

## **LỜI CAM ĐOAN**

Đề tài này là do tôi tự thực hiện dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Người thực hiện đề tài

**Lê Thanh Cao**

**Nguyễn Thanh Tuấn**

## LỜI CẢM ƠN

Để hoàn thành đồ án tốt nghiệp này, lời đầu tiên chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy **ThS. Nguyễn Văn Hiệp**, đã tận tình hướng dẫn, chia sẻ nhiều kinh nghiệm quý báu trong suốt quá trình chúng em thực hiện đồ án.

Chúng em cũng xin chân thành cảm ơn quý thầy, cô trường Đại Học Sư Phạm Kỹ Thuật TP.HCM, đặc biệt là các thầy cô trong khoa Điện-Điện tử, đã nhiệt tình giúp đỡ, truyền đạt kiến thức trong suốt quá trình học tập của nhóm tại trường. Vốn kiến thức được tiếp thu trong quá trình học tập không chỉ là nền tảng cho quá trình thực hiện đồ án tốt nghiệp mà còn là hành trang quý báu cho chúng em lập nghiệp sau này.

Bên cạnh đó, chúng em xin cảm ơn sự hỗ trợ và giúp đỡ của bạn bè trong thời gian học tập tại Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh và trong quá trình hoàn thành luận văn tốt nghiệp này.

Cuối cùng, chúng con chân thành cảm ơn sự động viên và hỗ trợ của gia đình trong suốt thời gian học tập. Đặc biệt, chúng con xin gửi lời cảm ơn trân trọng nhất đến cha mẹ, người đã sinh ra và nuôi dưỡng chúng con nên người. Sự quan tâm, lo lắng và hy sinh lớn lao của cha mẹ luôn là động lực cho chúng con cố gắng phấn đấu trên con đường học tập của mình.

Xin chân thành cảm ơn!

Người thực hiện đề tài

**Lê Thanh Cao**

**Nguyễn Thanh Tuấn**

# Mục lục

LỜI CAM ĐOAN .....	1
LỜI CẢM ƠN .....	2
<b>TÓM TẮT .....</b>	<b>10</b>
Chương 1: TỔNG QUAN .....	1
1.1.    ĐẶT VÂN ĐỀ .....	1
1.2.    MỤC TIÊU .....	2
1.3.    NỘI DUNG NGHIÊN CỨU .....	3
1.4.    GIỚI HẠN .....	3
1.5.    BÓ CỤC .....	4
Chương 2: CƠ SỞ LÝ THUYẾT.....	6
2.1    TỔNG QUAN HỆ THỐNG ĐỊNH VỊ TOÀN CẦU GPS .....	6
2.1.1.    Giới thiệu về GPS .....	6
2.1.2.    Cấu trúc của hệ thống định vị GPS .....	7
2.1.3.    Nguyên lý xác định vị trí bằng GPS .....	10
2.1.4.    Tín hiệu GPS .....	12
2.1.5.    Quá trình giải mã tín hiệu GPS .....	13
2.1.6.    Ứng dụng và những hạn chế của GPS .....	15
2.1.7.    Một số hệ thống định vị toàn cầu khác .....	17
2.2    TỔNG QUAN DỊCH VỤ BẢN ĐỒ TRỰC TUYẾN .....	19
2.2.1.    Tại sao lại có Google Maps .....	20
2.2.2.    Thu thập dữ liệu cho Google Maps .....	21
2.2.3.    Tương lai của ngành dịch vụ bản đồ số- bản đồ trực tuyến .....	24
2.3    CÁC CHUẨN TRUYỀN DỮ LIỆU ĐƯỢC SỬ DỤNG TRONG ĐỀ TÀI .....	25
2.3.1    Giao tiếp Serial – UART .....	25
2.3.2    Giao tiếp I2C .....	26
Chương 3: THIẾT KẾ .....	29
3.1    GIỚI THIỆU: .....	29
3.2    THIẾT KẾ SƠ ĐỒ KHỐI: .....	29
3.3    THIẾT KẾ CÁC KHỐI .....	30
3.3.1.    Khối xử lý trung tâm .....	30
3.3.2.    Khối động cơ: .....	34

3.3.3.	<b>Khối điều khiển động cơ</b>	35
3.3.4.	<b>Khối nhận tín hiệu GPS</b>	37
3.3.4.	<b>Khối cảm biến la bàn</b>	40
3.3.5.	<b>Khối cảm biến khoảng cách</b>	42
3.3.6.	<b>Khối định thời gian</b>	45
3.3.7.	<b>Khối nguồn</b>	48
3.3.8.	<b>Tính toán theo lý thuyết</b>	49
<b>3.4</b>	<b>SƠ ĐỒ NGUYÊN LÝ TOÀN MẠCH</b>	50
Chương 4: THI CÔNG		53
<b>4.1 GIỚI THIỆU</b>		53
<b>4.2 THI CÔNG MÔ HÌNH XE RÁC</b>		53
<b>4.3 LẬP TRÌNH CHO THÙNG RÁC</b>		56
4.3.1.	<b>Giới thiệu phần mềm lập trình Arduino IDE:</b>	56
4.3.2.	<b>Lưu đồ giải thuật</b>	59
Chương 5: KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ		65
<b>5.1 GIỚI THIỆU</b>		65
<b>5.2 KẾT QUẢ ĐẠT ĐƯỢC</b>		65
5.2.1 Biết cách sử dụng các cảm biến		65
5.2.2 Biết cách sử dụng Arduino Mega 2560		66
5.2.3 Biết cách sử dụng module GPS các loại		66
<b>5.3 KẾT QUẢ THỰC NGHIỆM</b>		67
5.3.1. <b>Định vị tọa độ trên Google Maps</b>		67
5.3.2. <b>Kết quả di chuyển thực tế</b>		71
<b>5.4 NHẬN XÉT – ĐÁNH GIÁ:</b>		72
5.4.1 Nhận xét		72
5.4.2 Đánh giá		73
Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		75
<b>6.1. KẾT LUẬN:</b>		75
<b>6.2. HƯỚNG PHÁT TRIỂN:</b>		76
6.2.1 Sử dụng nguồn điện bằng pin năng lượng mặt trời		76
6.2.2 Ứng dụng xử lý ảnh để nhận biết vật thể		77
6.2.3 Ứng dụng Máy học (Machine learning) để cho thùng rác học		77
6.2.4 Kết hợp thêm những cảm biến khác		78

<b>6.2.5</b>	<b>Ứng dụng IOT- Thùng rác thông minh-thành phố thông minh .....</b>	<b>79</b>
<b>TÀI LIỆU THAM KHẢO.....</b>		<b>81</b>
1.	Sách tham khảo .....	81
2.	Trang web tham khảo .....	81
<b>PHỤ LỤC: CODE ARDUINO.....</b>		<b>83</b>

## **LIỆT KÊ HÌNH VẼ**

<b>Hình</b>	<b>Trang</b>
Hình 2.1: Cấu trúc hệ thống định vị toàn cầu GPS .....	7
Hình 2.2: Các vệ tinh GPS và các quỹ đạo bay của nó .....	8
Hình 2.3: 5 Trạm kiểm soát trên thế giới .....	9
Hình 2.4: Biểu diễn của trilateration trên hình tròn.....	10
Hình 2.5: Vị trí máy thu GPS (màu xanh) là giao điểm của ba hình cầu (màu đỏ). Vệ tinh thứ tư (được hiển thị bằng vạch màu vàng) cho bạn thời gian. ....	11
Hình 2.6: Khoảng cách từ GPS tới máy thu (smartphone).....	12
Hình 2.7 Mỗi bộ thu GPS ghi lại các tín hiệu đến từ nhiều vệ tinh. ....	14
Hình 2.8: Bộ thu tín hiệu GPS so sánh các tín hiệu đến với các mã cho tất cả các vệ tinh .....	15
Hình 2.9: Uber, Grap là hai ứng dụng dịch vụ tốt nhất của GPS .....	16
Hình 2.10: Google Maps- bản đồ trực tuyến tốt nhất hiện nay .....	20
Hình 2.11: Google hiển thị những hình ảnh Street View của họ lên trên bản đồ cơ bản. ....	22
Hình 2.12: Google map được ứng dụng trên smartphone kết hợp GPS để chỉ đường. 23	
Hình 2.13: Người dùng Google Maps có thể chỉnh sửa, đóng góp của mình .....	23
Hình 2.14: Bing Map của Microsoft và Yandex của Nga là hai dịch vụ cạnh tranh mạnh mẽ với Google Maps .....	25
Hình 2.15: Truyền dữ liệu qua lại giữa 2 vi điều khiển và giữa vi điều khiển với PC. 26	
Hình 2.16: Giao thức truyền nhận dữ liệu của I2C .....	27
Hình 3.1: Sơ đồ khối của thùng rác thông minh.....	29
Hình 3.2: Vi điều khiển PIC 16F887 và AT89C52 .....	31
Hình 3.3: Bo mạch Raspberry Pi .....	31

Hình 3.4: Bo mạch Arduino Mega 2560 .....	32
Hình 3.5: Sơ đồ linh kiện trong Arduino Mega 2560.....	33
Hình 3.6: Một số hình ảnh về động cơ DC.....	34
Hình 3.7: Hình ảnh và sơ đồ chân module L298.....	35
Hình 3.8: Kết nối module L298 với Arduino và động cơ .....	36
Hình 3.9: Sơ đồ nguyên lý kết nối khối cảm biến với Arduino .....	36
Hình 3.10: Sơ đồ chân của module GPS Neo 6.....	38
Hình 3.11: Kết nối module GPS với Arduino .....	39
Hình 3.12: Sơ đồ nguyên lý kết nối khối nhận tín hiệu GPS với Arduino.....	39
Hình 3.13: Cảm biến la bàn số HMC5883L.....	40
Hình 3.14: Sơ đồ chân cảm biến la bàn số HMC5883L.....	40
Hình 3.15: Kết nối module la bàn số HMC5883L với Arduino.....	41
Hình 3.16: Sơ đồ nguyên lý kết nối khối cảm biến ba lèn .....	41
Hình 3.17: Cảm biến siêu âm SRF-04.....	42
Hình 3.18: Các chân của Cảm biến siêu âm SRF-04. ....	43
Hình 3.19: Kết nối module cảm biến khoảng cách HC-SR 04 với Arduino .....	44
Hình 3.20: Sơ đồ nguyên lý kết nối khối cảm cảm biến khoảng cách .....	45
Hình 3.21: Hình ảnh thực tế moduel RTC DS1307 .....	46
Hình 3.22: Kết nối module RTC DS1307 với Arduino.....	47
Hình 3.23: Sơ đồ nguyên lý kết nối khối định thời gian .....	47
Hình 3.24: Hình ảnh thực tế bộ Ac quy sử dụng trong đè tài.....	48
Hình 3.25: Sơ đồ nguyên lý toàn mạch .....	51
Hình 4.1: Hình ảnh thực tế thiết kế mặt đế thùng rác.....	53
Hình 4.2: Hình ảnh thực tế thiết kế khung bánh xe rác .....	54
Hình 4.3: Hình ảnh thực tế thiết kế thùng rác thông minh từ các hướng .....	55
Hình 4.4: Giao diện phần mềm Arduino IDE.....	57

Hình 4.5: Chức năng các nút lệnh trong Arduino IDE .....	58
Hình 4.6: Chọn Port kết nối.....	58
Hình 4.7: Lưu đồ giải thuật chương trình cho Arduino.....	60
Hình 4.8: Lưu đồ chương trình con Xoay .....	62
Hình 4.9: Lưu đồ chương trình con Chuyển hướng .....	63
Hình 5.1: Lấy tọa độ điểm xuất phát trên Google map .....	67
Hình 5.2: Lấy tọa độ điểm thứ nhất trên Google map.....	68
Hình 5.3: Lấy tọa độ điểm thứ hai trên Google map.....	68
Hình 5.4: Lấy tọa độ điểm thứ ba trên Google map.....	69
Hình 5.5: Lấy tọa độ điểm thứ tư trên Google map.....	69
Hình 5.6: Lấy tọa độ điểm thứ năm trên Google map.....	70
Hình 5.7: Lấy tọa độ điểm thứ sáu trên Google map .....	70
Hình 5.8: Hình ảnh thực tế thùng rác thông minh chạy ngoài trời.....	72

## **LIỆT KÊ BẢNG**

Bảng	Trang
Bảng 2.1: So sánh một số thông số kỹ thuật của ba hệ thống vệ tinh dẫn đường toàn cầu.	18
Hình 3.1: Thông số kỹ thuật Board Arduino Mega 2560.....	33
Hình 5.1: Dung lượng điện năng tiêu thụ thực nghiệm.....	74

## TÓM TẮT

Hiện nay, khi xã hội ngày càng phát triển, dân cư ngày càng đông đúc đặc biệt là ở các thành phố lớn, các khu công nghiệp từ đó dẫn đến lượng rác thải con người xả ra ngày càng tăng. Do đó, quản lý chất thải là một trong những vấn đề nóng hổi mà thế giới phải đổi mới không phân biệt đó là nước phát triển hay đang phát triển. Vấn đề chính là trong việc quản lý chất thải, thùng rác ở nơi công cộng luôn luôn trong tình trạng đầy và quá tải trước khi được xe rác tới lấy và đưa lượng rác đó ra một vị trí tập trung lượng rác lớn khác để xử lý. Việc thùng rác nơi công cộng luôn đầy và tràn ra lần lượt dẫn đến nhiều mối nguy hiểm khác nhau như bốc mùi, ô nhiễm môi trường cảnh quan và đó cũng là gốc rễ, nguyên nhân chính gây ra những căn bệnh lây nhiễm. Xuất phát từ nguyên nhân đó, “Thùng rác thông minh” được ra đời.

Với mục đích giúp cho môi trường xanh-sạch-đẹp và tự động theo xu hướng trường học thông minh trong môi trường trường Đại học đồng thời chúng em muốn tìm hiểu sâu hơn về mảng lập trình ứng dụng để tạo ra các sản phẩm có thể kết hợp với sản phẩm công nghệ như robot, mô hình xe, tính năng định vị toàn cầu GPS,... Dựa vào kiến thức đã học của môn lập trình vi xử lý và chuyên ngành Điện tử công nghiệp, chúng em cũng tạo ra được mô hình thùng rác thông minh có thể thông báo rác đầy ,tự di chuyển tới vị trí mà ta định vị sẵn trên GPS và chạy về vị trí cũ thông qua một nút nhấn trên thùng rác.Thùng rác cũng được trang bị realtime để thực hiện một số tính năng định thời gian.

Trong quá trình thực hiện đề tài “**Thùng rác thông minh**”, mặc dù nhóm thực hiện đề tài đã cố gắng hoàn thành nhiệm vụ đặt ra và đúng thời hạn nhưng do còn hạn chế về kiến thức và thời gian thực hiện nên chắc chắn sẽ không tránh khỏi những thiếu sót, hạn chế, mong quý Thầy/Cô và các bạn sinh viên thông cảm. Nhóm rất biết ơn và mong nhận được những ý kiến đóng góp của quý Thầy/Cô và các bạn sinh viên về đề tài này.

## **Nội dung chính của đề tài**

- Sử dụng Board Arduino Mega 2560 để làm khôi điều khiển trung tâm.
- Điều khiển được thùng rác chạy theo yêu cầu định vị trên GPS khi rác đầy
- Điều khiển được thùng rác chạy về vị trí ban đầu thông qua nút nhấn
- Ứng dụng đồng hồ thời gian thực để thùng rác luôn di chuyển về vị trí định sẵn mặc dù thùng rác chưa đầy vào mỗi tối.

# Chương 1: TỔNG QUAN

## 1.1. ĐẶT VĂN ĐỀ

Quản lý chất thải là một trong những vấn đề nóng hổi mà thế giới phải đối mặt không phân biệt đó là nước phát triển hay đang phát triển. Vấn đề chính là trong việc quản lý chất thải, thùng rác ở nơi công cộng được luôn luôn trong tình trạng đầy và quá tải trước khi được xe rác tới lấy và đưa lượng rác đó ra một vị trí tập trung lượng rác lớn khác để xử lý. Việc thùng rác nơi công cộng luôn đầy và tràn ra lần lượt dẫn đến nhiều mối nguy hiểm khác nhau như bốc mùi, ô nhiễm môi trường cảnh quan và đó cũng là gốc rễ, nguyên nhân chính gây ra những căn bệnh lây nhiễm. Điều này cũng đang dần xuất hiện trong môi trường học đường khi xuất phát từ chính trong ý thức của mỗi sinh viên chúng ta- thùng rác đã đầy nhưng chúng ta vẫn bỏ rác vào gây ra tình trạng rác văng ra ngoài. Để tránh tất cả nguy hiểm và duy trì sự sạch sẽ công cộng và sức khỏe cộng đồng trong trường học cũng như là ngoài xã hội, chúng ta cần phải thiết kế ra được một thùng rác mà nó có thể kiểm soát được lượng rác và đồng thời đưa ra những xử lý cần thiết nhằm giảm thiểu hiện trạng trên.

Nhìn chung, thùng rác thông minh không còn xa lạ gì với tất cả chúng ta. Đã có rất nhiều công ty start-up và những công ty thiết bị gia đình đã và đang đầu tư vào lĩnh vực thùng rác thông minh này. Điều đó là hoàn toàn hợp lý và có cơ sở, bởi lẽ thùng rác là một sản phẩm không thể thiếu được trong gia đình. Không chỉ làm nhiệm vụ đựng rác, nó còn thể hiện sự văn minh của gia đình hiện đại. Chính vì vậy, sự ra đời của chiếc thùng rác thông minh sẽ là sự lựa chọn hàng đầu cho căn bếp của chị em phụ nữ. Có thể kể đến một số hãng của thùng rác thông minh như: thùng rác thông minh Sensible Eco Living, Hòa Phát, Handy, Homematic,... và rất nhiều hãng khác. Tuy nhiên, đặc điểm chung của những hãng thùng rác thông minh nói trên đều được sử dụng trong hộ gia đình và có khả năng cảm ứng. Có nghĩa là nó được trang bị công nghệ cảm ứng tia hồng ngoại, tự động đóng mở nắp thùng, vì vậy bạn không cần phải dùng tay hay chân để đóng, mở nắp thùng rác – điều này đem

## **CHƯƠNG 1. TỔNG QUAN**

---

lại cảm giác vệ sinh trong gia đình bạn. Đó thực sự là một sự thiết kế tuyệt vời cho những ngôi nhà hiện đại.

