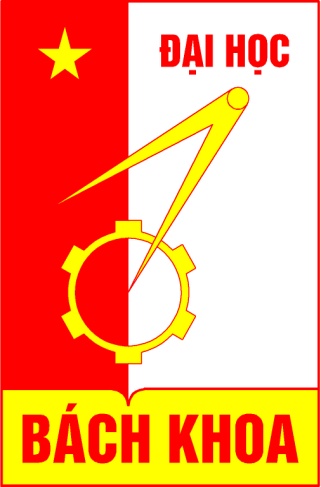
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

**---------o0o--------**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**“ĐIỀU KHIỂN XE BẰNG ĐIỆN THOẠI ANDROID”**

Giảng viên hướng dẫn : **TS. Hán Trọng Thanh**

Sinh viên thực hiện : **Vũ Văn Hài**

**ĐTVT 10- K56**

**20111468**

**Hà Nội, 03-2016**

# LỜI NÓI ĐẦU

Ngày nay nền công nghiệp nước ta đang phát triển mạnh. Các nhà máy, xí nghiệp, khu công nghiệp rộng lớn được xây dựng rất nhiều. Việc vận chuyển các thiết bị, hồ sơ tốn khá nhiều thời gian và tiêu hao nhân lực. Cần có biện pháp hỗ trợ để tiết kiệm thời gian và nhân lực nhằm tăng năng suất lao động cũng như hiệu quả công việc. Ngoài ra trong nghiên cứu khoa học và phát triển công nghệ, đôi khi con người phải làm việc trong các môi trường bị nhiễm xạ, môi trường bị nhiễm độc... Đòi hỏi phải có sự bảo hộ lao động cao. Việc con người vận chuyển từ bên ngoài vào môi trường nhiễm xạ, nhiễm độc khá mất thời gian và rườm rà vì phải thay đổi trang phục bảo hộ. Cần có phương tiện vận chuyển thay cho con người ở trong điều kiện đó.

Từ những yêu cầu thực tiễn trên em đã có ý tưởng thiết kế, chế tạo một chiếc xe điều khiển từ xa. Vấn đề đặt ra là chiếc xe phải có hiệu suất hoạt động cao, có nghĩa là thời gian nghỉ cũng như bỏ không của xe là ít. Do đó xe phải được sử dụng bởi nhiều người ở những lần khác nhau, và đương nhiên mỗi lần sử dụng chỉ có một người điều khiển nó. Vì vậy yêu cầu thiết bị điều khiển phải là thiết bị được nhiều người sử dụng.

Hiện nay điện thoại Smartphone đang rất được ưa chuộng và được sử dụng rất rộng rãi. Các thiết bị tích hợp trên Smartphone khá đa dạng, đặc biệt Smartphone có tích hợp sẵn wifi, bluetooth có khả năng truyền thông. Ngoài ra việc lập trình cho điện thoại smartphone được hỗ trợ rất nhiều. Ví dụ như Android, IOS, Window phone. Hệ điều hành phổ biến nhất là Android với hơn 82,8% trên tổng thiết bị Smartphone ( số liệu thống kê Quý 2 năm 2015).

Kết hợp những điều nêu trên em quyết định thiết kế, chế tạo một chiếc xe điều khiển từ xa được điều khiển bởi điện thoại Android.

Em xin chân thành cảm ơn sự hướng dẫn nhiệt tình của TS.Hán Trọng Thanh và PGS.TS.Đỗ Trọng Tuấn. Các thầy đã giúp đỡ em rất nhiều trong quá trình làm đề tài này.

Hà Nội, 03/2016

Sinh viên thực hiện

Vũ Văn Hài

# TÓM TẮT BÁO CÁO

Báo cáo đưa ra những yêu cầu của thực tiễn dẫn đến ý tưởng và xuất hiện đề tài, nêu lên sơ đồ tổng quan của hệ thống sau đó đi vào phân tích từng thành phần của hệ thống, từ những thành phần đó tiếp tục chia nhỏ và phân tích sâu hơn. Đưa ra các cách xây dựng hệ thống và các thành phần của hệ thống, các cơ sở lý thuyết, cách lắp đặt cũng như kết nối của từng thành phần, báo cáo nêu lên những kết quả đã đạt được, cũng như những gì chưa đạt được và đề xuất hướng đi tiếp theo.

MỤC LỤC

[LỜI NÓI ĐẦU 1](#_Toc446321254)

[TÓM TẮT BÁO CÁO 2](#_Toc446321255)

[CÁC HÌNH VẼ SỬ DỤNG TRONG BÁO CÁO 4](#_Toc446321256)

[PHẦN 1. GIỚI THIỆU TỔNG QUAN HỆ THỐNG 5](#_Toc446321257)

[1.1 Sơ đồ tổng quan hệ thống 5](#_Toc446321258)

[1.2 Phương thức hoạt động 5](#_Toc446321259)

[PHẦN 2 PHÂN TÍCH CÁC KHỐI CỦA SƠ ĐỒ TỔNG QUAN 7](#_Toc446321260)

[2.1 Kênh truyền bluetooth 7](#_Toc446321261)

[2.1.1 Khái niệm 7](#_Toc446321262)

[2.1.2 Đặc điểm 7](#_Toc446321263)

[2.1.3 Một số khái niệm 8](#_Toc446321264)

[2.1.4 Định nghĩa các liên kết vật lý trong Bluetooth 8](#_Toc446321265)

[2.1.5 Các chế độ kết nối 9](#_Toc446321266)

[2.2 Phần mềm điều khiển trên Android 9](#_Toc446321267)

[2.2.1 Giao diện điều khiển 9](#_Toc446321268)

[2.2.2 Giao diện tìm kiếm và ghép nối bluetooth 12](#_Toc446321269)

[2.2.3 Giao diện hiển thị khi tạo kết nối với thiết bị mới 13](#_Toc446321270)

[2.3 Cấu tạo chiếc xe 13](#_Toc446321271)

[2.3.1 Sơ đồ khối chiếc xe 13](#_Toc446321272)

[2.3.2 Khối nguồn 15](#_Toc446321273)

[2.3.3 Module bluetooth 15](#_Toc446321274)

[2.3.4 Arduino 19](#_Toc446321275)

[2.3.5 Mạch điều khiển động cơ 24](#_Toc446321276)

[2.3.6 Cảm biến siêu âm 26](#_Toc446321277)

[2.3.7 Kết nối giữa các module và mạch kiểm thử 28](#_Toc446321278)

[PHẦN 3. KẾT LUẬN 30](#_Toc446321279)

[3.1 Kết quả đạt được 30](#_Toc446321280)

[3.2 Vấn đề chưa được giải quyết 30](#_Toc446321281)

[3.3 Hướng phát triển 31](#_Toc446321282)

[TÀI LIỆU THAM KHẢO 31](#_Toc446321283)

