhóm lựa chọn sử dụng động cơ 5V DC.

- **Module Bluetooth HC-05:**



Đây là linh kiện hỗ trợ truyền thông không dây để trao đổi dữ liệu ở khoảng cách ngắn hay chính là công nghệ bluetooth đã trình bày ở phần đầu nội dung báo cáo. Module bluetooth HC05 dự kiến sử dụng được thiết kế dựa trên chip BC417. Linh kiện được sản xuất nhỏ gọn, tiện lợi trong việc cài cắm, sử dụng cũng như thao tác vô cùng dễ dàng khi mà nhà sản xuất đã tích hợp mọi thứ mà người dùng có thể cần với nó. Mạch bên trong được thiết kế để tương thích với việc cấp nguồn và giao tiếp thông qua 3.3VDC và 5VDC thích hợp cho nhiều ứng dụng khác nhau. Module có khả năng hoạt động ở 2 chế độ MASTER và SLAVE:

- **MASTER:** module sẽ tự động dò tìm thiết bị bluetooth khác và tiến hành pair chủ động mà không cần thiết lập gì từ máy tính hoặc Smartphone.
- **SLAVE:** người dùng 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, chúng ta đã có 1 cổng Serial từ xa hoạt động ở BaudRate 9600.

Các chân của module:

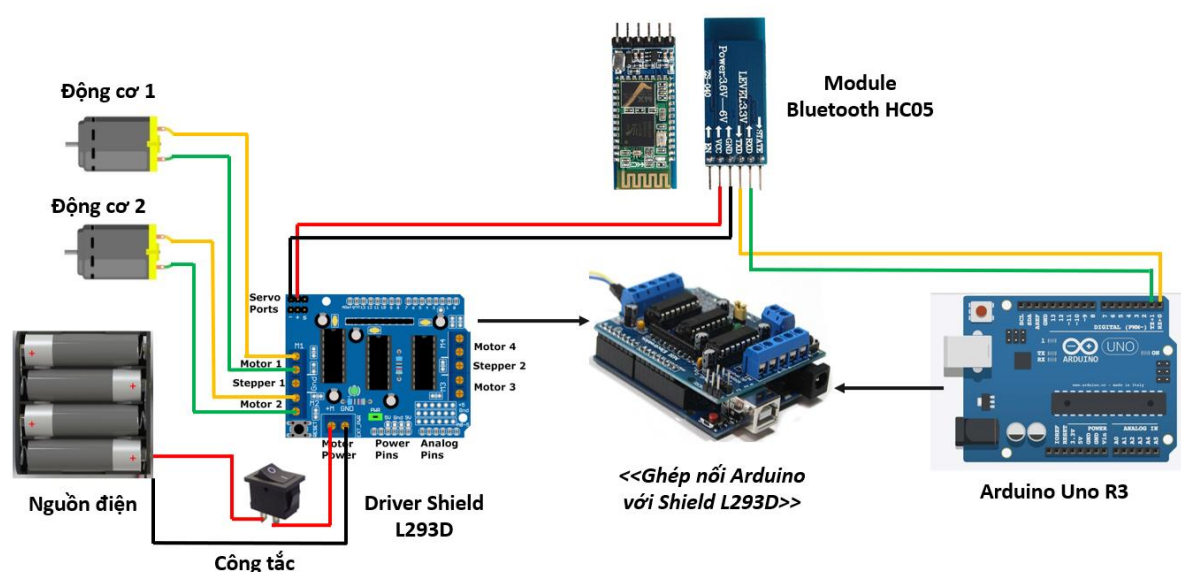
- **EN:** chân chọn chế độ hoạt động AT Mode hoặc Data Mode.
- **+5V (VCC):** chân cấp nguồn, nguồn điện dương với điện áp 5V.
- **GND:** Nối với chân nguồn GND (Mass) hay nối đất.
- **TXD và RXD:** đây là hai chân UART để giao tiếp logic.
- **STATE:** chân này dùng để báo trạng thái kết nối của bluetooth (ít được quan tâm).

Thông số kỹ thuật (tham khảo tại: dientutuonglai.com):

Điện áp hoạt động:	3.3 ~ 5VDC
Mức điện áp chân giao tiếp:	TTL tương thích 3.3VDC và 5VDC
Dòng điện khi hoạt động:	khi Pairing 30 mA, sau khi pairing hoạt động truyền nhận bình thường 8 mA
Tốc độ truyền khi giao tiếp UART có thể chọn được:	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 baud
Giao thức giao tiếp:	Chủ và tớ (Master and Slave)
Tần số:	2.4 GHz
Truyền điện:	≈ 4 dBm, class 2
Độ nhạy:	≈ -84 dBm tại 0.1% BER
Rate:	<ul style="list-style-type: none"> Không đồng bộ: 2.1 Mbps (max)/160 kbps Đồng bộ: 1 Mbps/1 Mbps
Bảo mật:	Xác thực và mã hóa
Kích thước:	15.2 x 35.7 x 5.6mm
Thiết lập UART mặc định:	tốc độ truyền 38400 hoặc 9600 (baud), N, 8, 1
Mật khẩu module HC-05 mặc định:	1234 hoặc 0000

2. Sơ đồ thiết kế mạch điện

Với sơ đồ khối đã phân tích và danh sách linh kiện được chuẩn bị như trên, phần thiết bị nhúng sẽ được ghép nối, kết nối với nhau theo sơ đồ nối các mạch điện như sau:



Để mô tả rõ hơn về các kết nối, chúng ta có thể theo dõi hình ảnh kèm với bảng chú thích dưới đây (đã có đi kèm các nội dung ghi chú)

Bảng thiết kế chi tiết:

- L293D đã ghép nối chân tương ứng với các chân của Arduino để tạo thành một khối như hình ảnh minh họa.
- Bảng nối mạch giữa động cơ với Driver Shield L293D:

Động cơ	L293D
Phải (Động cơ 1)	M1
Trái (Động cơ 2)	M2

(Cắm 2 dây động cơ vào tương ứng 2 chân ở cổng của L293D)

- Bảng nối mạch nguồn điện với Driver Shield L293D:

Nguồn điện	L293D
Cực dương	GND
Cực âm	+M

- Bảng nối mạch cho Module bluetooth HC-05

HC-05	L293D	Arduino Uno
RX (RXD)		TX1
TX (TXD)		RX0
+5V (VCC)	+ (SER1)	
GND	- (SER1)	

3. Quy ước dữ liệu điều khiển trao đổi qua bluetooth

Một nội dung nữa cần thống nhất trong bản thiết kế để chuẩn bị cho kế hoạch phân công triển khai cho các thành viên trong nhóm là thống nhất được việc trao đổi dữ liệu qua bluetooth giữa ứng dụng di động và phần lập trình xử lý trên xe, tạm gọi là Mobile App và Arduino Firmware.