Tuy nhiên, việc thùng rác thông minh sử dụng trong nhà lại là một vấn đề hoàn toàn khác so với những thùng rác mà chúng ta sử dụng ở những nơi công cộng. Đó chính là sự hạn chế về khả năng tự động và khả năng di động hơn. Những thùng rác được sử dụng ngoài trời sẽ có nhiều không gian và thời gian hơn để xử lý những tác vụ tự động, chẳng hạn như: khi rác đầy, có thể tự động di chuyển tới vị trí nơi chứa rác tổng qua việc định vị sẵn GPS hay hiện đại hơn là sự liên kết nhiều thùng rác lại với nhau để người lao công dễ dàng quản lý... là những tính năng mà thùng rác trong gia đình không thể làm được. Do đó, nhận thấy những tính năng và công dụng tuyệt vời của những “Thùng rác thông minh” trong nhà như vậy, kết hợp với những vấn đề những nhối trong việc quản lý rác thải ở những nơi công cộng. Do đó, việc thiết kế ra “Thùng rác thông minh” ứng dụng ở ngoài trời và được áp dụng công nghệ GPS để định vị là một nhu cầu hết sức cần thiết và đây chính là lý do mà nhóm nghiên cứu quyết định chọn đề tài này. Đề tài này không những là một thực tại khách quan mà nó còn đóng vai trò quan trọng thực sự trong tương lai sau này, đặc biệt là có thể ứng dụng rất tốt trong môi trường trường học và những nơi công cộng.

### **1.2. MỤC TIÊU**

- Sử dụng tính năng hệ thống định vị toàn cầu GPS để định hướng di chuyển và điểm đến cho thùng rác.
- Thiết kế được một chiếc thùng rác có khả năng nhận biết được rác đầy trong thùng.
- Khi phát hiện rác trong thùng đầy, thùng rác có thể tự di chuyển đến vị trí thùng rác tổng và có thể di chuyển về vị trí cũ điều khiển bằng một nút nhấn.
- Khi thùng rác đầy và trong lúc di chuyển ra thùng rác tổng, thùng rác sẽ phát tiếng kêu để báo động để mọi người né tránh.
- Trên đường đi, thùng rác có thể né được vật cản nhỏ

## **CHƯƠNG 1. TỔNG QUAN**

---

- Đến tối hoặc theo thời gian đã định sẵn, nếu rác chưa đầy thì thùng rác vẫn tự di chuyển tới vị trí thùng rác tổng và đứng sẵn ở đó tập trung cho đến khi có người bấm nút để quay lại vị trí cũ.

### **1.3. NỘI DUNG NGHIÊN CỨU**

- Nghiên cứu về hệ thống định vị toàn cầu GPS.
- Nghiên cứu về các chuẩn truyền thông I2C, UART.
- Nghiên cứu cách thức hoạt module GPS.
- Nghiên cứu cách thức hoạt động module cảm biến la bàn số.
- Nghiên cứu cách thức hoạt động module cảm biến siêu âm.
- Nghiên cứu cách thức hoạt động module thời gian thực.
- Viết chương trình cho Arduino Mega 2560.
- Thiết kế, thi công và lập trình khôi nhận tín hiệu.
- Thiết kế, thi công và lập trình khôi cảm biến la bàn.
- Thiết kế, thi công và lập trình khôi cảm biến khoảng cách.
- Thiết kế, thi công và lập trình khôi định thời gian.
- Thiết kế, thi công và lập trình khôi điều khiển động cơ.
- Thiết kế, thi công mô hình thùng rác với 4 bánh xe.
- Lắp ráp các khôi điều khiển vào mô hình.
- Chỉnh sửa các lỗi điều khiển, lỗi lập trình và lỗi của các thiết bị.
- Chạy thử nghiệm hệ thống bên ngoài trời.
- Cân chỉnh lại hệ thống.
- Viết sách luận văn.
- Báo cáo đề tài tốt nghiệp.

### **1.4. GIỚI HẠN**

- Kích thước thùng rác: 34.5 x 34 x 44 cm
- Khối lượng thùng rác khi không tải 4-5kg
- Khối lượng thùng rác khi có tải 9-10kg

## CHƯƠNG 1. TỔNG QUAN

---

- Hoạt động trong điều kiện thời tiết ổn định, ngoài trời, thoáng đãng, không mưa bão
- Vị trí đặt và đường đi của thùng rác phải thông thoáng, ít bị che khuất
- Sử dụng board Arduino Mega 2560 làm khôi điều khiển trung tâm
- Số lượng cảm biến: 5
  - + Cảm biến la bàn số HMC5883L: áp hoạt động 3.3V, dòng hoạt động: 0.1mA
  - + 4 Cảm biến siêu âm HC-SR04: áp hoạt động 5V, dòng hoạt động: 15mA
- Sử dụng module GPS NEO 6 để định vị vị trí: áp hoạt động 3.3V, dòng hoạt động: 50mA
- Sử dụng module RTC DS1307 để định thời gian thực realtime
- Có sử dụng buzzer để báo hiệu.
- Dùng 4 động cơ DC giảm tốc 250 vòng/ phút khi không tải
- Đề tài xây dựng dựa trên thùng rác có sẵn và để thùng rác làm bằng mica.

### 1.5. BỐ CỤC

Với đề tài “Thùng rác thông minh” thì bố cục của đồ án như sau:

- **Chương 1: Tổng quan.**

Chương này trình bày đặt vấn đề dẫn nhập lý do chọn đề tài, mục tiêu của đề tài, nội dung nghiên cứu, các giới hạn thông số và bố cục đề tài.

- **Chương 2: Cơ sở lý thuyết.**

Chương này trình bày giới thiệu chi tiết về hệ thống định vị toàn cầu GPS cũng như cách thức hoạt động của nó, những dịch vụ bản đồ trực tuyến và các chuẩn giao tiếp trong quá trình truyền nhận dữ liệu từ các khôi.

- **Chương 3: Thiết Kế**

Chương này trình bày về cách lựa chọn các linh kiện phù hợp với đề tài, vẽ sơ đồ khôi, chức năng từng khôi, sơ đồ nguyên lý của các board mạch của hệ thống và cách kết nối giữa các linh kiện: Mạch điều khiển trung tâm với mạch nhận tín hiệu GPS, cảm biến la bàn số, cảm biến siêu âm, RTC DS1307, cũng như kết nối với khôi

## **CHƯƠNG 1. TỔNG QUAN**

---

nguồn. Bên cạnh đó là sơ đồ nguyên lý toàn mạch để mọi người có cái nhìn tổng quan hơn về hệ thống.

- **Chương 4:** Thi công hệ thống.

Chương này trình bày về cách lắp ráp hệ thống, cách định vị các điểm trên dịch vụ bản đồ google map, cách lập trình, cách kiểm tra các khôi của hệ thống. Bên cạnh đó là hình ảnh thực tế, cũng như là kết quả hiện tại mà thùng rác đạt được.

- **Chương 5:** Kết quả, nhận xét và đánh giá.

Chương này trình bày các kiến thức, cũng như các kỹ năng mà nhóm đạt được sau khi thực hiện đề tài như: sử dụng các cảm biến, nhận thông tin qua GPS, cách lập trình cho arduino để thùng rác thực hiện một số tác vụ theo yêu cầu. Đồng thời là những hình ảnh thực tế của thùng rác chạy ngoài trời cũng như so sánh, đánh giá những kết quả thực tế đó so với những tính toán lý thuyết ban đầu.

- **Chương 6:** Kết luận và hướng phát triển.

Chương này trình bày những kết quả mà đề tài đạt được và những phần chưa đạt được. Đồng thời, đưa ra những giải pháp khắc phục và những hướng phát triển mới cho đề tài để có thể ứng dụng vào thực tiễn, đời sống.

# Chương 2: CƠ SỞ LÝ THUYẾT

## 2.1 TỔNG QUAN HỆ THỐNG ĐỊNH VỊ TOÀN CẦU GPS

Ngày nay khó có thể hình dung bất cứ một chiếc máy bay, tàu thủy, hay phương tiện thám hiểm nào mà không lắp đặt bộ thu tín hiệu GPS để thu những tín hiệu từ vệ tinh bay trên quỹ đạo quanh trái đất. Dù là một công nghệ đã phát triển từ rất lâu nhưng không vì thế GPS lại lỗi thời, ngược lại nó thực sự rất quan trọng trong mọi lĩnh vực của cuộc sống, chẳng hạn như trong quân sự, hàng không, tàu vũ trụ đến địa chất, trắc địa, bản đồ... hay gần đây những ứng dụng của nó đã được tích hợp trong cả Smartphone giúp theo dõi hoạt động người dùng, thú cưng hay tìm đường, định vị vị trí cho xe ô tô và những dịch vụ rất nổi tiếng như Uber, Grab. Chính vì thế, có thể nói GPS là một trong những đột phá công nghệ hữu ích nhất trong những năm gần đây: thay vì phải đọc những tấm bản đồ tồn thời gian, người tiêu dùng giờ đây đã có thể tìm được đường đi chỉ trong vòng một phút với hướng dẫn chi tiết cùng với số lượng địa điểm đồ sộ tăng dần theo thời gian.

### 2.1.1. Giới thiệu về GPS

**Hệ thống Định vị Toàn cầu** (tiếng Anh: *Global Positioning System - GPS*) hay còn gọi là NAVSTAR (NAVigation Satelllite Timing and Ranging) là hệ thống dẫn đường vệ tinh dùng để cung cấp thông tin về vị trí, tốc độ và thời gian cho các máy thu GPS ở khắp mọi nơi trên trái đất, trong mọi thời điểm và mọi điều kiện thời tiết. Hệ thống GPS có thể xác định vị trí sai số từ vài trăm mét tới vài centimet. Tất nhiên với độ chính xác càng cao thì cấu tạo máy thu tín hiệu GPS càng phức tạp và giá thành càng cao.

Hệ thống được phát triển bởi chính phủ Mỹ, quản lý bởi Không Lực Mỹ (U.S Air Force) và giám sát bởi ủy ban Định vị-Dẫn đường Bộ Quốc phòng Mỹ

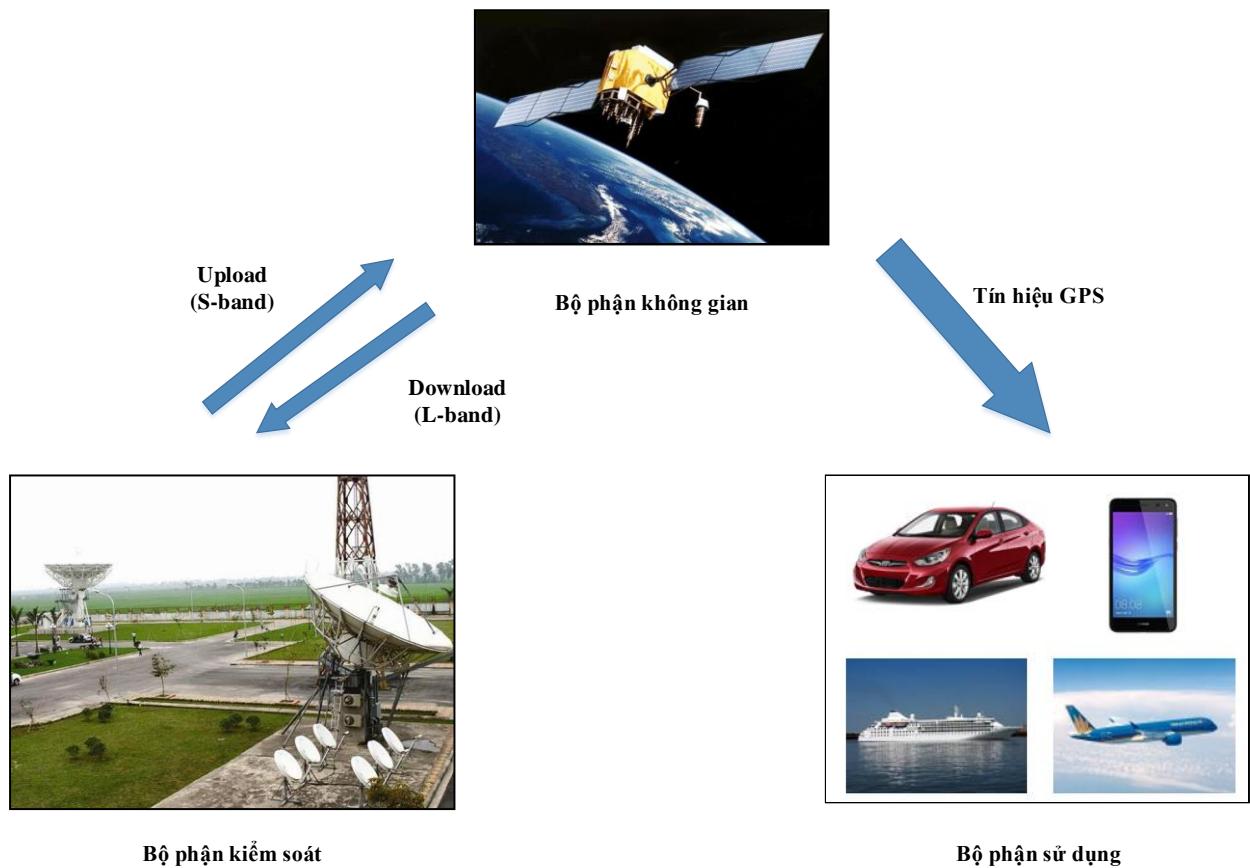
Hệ thống định vị toàn cầu của Mỹ là hệ dẫn đường dựa trên một mạng lưới 24 quả vệ tinh được Bộ Quốc phòng Hoa Kỳ đặt trên quỹ đạo không gian. Vệ tinh

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

đầu tiên được đưa vào quỹ đạo trái đất vào năm 1978. Hoàn chỉnh đầy đủ 23 vệ tinh vào năm 1994. Đến năm 2000, hệ thống này đã có 27 vệ tinh.

### 2.1.2. Cấu trúc của hệ thống định vị GPS

Hiện tại, hệ thống GPS được chia làm ba phần riêng biệt: phần không gian, phần kiểm soát và phần sử dụng.



Hình 2.1: Cấu trúc hệ thống định vị toàn cầu GPS

#### ❖ Phản hồi không gian:

Trong **phản không gian**, đó là một mạng lưới bao gồm 27 vệ tinh quay xung quanh trái đất. Trong số 27 vệ tinh này, 24 vệ tinh đang hoạt động, 3 vệ tinh còn lại đóng vai trò dự phòng trong trường hợp 1 trong số 24 vệ tinh chính bị hư hỏng.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Dựa vào cách sắp đặt của các vệ tinh này, khi đứng dưới mặt đất, bạn có thể nhìn được ít nhất là 4 vệ tinh trên bầu trời tại bất kì thời điểm nào. Từ đó, chúng ta có thể xác định được tọa độ của một điểm trên mặt đất bằng cách tính khoảng cách từ điểm đó tới 4 vệ tinh theo công thức:

$$S = v \cdot t \quad (2.1)$$

Trong đó S: là khoảng cách từ vệ tinh tới vị trí máy thu trên trái đất

v: tín hiệu radio di chuyển xấp xỉ với vận tốc ánh sáng 299,3 km/s

t: Thời gian từ lúc vệ tinh phát tín hiệu cho đến lúc nhận được



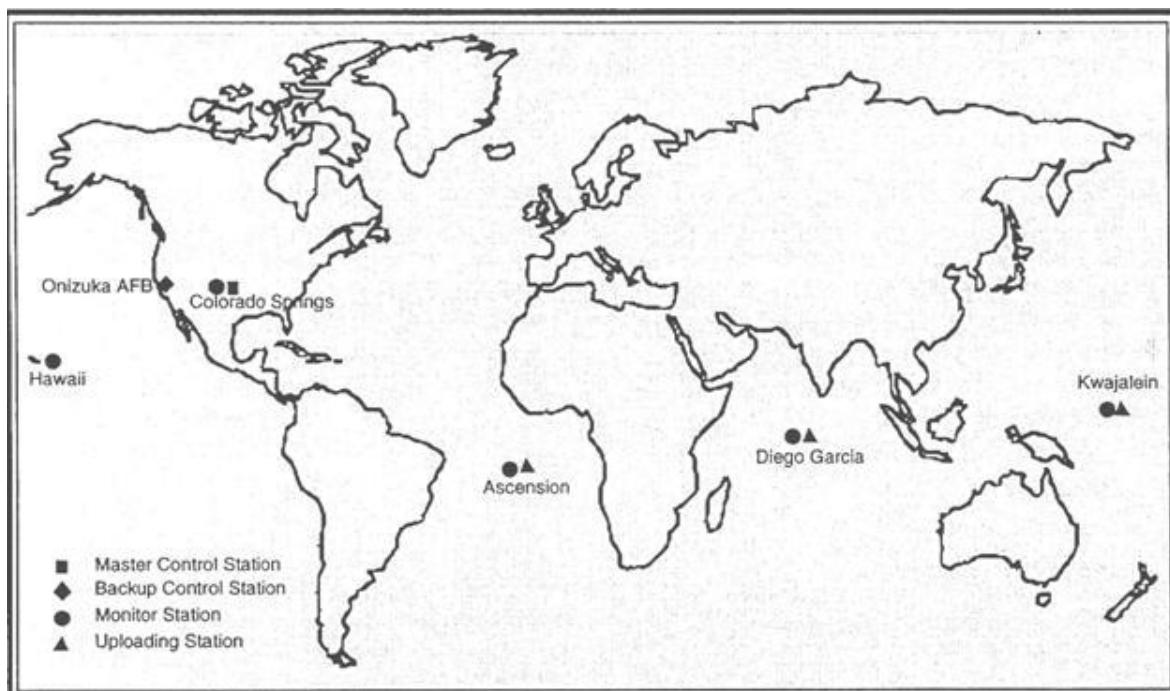
*Hình 2.2: Các vệ tinh GPS và các quỹ đạo bay của nó  
(nguồn từ Google)*

Các vệ tinh được cung cấp năng lượng mặt trời và vùng pin dự phòng để duy trì hoạt động khi chạy khuất vào vùng không có ánh sáng mặt trời, do đó các vệ tinh có thể hoạt động liên tục trong nhiều năm trước khi bị hư và đào thải ra khu vực rác thải trong không gian. Phía dưới mỗi vệ tinh được gắn một tên lửa nhỏ để giúp chúng bay đúng quỹ đạo đã định.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### ❖ *Phản kiểm soát:*

Phối hợp hoạt động với các vệ tinh quay xung quanh trái đất là 5 trạm theo dõi đặt trên mặt đất: trạm chủ được đặt tại Colorado (Mỹ) và 4 trạm khác (không có người điều khiển) được đặt tại các vị trí rất xa lạ, song lại rất gần với đường xích đạo (trong đó có Hawaii cũng ở Mỹ). Các trạm theo dõi này thu thập dữ liệu từ các vệ tinh và truyền dữ liệu về trạm chủ. Trạm chủ sau đó sẽ xử lý dữ liệu và đưa ra các thay đổi cần thiết để chuyển dữ liệu chuẩn về các vệ tinh GPS. Đây còn được gọi là **phản kiểm soát**. Mục đích trong phần này là kiểm soát vệ tinh đi đúng hướng theo quỹ đạo và thông tin thời gian chính xác.