# CÁC HÌNH VẼ SỬ DỤNG TRONG BÁO CÁO

[**Hình 1‑1 Sơ đồ tổng quan của hệ thống** 4](#_Toc445804805)

[**Hình 2-1 Giao diện điều khiển phần mềm trên Android** 10](#_Toc445804805)

[**Hình 2‑2 Giao diện tìm kiếm và ghép nối bluetooth** 12](#_Toc445804805)

[**Hình 2‑3 Giao diện hiển thị khi tạo kết nối với thiết bị mới** 13](#_Toc445804805)

[**Hình 2‑4 Sơ đồ khối của chiếc xe** 14](#_Toc445804805)

[**Hình 2‑5 Module Bluetooth** 15](#_Toc445804805)

[**Hình 2‑6 Sơ đồ khối Module Bluetooth HC 06** 17](#_Toc445804805)

[**Hình 2‑7 Sơ đồ các linh kiện của Arduino Mega2560** 21](#_Toc445804805)

[**Hình 2‑8 Xung PWM** 24](#_Toc445804805)

[**Hình 2‑9 Sơ đồ nguyên lý mạch điều khiển động cơ** 25](#_Toc445804805)

[**Hình 2‑10 Nguyên lý cảm biến siêu âm** 26](#_Toc445804805)

[**Hình 2‑11 Cảm biến siêu âm SRF05** 27](#_Toc445804805)

[**Hình 2‑12 Nguyên lý hoạt động của SRF05** 28](#_Toc445804805)

[**Hình 2‑13 Kết nối giữa HC 06 với Arduino**  29](#_Toc445804805)

[**Hình 2‑14 Kết nối SRF05 với Arduino** 29](#_Toc445804805)

[**Hình 2‑15 Mạch kiểm thử trên bo cắm** 30](#_Toc445804805)

# PHẦN 1. GIỚI THIỆU TỔNG QUAN HỆ THỐNG

## Sơ đồ tổng quan hệ thống



**Hình 1‑0‑1 Sơ đồ tổng quan của hệ thống**

**Điện thoại:** Có nhiệm vụ thiết lập kết nối bluetooth, gửi các tín hiệu điều khiển tới kênh truyền. Nhận tín hiệu phản hồi từ kênh truyền, xử lý và hiển thị thông tin lên màn hình.

**Kênh truyền bluetooth:** Có nhiệm vụ chuyển tín hiệu giữa điện thoại và chiếc xe.

**Xe điều khiển từ xa:** Có nhiệm vụ nhận tín hiệu từ kênh truyền, xử lý tín hiệu nhận được, sau đó điều khiển hướng đi cho phù hợp với tín hiệu. Gửi thông tin phản hồi vào kênh truyền.

## 1.2 Phương thức hoạt động

Khi người dùng muốn điều khiển chiếc xe, sẽ phải bật phần mềm điều khiển xe đã được cài trên điện thoại trước đó. Người dùng phải bật bluetooth của điện thoại thông qua giao diện phần mềm hoặc bật từ cài đặt hệ thống. Tiếp theo người dùng sẽ quét các thiết bị bluetooth đang bật trong phạm vi điện thoại có thể kết nối. Nếu tìm thấy thiết bị bluetooth của xe sẽ kết nối với nó bằng cách click chọn thiết bị sau đó nhập password, điện thoại sẽ thiết lập kết nối với chiếc xe. Sau khi kết nối được thiết lập, người dùng có thể điều khiển chiếc xe đi theo ý muốn bằng các nút bấm được thiết lập trên giao diện.

# PHẦN 2 PHÂN TÍCH CÁC KHỐI CỦA SƠ ĐỒ TỔNG QUAN

## Kênh truyền bluetooth

### 2.1.1 Khái niệm

* Bluetooth là công nghệ không dây cho phép các thiết bị điện, điện tử giao tiếp với nhau trong khoảng cách ngắn, bằng sóng vô tuyến qua băng tần chung ISM (Industrial, Scientific, Medical) trong dải tần số 2.40 – 2.48 GHz. Đây là dải băng tần không cần đăng ký, được dành riêng để dùng cho các thiết bị không dây trong công nghệ, khoa học, y tế.
* Bluetooth được thiết kế nhằm mục đích thay thế dây cable giữa máy tính và các thiết bị truyền thông cá nhân, kết nối vô tuyến giữa các thiết bị điện tử lại với nhau một cách thuận lợi với giá thành rẻ.
* Khi được kích hoạt, Bluetooth có thể tự động định vị những thiết bị khác có chung công nghệ trong vùng xung quanh và bắt đầu kết nối với chúng. Nó được định hướng sử dụng cho việc truyền dữ liệu lẫn tiếng nói.

### 2.1.2 Đặc điểm

* Tiêu thụ năng lượng thấp, cho phép ứng dụng được trong nhiều loại thiết bị.
* Giá thành rẻ.
* Khoảng cách giao tiếp cho phép: Tùy tần số sử dụng, tần số càng cao suy hao càng lớn, khoảng cách truyền đi càng giảm. Trong các điều kiện lý tưởng thì Bluetooth có thể truyền tối đa là 100m và truyền gần không dưới 10m.
* Bluetooth sử dụng băng tần không đăng ký 2.4 GHz trên dải băng tần ISM. Tốc độ truyền dữ liệu có thể đạt tới mức tối đa 1 Mbps (tần số càng cao tốc độ truyền càng lớn) mà các thiết bị không cần phải thấy trực tiếp nhau.
* Bluetooth kết nối một ứng dụng này với một ứng dụng khác thông qua các chuẩn chung. Do đó có thể độc lập về phần cứng cũng như hệ điều hành sử dụng. Từ đó dễ dàng phát triển các ứng dụng.
* Một kết nối được chia làm 10 kênh, trong đó 3 kênh để truyền tiếng nói, 7 kênh để truyền dữ liệu.
* An toàn và bảo mật: được tích hợp với sự xác nhận và mã hóa.
* Tương thích cao, được nhiều nhà sản xuất phần cứng cũng như phần mềm hỗ trợ.

### 2.1.3 Một số khái niệm

* *Master Unit*: Là thiết bị duy nhất trong 1 Piconet, Master thiết lập đồng hồ đếm xung kiểu bước nhảy (hopping) để đồng bộ tất cả các thiết bị trong cùng Piconet mà nó đang quản lý, thường là thiết bị đầu tiên chuyển đổi dữ liệu. Master cũng quyết định số kênh truyền thông. Mỗi Piconet có một kiểu hopping duy nhất.
* *Slave Unit*: Là toàn bộ các thiết bị không phải Master. Được quản lý bởi Master. Có khả năng trao đổi dữ liệu với Master. Nếu muốn trao đổi với các Slave khác thì phải thông qua Master.
* *Piconet*: Piconet là tập hợp các thiết bị được kết nối thông qua kỹ thuật Bluetooth theo mô hình Ad-Hoc (kiểu mạng được thiết lập cho nhu cầu truyền dữ liệu hiện hành và tức thời, tốc độ nhanh và kết nối sẽ tự động hủy sau khi truyền xong). Trong 1 Piconet thì chỉ có 1 thiết bị là Master. Đây thường là thiết bị đầu tiên tạo kết nối, nó có vai trò quyết định số kênh truyền thông và thực hiện đồng bộ giữa các thành phần trong Piconet, các thiết bị còn lại là Slave. Đó là các thiết bị gửi yêu cầu đến Master.