Theo đó, thông qua phân tích, thảo luận và đánh giá, nhóm đi đến quyết định lựa chọn dữ liệu trao đổi sẽ là các ký tự đại diện cho các trường hợp điều khiển như sau:

- Bên gửi dữ liệu: Mobile App
- Bên nhận và xử lý dữ liệu: Phần lập trình Arduino Firmware
- Kênh truyền thông: Kết nối không dây bluetooth
- Định dạng dữ liệu: một ký tự duy nhất
- Kiểu dữ liệu: **char** (cho cả lập trình C và Java)

- Quy ước:

Dữ liệu trao đổi dùng trong lập trình	Ký tự tương minh	Ý nghĩa gọi nhớ (viết tắt của từ)	Trường hợp điều khiển
's'	s	"Stop"	Xe dừng lại, đứng yên
'f'	f	"Forward"	Xe chạy tiến về phía trước
'b'	b	"Backward"	Xe chạy lùi về phía sau
'l'	l	"Left"	Xe quay đầu sang trái
'r'	r	"Right"	Xe quay đầu sang phải
'L'	L	"Left" (khác l)	Xe rẽ hướng di chuyển sang trái khoảng 90 độ
'R'	R	"Right" (khác r)	Xe rẽ hướng di chuyển sang phải khoảng 90 độ

(Các trường hợp điều khiển với ký tự 'L' và 'R' (in hoa) được bổ sung để cải tiến xe cho tính năng rẽ trái và rẽ phải phù hợp hơn sau một thời gian quan sát kết quả thử nghiệm)

Đến đây thì mọi thứ đều đã sẵn sàng, giờ bắt tay vào triển khai xây dựng sản phẩm theo kế hoạch mà nhóm đã thống nhất từ trước.

IV. Tiến hành xây dựng dự án

1. Xây dựng ứng dụng điều khiển trên điện thoại (Android)

Thành phần trực quan trong dự án mà khi sản phẩm được hoàn thành, người dùng sẽ trực tiếp tương tác với nó là phần ứng dụng điều khiển trên giao diện mobile. Ở đây, để dễ dàng xây dựng nhất với khả năng và kiến thức của các thành viên trong nhóm, nhóm đã quyết định lựa chọn nền tảng Android. Vì kiến thức còn hạn chế nên để kịp thời xây dựng ứng dụng, sau một thời gian tìm hiểu, nhóm đã lựa chọn nền tảng MIT App Inventor để tạo ra ứng dụng Android. Ứng dụng này cần có khả năng cung cấp giao diện nhận các thao tác điều khiển hoặc tiếng nói từ người dùng để từ đó gửi các dữ liệu điều khiển tương ứng qua bluetooth đến thiết bị nhúng theo quy ước dữ liệu đã được thiết kế.

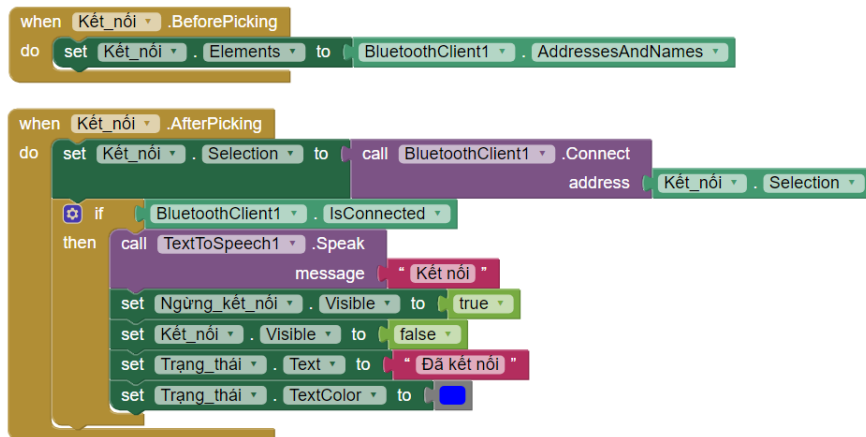
Truy cập vào địa chỉ <http://ai2.appinventor.mit.edu>. Tiếp đó tiến hành đăng nhập vào tài khoản Googhe để quản lý các project và tiến hành tạo một dự án mới:

Đầu tiên, để thiết kế giao diện hiển thị trên màn hình ứng dụng, chúng ta sử dụng tính năng thiết kế (nút Design) và tiến hành kéo thả với các thành phần như sau:

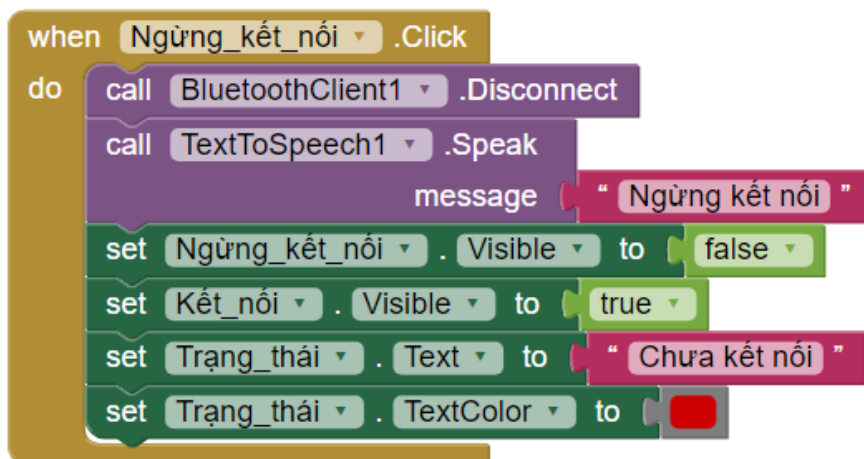
- Nút kết nối và ngưng kết nối: sử dụng hình ảnh biểu tượng bluetooth (hình ảnh màu xanh cho nút kết nối và màu đỏ cho hủy kết nối).
- Một label (nhãn văn bản) để hiển thị trạng thái kết nối.
- Có 4 nút nhấn giữ di chuyển trái, phải, tiến, lùi tương ứng là các nút mũi tên điều hướng.
- Một nút bấm với biểu tượng micro cho chức năng nghe tiếng nói và chuyển thành văn bản (SpeechToText hay SpeechOrganizer).
- Một label ghi nhận văn bản thu được sau khi lắng nghe tiếng nói của người dùng.

Để ứng dụng có thể hoạt động, dưới các giao diện hiển thị cần có các trao đổi, xử lý logic. Các khối lập trình (Blocks) trong dự án này bao gồm:

- Khi click vào nút kết nối sẽ hiện danh sách các bluetooth để kết nối khi chọn bluetooth của chiếc xe điều khiển thì ta sẽ tạo kết nối với địa chỉ đã chọn và chuyển trạng thái là “Đã kết nối” (màu xanh) đồng thời ẩn nút kết nối và hiện nút ngắt kết nối.