Hình 2.3: 5 Trạm kiểm soát trên thế giới  
(Nguồn từ Wikipedia)

### ❖ *Phản sử dụng:*

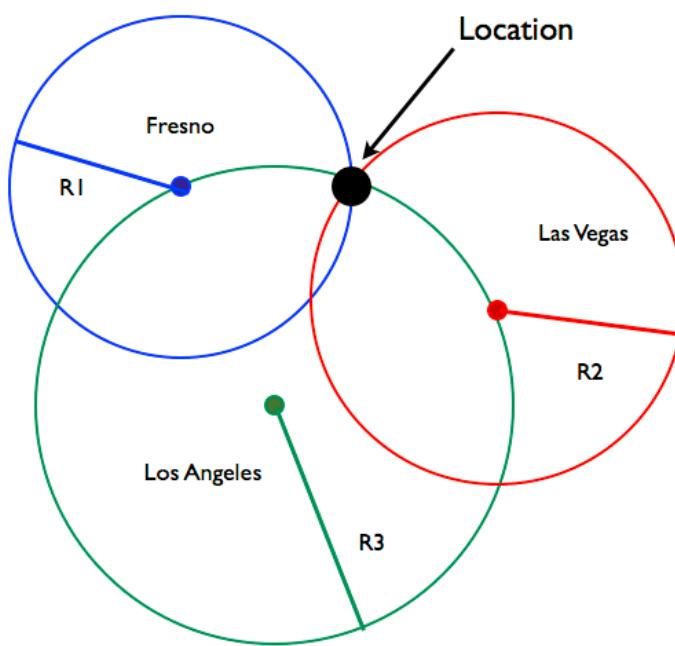
Là thiết bị nhận tín hiệu vệ tinh GPS và người sử dụng thiết bị này.

GPS ban đầu chỉ dành cho các mục đích quân sự, nhưng từ năm 1980 chính phủ Mỹ cho phép sử dụng trong dân sự. GPS hoạt động trong mọi điều kiện thời tiết, mọi nơi trên Trái Đất, 24 giờ một ngày. Không mất phí thuê bao hoặc mất tiền trả

cho việc thiết lập sử dụng GPS nhưng phải tốn tiền không rẻ để mua thiết bị thu tín hiệu và phần mềm nhúng hỗ trợ.

### 2.1.3. Nguyên lý xác định vị trí bằng GPS

Một máy thu GPS xác định vị trí của nó bằng cách sử dụng một quá trình gọi là thuật toán **trilateration** [13] - là quá trình xác định vị trí tuyệt đối hoặc tương đối của các điểm bằng cách đo khoảng cách, sử dụng hình học của hình tròn, hình cầu hoặc hình tam giác. Để dễ hình dung và hiểu một cách đơn giản nhất về nguyên tắc xác định vị trí này, chúng ta nên thông qua một ví dụ như ở *Hình 2.4*

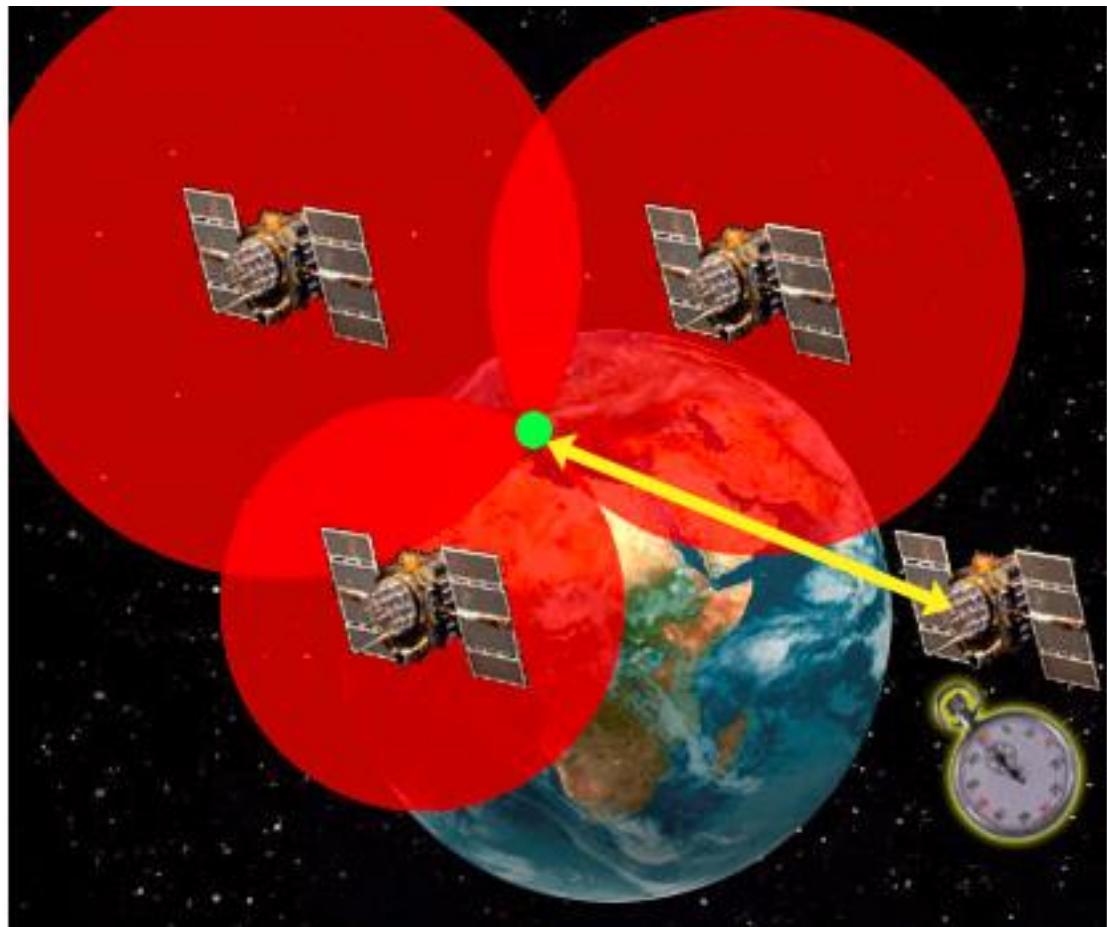


*Hình 2.4: Biểu diễn của trilateration trên hình tròn*

Giả sử, tín hiệu GPS đang được truyền từ các đài phát thanh ở Fresno, Los Angeles và Las Vegas. Giả sử bạn có thể giải mã tín hiệu để biết bạn ở bao xa từ mỗi tháp được truyền đến ( $R_1, R_2, R_3$ ). Sử dụng khoảng cách đã biết làm bán kính để vẽ một vòng tròn xung quanh mỗi tháp. Nếu bạn chỉ có tín hiệu cho Las Vegas và Los Angeles thì bạn có thể ở một trong hai điểm giao nhau giữa hai vòng tròn đó. Những nếu bạn thêm vệ tinh từ Fresno, bạn có thể tìm ra nơi chính xác nhất bởi vì

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

chỉ có duy nhất một điểm giao nhau giữa ba vòng tròn. *Hình 2.5* sẽ cho bạn thấy được phiên bản GPS thực sự của quá trình gọi là **trilateration**. Thay vì là những vòng tròn trong *Hình 2.4*, hãy nghĩ đến mỗi tín hiệu GPS mà vệ tinh phát ra là một dạng hình cầu [14].



*Hình 2.5:* Vị trí máy thu GPS (màu xanh) là giao điểm của ba hình cầu (màu đỏ). Vệ tinh thứ tư (được hiển thị bằng vạch màu vàng) cho bạn thời gian.

### ❖ Công thức Haversine:

Từ kinh độ và vĩ độ giữa điểm thứ nhất và điểm thứ hai, ta tính được khoảng cách giữa hai điểm đó ( $d$ ) và góc từ điểm thứ nhất đến điểm thứ hai ( $\theta$ ) **Haversine**

$$a = \sin^2(\Delta\varphi / 2) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2(\Delta\lambda / 2) \quad (2.2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (2.3)$$

$$d = R \cdot c$$

(2.4)

Trong đó:

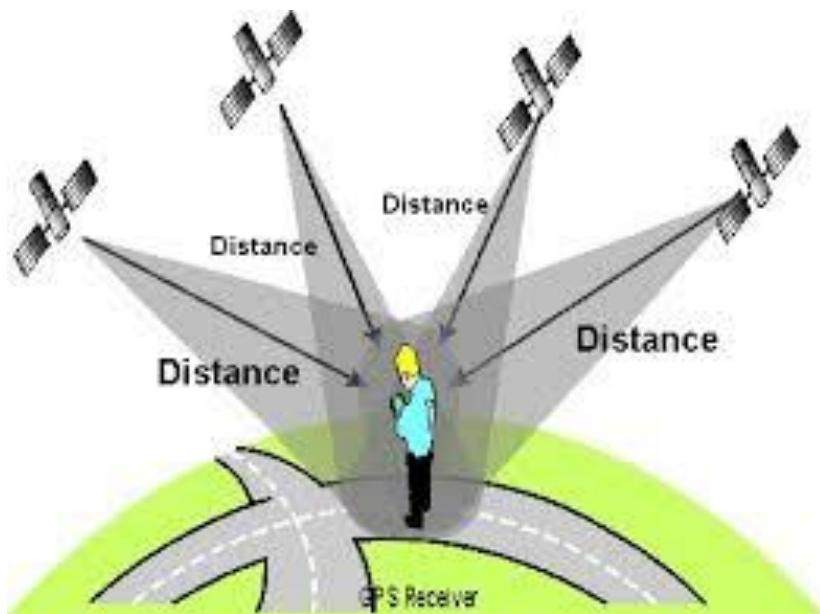
$\phi$  là vĩ độ,  $\lambda$  là kinh độ,  $R$  là bán kính trái đất (bán kính trung bình = 6,371 km);

Lưu ý rằng các góc cần được tính bằng đơn vị radian

Để tính được góc lệch giữa 2 vị trí kinh độ vĩ đó đó, chúng ta sử dụng công thức:

$$\theta = \text{atan2}(\sin \Delta \cdot \cos \varphi_2, \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta\lambda) \quad (2.5)$$

Trong đó  $\varphi_1, \lambda_1$  là điểm bắt đầu,  $\varphi_2, \lambda_2$  là điểm kết thúc ( $\Delta\lambda = \lambda_2 - \lambda_1$  là chênh lệch vĩ kinh độ)



Hình 2.6: Khoảng cách từ GPS tới máy thu (smartphone)

### 2.1.4. Tín hiệu GPS

Các vệ tinh GPS phát hai tín hiệu vô tuyến công suất thấp dải L1 và L2. (dải L là phần sóng cực ngắn của phổ điện tử trải rộng từ 0,39 tới 1,55 GHz). GPS dân sự dùng tần số L1 1575.42 MHz trong dải UHF (ultra high frequency). Tín hiệu truyền trực tiếp, có nghĩa là chúng sẽ xuyên qua mây, thuỷ tinh và nhựa nhưng không qua phần lớn các đối tượng cứng như núi và nhà.

L1 chứa hai mã "giả ngẫu nhiên" (pseudo random), đó là mã Protected (P) và mã Coarse/Acquisition (C/A). Mỗi một vệ tinh có một mã truyền dẫn nhất định, cho

phép máy thu GPS nhận dạng được tín hiệu. Mục đích của các mã tín hiệu này là để tính toán khoảng cách từ vệ tinh đến máy thu GPS.

Tín hiệu GPS chứa ba mẩu thông tin khác nhau – mã giả ngẫu nhiên, dữ liệu thiên văn và dữ liệu lịch. Mã giả ngẫu nhiên đơn giản chỉ là mã định danh để xác định được quả vệ tinh nào là phát thông tin nào. Có thể nhìn số hiệu của các quả vệ tinh trên trang vệ tinh của máy thu Garmin để biết nó nhận được tín hiệu của quả nào.

Dữ liệu thiên văn cho máy thu GPS biết quả vệ tinh ở đâu trên quỹ đạo ở mỗi thời điểm trong ngày. Mỗi quả vệ tinh phát dữ liệu thiên văn chỉ ra thông tin quỹ đạo cho vệ tinh đó và mỗi vệ tinh khác trong hệ thống.

Dữ liệu lịch được phát đều đặn bởi mỗi quả vệ tinh, chứa thông tin quan trọng về trạng thái của vệ tinh (lành mạnh hay không), ngày giờ hiện tại. Phần này của tín hiệu là cốt lõi để phát hiện ra vị trí.

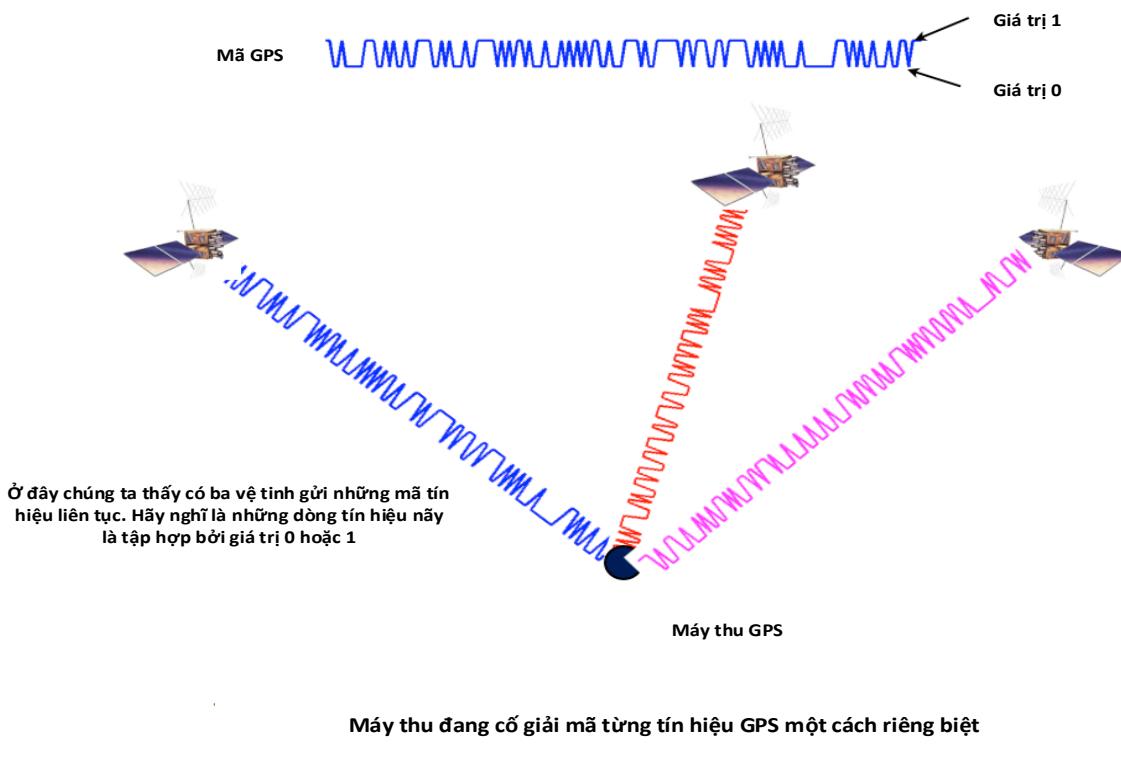
### 2.1.5. Quá trình giải mã tín hiệu GPS

Nếu xem xong những thông tin đã được đề cập phía trên, chắc chắn bạn phải tò mò rằng: Làm thế nào để biết được bán kính của những hình cầu đó như thế nào? Hay làm thế nào để bạn biết được nơi các vệ tinh ở đâu khi nó gửi các tín hiệu? Hay quá trình đọc và giải mã các tín hiệu GPS nó thực hiện như thế nào.

Tín hiệu GPS được thiết kế đặc biệt để trả lời những câu hỏi đó. Mỗi vệ tinh GPS sẽ gửi một mã duy nhất. Mã đó là một dãy số 1 và số 0 (*Hình 2.7*). Bộ phận thu GPS trong ô tô hoặc điện thoại của bạn có bản sao của mỗi mã vệ tinh. Khi tín hiệu GPS đi vào, thiết bị điện tử của máy thu phải tìm ra các vệ tinh nào đã gửi chúng. Nó cẩn thận so sánh các tín hiệu GPS nhận được với các mã cho tất cả các vệ tinh mà nó biết. Khi máy thu đã xác định từng vệ tinh, nó sẽ dịch mã để xếp mã nhận được với mã bên trong bộ nhớ của nó. Sự dịch chuyển đại diện cho lượng thời gian mã vệ tinh đã đạt tới máy thu. Chúng tôi sử dụng thời gian được dịch chuyển này để tính toán khoảng cách giữa vệ tinh khi nó gửi tín hiệu. Ta xem mỗi dấu tích trên mã

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

đại diện cho 1 mili giây. Nếu máy thu tín hiệu GPS phải thay đổi 70 mã vạch, tín hiệu mất 70 mili giây để di chuyển từ vệ tinh đến Trái đất. Để chuyển đổi nó thành khoảng cách, chúng ta nhân với tốc độ ánh sáng ( $\sim 3 \times 10^8$  m/s). Điều đó cho chúng ta biết được là khoảng cách từ vệ tinh tới trái đất là 21.000.000 mét.



Hình 2.7: Mỗi bộ thu GPS ghi lại các tín hiệu đến từ nhiều vệ tinh.