Cách hình thành 1 Piconet: 1 Piconet bắt đầu với 2 thiết bị kết nối với nhau, như laptop PC với 1 Mobilephone. Giới hạn 8 thiết bị trong 1 Piconet (3 bit MAC cho mỗi thiết bị). Tất cả các thiết bị Bluetooth đều ngang hàng và mang chức năng xác định. Tuy nhiên khi thành lập 1 Piconet, 1 thiết bị sẽ đóng vai Master để đồng bộ về tần số và thời gian truyền phát, và các thiết bị khác là Slave.

### 2.1.4 Định nghĩa các liên kết vật lý trong Bluetooth

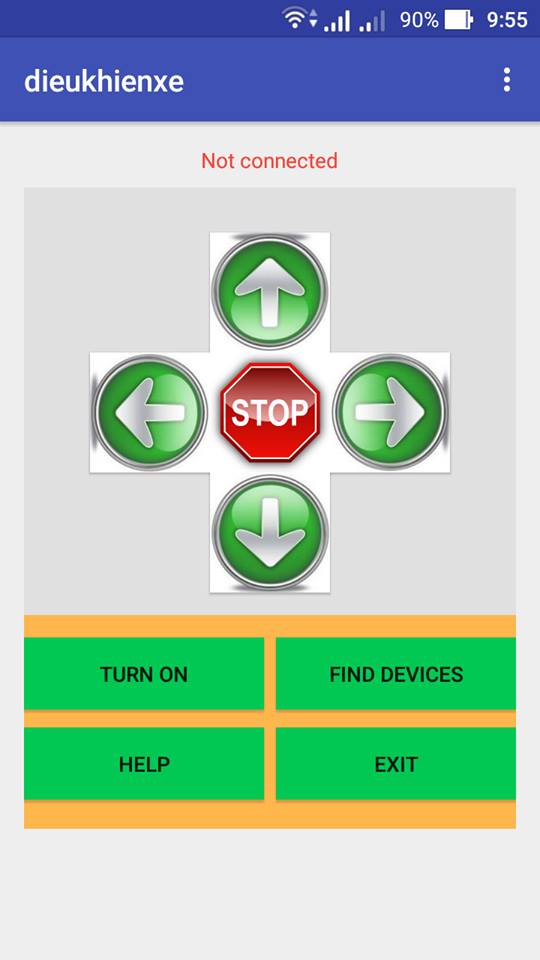
* Asynchronous connectionless (ACL): được thiết lập cho việc truyền dữ liệu, những gói dữ liệu cơ bản. Là một kết nối point-to-multipoint giữa Master và tất cả các Slave tham gia trong piconet. Chỉ tồn tại duy nhất một kết nối ACL. Chúng hỗ trợ những kết nối chuyển mạch gói đối xứng và không đối xứng, Những gói tin đa khe dùng ACL link và có thể đạt tới khả năng truyền tối đa 723 Kbps ở một hướng và 57.6 Kbps ở hướng khác. Master điều khiển độ rộng băng tần của ACL link và sẽ quyết định xem trong 1 Piconet, 1 Slave có thể dùng băng tần rộng bao nhiêu. Những gói tin broadcast truyền bằng ACL link, từ Master đến tất cả các Slave. Hầu hết các gói tin ACL đều có thể truyền lại.
* Synchronous connection-oriented (SCO): hỗ trợ kết nối đối xứng, chuyển mạch mạch, point-to-point giữa một Master và một Slave trong 1 Piconet. Kết nối SCO chủ yếu dùng để truyền dữ liệu tiếng nói. Hai khe thời gian liên tiếp đã được chỉ định trước sẽ được dành riêng cho SCO link. Dữ liệu truyền theo SCO link có tốc độ 64 Kbps. Master có thể hỗ trợ tối đa 3 kết nối SCO đồng thời. SCO packet không chứ CRC (Cyclic Redundancy Check) và không bao giờ truyền lại. Liên kết SCO được thiết lập chỉ sau 1 thiết kế ACL đầu tiên được thiết lập.

### 2.1.5 Các chế độ kết nối

* Active mode: trong chế độ này, thiết bị Bluetooth tham gia vào hoạt động của mạng. Thiết bị Master sẽ điều phối lưu lượng và đồng bộ hóa cho các thiết bị Slave.
* Sniff mode: là 1 chế độ tiết kiệm năng lượng của thiết bị đang ở trạng thái active. Ở Sniff mode, thiết bị Slave lắng nghe tín hiệu từ mạng với tần số giảm hay nói cách khác là giảm công suất. Tần số này phụ thuộc vào tham số của ứng dụng. Đây là chế độ ít tiết kiệm năng lượng nhất trong 3 chế độ tiết kiệm năng lượng.
* Hold mode: là 1 chế độ tiết kiệm năng lượng của thiết bị đang ở trạng thái active. Master có thể đặt chế độ Hold mode cho Slave của mình. Các thiết bị có thể trao đổi dữ liệu ngay lập tức ngay khi thoát khỏi chế độ Hold mode. Đây là chế độ tiết kiệm năng lượng trung bình trong 3 chế độ tiết kiệm năng lượng.
* Park mode: là chế độ tiết kiệm năng lượng của thiết bị vẫn còn trong mạng nhưng không tham gia vào quá trình trao đổi dữ liệu (inactive). Thiết bị ở chế độ Park mode bỏ địa chỉ MAC, chỉ lắng nghe tín hiệu đồng bộ hóa và thông điệp broadcast của Master. Đây là chế độ tiết kiệm năng lượng nhất trong 3 chế độ tiết kiệm năng lượng.