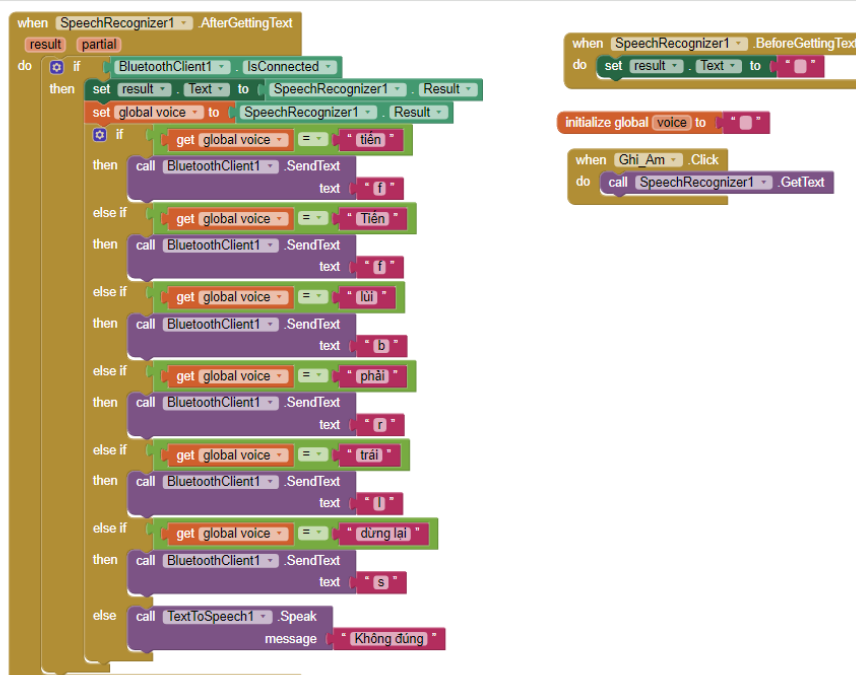
- Khi click vào nút ngừng kết nối thì App sẽ ngắt kết nối đồng thời chuyển trạng thái sang “Chưa kết nối” (màu đỏ) và ẩn nút ngừng kết nối và hiện nút kết nối.



- Khi nhấn giữ các nút điều khiển (TouchDown) chính là các biểu tượng mũi tên điều hướng tương ứng trái, phải, tiến và lùi thì App sẽ kiểm tra bluetooth có kết nối hay không nếu kết nối thì sẽ gửi tín hiệu text (tương ứng) qua đường ống kết nối (Connection Pipe) này. Khi TouchUp (người dùng thả tay khỏi màn hình hay khỏi nút điều khiển họ đang giữ) ứng dụng sẽ kiểm tra kết nối nếu có kết nối thì sẽ gửi tín hiệu text là “s” cho bluetooth để xe dừng lại.



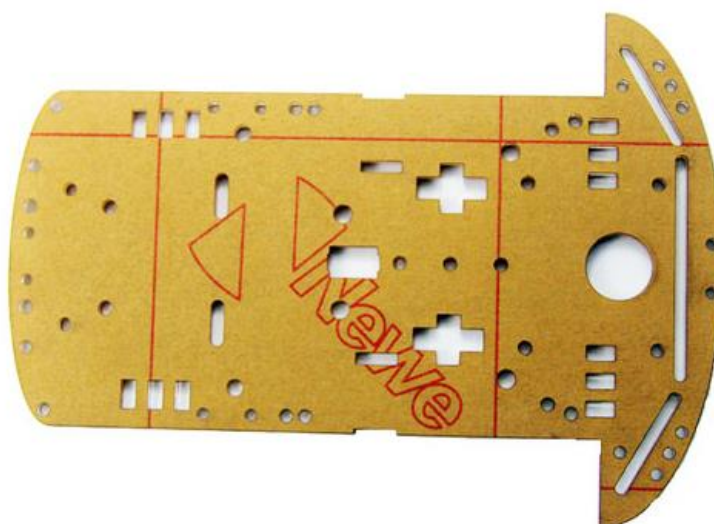
- Dùng biến **voice** để lấy giá trị khi người dùng nói. Rồi tiến hành kiểm tra, nếu giá trị **voice** mà bằng với “Tiền”, “lùi”, “trái”, “phải” hoặc “dừng lại” thì gửi ký tự “f”, “b”, “L”, “R” hay “s” tương ứng theo quy ước. Trong trường hợp tiếng nói ghi nhận từ người dùng khác với các từ kể trên thì ứng dụng sẽ phát đi thông báo “Không đúng” nghĩa là người dùng nói từ chưa đúng bất cứ yêu cầu điều khiển nào.



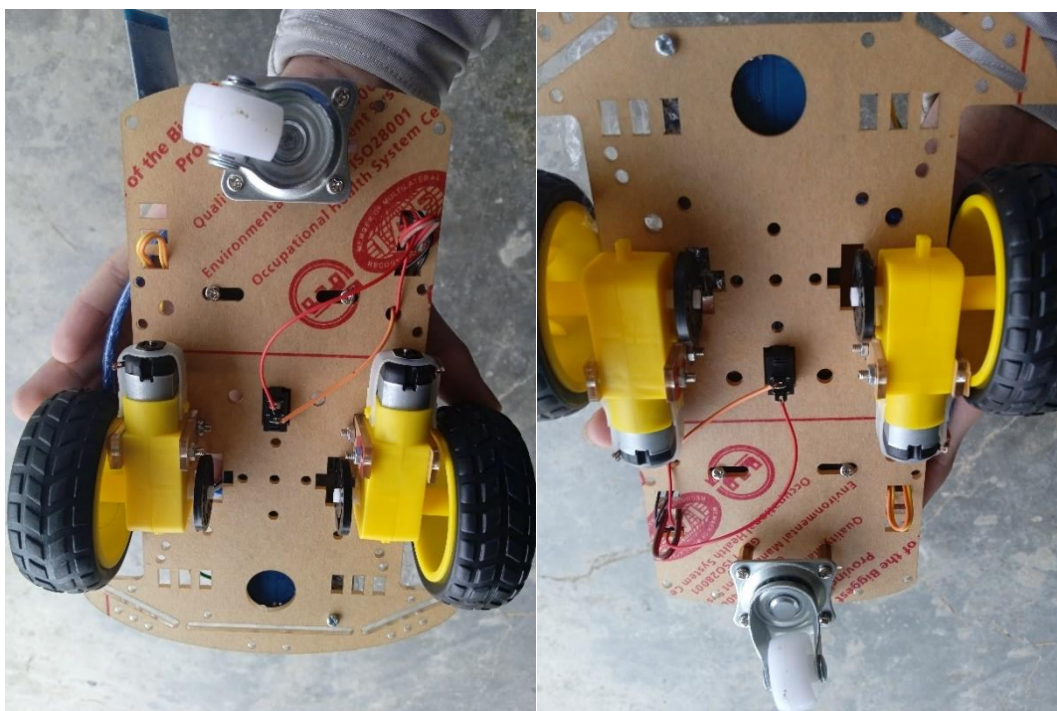
Tiến hành xây dựng ứng dụng như mô tả kể trên, chúng ta thu được một App android hoàn chỉnh, xuất file .apk để cài đặt và sử dụng ứng dụng.

2. Xây dựng thiết bị nhúng (xe điều khiển) và lập trình firmwre

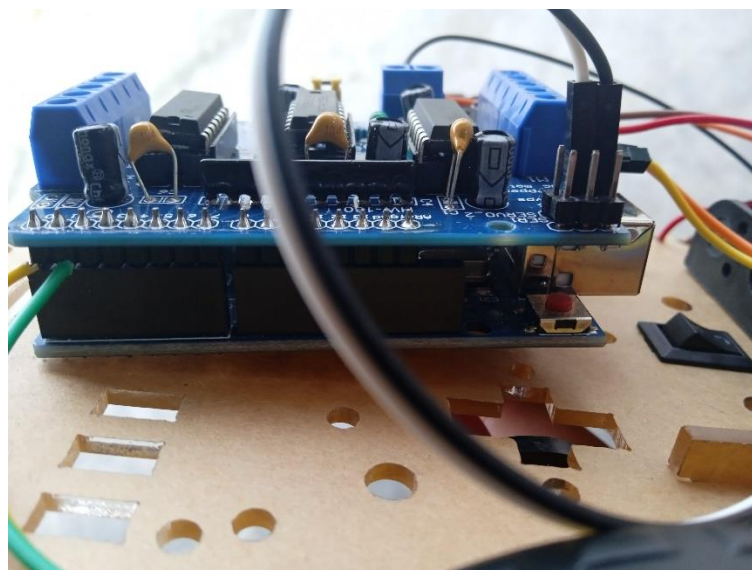
Chúng ta đã có một bản phân tích – thiết kế khá chi tiết, vậy nên việc tiến hành triển khai xây dựng từng thành phần trở nên vô cùng đơn giản. Sau khi đã chuẩn bị đủ các linh kiện theo danh sách kể trên, nhóm đã tiến hành lắp đặt. Để thuận tiện cho việc này, nhóm đã sử dụng một khung xe Mica đã được sản xuất cho người sử dụng dễ dàng đặt các linh kiện lên theo vị trí phù hợp và cố định lại chúng.