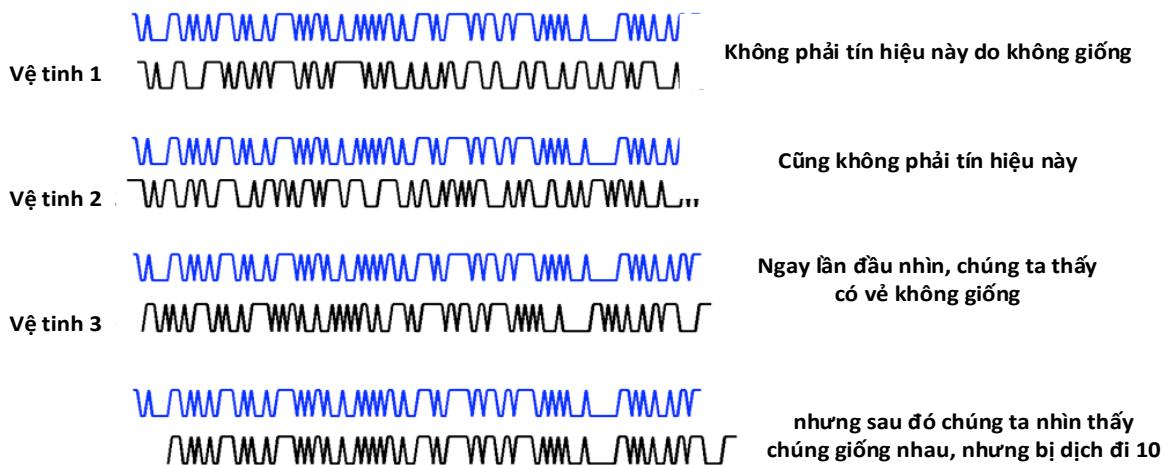
Bây giờ bộ phận thu tín hiệu đã biết khoảng cách của mỗi vệ tinh khi nó gửi tín hiệu về. Không chỉ vậy, hãy nhớ rằng, mọi vệ tinh đều có đồng hồ riêng của mình. Điều đó có nghĩa là bán kính của mỗi quả cầu trong *Hình 2.5* sẽ hơi lệch, tùy thuộc vào việc đồng hồ vệ tinh đó chạy nhanh hay chậm. Đây không phải là một vấn đề tầm thường vì sự thay đổi đồng hồ vệ tinh có thể tạo ra lỗi lớn tới 150 km. Vấn đề này được giải quyết bằng phân đoạn điều khiển mà chúng ta đã nói đến trên trang Satellites, Controllers, Users [14]. Trạm điều khiển chính đồng bộ hóa tất cả các đồng hồ vệ tinh và truyền các chỉnh sửa của chúng cho người dùng.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Cuối cùng, chúng ta cần phải biết vệ tinh đang nằm ở đâu khi nó gửi tín hiệu. Vệ tinh GPS đang di chuyển với tốc độ ~ 4 km / giây, vì vậy điều quan trọng là phải biết vị trí của tọa độ cực kỳ tốt. May mắn thay, trung tâm điều khiển chính có thể cung cấp thông tin đó trong thời gian thực với độ chính xác vài mét. Điều cuối cùng mà máy thu GPS cần làm là đồng bộ hóa đồng hồ của nó với đồng hồ của GPS. Nó sử dụng các phép đo từ vệ tinh thứ tư để làm điều đó.

**Ở đây thiết bị thu GPS sẽ so sánh mã tín hiệu màu xanh với tất cả những mã tín hiệu mà nó biết**



Đó là khoảng thời gian dịch chuyển mà máy thu sử dụng dùng để tìm ra khoảng cách vệ tinh 3 đến vị trí máy thu- và độ lớn bán kính cho hình cầu hình 2.5

Hình 2.8: Bộ thu tín hiệu GPS so sánh các tín hiệu đến với các mã cho tất cả các vệ tinh

### 2.1.6. Ứng dụng và những hạn chế của GPS

#### a) Ứng dụng:

Như đã đề cập ở phần tóm tắt, GPS được coi là một trong những công nghệ đột phá, hữu ích trong cuộc sống con người. Càng ngày, GPS càng thể hiện được vai trò đặc biệt của nó trong hầu hết các lĩnh vực. Ngày nay, các hệ thống giám sát được xây dựng có sẵn nhiều định dạng và kích cỡ với nhiều tính năng, từ thiết bị GPS theo dõi tiêu chuẩn tới các thiết bị tích hợp trong điện thoại di động. Một số ứng dụng điển hình của công nghệ GPS mang lại như:

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- *Trong giáo dục:* Chương trình thiết bị bản đồ & GIS Giáo dục (Mapping & GIS Educator) giới thiệu những giải pháp đơn giản và đa dạng tạo điều kiện thuận lợi nhất cho các tổ chức giáo dục, thực hiện giảng dạy về công nghệ GPS và GIS cho học viên dựa trên những công nghệ mới nhất của Trimble
- *Trong quân sự:* Đây là lĩnh vực đã hình thành nên hệ thống định vị toàn cầu GPS. Vì vậy hệ thống GPS đóng một vai trò đặc biệt quan trọng trong lĩnh lực này như là giám sát an ninh, phục vụ cho quân đội, khảo sát vẽ bản đồ, dẫn đường cho tên lửa hay máy bay chiến đấu...
- *Trong giao thông:* Dùng trong việc dẫn đường các phương tiện giao thông. Ngoài ra còn được sử dụng trong các hệ thống chống trộm nhằm bảo vệ tài sản cho các phương tiện giao thông, giúp các đơn vị vận tải theo dõi và nắm bắt thông tin chính xác về hoạt động của các xe khi hoạt động trên đường như: Giám sát lộ trình đường đi của phương tiện theo thời gian thực vận tốc, hướng di chuyển và trạng thái tắt/mở máy, quá tốc độ của xe...., xác định vị trí xe chính xác ở từng góc đường (vị trí xe được thể hiện nháy nháy trên bản đồ), xác định vận tốc và thời gian xe dừng hay đang chạy, biết được lộ trình hiện tại xe đang đi (real time)



Hình 2.9: Uber, Grab là hai ứng dụng dịch vụ tốt nhất của GPS

### b) Hạn chế

Hoạt động của GPS có thể bị ảnh hưởng bởi các yếu tố sau:

- Khi các vệ tinh ở quá gần nhau, chúng sẽ khiến cho việc xác định một vị trí chính xác trở nên khó khăn hơn.
- Vì tín hiệu radio đi từ vệ tinh xuyên qua tầng điện ly và tầng đối lưu, tốc độ cần thiết để tín hiệu truyền tới thiết bị nhận sẽ bị chậm đi. Hệ thống GPS có dự phòng điều đó bằng cách tính thêm khoảng thời gian chậm trễ trung bình, nhưng cũng không được hoàn toàn chính xác.
- Chướng ngại lớn như các dãy núi hay các tòa nhà cao tầng hoặc là thời tiết xấu cũng làm cho thông tin bị sai lệch.
- Giữa thiết bị nhận (nhất là của người dùng cá nhân) với vệ tinh (có thể không hoàn toàn trùng khớp về mặt thời gian, và các vệ tinh đôi khi chạy lệch khỏi quỹ đạo.

#### 2.1.7. Một số hệ thống định vị toàn cầu khác

Ngoài sự thông dụng của hệ thống định vị toàn cầu GPS ra, ở một nơi trên thế giới họ cũng không muốn lệ thuộc nhiều vào hệ thống định vị này của Mỹ. Chính vì thế, một số quốc gia phát triển khác đã tự tạo và phóng những tên lửa vệ tinh định vị của chính họ để sử dụng như:

- **Hệ Thống GLONASS**

Hệ Thống GLONASS (Global Orbiting Navigation Satellite System) là hệ thống thuộc quyền sở hữu của Nga, gồm có 30 vệ tinh chuyển động trong ba mặt phẳng quỹ đạo xung quanh trái đất

- **Hệ thống Galileo**

Hệ thống định vị Galileo là một hệ thống vệ tinh định vị toàn cầu (GNSS) được xây dựng bởi Liên minh châu Âu. Galileo khác với GPS của Hoa Kỳ và GLONASS của Liên bang Nga ở chỗ nó là một hệ thống định vị được điều hành và quản lý bởi

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

các tổ chức dân dụng, phi quân sự. Galileo theo kế hoạch chính thức hoạt động vào năm 2011-2012, muộn 3-4 năm so với kế hoạch ban đầu.

Hệ thống định vị GPS được đặt theo tên của nhà thiên văn học người Ý (Galilei Galileo) nhằm tưởng nhớ những đóng góp của ông.

**Bảng 2.1** So sánh một số thông số kỹ thuật của ba hệ thống vệ tinh dẫn đường toàn cầu (Yasuda, 2001)

Hạng mục	GPS	GLONASS	GALILEO
Số vệ tinh	28 (tính đến 2000)	30	30
Số mặt phẳng quỹ đạo	6MEO	3MEO	3MEO
Độ nghiêng MPQĐ	55°	64.8°	56°
Bán kính quỹ đạo	26.560 km	25.510 km	29.980 km
Chu kỳ	11 giờ 58 phút 2 giây	11 giờ 15 phút 40 giây	14 giờ 21 phút 36 giây
Tần số sóng mang	L1: 1575.42 MHz L2: 1227.60 MHz L5: 1176.45 MHz	G1: 1602 + Kx0.5625 MHz G2: 1246 + Kx0.5625 MHz K = -7~24 G2 = G1x7/9	E1: 1589.742 MHz E2: 1561.098 MHz E5: 1202.025 MHz E6: 1278.75 MHz C1: 5019.86 MHz
Phương trình	CDMA	FDMA	CDMA
Dạng mã số		Chuỗi M	
Độ dài mã số	1023 bit $2.35 \times 10^{14}$	511 bit 5110000	N/A
Tốc độ mã số (C/A L1, P L1, L2)	1.023 Mcps	0.511 Mcps	E1, E2: 2.046 Mcps

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

	10.23 Mcps	5.11 Mcps	E5: 10.23/1.023 Mcps
Thời gian chuẩn	UTC (USNO)	UTC (Nga)	UTC
Sai số chủ định	SA (đã bỏ 2000)	Không có	Không có

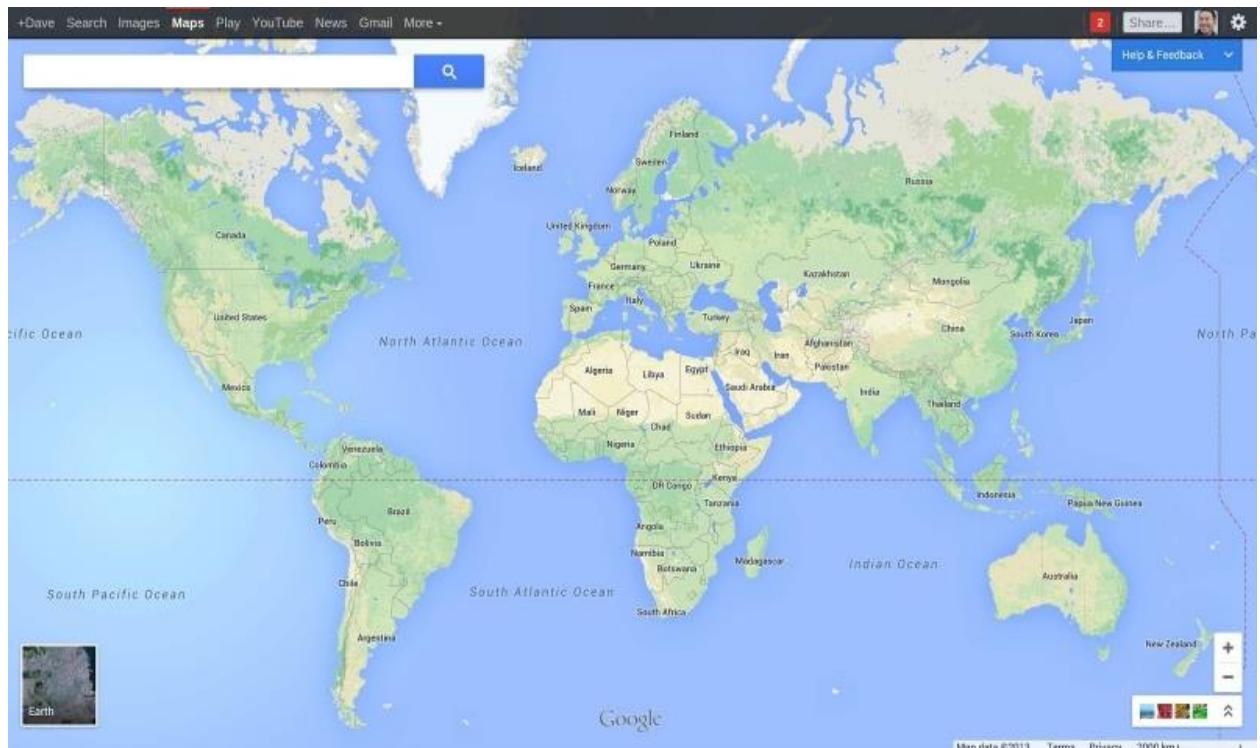
Ngoài ra còn có hệ thống định vị Bắc Đẩu – một hệ thống định vị của Trung Quốc và chỉ phủ sóng một phạm vi ngắn trong Trung Quốc. Theo kế hoạch hệ thống sẽ cung cấp dịch vụ cho khách hàng trong khu vực châu Á-Thái Bình Dương vào năm 2012 và các hệ thống toàn cầu sẽ được hoàn thành vào năm 2020, sau khi sở hữu 35 vệ tinh.

### 2.2 TỔNG QUAN DỊCH VỤ BẢN ĐỒ TRỰC TUYẾN

Internet đã đánh dấu một thời kỳ hưng thịnh của công nghệ dịch vụ trực tuyến khi được ra đời và phổ biến ở Việt Nam vào 20 năm trước. Internet đúng là một bước tiến lớn và đánh dấu sự phát triển và thay đổi loài người mọi mặt trong cuộc sống xã hội. Một trong số đó là sự ra đời của bản đồ trực tuyến- đó chính là cầu nối giữa những tấm bản đồ thô sơ với công nghệ hiện đại. Thay vì phải loay hoay với việc định hướng, tìm đường trên tấm bản đồ khô khan, người dùng có thể thoải mái đến bất kì đâu mà mình muốn chỉ với một vài thao tác đơn giản.

Có thể nói, đây là dịch vụ gần như không thể thiếu trong cuộc sống công nghệ hiện nay, chỉ đứng đầu sau các dịch vụ cung cấp email và Google. Và trên thị trường hiện nay, cũng không thiếu các nhà cung cấp dịch vụ bản đồ trực tuyến, tuy nhiên nổi trội nhất vẫn là sản phẩm từ các nhà cung cấp có tên tuổi: Google Maps, MapQuest và Microsoft Bing Maps. Tuy vậy, ở quốc gia như Việt Nam, Google Maps vẫn là số một và là lựa chọn hàng đầu.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT



Hình 2.10: Google Maps- bản đồ trực tuyến tốt nhất hiện nay

### 2.2.1. Tại sao lại có Googles Maps

Google Maps là một phần của sứ mệnh tổ chức thông tin toàn thế giới để mọi người có thể dễ dàng tiếp cận từ mọi nơi. Google tự nhận thực hiện sứ mệnh đó. Nhưng thông tin Google đang tổ chức không chỉ ở dạng online. Một số lượng lớn người dùng không thường xuyên online, và Google Maps chính là cầu nối giữa cái mà chúng ta nhìn thấy trong đời thực với thế giới trực tuyến.

Có rất nhiều dạng thông tin offline như hệ thống đường cao tốc, biển báo, tên đường, tên công ty... Nhiệm vụ của Google Maps là đưa các thông tin này lên mạng và còn nhiều hơn thế. Chính vì thế Google đang thu thập, tổ chức và biên dịch hàng chục triệu gigabyte dữ liệu cho Google Maps. Ảnh tượng ở chỗ không có dữ liệu nào trong số này cũ tới 3 năm.

Để hỗ trợ quá trình thu thập dữ liệu không lồ trên, Google đã hợp tác với nhiều đối tác khác nhau trong khuôn khổ Chương trình Đối tác Bản đồ Cơ sở (BMPP).

Các dữ liệu do đối tác cung cấp có thể bao gồm những thay đổi về đường biên giới, tuyến đường biển, đường xe đạp và rất nhiều thứ khác.

### 2.2.2. Thu thập dữ liệu cho Google Maps

Khi nói đến việc thu thập dữ liệu để giúp duy trì và cải thiện Google Maps, có vẻ như không bao giờ là đủ- và còn ấn tượng hơn khi không có thông tin nào có tuổi đời quá 3 năm. Đây là một dự án thực sự rất lớn [15].

- **Map Partner**

Để giúp sức cho sự cố gắng này, Google cộng tác với: “những nguồn dữ liệu toàn diện và chính xác nhất” thông qua chương trình **Base Map Partner Program**. Một lượng lớn các cơ quan nộp những dữ liệu vector chi tiết đến Google, một vài cái tên có thể kể đến là: Cục Kiểm lâm Hoa Kỳ, Cục Công viên Quốc gia Hoa Kỳ, Cục Khảo sát Địa chất Hoa Kỳ, bên cạnh đó còn rất nhiều đơn vị khác.

Những dữ liệu này được dùng để xác định những đường ranh giới và sông ngòi bị thay đổi, hiển thị những con đường mới và nhiều thứ khác, và điều này giữ “bản đồ cơ bản” luôn được cập nhật tốt nhất có thể

- **Street View**

**Google Street View** là một chuyến hành trình không bao giờ kết thúc. Với một lượng cực lớn những phương tiện di chuyển trên toàn thế giới, mục tiêu của họ là lặp lại việc di chuyển trên tất cả những con đường mà họ tìm thấy và chụp những bức ảnh 360 độ mọi nơi mà họ đến.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---



Hình 2.11: Google hiển thị những hình ảnh Street View của họ lên trên bản đồ cơ bản.

Dựa trên **công nghệ GPS** được tích hợp trên những phương tiện đó, Google hiển thị những hình ảnh Street View của họ lên trên bản đồ cơ bản.

Street View làm được nhiều điều hơn chỉ là một bức tranh toàn cảnh về những con đường và địa điểm được khâu lại. Sử dụng công nghệ nhận diện ký tự quang học (OCR) luôn được cải thiện, Google có thể “đọc” được những thứ như bảng hiệu đường, biển báo giao thông, và tên doanh nghiệp.

Những thứ OCR đọc được sẽ được xử lý và chuyển thành dữ liệu điều hướng mà Google Maps có thể kết hợp vào cơ sở dữ liệu của nó. Nếu tên của một con đường đã bị thay đổi kể từ lần cuối nó được chụp, một bức ảnh Street View mới hơn sẽ phát hiện ra. Đây cũng (một phần) là cách mà Google xây dựng một cơ sở dữ liệu chi tiết về doanh nghiệp địa phương không lồ.

- **Những vệ tinh**

Một lớp khác của Google Maps và tầm nhìn từ vệ tinh. Đây là một bản phối gần gũi với Google Earth, kết hợp những tấm ảnh chụp từ vệ tinh có độ phân giải cao với nhau.