## Phần mềm điều khiển trên Android

### Giao diện điều khiển



**Hình 2-1 Giao diện điều khiển của phần mềm trên Android**

Cách sử dụng:

* Khi ta bấm các nút mũi tên lên, sang phải, xuống, sang trái điện thoại sẽ gửi các ký tự tương ứng là 1, 2, 3, 4 đến cho Arduino. Arduino sẽ có nhiệm vụ điều khiển sao cho xe di chuyển đúng chiều như chiều mũi tên.
* Phím Stop: Điện thoại gửi ký tự 0 đến cho Arduino. Arduino sẽ điều khiển xe dừng lại.
* Khi thiết bị đang bật bluetooth thì phím TURN OFF sẽ xuất hiện, khi nhấn vào đó phần mềm sẽ tắt thiết bị bluetooth trên điện thoại. Nếu bluetooth chưa được bật thì sẽ hiển thị phím TURN ON, khi nhấn vào đó phần mềm sẽ bật bluetooth của điện thoại lên.
* Phím FIND DEVICES: Bluetooth đang được bật, nếu nhấn vào đó sẽ hiển thị ra giao diện để tìm kiếm các thiết bị đang bật bluetooth khác.
* Phím HELP: Cung cấp thông tin cách sử dụng và chức năng các phím.
* Phím EXT: Thoát khỏi chương trình.

### Giao diện tìm kiếm và ghép nối bluetooth



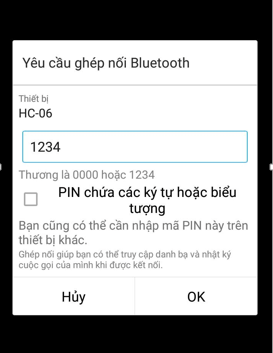
**Hình 2-2 Giao diện tìm kiếm và ghép nối bluetooth**

Sau khi tiến hành tìm kiếm. Các thiết bị đã từng ghép nối trước đó sẽ hiển thị bên dưới dòng Paired Devices. Các thiết bị phát hiện mới sẽ hiển thị dưới dòng New Devices. Để cập nhật lại danh sách các thiết bị trong phạm vi kết nối thì nhấn nút **SCAN**.

Khi muốn kết nối với thiết bị có trong danh sách đã từng kết nối chỉ cần nhấn vào tên thiết bị, sau đó kết nối sẽ được thiết lập.

Nếu muốn kết nối với thiết bị mới sẽ có giao diện hiện ra yêu cầu người dùng nhập password để kết nối.

### Giao diện hiển thị khi tạo kết nối với thiết bị mới



**Hình 2-3 Giao diện hiển thị khi tạo kết nối với thiết bị mới**

Khi nhấn vào tên thiết bị mới để kết nối, giao diện này sẽ hiện ra yêu cầu người sử dụng nhập đúng password của thiết bị kia yêu cầu mới có thể kết nối. Nếu người dùng nhập đúng kết nối được tạo thành, nếu sai quay lại giao diện tìm kiếm và ghép nối trước đó.

## Cấu tạo chiếc xe

### 2.3.1 Sơ đồ khối chiếc xe



**Hình 2-4 Sơ đồ khối của chiếc xe**

Nhiệm vụ chính của các khối:

**Khối nguồn:** Cung cấp năng lượng cho mạch hoạt động.

**Module bluetooth:** Nhận tín hiệu điều khiển từ kênh truyền sau đó chuyển tín hiệu sang Arduino. Đồng thời cũng làm nhiệm vụ chuyển tín hiệu phản hồi từ Arduino đến kênh truyền.

**Arduino:** Xử lý tín hiệu điều khiển nhận được, gửi tín hiệu xung PWM (Pulse Width Modulation) đến mạch điều khiển động cơ. Đồng thời nhận tín hiệu gửi đến từ cảm biến siêu âm để xử lý, sau đó gửi tín hiệu phản hồi vào kênh truyền thông qua module bluetooth.

**Mạch điều khiển động cơ:** Nhận tín hiệu xung PWM từ Arduino sau đó điều chỉnh dòng điện từ nguồn cấp vào động cơ.

**Cảm biến siêu âm:** Phát tín hiệu siêu âm xác định vật cản, sau đó gửi tín hiệu cho Arduino.

### 2.3.2 Khối nguồn

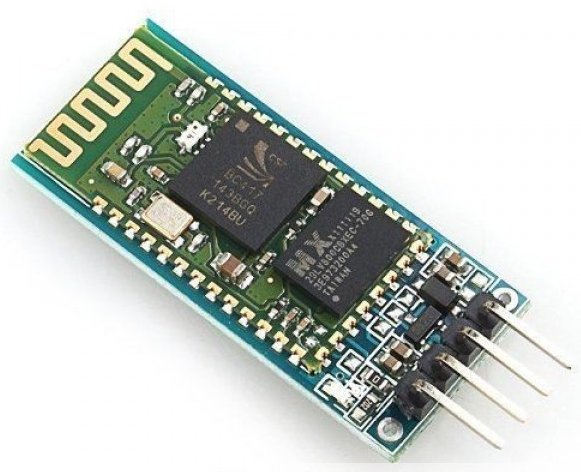
#### 2.3.2.1 Nguồn Acquy

Nguồn Acquy cấp nguồn 12V cho 2 động cơ thông qua mạch điều khiển động cơ. Acquy được sử dụng loại 12V-7.5A

#### 2.3.2.2 Nguồn pin 9V

Pin cấp nguồn cho Arduino hoạt động, đồng thời cấp nguồn cho module bluetooth thông qua Arduino.

### 2.3.3 Module bluetooth



**Hình 2-5 Module Bluetooth**

#### 2.3.3.1 Chức năng

Module bluetooth có chức năng nhận và phát sóng bluetooth để tương tác với thiết bị Smart phone. Dữ liệu nhận được từ module này chuyển đến Arduino hay dữ liệu từ Arduino chuyển đến module được truyền nối tiếp thông qua dây dẫn. Sau đây chúng ta sẽ xem xét module bluetooth HC06-module mà ta đã chọn để thực hiện.