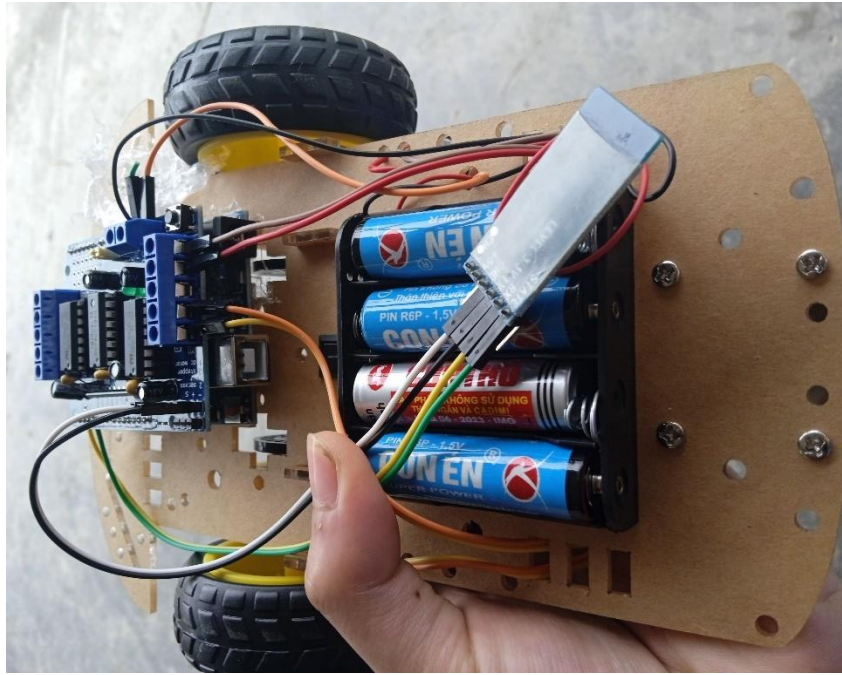
Tiếp đến, bắt đầu lắp đặt phần động cơ lên khung xe và cố định đồng thời lắp bánh xe vào động cơ, đặt thêm một bánh xe tự do (bánh xe phụ hay là bánh sau trong một xe 3 bánh), bánh xe này sẽ không tham gia vào việc điều khiển và tạo chuyển động mà có khả năng lăn bánh và xoay linh hoạt do được hai bánh trước gắn với động cơ kéo nó chạy theo.



Ở mặt trên của khung xe, tiến hành cố định phân công tắc và khung nguồn điện để cắm pin vào. Ở đây nhóm lựa chọn sử dụng bộ nguồn với 4 pin con thỏ để cấp điện cho xe. Cùng với đó, ghép board mạch Arduino Uno R3 với Driver Shield L293D như hình ảnh minh họa ở phần thiết kế mạch điện.



Sau khi đã cố định lại được mọi thứ, tiến hành nối mạch theo đúng sơ đồ đã thiết kế để thu được phần xe hoàn chỉnh và bắt tay vào lập trình firmware cho Arduino.



Phần cứng đã xong xuôi, để lập trình firmware, chúng ta có thể lựa chọn Arduino IDE ver1.8.19 cho Windows. Về bản chất, khi lập trình với Arduino, để điều khiển được động cơ theo ý đồ của người lập trình, chúng ta cần nắm bắt được rằng, Arduino chỉ phát ra các tín hiệu HIGH hoặc LOW cho các chân digital và có các chân analog phát tín hiệu tương tự. Nên khi dùng một driver thông thường, chúng ta sẽ hướng đến lập trình với tư tưởng là viết các tín hiệu ra các chân tương ứng của Driver, các chân này đã nối với các động cơ. Ví dụ, trong trường hợp thông thường, động cơ đã kết nối vào chân inABCXYZ nào đó. Chúng ta có thể có đoạn code như sau:

```
analogWrite(ENA, 100);  
digitalWrite(inABCXYZ, HIGH);
```

Hoặc

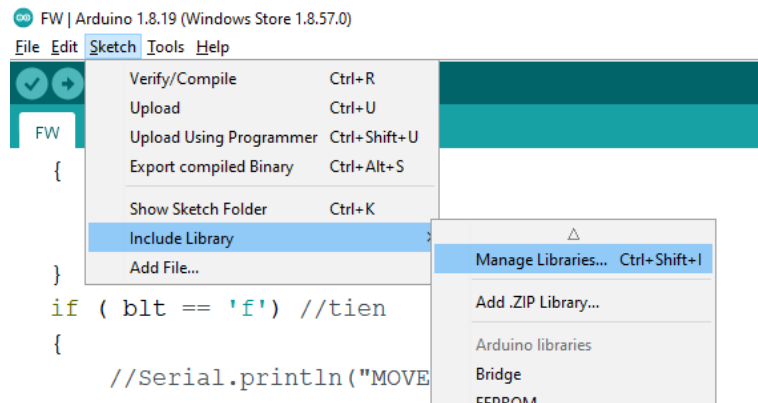
```
analogWrite(ENB, 255);  
digitalWrite(inABCXYZ, LOW);
```

Những dòng code như vậy sẽ gửi các giá trị tín hiệu HIGH (1) hay LOW (0) đến các chân tạo ra các chuyển động tiến, lùi (quay theo chiều thuận và chiều ngược lại) của động cơ. Đồng thời, giá trị analog đã đặt tốc độ của động cơ là $100/255 = 39.22\%$ tốc độ quay cực đại hoặc tốc độ chính bằng 255 là tốc độ quay lớn nhất mà động cơ có thể đạt được.