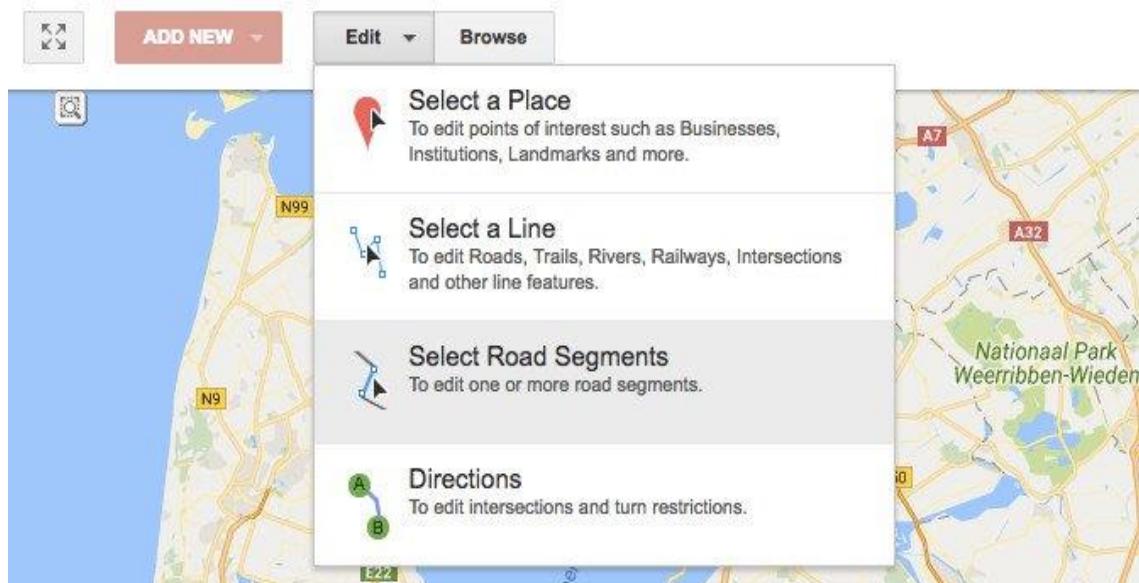
## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Những hình ảnh này được đổi chiều chéo với những lớp khác của dữ liệu, như của Street View và các Map Partners. Điều này giúp Maps thu thập những thay đổi về địa lý, những công trình mới hay được tu sửa, vân vân...



Hình 2.12: Google map được ứng dụng trên smartphone kết hợp GPS để chỉ đường

- **Những người dùng Google Maps**



Hình 2.13: Người dùng Google Maps có thể chỉnh sửa, đóng góp của mình

**Google Map Maker** là một phương tiện khác của Google để thực hiện thu thập dữ liệu đám đông cho Maps, và đây là một chương trình đã có từ 2008.

Hoạt động với nhiều điểm giống với **OpenStreetMap**. Google Map Maker cho phép bất cứ ai cũng có thể đóng góp những hiểu biết địa phương của họ cho Google Maps. Tin tốt là hầu hết chức năng này đã được tích hợp vào Maps, và Maps Maker sẽ được đóng lại vào năm nay khi sự chuyển đổi đã hoàn tất.

Một cách ngắn gọn, người dùng có thể chỉnh sửa bản đồ của Google với sự đóng góp của chính mình. Bạn có thể thêm và chỉnh sửa những địa điểm, những con đường mới và nhiều thứ khác. Và nếu bạn nghĩ mình có thể trốn thoát sau khi cõi tình phá hoại, hãy nghĩ lại: những chỉnh sửa của người dùng có thể được xem lại bởi những người dùng khác.

Điều này có nghĩa là có một lực lượng cực lớn những người biên tập giữ cho Google Maps được cập nhật 24/7. Điều này đặc biệt hữu dụng để đưa những nơi khó đến lên bản đồ và để thu thập những hiểu biết mà mặt khác nằm ngoài tầm với hoặc để ý của Google.

### 2.2.3. Tương lai của ngành dịch vụ bản đồ số- bản đồ trực tuyến

Theo các chuyên gia, trong khoảng 5 năm tới, dịch vụ bản đồ sẽ tích hợp thêm nhiều công nghệ cũng như cảm biến mới nhằm thay thế hệ thống GPS và tín hiệu di động hiện nay.

Ngoài ra, tương lai của bản đồ sẽ không chỉ đơn thuần là dữ liệu điều hướng mà còn là cách mà chúng ta sử dụng các dịch vụ này ra sao.

Đơn cử như Bluetooth Beacon có thể được tích hợp sâu và hệ thống bản đồ giúp người dùng và các cửa hàng có thể dễ dàng đến với nhau, từ đó, thúc đẩy việc kinh doanh, mua sắm.

Hoặc trong tương lai, các hệ thống thực tại ảo có thể tích hợp vào các dịch vụ điều hướng, từ đó, tái tạo, mô phỏng các không gian 3 chiều trong đời thực và tương lai.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Ngoài ra, chúng ta còn thấy sự nổi lên và cạnh tranh thị trường của rất nhiều hãng, trang web bản đồ trực tuyến như Wikimapia.org, Yahoo Map, Bing Map, Yandex Map, OpenStreetMap.org, MapQuest.com... Thê nên, những dịch vụ nào biết cách vận dụng được những công nghệ mới cũng như tạo dựng được thương hiệu sẽ lợi thế trong cuộc cạnh tranh này

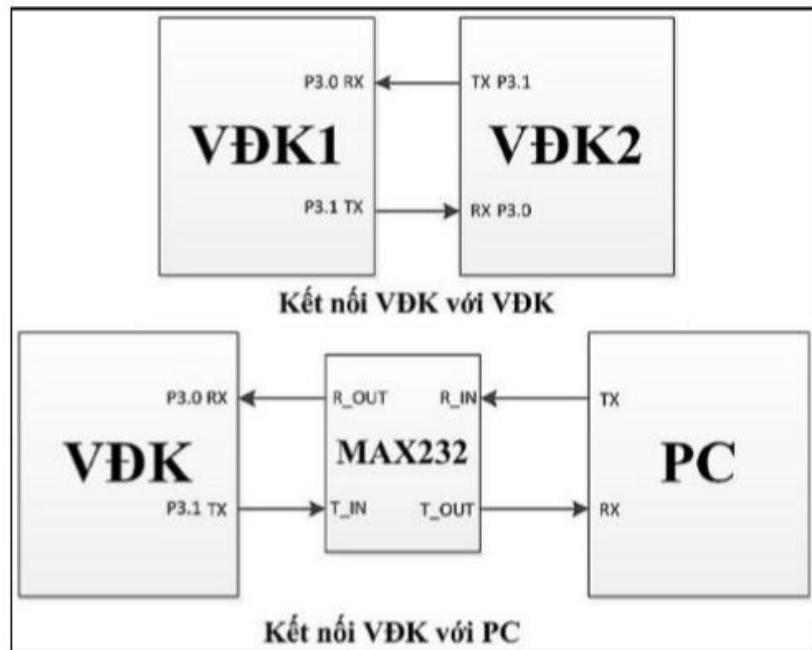


Hình 2.14: Bing Map của Microsoft và Yandex của Nga là hai dịch vụ cạnh tranh mạnh mẽ với Google Maps

### 2.3 CÁC CHUẨN TRUYỀN DỮ LIỆU ĐƯỢC SỬ DỤNG TRONG ĐỀ TÀI

#### 2.3.1 Giao tiếp Serial – UART

UART là viết tắt của Universal Asynchronous Receiver – Transmitter Là kiểu truyền thông tin nối tiếp không đồng bộ thường là một mạch tích hợp. Mục đích của UART là để truyền tín hiệu qua lại lẫn nhau (ví dụ truyền tín hiệu từ Laptop vào Modem hay ngược lại) hay truyền từ vi điều khiển tới vi điều khiển, từ laptop tới vi điều khiển.



Hình 2.15: Truyền dữ liệu qua lại giữa 2 vi điều khiển và giữa vi điều khiển với PC

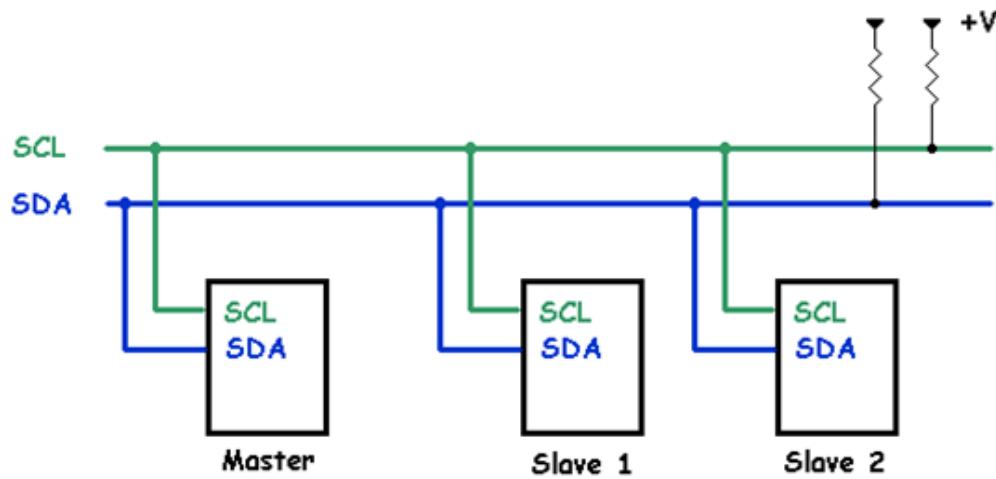
### 2.3.2 Giao tiếp I2C

I2C là tên viết tắt của cụm từ Inter-Integrated Circuit. Đây là đường Bus giao tiếp giữa các IC với nhau. Bus I2C được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, AVR, ARM... chip nhớ như: RAM tĩnh (Static Ram), EEPROM, bộ chuyển đổi tương tự số (ADC), số tương tự (DAC), IC điều khiển LCD, LED...

#### ❖ Nguyên lý hoạt động:

I2C sử dụng hai đường truyền tín hiệu:

- Một đường xung nhịp đồng hồ (SCL) chỉ do Master phát đi (thông thường ở 100kHz và 400kHz. Mức cao nhất là 1Mhz và 3.4MHz).
- Một đường dữ liệu(SDA) theo 2 hướng.



Hình 2.16: Giao thức truyền nhận dữ liệu của I2C

Như hình vẽ trên, khi một thiết bị ngoại vi kết nối vào đường bus I2C thì chân SDA của nó sẽ nối vào dây SDA của bus, chân SCL sẽ nối vào dây SCL.

Có rất nhiều thiết bị có thể cùng được kết nối vào một bus I2C, tuy nhiên sẽ không xảy ra chuyện nhầm lẫn giữa các thiết bị, bởi mỗi thiết bị sẽ được nhận ra bởi một địa chỉ duy nhất với một quan hệ chủ/tớ tồn tại trong suốt thời gian kết nối. Mỗi thiết bị có thể hoạt động như là thiết bị nhận hoặc truyền dữ liệu hay có thể vừa truyền vừa nhận. Hoạt động truyền hay nhận còn tùy thuộc vào việc thiết bị đó là chủ (master) hay tớ (slave).

Một thiết bị hay một IC khi kết nối với bus I2C, ngoài một địa chỉ (duy nhất) để phân biệt, nó còn được cấu hình là thiết bị chủ hay tớ. Tại sao lại có sự phân biệt này? Đó là vì trên một bus I2C thì quyền điều khiển thuộc về thiết bị chủ. Thiết bị chủ nắm vai trò tạo xung đồng hồ cho toàn hệ thống, khi giữa hai thiết bị chủ-tớ giao tiếp thì thiết bị chủ có nhiệm vụ tạo xung đồng hồ và quản lý địa chỉ của thiết bị tớ trong suốt quá trình giao tiếp. Thiết bị chủ giữ vai trò chủ động, còn thiết bị tớ giữ vai trò bị động trong việc giao tiếp.

## **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

---

Điểm mạnh của PC chính là hiệu suất và sự đơn giản của nó: một khối điều khiển trung tâm có thể điều khiển cả một mạng thiết bị mà chỉ cần hai lối ra điều khiển.

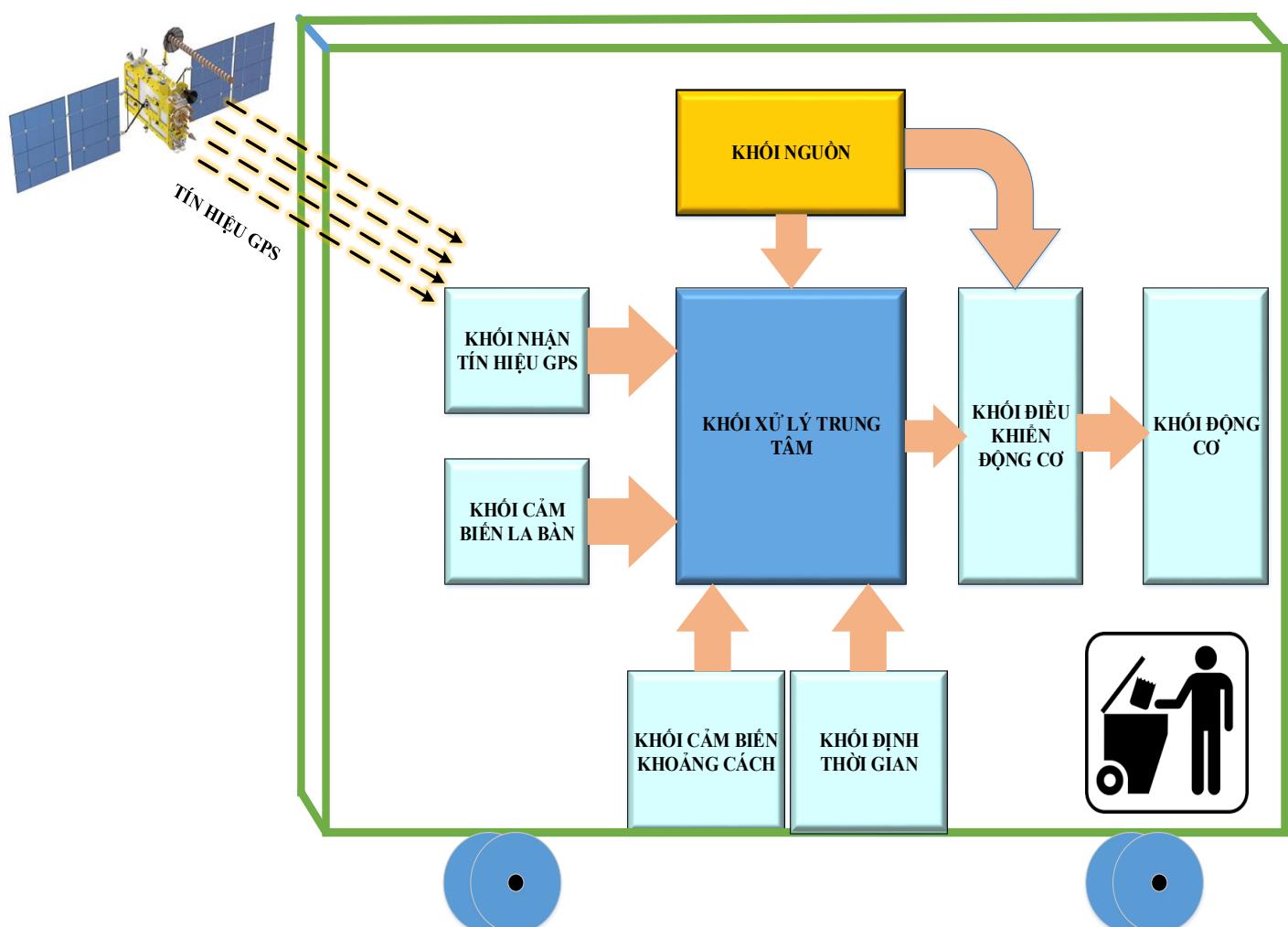
## **Chương 3: THIẾT KẾ**

### **3.1 GIỚI THIỆU:**

Trong chương này, trình bày về sơ đồ khói, lựa chọn linh kiện phù hợp với đề tài, sơ đồ nguyên lý của các board mạch, cách kết nối giữa các module với nhau.

### **3.2 THIẾT KẾ SƠ ĐỒ KHỐI:**

Với các yêu cầu đã đưa ra, nhóm thực hiện đã hình thành sơ đồ khói cho thùng rác như sau:



*Hình 3.1: Sơ đồ khói của thùng rác thông minh.*

### Chức năng từng khối:

- **Khối nguồn:** Có chức năng cấp nguồn cho toàn mạch hoạt động.
- **Khối xử lý trung tâm:** Có chức năng nhận vào và xử lý các tín hiệu từ khối nhận tín hiệu GPS, khối định thời gian, khối cảm biến khoảng cách và khối cảm biến la bàn số. Sau đó truyền các tín hiệu điều khiển ra khối điều khiển động cơ để làm cho động cơ hoạt động.
- **Khối nhận tín hiệu GPS:** Nhận tín hiệu từ vệ tinh GPS để từ đó xác định được đúng kinh độ, vĩ độ mà ta đã định sẵn từ đó giúp cho bộ xử lý trung tâm điều khiển khói động cơ hoạt động đúng theo yêu cầu định vị vị trí cho sẵn.
- **Khối cảm biến la bàn:** Việc sử dụng cảm biến la bàn kết hợp với vị trí định vị từ GPS để việc xác định đúng phương hướng và góc rẽ tới các tọa độ đã biết một cách chính xác.
- **Khối Cảm Biến khoảng cách:** Có chức năng cung cấp cho hệ thống khả năng đọc khoảng cách tới vật cản.
- **Khối điều khiển động cơ:** Giúp cho dễ dàng trong việc xử lý các thao tác điều khiển độc lập các motor, đồng thời cho phép khói xử lý trung tâm điều khiển được nhiều động cơ hơn, với công suất lớn hơn.
- **Khối động cơ:** Bao gồm các motor đảm nhiệm công việc giúp xe di chuyển.