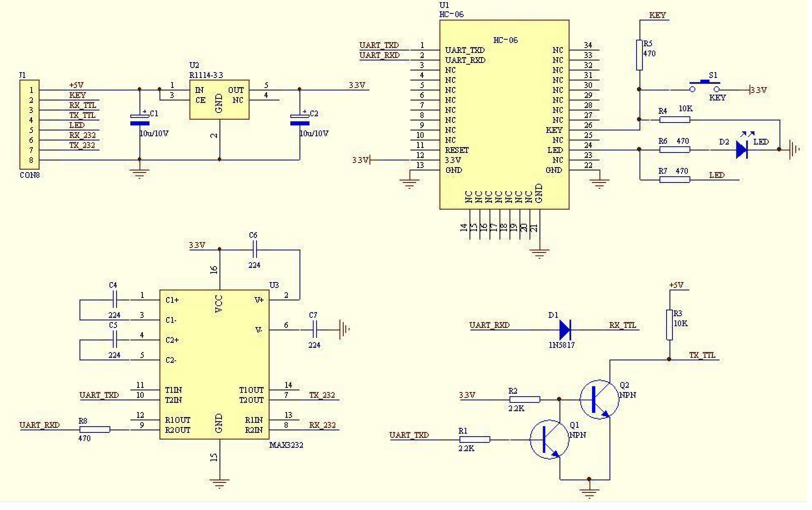
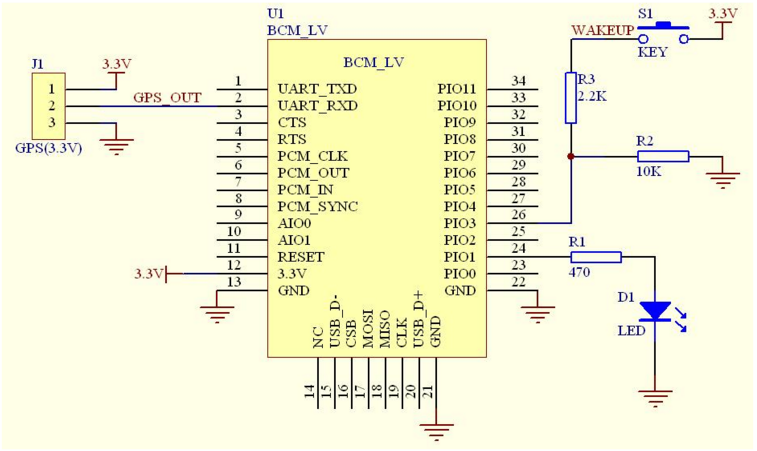
#### 2.3.3.2 Thông số kỹ thuật

* Sử dụng CSR mainstream bluetooth chip, bluetooth V2.0 protocol standards.
* Điện thế hoạt động của UART từ 3.3-5 V.
* Dòng điện hoạt động khi đang kết nối là 30 mA, sau khi kết nối và hoạt động truyền nhận dữ liệu bình thường là 8mA.
* Baudrate UART là giá trị đặt sẵn, có thể chọn một trong những giá trị: 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200.
* Kích thước module chính: 28mm×15mm×2.35mm.
* Dải tần sóng hoạt động: 2.4 GHz.
* Nhiệt độ lưu trữ: -40oC đến 85oC.

**Một số thiết lập mặc định:**

* Baudrate 9600, N, 8, 1.
* Mật khẩu: 1234.

#### 2.3.3.3 Sơ đồ khối của module bluetooth HC 06



**Hình 2-6 Sơ đồ khối Module bluetooth HC 06**

#### 2.3.3.4 Giao tiếp với Module bluetooth HC 06

* Module bluetooth HC06 được thiết kế để có thể hoạt động được ở mức điện áp 3.3V đến 5V. Ta có thể kết nối module này với máy tính thông qua cáp nối USB. Khi đó HC06 được sử dụng như một cổng COM ảo, việc truyền nhận với COM ảo giống như việc truyền nhận dữ liệu trực tiếp với UART trên module.
* Có một số điểm cần lưu ý như, khi thay đổi Baudrate cho COM ảo không làm thay đổi baudrate của UART, baudrate UART chỉ có thể thay đổi bằng AT command trên module. Module HC-06 được setup mặc định là Slave không thể thay đổi được nên chỉ có thể giao tiếp với các thiết bị bluetooth ở dạng master như Smart phone, HC-05 master... Hai module được setup là Slave không thể giao tiếp được với nhau.

#### 2.3.3.5 Thay đổi thông số cho Module bluetooth

Thay đổi thông số cho module bluetooth bằng cách kết nối module bluetooth với máy tính bằng cáp USB sau đó dùng AT command để thiết lập thông số.

Các bước thực hiện:

* Kết nối Module với USB TO COM Set KEY xuống mức thấp ( GND)
* Cấp nguồn cho Module ( Lấy nguồn từ USB TO COM)
* Chuyển Key lên 3.3V
* Test:

Gõ lệnh: AT

Màn hình trả về OK

* Thay đổi tốc độ:

Gõ: AT+BAUD2

Màn hình trả về OK2400(module đã được thiết lập tốc độ 2400).

* Tương tự với các tốc độ khác ta dùng lệnh AT+BAUDx, với x là các số tương ứng với tốc độ baud dưới đây.

1: 1200.

2: 2400.

3: 4800.

4: 9600.

5: 19200.

6: 38400.

7: 57600.

8: 115200.

9: 230400

A: 460800.

B: 921600.

C: 1382400.

Lưu ý module hoạt động ổn định với tốc độ ≤ 115200.

* Thay đổi tên:

Gõ : AT+NAMEname

Màn hình trả về: Okname

Chú ý tên không vượt quá 20 ký tự.

* Thay đổi pass:

Gõ AT+PINxxxx

Màn hình trả về: Oksetpin

xxxx là mật khẩu, độ dài 4 ký tự.

### 2.3.4 Arduino

#### 2.3.4.1 Chức năng của Arduino trong mạch

* Arduino có nhiệm vụ nhận các tín hiệu điều khiển mà module bluetooth thu được, sau đó xử lý các tín hiệu, cấp xung PWM đến mạch điều khiển động cơ để điều khiển xe theo ý muốn.
* Nhận các tín hiệu từ cảm biến siêu âm, sau đó xử lý các tín hiệu rồi truyền tín hiệu cho module bluetooth đưa tín phản hồi vào kênh truyền.