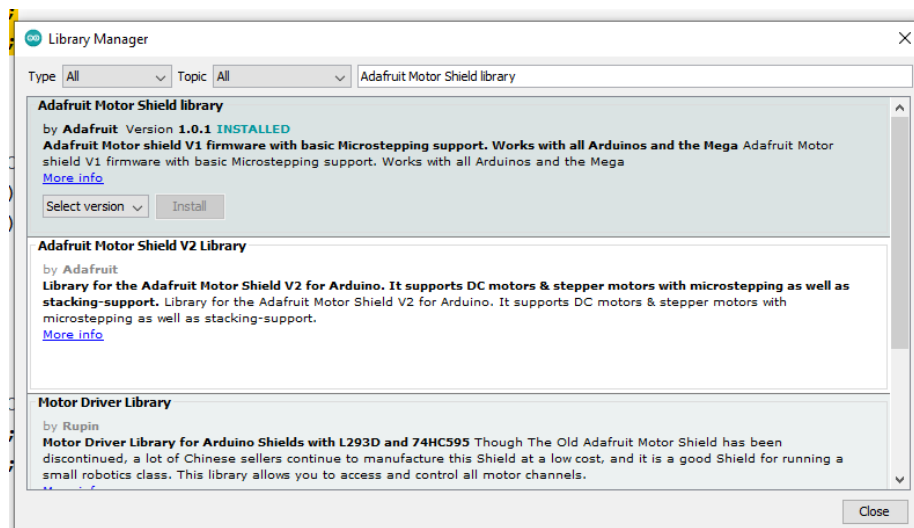
Nhưng trong dự án này, sau một quá trình tìm hiểu thông tin, nhóm đi đến lựa chọn một cách triển khai đơn giản, mã nguồn dễ đọc, dễ hiểu hơn, đó là sử dụng thư viện. Chúng ta đang có Driver cho động cơ là Board mở rộng Arduino Motor Shield L293D với các vị trí được đánh dấu M1, M2, M3 và M4, mỗi vị trí có 2 chân để đấu nối với 2 dây của động cơ DC. Như đã giới thiệu trong phần nói

về các linh kiện, bên cạnh việc tạo ra sự dễ dàng, thuận tiện cho việc ghép nối các linh kiện, kết nối Arduino Uno với động cơ, thì Shield L293D còn cho chúng ta sự tương thích với một số thư viện giúp dễ dàng hơn trong lập trình. Thư viện mà nhóm đang muốn đề cập đến ở đây là AFMotor, trong lập trình là `#include <AFMotor.h>` (Adafruit Motor Shield library).

Vào giao diện Arduino IDE và tiến hành như sau: Chọn vào **Sketch → Include Library → Manage Libraries** hoặc nhấn tổ hợp phím **Ctrl + Shift + I**.



Màn hình hiện ra, nhập từ khóa “**Adafruit Motor Shield library**” vào ô **Filter your search**, lúc này chọn vào thư viện tương ứng hiện ra (thường nằm ở kết quả đầu tiên) và bấm **Install**.



Được rồi, bây giờ chúng ta có thể include thư viện AFMotor.h vào mã nguồn để tiến hành lập trình với các hàm đã được hỗ trợ cho việc điều khiển động cơ chuyển động quay theo hướng được quy định là FORWARD (tiền) và BACKWARD (lùi) cũng như đặt tốc độ quay của động cơ tương ứng.

Mã nguồn chương trình sẽ như sau: khai báo thư viện, khai báo một biến **blt** kiểu **char** và khởi tạo với giá trị mặc định là 's' tương ứng trạng thái xe đứng yên (như đã mô tả chi tiết ở phần quy ước trao đổi dữ liệu trong nội dung thiết kế),

khai báo 2 động cơ tương ứng vị trí M1 và M2 trên Shield L293D cho 2 motor gắn với 2 bánh xe (bánh trước) của xe.

```
#include <AFMotor.h>

char blt = 's'; //Biến nhận dữ liệu qua bluetooth
AF_DCMotor motor1(1); //Khai báo động cơ 1
AF_DCMotor motor2(2); //Khai báo động cơ 2
```

Tiếp đến là hàm setup(), hàm này sẽ được thực hiện một lần duy nhất trong vòng đời hoạt động của thiết bị khi đã nạp code và vận hành với Arduino. Hàm phù hợp để khởi tạo một số giá trị cũng như đặt vào đó những nội dung ít thay đổi

```
void setup()
{
    // put your setup code here, to run once:
    Serial.begin(9600); //Mở cổng Serial
    motor1.setSpeed(255); //Đặt tốc độ động cơ
    motor2.setSpeed(255);
}
```

Ở đây, chúng ta sẽ mở cổng Serial là cổng sẽ trao đổi, gửi nhận dữ liệu qua module bluetooth, đặt BaudRate là 9600 cho cổng này để nhận gửi nhận dữ liệu theo tiêu chuẩn tránh bị sai lệch bit nào không mong muốn trong quá trình trao đổi thông tin. Tiếp đến là đặt tốc độ 255 tương ứng với tốc độ quay lớn nhất của 2 động cơ.

Phần xử lý quan trọng nhất trong lập trình firmware là hàm **loop()**, đây là hàm sẽ thực hiện lặp đi lặp lại mãi trong toàn bộ vòng đời của hệ thống nhúng với Arduino. Trong hàm này, mỗi lần lặp, chúng ta sẽ nhận dữ liệu từ Serial:

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

Mỗi lần nhận được dữ liệu, nếu là dữ liệu hợp lệ (một ký tự kiểu **char** như đã quy ước), chúng ta tiến hành ghi dữ liệu nhận được vào biến **blt** đã khai báo ở đầu. Đồng thời ở mỗi lần lặp như vậy, chúng ta lại kiểm tra xem ký tự này tương ứng ca điều khiển nào và cho xe chạy (động cơ quay) theo yêu cầu tương ứng.

Nếu nhận được ký tự f (forward), chúng ta sẽ tiến về trước, ký tự b (backward) thì lùi:

```
if ( blt == 'f' ) //tiền
{
    //Serial.println("MOVE FORWARD!");
    motor1.run(BACKWARD); // động cơ tiến
}
```



```

    motor2.run(BACKWARD);
    //blt = 's';
    delay(5);
}
if (blt == 'b') //lui
{
    //Serial.println("MOVE FORWARD!");
    motor1.run(FORWARD);      // động cơ lui
    motor2.run(FORWARD);
    //blt = 's';
    delay(5);
}

```

Ở đây, chắc hẳn sẽ có ai đó thắc mắc, tại sao động cơ tiến là là **.run(BACKWARD)**; và lùi là **.run(FORWARD)**; rõ ràng đây cũng là hàm của thư viện AFMotor.h, không thể nào người ta lại xây dựng thư viện lỗi được, hay anh đã lập trình sai? Câu trả lời là không sai đâu ạ, với đoạn code như vậy, xe mà nhóm đã lắp đặt tương ứng sẽ chạy về phía trước cũng như lùi về phía sau, vị trí trên Shield để nối động cơ cũng đã chọn đúng với M1 và M2, lắp cố định linh kiện cũng đã đúng chiều, nhưng trong quá trình lắp đặt rất tiếc là thành viên của nhóm đã đấu nối ngược chiều dây gây ra logic bị ngược như thế này. Nhưng không sao cả, dù sao người lập trình hiểu vấn đề và đã lập trình sao cho xe chạy đúng yêu cầu, như vậy là thành công rồi.