### 3.3 THIẾT KẾ CÁC KHỐI

#### 3.3.1. Khối xử lý trung tâm

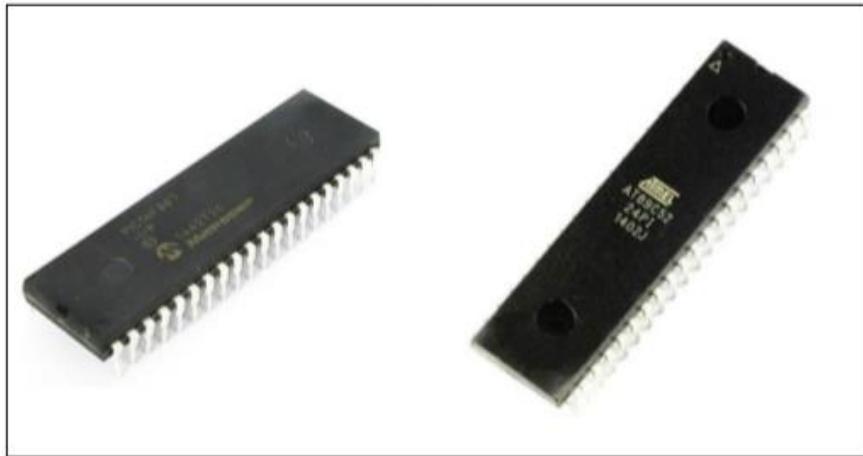
Có chức năng nhận vào và xử lý các tín hiệu từ Module nhận tín hiệu GPS, cảm biến khoảng cách, module định thời gian thực và cảm biến la bàn số. Sau đó truyền các tín hiệu điều khiển ra khói Module điều khiển động cơ để làm cho động cơ hoạt động theo cách người sử dụng lập trình.

Về việc đi thiết kế khói xử lý trung tâm thì thị trường hiện nay đáp ứng cho chúng ta rất nhiều các phương án khác như: vi điều khiển hãng Microchip (tiêu biêu

## CHƯƠNG 3. THIẾT KẾ

PIC 16F887, 18F4550, vi điều khiển hãng ATMEL (AT89C52), RASPBERRY PI, ARDUINO.

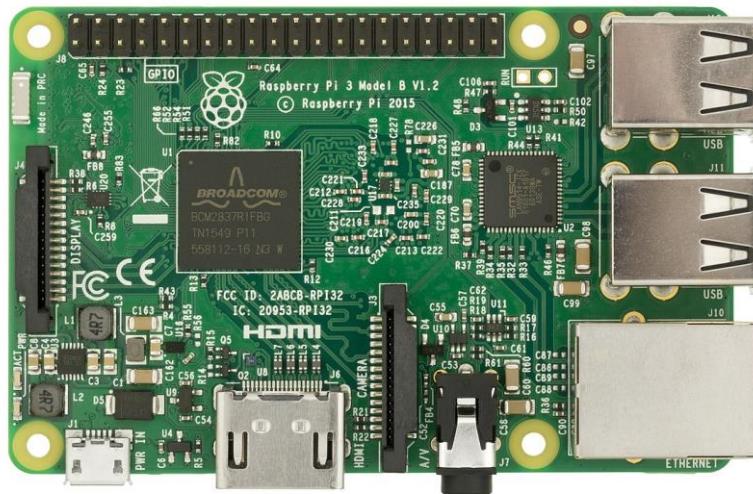
### a) Vi điều khiển Pic và AT89C52



Hình 3.2: Vi điều khiển PIC 16F887 và AT89C52

PIC 16F và 18F rất phổ biến, sử dụng nhiều, nhưng kết nối chân hơi phức tạp, phải hiểu cấu trúc và nguyên lý của PIC, không những vậy chúng ta cần phải thiết kế mạch in khá tốn thời gian, không tiện cho người sử dụng.

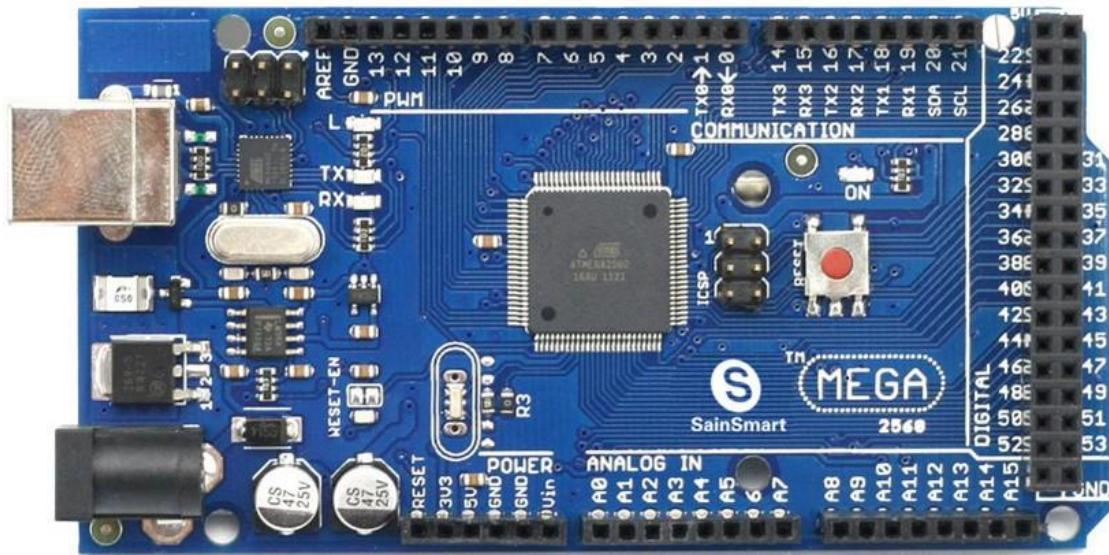
### b) Raspberry Pi



Hình 3.3: Bo mạch Raspberry Pi

RASPBERRY PI như là một máy tính nhỏ, mạnh mẽ với những ứng dụng cao và đi cùng với đó là chi phí bỏ ra cũng không hề nhỏ.

### c) Arduino



Hình 3.4: Bo mạch Arduino Mega 2560

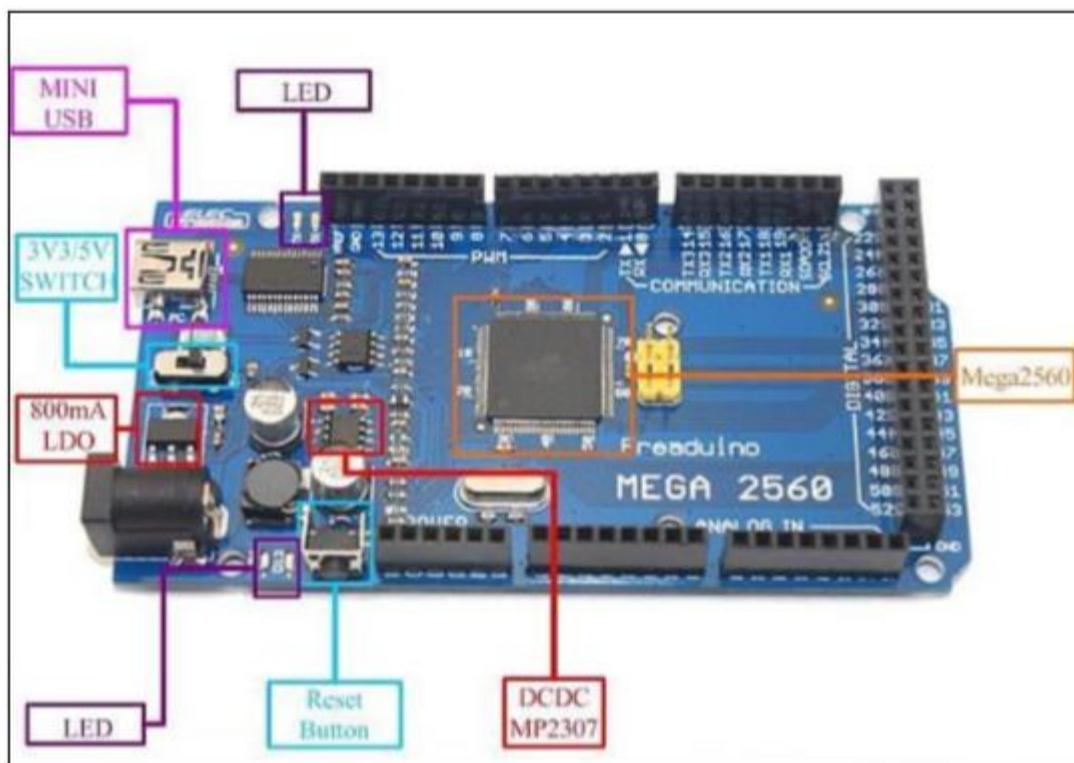
Trong những năm gần đây, sự phát triển mạnh mẽ của Arduino đã cho thấy tầm ảnh hưởng rất nhiều tới thế giới hiện tại, với cộng đồng nguồn tài liệu phong phú trên mạng, thư viện hỗ trợ hầu hết các Module, các sản phẩm thiết bị ngoại vi khác, thao tác dễ xử lý và bo mạch gọn nhẹ là ưu điểm sử dụng với đòn bẩy tài chính này.

Với những yêu cầu của đòn bẩy tài chính này là thực hiện các công việc điều khiển đơn giản và dung lượng chương trình không quá lớn nên ở đây lựa chọn sử dụng Arduino là đủ để đáp ứng cho hệ thống hoạt động hiệu quả ngoài ra các dòng Arduino trên thị trường hiện nay còn có độ thông dụng cao, giá thành rẻ hơn (chỉ từ 75 nghìn vnđ) so với các bộ Kit raspberry (hơn 1 triệu vnđ) hay ARM (có giá từ 400 nghìn vnđ). Cụ thể là lựa chọn dòng Arduino Mega để đi thiết kế vì so với Arduino Uno thì Arduino Mega hỗ trợ nhiều chân hơn, nhờ thế chúng ta không cần phải lo lắng về vấn đề thiếu chân nối. **Vì thế nhóm chọn Arduino Mega để lập trình, điều khiển.**

Arduino được sử dụng trong đòn bẩy tài chính này là Arduino Mega 2560 với thiết kế phần cứng bao gồm 54 chân digital (15 chân có thể sử dụng như các chân PWM, 16

## CHƯƠNG 3. THIẾT KẾ

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. Ngoài ra Arduino Mega 2560 cơ bản là giống với Arduino Uno R3 mà ta vẫn thường hay sử dụng, chỉ khác về số chân và tính năng, do đó ta có thể lập trình tương tự như lập trình cho Arduino Uno R3. Sơ đồ linh kiện và chân của Arduino Mega 2560:



Hình 3.5: Sơ đồ linh kiện trong Arduino Mega 2560

Bảng 3.1: Thông số kỹ thuật Board Arduino Mega 2560.

Vị trí điều khiển	ATmega2560
Điện áp hoạt động	5V
Điện áp đầu vào (được đề nghị)	7-12V
Điện áp đầu vào (giới hạn)	6-20V
Số lượng chân I / O	54 (trong đó có 15 cung cấp sản lượng PWM)
Số lượng chân Input Analog	16

## CHƯƠNG 3. THIẾT KẾ

Dòng điện DC mỗi I / O	20 mA
Dòng điện DC với chân 3.3V	50 mA
Dòng điện DC với chân 5V	500 mA
Bộ nhớ flash	256 KB trong đó có 8 KB sử dụng bởi bộ nạp khởi động
SRAM	8 KB
EEPROM	4 KB

### 3.3.2. Khối động cơ:

Bao gồm 2 motor đảm nhiệm công việc giúp xe di chuyển.

Với yêu cầu là để chạy xe với công suất vừa và tốc độ trung bình nên có thể lựa chọn các động cơ DC. Động cơ DC không chỉ có độ thông dụng cao, giá thành rẻ mà còn dễ dàng trong lập trình điều khiển để làm các mô hình. Ở đây lựa chọn loại động cơ DC giảm tốc JGA25-370 12V-250RPM.



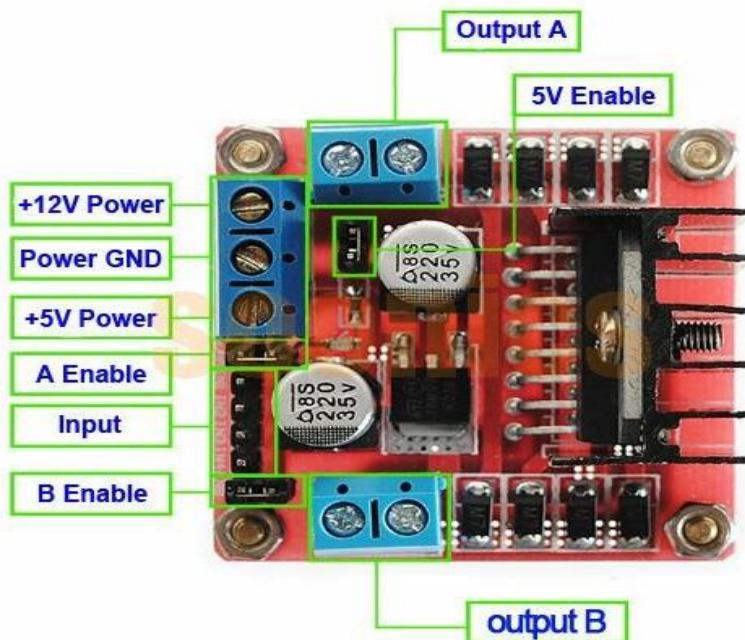
Hình 3.6: Một số hình ảnh về động cơ DC.

#### ❖ Thông số kỹ thuật:

- Điện áp hoạt động: 6-12V.
- Dòng điện tiêu thụ: 340mA – 370 mA.
- Số vòng/phút: 250 vòng/ phút khi không có tải

### 3.3.3. Khối điều khiển động cơ

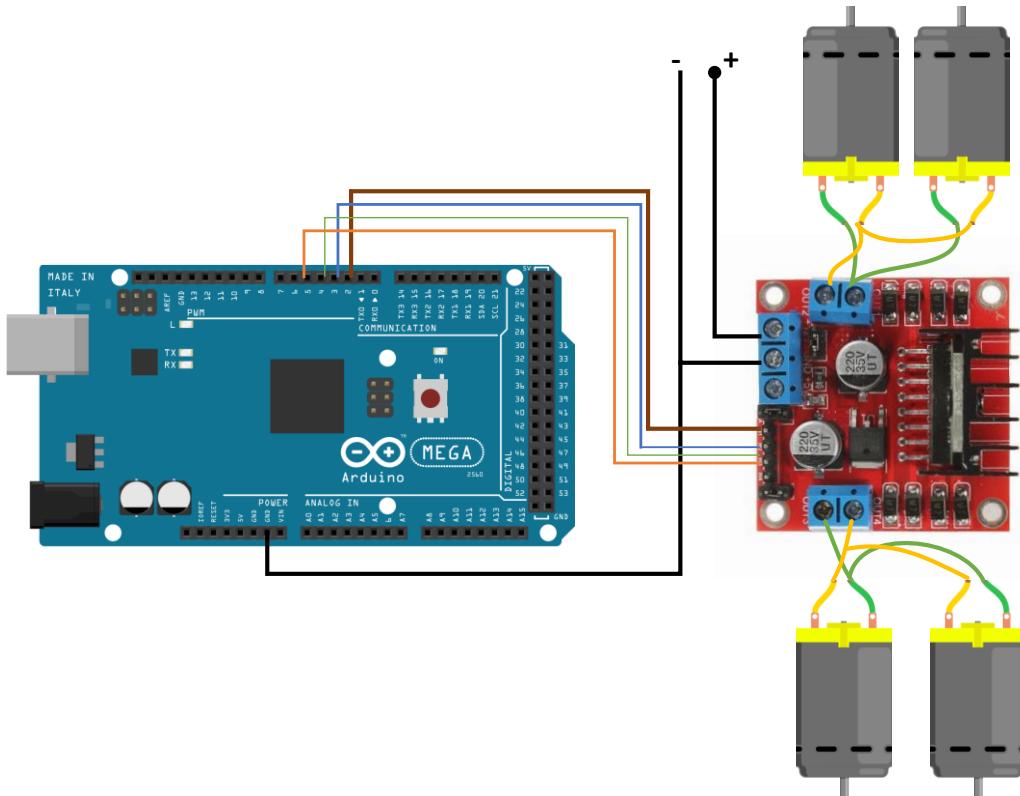
Với yêu cầu của đề tài là điều khiển nhiều động cơ DC 12V và không cần thiết đảo chiều quay (để làm cho xe chạy lùi) thì ở đây có thể lựa chọn giải pháp là sử dụng các module mạch cầu H để điều khiển. Trên thị trường hiện nay thì có rất nhiều các mạch điều khiển động cơ DC như Module DC L298, L293, L9110, hay các dòng mini DRV8833, H1, TB6612... Tuy nhiên để có thể giao tiếp dễ dàng với board Arduino Mega 2560 và điều khiển đồng thời 2 động cơ DC theo từng cặp theo yêu cầu ở đề tài này thì việc lựa chọn sử dụng mạch điều khiển động cơ L298 là một giải pháp thích hợp. Module điều khiển động cơ L298 còn có độ thông dụng cao, dễ lắp đặt, dễ mua trên thị trường với giá thành rẻ (chỉ từ khoảng 50 ngàn vnd).



Hình 3.7: Hình ảnh và sơ đồ chân module L298

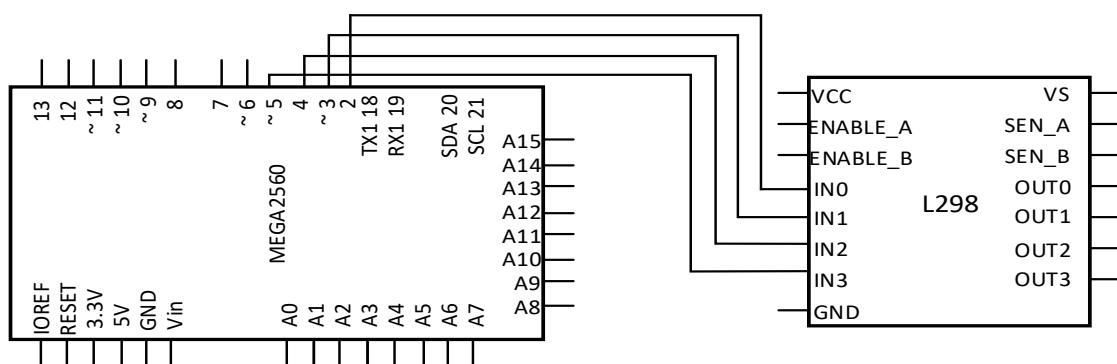
## CHƯƠNG 3. THIẾT KẾ

### ❖ Sơ đồ kết nối với Arduino



Hình 3.8: Kết nối module L298 với Arduino và động cơ

### ❖ Mạch nguyên lý L298 kết nối với Arduino



Hình 3.9: Sơ đồ nguyên lý kết nối khỏi cảm biến với Arduino

### 3.3.4. Khối nhận tín hiệu GPS

Module GPS là được coi như là một máy thu của vệ tinh GPS, điều này là rất cần thiết để làm các dự án về định vị vị trí và chuyển động. Nó có tốc độ cập nhật rất nhanh, trả về tọa độ rất chính xác, đã được tích hợp cả anten nên GPS độ nhạy cao trên mạch, kết nối và sử dụng rất đơn giản là những ưu điểm của loại module này. Thay vì chọn GPS Neo 7 và GPS Neo 8 thì GPS Neo 6 mặc dù là phiên bản cũ, có tốc độ cập nhật tọa độ chậm hơn, tuy vậy có giá thành rẻ và công suất tiêu thụ ít hơn và xác định tọa độ cũng khá chính xác sau một vài lần test, phù hợp với yêu cầu của đề tài nên được dùng để sử dụng. Một số tính năng chúng ta có thể dễ nhận thấy như:

- Xác định tọa độ (kinh tuyến, vĩ tuyến) hiện tại của module trên bề mặt trái đất với sai số nhỏ nhất  $< 1m$ .
- Xác định thời gian quốc tế được cấp bởi đồng hồ nguyên tử trên vệ tinh gửi về. Từ đó bạn cũng có thể suy ra thời gian đồng hồ nơi ở của bạn theo tắc trừ múi giờ. Khỏi cần module RTC.
- Chỉ cần 3 vệ tinh là bạn có thể xác định được tọa độ, chỉ cần 4 vệ tinh là bạn có thể xác định được độ cao hiện tại so với mực nước biển.
- Có thể tính toán ra tốc độ di chuyển, hướng di chuyển của vật thể được gắn module GPS.
- Giải các bài toán về tính toán giữa 2 điểm bất kì, tính diện tích ở một không gian cực kì rộng lớn.