#### 2.3.4.2 Giới thiệu chung về Arduino

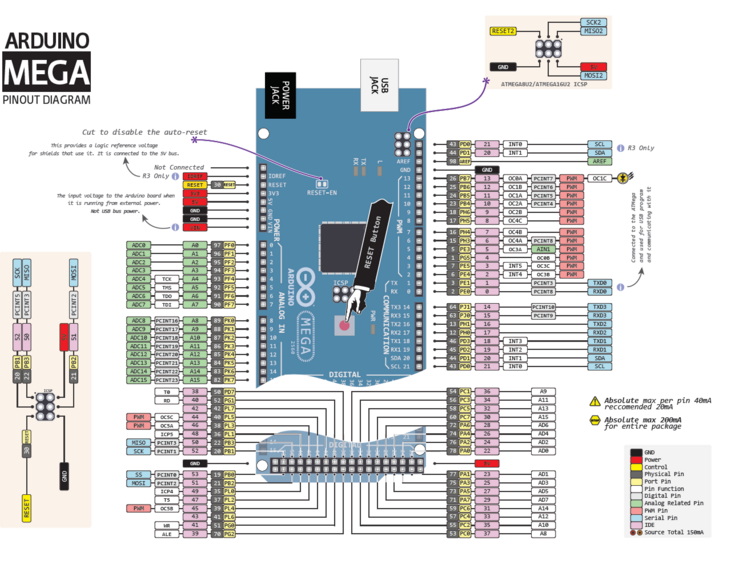
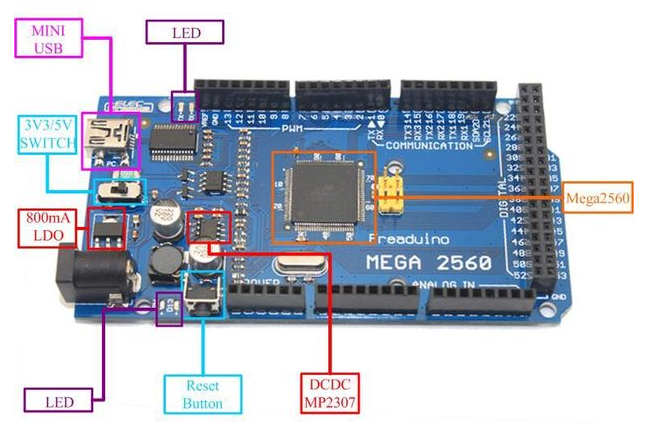
* Arduino là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, các chân đầu vào analog, các chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau.
* Arduino được giới thiệu vào năm 2005. Đi cùng với nó là môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân thông thường và cho phép người dùng viết các chương trình cho Arduino bằng ngôn ngữ C hoặc C++.
* Arduino có các kết nối tiêu chuẩn, cho phép người dùng kết nối CPU của board với các module thêm vào và có thể dễ dàng chuyển đổi. Arduino chính thức thường sử dụng các dòng chip megaAVR, đặc biệt là Atmega8, ATmega168, ATmega328, ATmega1280, và ATmega2560. Hầu hết các mạch có bộ điều chỉnh tuyến tính 5V và một thạch anh dao động 16MHz.
* Các board Arduino hiện tại được lập trình thông qua cổng USB, thực hiện thông qua chip chuyển đổi USB-to-serial như là FTDI, FT232. Vài biến thể, như Arduino Mini và Boarduino không chính thức, sử dụng một board adapter hoặc cáp nối USB-to-serial có thể tháo rời được, Bluetooth hoặc các phương thức khác (Khi sử dụng một công cụ lập trình vi điều khiển truyền thống thay vì ArduinoIDE, công cụ lập trình AVR ISP tiêu chuẩn sẽ được sử dụng.)
* Board Arduino sẽ đưa ra hầu hết các chân I/O của vi điều khiển để sử dụng cho những mạch ngoài. Hai loại chính của các chân I/O là chân I/O kỹ thuật số (trong đó có các chân có thể tạo xung PWM-điều chế độ rộng xung, hoặc không) và chân input analog, có thể sử dụng như là chân I/O.

#### 2.3.4.3 Arduino Mega2560

Trong đề tài em sử dụng Arduino Mega2560 để điều khiển, vì thế em sẽ tập trung phân tích Arduino Mega2560.

* Các thành phần chính của Arduino Mega2560:
  + Bộ xử lý trung tâm là vi điều khiển AVR ATmega2560.
  + 54 chân digital trong đó 15 chân có thể được sử dụng như các chân PWM.
  + 16 đầu vào analog.
  + 4 UARTs (cổng nối tiếp phần cứng).
  + 1 thạch anh 16 MHz.
  + 1 cổng kết nối USB.
  + 1 jack cắm điện.
  + 1 đầu ICSP.
  + 1 nút reset.

Sơ đồ linh kiện của Arduino Mega2560 được trình bày ở hình dưới đây



**Hình 2-7 Sơ đồ các linh kiện của Arduino Mega2560**

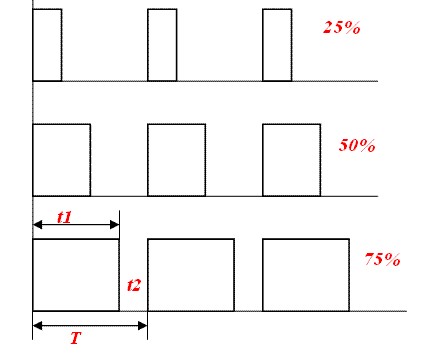
* Các thông số của Arduino Mega 2560

**Bảng 2-1 Các thông số cơ bản của Arduino Mega2560**

|  |  |
| --- | --- |
| Vi điều khiển | Atmega2560 |
| Điện áp hoạt động | 5V – DC |
| Tần số hoạt động | 16 MHz |
| Điện áp vào khuyên dùng | 7-12V – DC |
| Điện áp vào giới hạn | 6-20V – DC |
| Số chân Digital I/O | 54 ( 15 chân PWM) |
| Số chân Analog | 16 |
| Dòng tối đa trên mỗi chân I/O | 40 mA |
| Dòng ra tối đa (3.3V) | 50 mA |
| Bộ nhớ flash | 256 KB (Atmega2560) với 8KB dùng bởi bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |

* Năng lượng:
  + Arduino Mega2560 có thể được cấp nguồn 5V thông qua kết nối USB hoặc với một nguồn cung cấp điện bên ngoài với điện áp khuyên dùng là 7-12V DC và giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lý nếu không có sẵn nguồn từ cổng USB.
  + Các chân năng lượng:
    - GND (Ground): Arduino Mega2560 có 5 chân GND, khi ta sử dụng những nguồn điện riêng biệt thì những chân này phải nối với nhau.
    - 5V: Có 3 chân 5V, cấp điện áp 5V đầu ra.
    - 3.3V: Có 1 chân 3.3V, cấp điện áp 3.3V, dòng cho phép tối đa ở chân này là 50mA.
    - Vin (Voltage Input): 1 chân Vin, để cấp nguồn ngoài cho Arduino Mega2560, ta nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
    - IOREF: 1 chân IOREF, điện áp hoạt động của vi điều khiển trên Arduino Mega2560 có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy ta không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
    - RESET: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.
* Các cổng vào ra cơ bản
  + Serial: 0 (RX) và 1 (TX) nối tiếp 1; 19 (RX) và 18 (TX) nối tiếp 2; 17 (RX) và 16 (TX) nối tiếp 3; 15 (RX) và 14 (TX) nối tiếp 4; được sử dụng để nhận (RX) và truyền (TX) dữ liệu nối tiếp. Pins 0 và 1 cũng được kết nối với các chân tương ứng của ATmega16U2 USB-to-TTL nối tiếp chip.
  + Arduino Mega2560 có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit (0 → 210-1) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân AREF trên board, ta có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu ta cấp điện áp 2.5V vào chân này thì ta có thể dùng các chân analog để đo điện áp trong khoảng từ 0V → 2.5V với độ phân giải vẫn là 10bit.
  + Chân PWM: các chân 2-13 và 44-46 cho phép ta xuất ra xung PWM với độ phân giải 8bit. Ta sẽ cấp xung PWM vào mạch điều khiển động cơ để điều khiển. Vì thế ta sẽ phân tích một chút về xung PWM.

PWM là sự điều chế độ rộng của chuỗi xung vuông, dẫn đến sự thay đổi điện áp ra. Các PWM khi biến đổi thì có cùng một tần số và chỉ khác nhau về độ rộng của sườn âm hay sườn dương.