Tương tự với phần điều khiển rẽ trái và rẽ phải, chúng ta có 2 bánh trước tạo chuyển động và có thể điều khiển, động cơ thì có 2 chiều quay duy nhất, vậy để rẽ trái, rẽ phải thì cũng khá đơn giản, chúng ta sẽ cho bánh bên trái (động cơ motor1) quay theo chiều lùi đồng thời bánh xe bên phải (động cơ motor2) quay theo chiều tiến về phía trước, cùng với bánh xe phụ tự do ở sau xe có khả năng lăn bánh và xoay linh hoạt theo trục thẳng đứng, xe sẽ tạo ra được chuyển động rẽ trái. Tương tự tiến hành ngược lại với tình huống rẽ phải:

```

if (blt == 'l') //xoaytrai
{
    motor1.run(FORWARD);      // động cơ quay trái
    motor2.run(BACKWARD);
    //blt = 's';
    delay(5);
}
if (blt == 'r') //xoayphai
{
    motor1.run(BACKWARD);     // động cơ quay phải
    motor2.run(FORWARD);
    //blt = 's';
    delay(5);
}

```

Đến đây, các chuyển động cơ bản đã hoàn tất, nhưng qua một quá trình nạp code, chạy thử nghiệm và quan sát kết quả, nhận thấy có những chỗ không phù hợp khi mà việc người dùng điều khiển tiếng nói cho việc quay trái, quay phải thì xe đang nhận ký tự 'l' và 'r' tương ứng rồi tiến hành quay đầu sang trái mãi tạo thành một chuyển động quay tròn lặp đi lặp lại liên tục. Để khắc phục điều này, nhóm tiến hành chuyển đổi ký tự điều khiển tương ứng yêu cầu bằng tiếng nói “trái” và “phải” thành ký tự ‘L’ và ‘R’ in hoa, lúc này chúng ta sẽ tiến hành kiểm tra và xây dựng các rẽ nhánh cho việc điều khiển:

```
if (blt == 'L')//Re trái 90 độ
{
    int i = 1500;
    while(i--)
    {
        motor1.run(FORWARD);    // động cơ quay trái
        motor2.run(BACKWARD);
    }
    blt = 'f';
    delay(5);
}
if (blt == 'R')//Re phải 90 độ
{
    int i = 1500;
    while(i--)
    {
        motor1.run(BACKWARD);    // động cơ quay phải
        motor2.run(FORWARD);
    }
    blt = 'f';
    delay(5);
}
```

Trong suốt quá trình lập trình cho việc rẽ trái và rẽ phải này, thông qua quan sát với các lần thử nghiệm, nhóm đi đến lựa chọn cho xe tiến hành hoạt động quay trái và quay phải (hoạt động này được mô tả y hệt như ca điều khiển phía trên), tuy nhiên, lúc này, hoạt động “quay xe” này sẽ diễn ra có kiểm soát. Cụ thể, nhóm đã lựa chọn được thông số là thực hiện yêu cầu quay trái, quay phải tương ứng cho động cơ điều khiển bánh xe trong khoảng 1500 lần lặp cơ bản của chương trình sẽ tạo cho xe một góc quay khá phù hợp là một góc ước lượng khoảng 90 độ tương đối. Sau khi xe đã quay đủ một góc ưng ý thì không lặp lại yêu cầu “quay xe” thêm nữa và tiến hành cho xe tiếp tục di chuyển về phía trước theo hướng mà xe đã nhận được bằng cách đặt lại biến blt = ‘f’ tương ứng với yêu cầu tiến về phía trước cho đến khi có yêu cầu dừng lại (Đây là phần mới được cải tiến trong bản cập nhật ở buổi báo cáo cuối cùng của dự án).

Và khi nhận được tín hiệu yêu cầu dừng lại (ký tự 's') thì dùng ta đặt động cơ về trạng thái RELEASE:

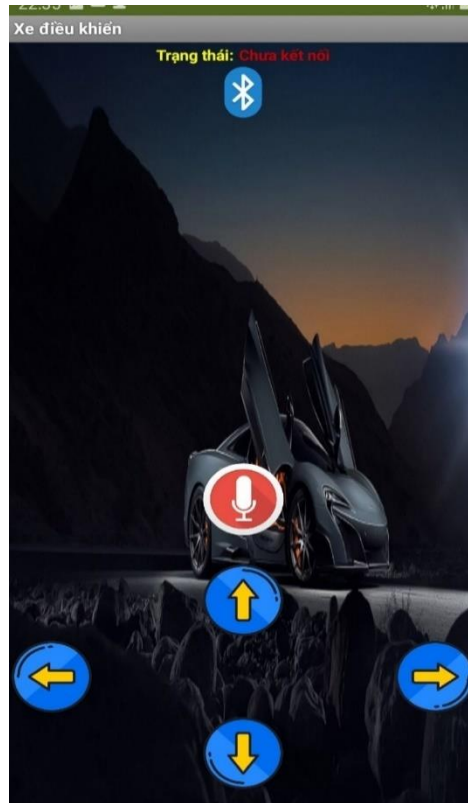
```
if (blt == 's')//Dung dong co
{
    //Serial.println("STOP!");
    motor1.run(RELEASE);
    motor2.run(RELEASE);
    //blt = 's';
    delay(5);
}
```

Như vậy là hoàn thành phần lập trình firmware cho Arduino để điều khiển các động cơ, ở đây, qua quá trình lập trình và quan sát kết quả, ở mỗi trường hợp chuyển động, nhóm quyết định thêm lệnh **delay(5);**, tại sao lại vậy? Tại vì, ban đầu khi chỉ lập trình cho việc kiểm tra dữ liệu rồi thực hiện chuyển động tương ứng ở động cơ, các thành viên phát triển nhận ra rằng, dữ liệu trao đổi qua bluetooth là các ký tự đến một cách rời rạc, mặc dù khoảng thời gian giữa 2 lần nhận một ký tự là rất rất nhỏ nhưng vì nhận rời rạc và Arduino tiến hành kiểm tra rồi phát đi yêu cầu điều khiển thì xe chuyển động nhưng gây ra những cảm giác không được mượt mà. Mà khi xe đang chạy, ví dụ như chạy về phía trước, thì xe sẽ nhận được một loạt các ký tự 'f' trong các lần lặp, khi không có ký tự này xe sẽ cố gắng dừng chuyển động hoặc đổi sang một trạng thái nào đó. Vậy việc đặt **delay(5);** sẽ giúp trạng thái chuyển động hay dừng mà xe đang thực hiện được giữ trong vòng 5ms, thời gian này rất nhỏ nên không gây sai sót trong điều khiển mà sẽ giúp trạng thái của động cơ được giữ trong khoảng thời gian chờ nhận ký tự ở lần lặp tiếp theo, từ đó chuyển động của xe mang đến cảm giác mượt mà hơn.