## CHƯƠNG 3. THIẾT KẾ

### ❖ Sơ đồ chân:



Hình 3.10: Sơ đồ chân của module GPS Neo 6

### Tương ứng

- E: Để hở chân này (không nối)
- G (GND): Nối với GND của arduino.
- R: là chân RX của module này.
- T: là chân TX của module này.
- V: nối pin này với nguồn 3.3v (đến 5v)
- P: chân này để đặt ngắt báo module đã cập nhật xong vị trí sẵn sàng truyền dữ liệu (để hở pin này nếu không biết dùng).

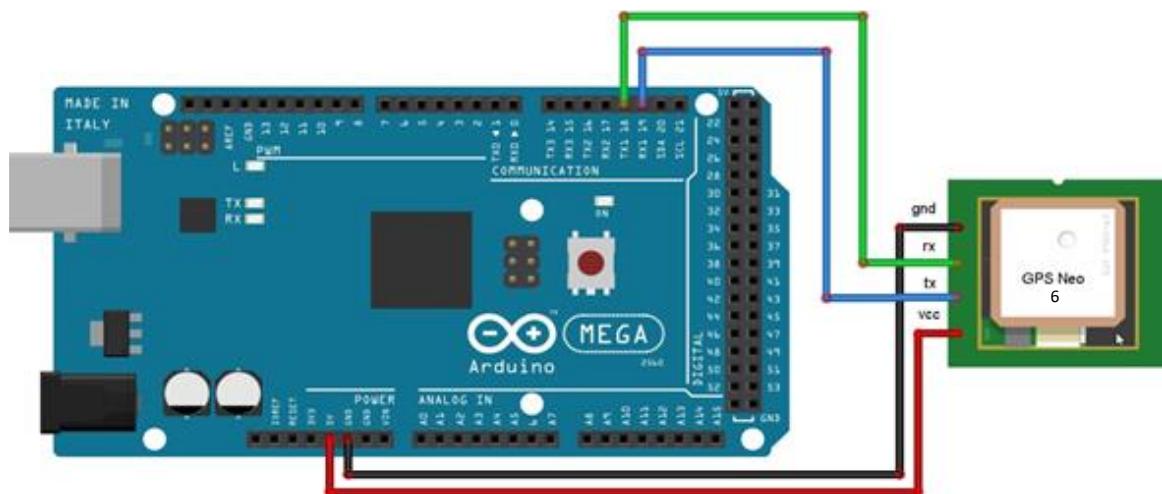
### ❖ Thông số kỹ thuật:

- Dòng hoạt động 50Ma
- Điện áp sử dụng: 3-5 VDC
- Ic chính: Neo 6 nguyên chiếc của Ublox.
- Hỗ trợ GPS-QZSS-GLONASS
- Giao tiếp: Serial (UART TTL)
- Anten rời hoặc Anten gồm- dán trực tiếp lên mạch để bắt sóng GPS tốt hơn
- Chu kỳ cập nhật tọa độ 0,1 s đến 1s tùy từng địa hình có thuận lợi hay không.

## CHƯƠNG 3. THIẾT KẾ

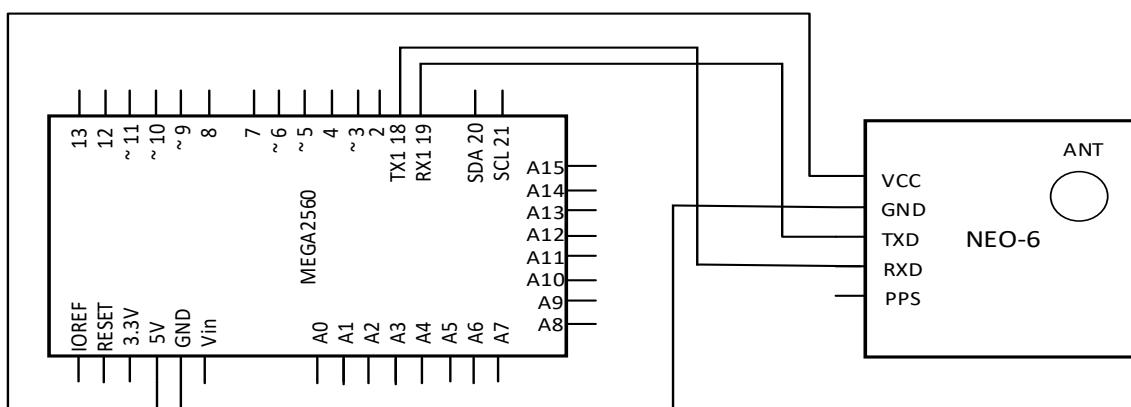
Module GPS-Neo 6 được thiết kế với hai chân TXD và RXD chuẩn UART để giao tiếp với chân TX và RX của Arduino nên rất tiện lợi cho việc kết nối. Tín hiệu mà module GPS-Neo 6 thu được phụ thuộc khá nhiều về thời gian và không gian. Trong ngày những khoảng thời gian khác nhau thì GPS sẽ trả về những tọa độ tương ứng khác nhau, sai số xấp xỉ khoảng 3-5m. Đồng thời, những nơi quang đãng, thoáng, ít vật cản sẽ là một yếu tố quan trọng để vệ tinh truyền về tín hiệu chính xác nhất.

- ❖ Sơ đồ kết nối với Arduino



Hình 3.11: Kết nối module GPS với Arduino

- ❖ Mạch nguyên lý module GPS kết nối với Arduino



Hình 3.12: Sơ đồ nguyên lý kết nối khôi nhận tín hiệu GPS với Arduino

### 3.3.4. Khối cảm biến la bàn

Khi nhắc tới la bàn thì chắc chắn chúng ta có thể ngầm suy đoán được chức năng xác định phương hướng của loại dụng cụ này. Áp dụng theo nguyên tắc của những con la bàn định hướng bình thường, người ta đã sáng chế ra module la bàn số. Hiện tại module này chỉ có HMC5883L là đáp ứng tốt nhất những yêu cầu của đế tài.

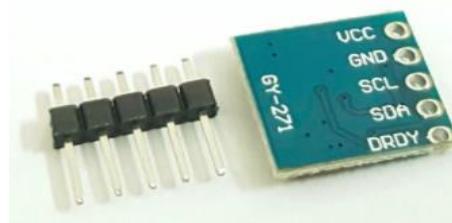
Cảm biến la bàn số HMC5883L hoạt động ở điện áp 3.3V và dòng hoạt động và tiêu thụ thấp 0.5mA. Với việc hoạt động dựa trên chuẩn giao tiếp I2C nên dễ dàng kết nối với Arduino và có hỗ trợ nhiều thư viện.



Hình 3.13: Cảm biến la bàn số HMC5883L

Cảm biến la bàn số HMC5883L còn có thể dùng để đo từ trường thô hoặc các nguồn từ trường mạnh hơn gần nó, cảm biến có thể cảm nhận được nguồn từ trường xung quanh nó như của nam châm hoặc điện trường, khi phát hiện được từ trường từ bên ngoài, nó có thể xác định được khoảng cách tương đối hoặc chiều đến vật phát ra từ trường đó.

❖ **Sơ đồ chân:**



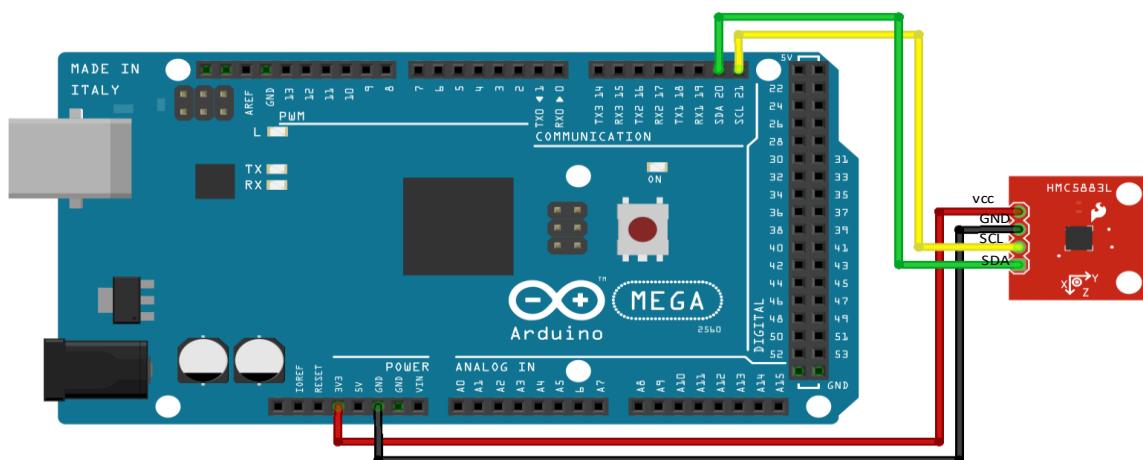
Hình 3.14: Sơ đồ chân cảm biến la bàn số HMC5883L

## CHƯƠNG 3. THIẾT KẾ

### ❖ Thông số kỹ thuật:

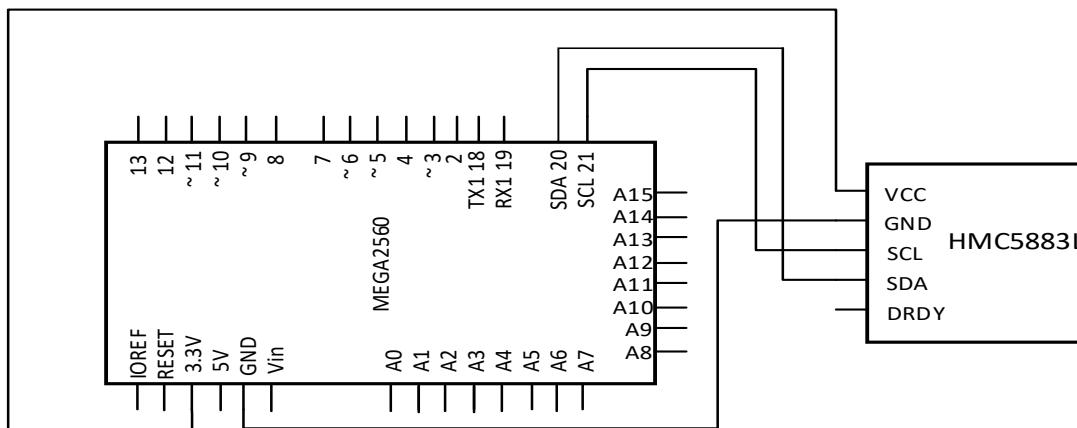
- Sử dụng Chip: HMC5883L.
- Điện áp sử dụng: Giao thức truyền thông IIC tiêu chuẩn (3.3V ~ 5V TTL)
- Giao tiếp: I2C
- Dải đo:  $\pm 1,3\text{-}8$  Gauss
- Kích thước: 14mm x 13mm

### ❖ Sơ đồ kết nối với Arduino



Hình 3.15: Kết nối module la bàn số HMC5883L với Arduino

### ❖ Mạch nguyên lý module HMC5883L kết nối với Arduino



Hình 3.16: Sơ đồ nguyên lý kết nối khói cảm biến ba lá

### 3.3.5. Khối cảm biến khoảng cách

Có chức năng cung cấp cho hệ thống khả năng đọc khoảng cách tới vật cản.

Để thiết kế khối cảm biến có thể lựa chọn sử dụng các loại cảm biến khoảng cách sử dụng sóng hồng ngoại (cảm biến DS 30C4, E18-D80NK...), sóng radar (cảm biến RCWL -0516, HB100...), sóng siêu âm (cảm biến SRF-04, SRF-05, US-115...) hoặc các cảm biến line. Mạch chỉ có nhu cầu đọc khoảng cách với tốc độ chuyển đổi và độ chính xác không quá cao nên ở đây lựa chọn cảm biến siêu âm SRF-04 là đủ để đáp ứng. Cảm biến siêu âm SRF-04 còn có kích thước nhỏ gọn và có thiết kế các chân dễ dàng kết nối và tương thích với thiết kế chân cắm trên board Arduino.

Tuy nhiên việc lựa chọn cảm biến siêu âm chỉ là một giải pháp phù hợp trong đề tài này vì tiết kiệm được chi phí. Theo các nghiên cứu, sóng siêu âm có nhiều ảnh hưởng không tốt đến sức khỏe con người, đặc biệt là với trẻ nhỏ. Vì vậy để tạo ra một sản phẩm thùng rác an toàn cho cả trẻ em thì người thiết kế nên lựa chọn các loại cảm biến sử dụng những sóng không gây ảnh hưởng đến sức khỏe người sử dụng, trong đó có thể kể đến các cảm biến sử dụng sóng hồng ngoại.

#### *Khảo sát cảm biến siêu âm SRF-04*

Cảm biến khoảng cách siêu âm HC-SR04 được sử dụng rất phổ biến để xác định khoảng cách vì rẻ và chính xác. Cảm biến sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 đến 300 cm, nếu đo được chính xác thời gian và không có nhiễu, mạch cảm biến siêu âm trả về kết quả cực kì chính xác. Điều này phụ thuộc vào cách viết chương trình không sử dụng các hàm delay.



Hình 3.17: Cảm biến siêu âm SRF-04.

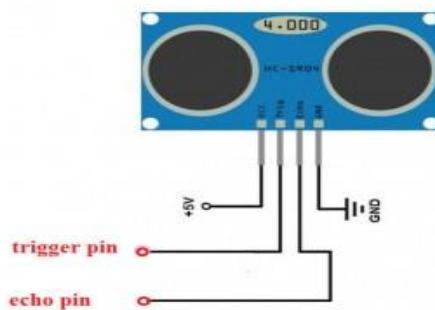
## CHƯƠNG 3. THIẾT KẾ

---

Sóng siêu âm chỉ bị dội lại khi gặp 1 số loại vật cản, nếu phát sóng siêu âm vào chăn, nệm bạn sẽ không nhận được sóng phản hồi.

### *Sơ đồ chân Cảm biến siêu âm SRF-04:*

Cảm biến HC-SR04 có 4 chân là: Vcc, Trig, Echo, GND.



Hình 3.18: Các chân của Cảm biến siêu âm SRF-04.

#### ❖ Thông số kỹ thuật:

- Nguồn cung cấp: 5V DC
- Dòng tiêu thụ: nhỏ hơn 2mA
- Tần số hoạt động: 40KHz
- Khoảng cách lớn nhất đo được: ~300 cm
- Khoảng cách nhỏ nhất đo được: ~ 2 cm
- Góc quét: nhỏ hơn 30 °.
- Độ chính xác: 0,3 cm.

#### ❖ Nguyên lý hoạt động:

Để đo khoảng cách, ta sẽ phát 1 xung rất ngắn (nhỏ hơn 10  $\mu$ s) từ chân Trig. Sau đó, cảm biến sẽ tạo ra 1 xung HIGH ở chân Echo cho đến khi nhận lại được sóng phản xạ ở chân này. Chiều rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và quay trở lại.

Từ thời gian thu được từ xung echo bên trên ta suy ra khoảng cách vật cản dựa trên công thức sau:

$$\text{Distance (mm)} = (\text{Measured Echo Time} * 344) * 1000 / 2 \quad (3.1)$$

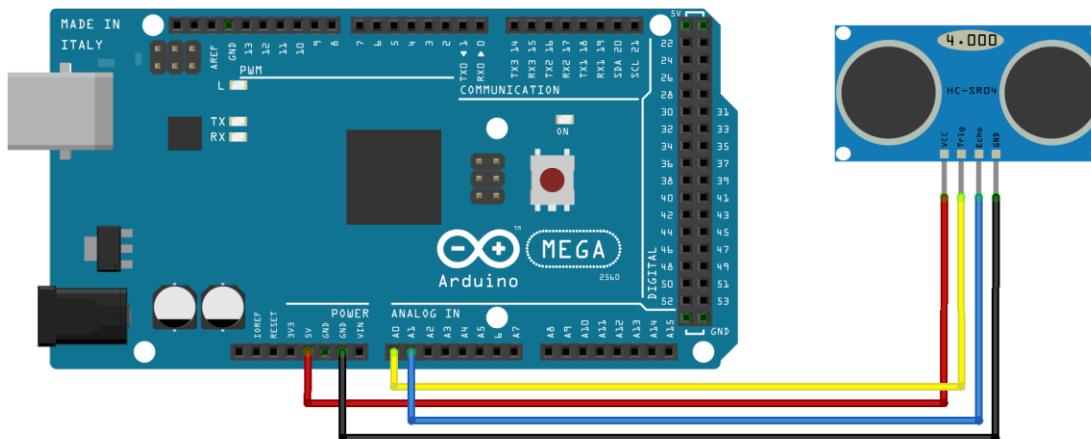
Trong đó:

- Measured Echo Time là thời gian đo xung mức cao từ chân echo, đơn vị là giây (s).
- 344 là vận tốc âm thanh lan truyền trong không khí.
- 1000 là hệ số nhân để đưa khoảng cách đo được về đơn vị mm.
- là hệ số chia do chỉ lấy  $\frac{1}{2}$  quãng đường đo được tức là chỉ từ vật cản đo đến module.

Với nguyên lý hoạt động dễ hiểu trong việc truyền nhận tín hiệu, module cảm biến khoảng cách HC-SR04 được sử dụng cho rất nhiều mục đích khác nhau. Và trong đề tài “Thùng rác thông minh” đã chứng minh thấy được sự đa năng trong module này.