**Hình 2-8 Xung PWM**

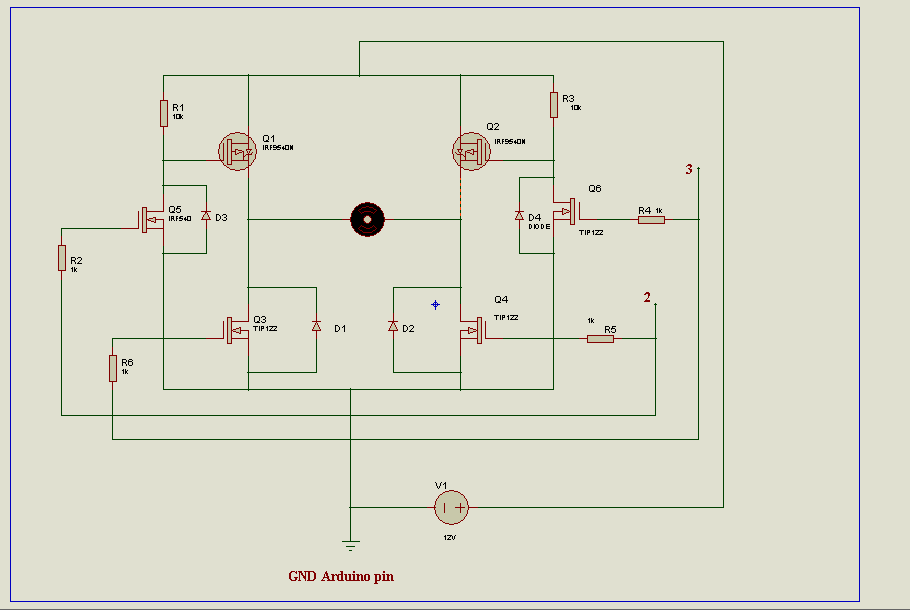
### 2.3.5 Mạch điều khiển động cơ

#### 2.3.5.1 Chức năng của mạch điều khiển động cơ

* Mạch điều khiển động cơ có nhiệm vụ thay đổi tốc độ động cơ, dựa vào việc thay đổi tốc độ động cơ để điều chỉnh hướng đi của xe.
* Có khả năng đảo chiều quay của động cơ khiến xe có khả năng chạy lùi.

#### 2.3.5.2 Sơ đồ nguyên lý của mạch điều khiển động cơ

Ta sẽ phân tích sơ đồ nguyên lý điều khiển một động cơ, với động cơ còn lại ta làm tương tự. Mạch điều khiển động cơ sử dụng mạch cầu H để điều khiển chiều động cơ, với một động cơ sử dụng 2 chân xung PWM để cấp xung PWM điều khiển mạch.



**Hình 2-9 Sơ đồ nguyên lý của mạch điều khiển động cơ**

#### 2.3.5.3 Phân tích sơ đồ nguyên lý

2 chân được đánh số trong sơ đồ được nối với các chân tương ứng của Arduino.

Chân GND của Arduino được nối với chân âm của nguồn.

Khi các chân 2, 3 chưa được cấp xung PWM, không có sự chênh lệch điện áp giữa chân G và chân S của các mosfet, do đó các mosfet không cho dòng điện chạy qua, 2 động cơ đứng yên.

Khi chân 2 được cấp xung PWM, Q4 dẫn, Q5 dẫn làm cho điện áp chân S của Q1 cao hơn điện áp chân G (Q1 là mosfet loại p) dẫn đến Q1 dẫn, động cơ quay theo chiều kim đồng hồ.

Chân 3 được cấp xung PWM, Q3 và Q6 dẫn, Q6 dẫn làm điện áp chân S của Q2 cao hơn điện áp chân G (Q2 là mosfet loại p) làm cho Q2 dẫn. động cơ quay ngược chiều kim đồng hồ.

Tương tự với động cơ còn lại, ta sẽ có mạch điều khiển hai động cơ.

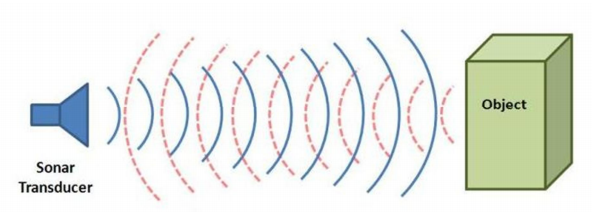
Khi cấp xung PWM có thời gian xung dương càng lớn trong một chu kỳ thì thời gian dẫn của các mofest càng lớn, tốc độ động cơ quay càng nhanh.

Từ các trường hợp trên ta sẽ lập trình cho arduino cấp xung PWM sao cho động cơ quay phù hợp với tín hiệu điều khiển.

### 2.3.6 Cảm biến siêu âm

#### 2.3.6.1 Nguyên lý hoạt động của cảm biến siêu âm

Cảm biến siêu âm là một loại cảm biến khoảng cách dựa trên nguyên lý thu phát siêu âm. Cảm biến gồm một bộ phát và một bộ thu sóng siêu âm. Sóng siêu âm từ đầu phát truyền đi trong không khí, gặp vật cản sẽ phản xạ trở lại và được đầu thu ghi lại.

**Hình 2-10 Nguyên lý cảm biến siêu âm**

#### 2.3.6.2 Cấu tạo và cách thức hoạt động của cảm biến siêu âm SRF05



**Hình 2-11 Cảm biến siêu âm SRF05**

Chức năng của các chân:

**VCC:** Cấp nguồn cho cảm biến.

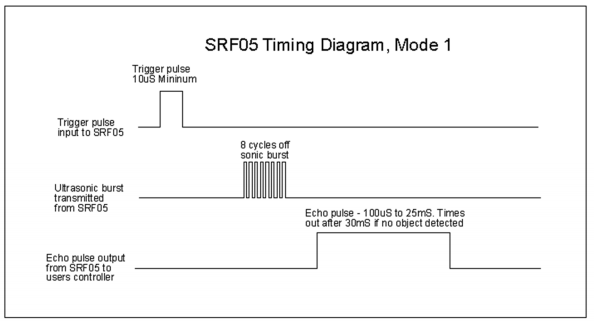
**Triger:** Kích hoạt quá trình phát sóng âm.

**Echo:** Bình thường sẽ ở trạng thái 0V, được kích hoạt lên 5V ngay sau khi có tín hiệu trả về , sau đó trở về 0V.

**GND:**  Nối với cực âm của mạch.

**OUT:** Không sử dụng.

Để điều khiển SRF05 ta cần cấp cho chân Triger một xung điều khiển với độ rộng tối thiểu 10µs. Sau một khoảng thời gian, đầu phát sóng sẽ phát ra sóng siêu âm. Vi xử lý tích hợp trên module sẽ tự xác định thời điểm phát sóng siêu âm và thu sóng siêu âm. Vi xử lý này đưa kết quả ra chân ECHO. Độ rộng xung vuông ECHO tỉ lệ với khoảng cách từ cảm biến đến vật thể. Khoảng cách đo được của SRF05 nằm trong khoảng từ 4cm đến 300cm.