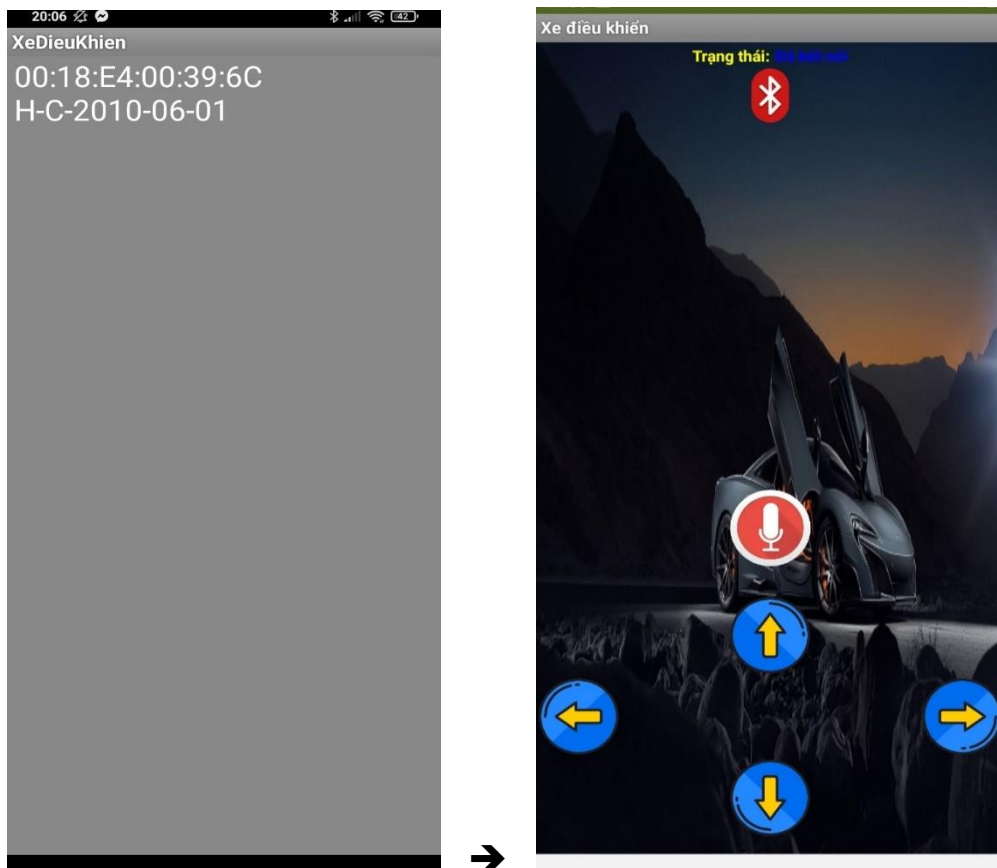
Bây giờ thì mọi thành phần đều đã hoàn tất, tiến hành nạp code vào Arduino Uno của xe và tiến hành ghi nhận kết quả thu được. Mã nguồn hoàn chỉnh của các thành phần trong toàn bộ dự án sẽ được ghi chú ở phần phụ lục.

V. Kết quả thu được

Đầu tiên chúng ta sẽ xem kết quả thu được trên ứng dụng mobile (Android), khi khởi động ứng dụng, màn hình sẽ hiện ra như sau:



Trên màn hình, có thể dễ dàng nhận ra 4 nút mũi tên điều hướng là các nút dùng để điều khiển xe tương ứng tiến, lùi, rẽ trái và rẽ phải. Ở gần trên cùng giao diện là một dòng thông báo “Trạng thái:” với dòng chữ đỏ đang hiển thị là “Chưa kết nối”, đây là trạng thái kết nối hay ghép nối bluetooth của điện thoại. Chúng ta tiến hành bấm vào biểu tượng Bluetooth (nền xanh, logo trắng) ở ngay dưới dòng thông báo này, màn hình sẽ hiện ra các điểm bluetooth mà máy có thể kết nối, bấm tương ứng vào thông tin bluetooth mà mình muốn để tiến hành ghép nối. Khi đã kết nối bluetooth thành công, dòng trạng thái chuyển thành màu xanh và nội dung đổi thành “Đã kết nối”:

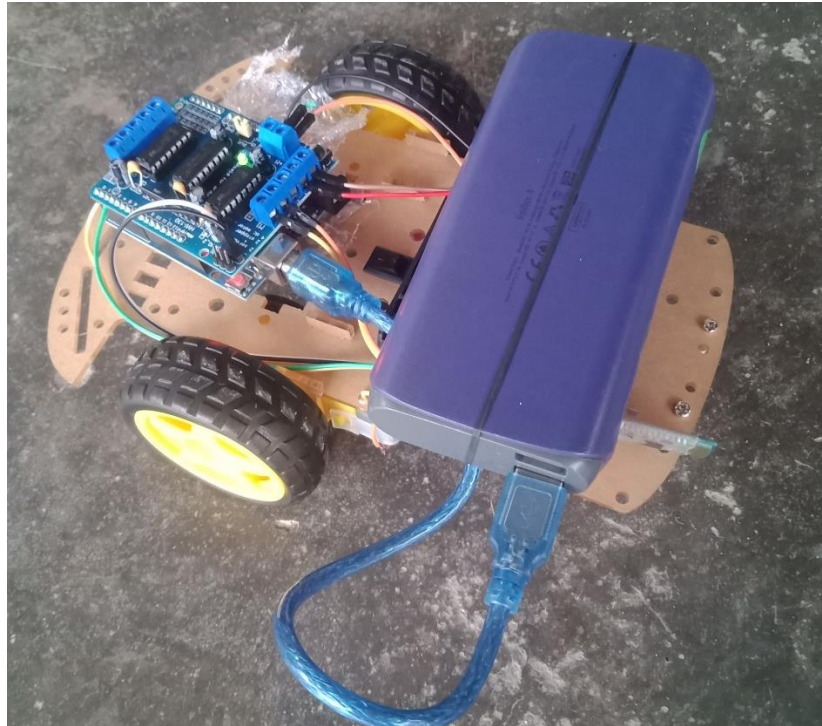


Bây giờ, chúng ta có thể sử dụng tính năng điều khiển bằng tiếng nói thông qua nút bấm biểu tượng chiếc micro đỏ trắng trên màn hình và nói các câu cho yêu cầu điều khiển tương ứng:

- “Tiền” để xe bắt đầu chạy về phía trước
- “Lùi” để xe chạy ngược về phía sau
- “Trái” để xe thực hiện chuyển động quay hướng đầu xe sang trái
- “Phải” để xe chuyển động quay theo hướng đầu xe sang phải.
- “Dừng lại” để xe dừng khi đang chuyển động.

Ngoài ra, cũng có thể sử dụng các nút bấm điều hướng bằng cách nhấn giữ các nút cho xe chạy theo hướng tương ứng và xe sẽ tự động dừng khi thả tay ra khỏi nắm bấm điều khiển (giữ nút điều hướng để giữ trạng thái chuyển động của xe được liên tục).

Dưới đây là hình ảnh thực tế xe hoàn chỉnh khi đã cấp nguồn, nạp code và đem ra chạy thử nghiệm, khi thực hiện thao tác điều khiển (tay hoặc tiếng nói) trên ứng dụng Android kể trên của điện thoại đã ghép nối bluetooth với xe, xe sẽ thực hiện chuyển động tiến, lùi, rẽ hay dừng theo đúng yêu cầu của người điều khiển:



Chi tiết về phần điều khiển xe chạy thử nghiệm, nhóm đã ghi lại trong video đã tải lên youtube, đường dẫn đã được ghi chú dưới phần phụ lục, Nhìn chung kết quả thu được là xe đã hoạt động, di chuyển theo đúng yêu cầu nhóm đã đặt ra khi xác định nội dung đề tài.

VI. Tổng kết

Thông qua đề tài đã xây dựng, điều đầu tiên chúng ta có thể ghi nhận là nhóm đã xây dựng và phát triển được một sản phẩm, một hệ thống - ứng dụng hoàn chỉnh đáp ứng được các tiêu chí đã đặt ra đó là một ứng dụng điều khiển trên giao diện màn hình điện thoại Android với khả năng thao tác trên các nút bấm, ghi nhận tiếng nói để phát đi các yêu cầu điều khiển, các yêu cầu này được truyền qua kết nối bluetooth dưới dạng dữ liệu là các ký tự, phần xe mô phỏng bằng Arduino, động cơ cùng các linh kiện đi kèm sẽ nhận những dữ liệu này và kiểm tra để tiến hành cho xe chạy (hay dừng) theo đúng yêu cầu tương ứng từ người điều khiển. Thông qua dự án, 2 thành viên trong nhóm đã được củng cố và thực hành kiến thức về lập trình Arduino cũng như làm quen với một số linh kiện, lắp đặt và xây dựng dự án với chúng như nội dung đã được học trong học phần Hệ Nhúng, đồng thời dự án mang đến góc nhìn trực quan hơn về tri thức thay vì chỉ ngồi nghe giảng và nhìn qua màn hình học trực tuyến trước bối cảnh dịch bệnh.