**Hình 2-12 Nguyên lý hoạt động của SRF05**

### 2.3.7 Kết nối giữa các module và mạch kiểm thử

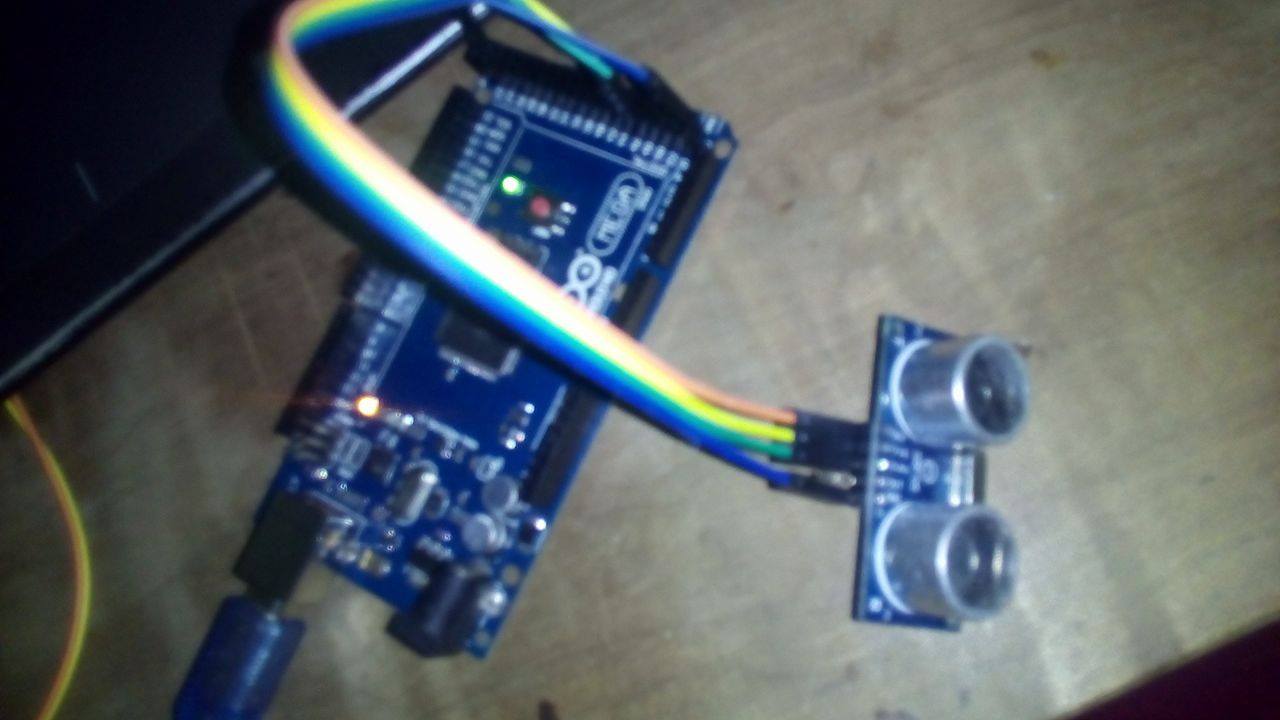
#### 2.3.7.1 Kết nối giữa module bluetooth HC06 với Arduino.

* GND của HC-06 được nối với các chân GND của Arduino.
* VCC của HC-06 được nối với chân 5V của Arduino. HC-06 được cấp nguồn 5V DC.
* RX của HC-06 được nối với chân TX ( chân 1) của Arduino.
* TX của HC-06 được nối với chân RX (chân 0) của Arduino.

**Hình 2-13 Kết nối giữa HC 06 với Arduino**

#### 2.3.7.2 Kết nối giữa SRF05 với Arduino

* VCC của SRF05 kết nối với chân 5V của Arduino.
* GND của SRF05 kết nối với chân GND của Arduino.
* Triger của SRF05 kết nối với một chân pin của Arduino (chọn chân 44).
* Echo của SRF05 kết nối với một chân pin của Arduino (chọn chân 46).



**Hình 2-14 Kết nối SRF05 với Arduino**

#### 2.3.7.3 Mạch kiểm thử trên bo cắm



**Hình 2-15 mạch kiểm thử trên bo cắm**

# PHẦN 3. KẾT LUẬN

## 3.1 Kết quả đạt được

* Xây dựng xong phần mềm chạy trên Android.
* Lắp mạch kiểm thử trên bo cắm với động cơ 9V.
* Tiến hành chạy trên động cơ công suất lớn.
* Khi chạy phần mềm trên Android và mạch kiểm thử, động cơ quay theo đúng theo tín hiệu điều khiển khi điều khiển trên Android.
* Khi động cơ đang quay ổn định, tín hiệu điều khiển yêu cầu động cơ quay ngược lại sẽ không cấp dòng ngược chiều ngay lập tức mà ngắt dòng 1s sau đó cấp ngược lại.
* Sau một thời gian nhất định (5s) nếu không có tín hiệu điều khiển đến động cơ sẽ dừng lại, tránh trường hợp mất kết nối động cơ chạy mãi, không kiểm soát được.

## 3.2 Vấn đề chưa được giải quyết

* Chưa sử dụng cảm biến khoảng cách để xác định vật cản.
* Động cơ khi điều khiển chạy thẳng còn di chuyển cong do ma sát của hai động cơ là khác nhau.
* Chưa làm mạch in.

## 3.3 Hướng phát triển

* Điều chỉnh để động cơ có thể chạy thẳng.
* Sử dụng cảm biến khoảng cách để dò tìm vật cản.
* Gửi tín hiệu phản hồi cho Android.
* Xây dựng định vị xe qua Internet.

# TÀI LIỆU THAM KHẢO

[1] <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> Truy nhập cuối cùng ngày 14/03/2016.

[2] <http://www.titans.com.vn/public_files/AN/HMAR02.pdf> Truy nhập cuối cùng ngày 14/03/2016.

[3] PGS.TS.Nguyễn Đức Thuận, THS.Vũ Duy Hải, *Giáo trình cấu kiện điện tử*, Trường Đại Học Bách Khoa Hà Nội, 2009.

[4] <https://www.arduino.cc/en/Main/ArduinoBoardMega2560> Truy nhập cuối cùng ngày 14/03/2016.

[5] <http://www.hocavr.com/index.php/hardware/hbridge> Truy nhập cuối cùng ngày 14/03/2016.