Những ưu điểm nổi bật mà sản phẩm có thể kể đến bao gồm: xe và ứng dụng đều đã hoàn chỉnh, hoạt động tốt, đúng và đầy đủ yêu cầu như phân phân tích và thiết kế đã đặt ra. Xe điều khiển đã có những cải tiến rõ ràng với các đề tài mô phỏng xe điều khiển qua bluetooth bằng Arduino truyền thống đó là việc cụ thể hóa ứng dụng điều khiển trên ứng dụng Android, có khả năng điều khiển bằng tiếng nói, giao diện ứng dụng thân thiện với người dùng, dễ dàng sử dụng,... Bên cạnh đó, cũng không thể bỏ qua một vài vấn đề còn tồn tại như sau: Rõ ràng nhất là dung lượng (năng lượng điện) mà nguồn cung cấp cho toàn bộ xe vận hành còn chưa đảm bảo chỉ với bộ 4 pin con thỏ như đã trình bày, thông qua quá trình thực nghiệm, nhận thấy nguồn định bị yếu làm cho xe hoạt động không ổn định nên nhóm đã phải cắm thêm một nguồn điện bổ sung vào bằng cách cắm chân COM (chính là chân nạp code) vào sạc dự phòng thường dùng của điện thoại, tất nhiên đây chỉ là giải pháp tình thế khi mà cân nặng của sạc dự phòng này khá là nặng so với xe nhưng để cấp điện thì trong quá trình lăn bánh, xe phải chở bộ sạc này trên lưng khá cồng kềnh (như hình ảnh xe thu được cuối cùng trong nội dung trình bày kết quả). Điểm hạn chế cũng làm cho nhóm chưa thực sự hoàn toàn hài lòng nữa là sự giới hạn trong việc điều khiển bằng tiếng nói, do phải sử dụng API tiếng nói của Google đã được xây dựng thành tính năng trên MIT App Inventor, nên việc điều khiển bằng tiếng nói vẫn chưa thực sự linh hoạt mà phải qua một nút bấm cũng như phải nói đúng chính xác một số câu cơ bản đã được chuẩn bị. Một vấn đề nữa là kết nối được sử dụng là công nghệ truyền thông qua bluetooth nên khoảng cách còn hạn chế và tất nhiên còn nhiều vấn đề phát sinh gây khó khăn trong quá trình xây dựng đề tài, chủ yếu đến từ những hạn chế trong bối cảnh dịch bệnh cũng như kiến thức của các thành viên còn hạn chế.

Năm bắt được những vấn đề còn tồn tại, trong tương lai gần, nhóm hướng đến giải quyết sớm những hạn chế mà dự án đang gặp phải như về nguồn điện, nên tìm kiếm một nguồn điện đủ tốt và ổn định hơn so với bộ nguồn 4 pin con thỏ. Ngoài ra, có thể tích hợp thêm wifi để linh hoạt hơn trong kết nối, xây dựng khả năng kết nối điều khiển và giám sát từ mọi nơi thông qua Internet, hay tích hợp thêm cảm biến khoảng cách để phát hiện vật cản và tránh trong quá trình xe di chuyển, cảm biến hồng ngoại để xử lý trong việc dò đường trên các bản đồ di chuyển đơn giản của xe,... tất cả sẽ giúp xe được hoàn thiện và đa năng hơn. Xa hơn một chút, nhóm sẽ hướng đến tìm kiếm một giải pháp xử lý tiếng nói tốt hơn để trải nghiệm người dùng trong việc điều khiển xe được linh hoạt hơn, cũng cố thêm kiến thức về xây dựng ứng dụng Mobile để có thể cải tiến ứng dụng theo ý riêng của mình tốt hơn. Cuối cùng, trong suốt quá trình phát triển dự án, nhóm không thể tránh khỏi những thiếu sót nhìn ra để khắc phục, rất mong nhận được thêm nhiều những tư vấn, góp ý để đề tài được hoàn thiện tốt hơn.

Lời cuối cùng, các thành viên nhóm 102 xin chân thành cảm ơn!

PHỤ LỤC

1. Phân công nhiệm vụ

Trong quá trình phát triển đề tài cho đến khi ra được sản phẩm theo yêu cầu, 2 thành viên trong nhóm đã thường xuyên liên lạc và trao đổi với nhau để thực hiện toàn bộ công việc tốt nhất, các phân công dưới đây mang tính chất tương đối:

- **Nguyễn Thị Trang:** phụ trách xây dựng ứng dụng mobile Android, tham gia lên ý tưởng, thảo luận phân tích - thiết kế, tham gia góp ý trong quá trình lập trình firmware Arduino.
- **Nguyễn Duy Khai:** tham gia lên ý tưởng, thảo luận và phân tích – thiết kế, vẽ các sơ đồ, chuẩn bị linh kiện, lắp ráp, kết nối mạch điện cho các linh kiện thành xe hoàn chỉnh theo thiết kế, lập trình firmware Arduino, nạp code và chạy thử nghiệm.

Còn nội dung báo cáo cả 2 thành viên cùng trao đổi xây dựng.

2. Tài liệu tham khảo

- Slide bài giảng học phần Hệ Nhúng (IT4210) - TS. Đỗ Công Thuận, Bộ môn Kỹ thuật máy tính – Trường CNTT&TT – Đại học Bách Khoa Hà Nội.
- Hướng dẫn xây dựng ứng dụng trên nền tảng MIT App Inventor: <https://www.youtube.com/watch?v=kLPuNPZDe7A&t=661s>
- Tài liệu tham khảo về công nghệ bluetooth: <https://binhminhdigital.com/tin/tim-hieu-ve-ket-noi-khong-day-bluetooth.html>
- Tham khảo về thông số kỹ thuật của module bluetooth HC05: <https://dientutuonglai.com/mach-thu-phat-bluetooth-hc-05-da-ra-chan.html>
- Thông số kỹ thuật Arduino Uno R3: <https://arduinookit.vn/mach-arduino-uno-la-gi/>
- Các bài viết về điện tử căn bản và lập trình Arduino Tiếng Việt: <http://arduino.vn/>
- Thư viện AFMotor.h (Adafruit Motor Shield library): <https://learn.adafruit.com/adafruit-motor-shield>

3. Mã nguồn (source code) và các resource của dự án

- Mã nguồn, các file project App Inventor và file cài của ứng dụng Android trong toàn bộ dự án:
<https://github.com/itKhaiNDdotDev/ImprovedMiniControlCar>
- Link video demo: <https://youtu.be/1lvrr-oPCrU>
- Link video demo đã cập nhật: <https://youtu.be/S8e021TuMTY>
- Công cụ vẽ sơ đồ khối: <https://www.diagrams.net/>
- Công cụ vẽ bản thiết kế mạch điện: Microsoft PowerPoint
- Lập trình firmware: Arduino IDE
- Xây dựng ứng dụng Android: <http://ai2.appinventor.mit.edu>