Vì có sự trùng lặp nên những hình sơ đồ kết nối và sơ đồ kết nối dưới đây sẽ kết nối tượng trưng một cảm biến khoảng cách với arduino, các cảm biến khoảng cách khác cũng được kết nối với các chân Analog của arduino tương tự như vậy.

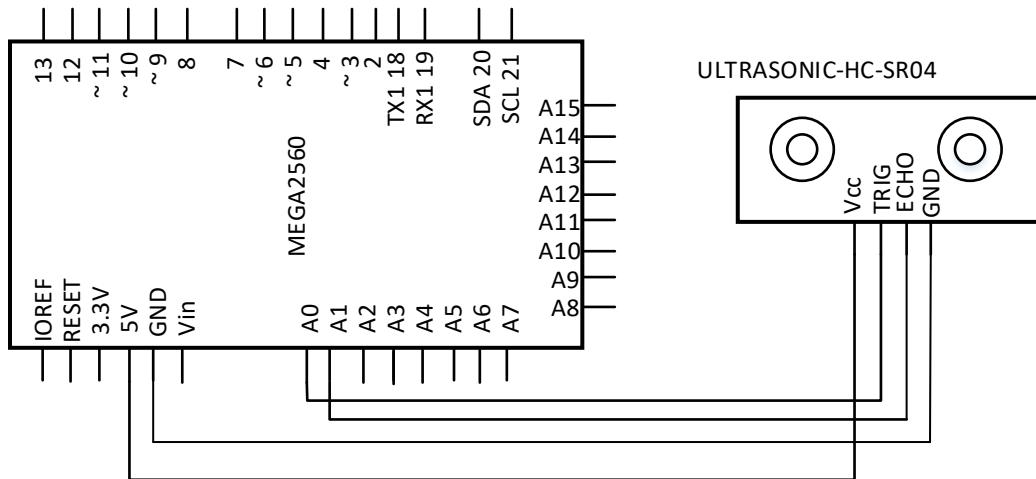
### ❖ Sơ đồ kết nối với Arduino



Hình 3.19: Kết nối module cảm biến khoảng cách HC-SR 04 với Arduino

## CHƯƠNG 3. THIẾT KẾ

- ❖ Mạch nguyên lý module khoảng cách HC-SR04 kết nối với Arduino



Hình 3.20: Sơ đồ nguyên lý kết nối khối cảm biến khoảng cách

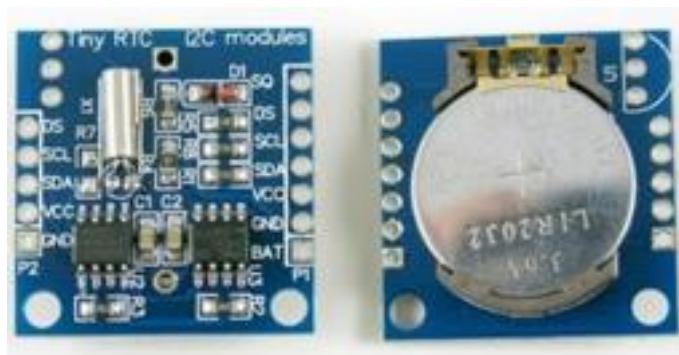
### 3.3.6. Khởi động thời gian

Với yêu cầu của đè tài là thùng rác thông minh sẽ tự động di chuyển tới thùng rác tổng vào đúng một thời gian đã định nào đó từ trước hàng ngày (giả sử là tầm khoảng 8h tối) thì đó là lý do mà khởi động này được ra đời.

Về vấn đề này chắc chắn chúng ta phải dùng đến những đồng hồ thời gian thực (Real Time Clock) hay còn gọi là RTC để đo và tính chính xác thời gian: Thứ, ngày, tháng, năm, phút, giây.

Những tính năng thời gian thực đó còn được tích hợp trong module GPS- Neo 6 như đã khảo sát ở trên. Tuy vậy, khả năng đáp ứng chậm cùng với những sai số tương đối lớn và khả năng điều khiển khó nên ứng dụng này trên GPS không được triển khai

Thay vào đó là các module riêng biệt làm công việc tính real time như RTC DS1307, RTC3231. Ở đây, lựa chọn module RTC DS1307 bởi tính năng của nó và giá thành rẻ, công suất tiêu thụ nhỏ, đáp ứng nhu cầu sử dụng trong đè tài.



Hình 3.21: Hình ảnh thực tế moduel RTC DS1307

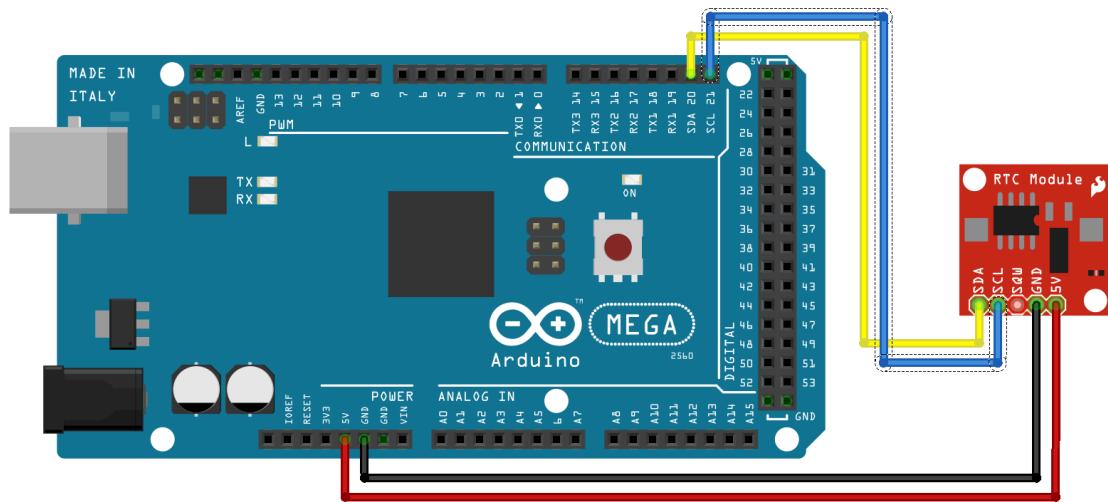
Module thời gian thực DS1307 (RTC) có chức năng lưu trữ thông tin ngày tháng năm cũng như giờ phút giây, nó sẽ hoạt động như một chiếc đồng hồ và có thể xuất dữ liệu ra ngoài qua giao thức I2C. Module được thiết kế kèm theo một viên pin đồng hồ vì thế có thể sử dụng cả khi không có điện áp cung cấp, có khả năng lưu trữ thông tin lên đến 10 năm mà không cần cấp nguồn 5V từ bên ngoài. Module đi kèm với EEPROM AT24C32 có khả năng lưu trữ thêm thông tin lên đến 32Kbit.

❖ **Thông tin kỹ thuật:**

- Nguồn cung cấp: 3.3- 5VDC.
- Bao gồm 1 IC thời gian thực DS1307
- Khả năng lưu trữ 32K bit với EEPROM AT24C32.
- Lưu trữ thông tin giờ phút giây AM/PM.
- Lịch lưu trữ chính xác lên đến năm 2100.
- Có pin đồng hồ lưu trữ thông tin.
- 5-pin bao gồm giao thức I2C sẵn sàng giao tiếp: INT (QWO), SCL, SDA, VCC và GND
- Có ngõ ra tần số 1Hz.

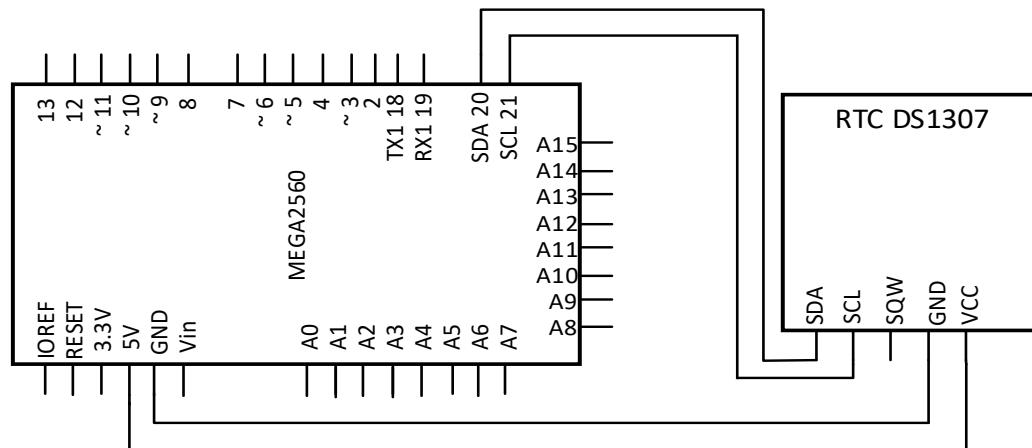
## CHƯƠNG 3. THIẾT KẾ

- #### ❖ Sơ đồ kết nối với Arduino:



Hình 3.22: Kết nối module RTC DS1307 với Arduino

- ❖ Mạch nguyên lý module RTC DS1307 kết nối với Arduino



Hình 3.23: Sơ đồ nguyên lý kết nối khói định thời gian

### 3.3.7. Khối nguồn

Có chức năng cấp nguồn cho toàn mạch hoạt động.

Nguồn sử dụng trong đề tài này ngoài việc cung cấp điện áp hoạt động cho board Arduino Uno R3 và module điều khiển động cơ L298 ra thì còn đòi hỏi sự linh động, gọn gàng vì được gắn cho đề của thùng rác đặt ngoài trời nên không thể kết nối với nguồn điện thông qua ổ cắm cố định. Vậy giải pháp ở đây là có thể sử dụng Pin để thay thế cho nguồn điện.

Có nhiều loại pin sạc như chúng ta đã thấy hiện nay. Tuy nhiên, việc sử dụng pin sạc sẽ ứng dụng rất tốt với nhu vật nhỏ gọn và sử dụng trong gia đình.

Với yêu cầu là một thùng rác thông minh chạy ngoài trời, việc sử dụng pin sạc sẽ khá tốn năng lượng. Nếu chúng ta sử dụng ngoài trời thì sẽ chỉ trong khoảng thời gian ngắn từ 1-2 tiếng sẽ hết năng lượng do dòng điện tiêu thụ của Arduino Mega 2560 và động cơ là tương đối lớn. Do đó, sử dụng một bình ác quy sẽ là một thiết kế phù hợp với loại đề tài này hơn.



Hình 3.24: Hình ảnh thực tế bộ Ac quy sử dụng trong đề tài

Ác quy này khi được sử dụng sẽ được cố định và đi dây chắc chắn, để tránh trường hợp lỏng hoặc chập điện.

❖ **Thông tin kỹ thuật:**

- Ác quy loại khô
- Dung lượng: 12V- 6Ah (10HR)
- Kích thước (mm): 121 x 62 x 131 (dài x rộng x cao)

Được dùng cho các dòng xe máy thông dụng hiện nay

### 3.3.8. Tính toán theo lý thuyết

Từ những thông số và những linh kiện đã chọn ở trên, chúng ta có những thông số cần tính toán như:

❖ **Tính dung lượng tiêu thụ trung bình (lý thuyết):**

Giả sử vị trí đặt thùng rác cách điểm đổ rác 153m. Đoạn đường đi và về sẽ là 306m

Tổng thời gian đi về:  $t = s / v = 306 / 0,515 = 594$  giây = 10 phút

Số giờ sử dụng bình acquy  $6000 / 672 = 8,93$  giờ.

Giả sử 1 ngày thùng rác đi về 2 lần

=> Số ngày sử dụng bình acquy là  $8,92 \times 3 = 26$  ngày

=> Sau 26 ngày ta cần sạc bình 1 lần.

❖ **Tính vận tốc xe (lý thuyết):**

Sử dụng động cơ giảm tốc GA25 250 vòng/phút

Tốc độ không tải 250 vòng/phút

Tốc độ có tải 175vòng/phút, dòng điện tiêu thụ 240mA

Ta cho động cơ chạy với 70% công suất tương đương 123vòng/phút và dòng điện tiêu thụ 168mA

Tổng cộng 4 động cơ:  $I_{DC} = 168 \times 4 = 672$  mA

Đường kính bánh xe 8cm => chu vi bánh xe  $C = \pi \times d = \pi \times 0,08 = 0,251$  m  
(Động cơ quay 1 vòng thì thùng rác đi được 0,251 m)

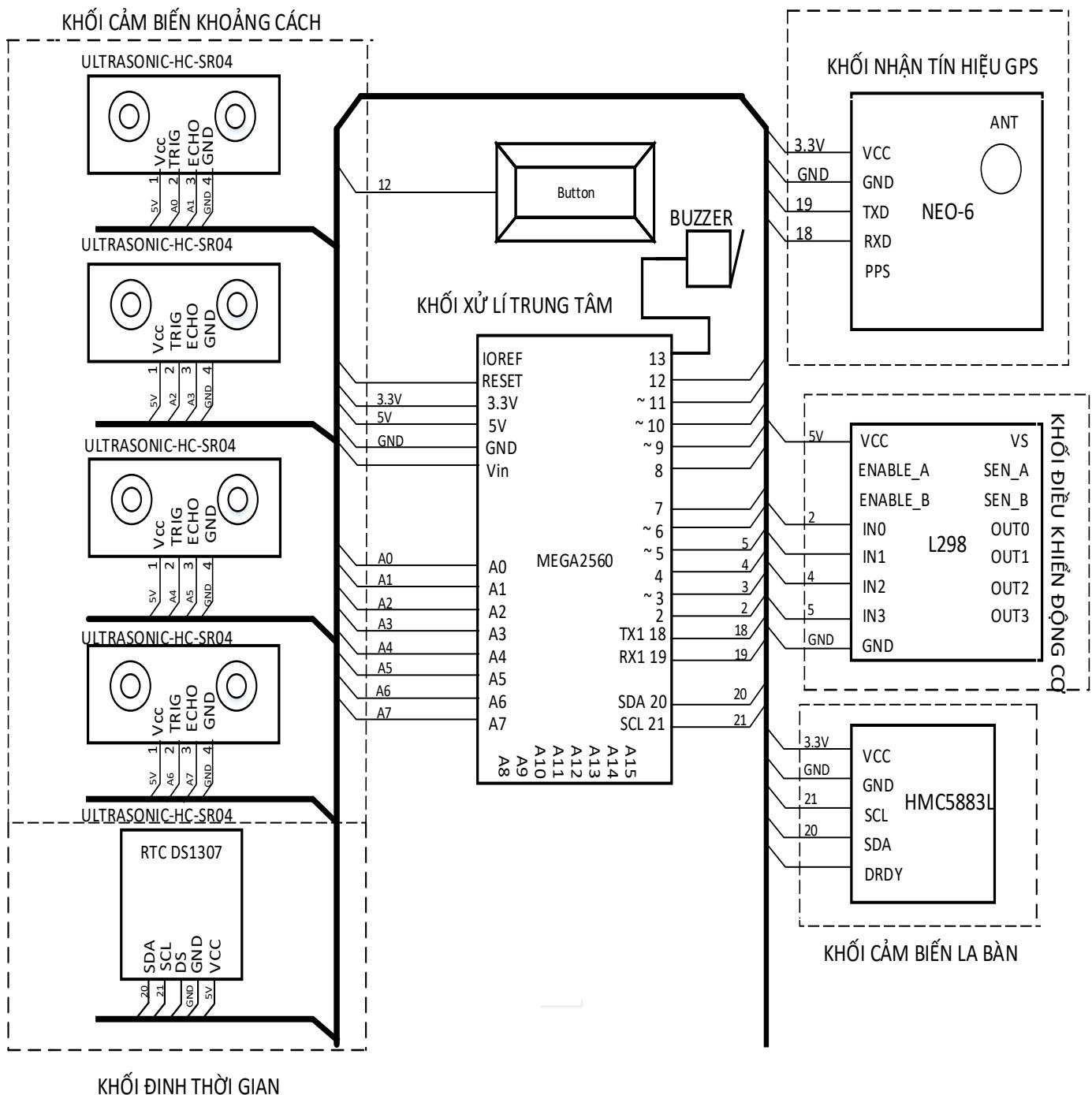
=> Trong 1 phút thùng rác đi được đoạn đường  $s = 123 \times 0,251 = 30,9$  m

=> Vận tốc di chuyển của thùng rác  $v = s / t = 30,9/60 = 0,515$  m/s

### 3.4 SƠ ĐỒ NGUYÊN LÝ TOÀN MẠCH

Với yêu cầu là thiết kế “Thùng rác thông minh” với những tính năng tự động di chuyển đến vị trí rác tổng khi rác đầy hoặc tự di chuyển khi tới một thời gian cho phép nào đó và tự chạy về bằng vị trí cũ bằng một nút nhấn cùng với đó là việc kết hợp các cảm biến siêu âm, la bàn để thùng rác di chuyển chính xác. Sau khi khảo sát và lựa chọn linh kiện, chúng ta có sơ đồ nguyên lý toàn mạch như sau:

### CHƯƠNG 3. THIẾT KẾ



Hình 3.25: Sơ đồ nguyên lý toàn mạch

## CHƯƠNG 3. THIẾT KẾ

---

### ❖ Giải thích sơ đồ nguyên lý toàn mạch:

- Dùng nguồn Ac quy, cấp nguồn 12V vào Arduino và khởi động cơ
- Ban đầu khi cấp nguồn bằng Switch, module GPS sẽ được kích hoạt, cấp nguồn từ Arduino 3.3V vào và thiết lập khởi động trong khoảng một phút. Sau khi đèn module GPS-Neo 6 nhấp nháy 3 lần là lúc nó đã được thiết lập xong. Lúc này module GPS sẽ nhận tọa độ đã được định vị và gán sẵn trong code qua hai chân TX và RX để nhận và gửi dữ liệu qua lại giữa Arduino và module
- Khi lượng rác trong thùng được nhận thấy đã đầy khi tín hiệu nhận ra ở chân Trig (nối vào A0 của Arduino) và truyền ngược lại về chân Echo (nối vào chân A1 của Arduino), từ đó bằng cảm biến siêu âm bên trong thì lúc này module GPS sẽ chuyển qua đọc tọa độ vị trí đầu tiên.
- Tùy theo số lượng vị trí đã được định vị trên google map và được gán từ lúc đầu. Tín hiệu GPS được thu từ anten thu của module GPS sẽ cập nhật những tọa độ đó để di chuyển tới. Muốn di chuyển đúng hướng, ta phải kết hợp thêm module la bàn số để thùng rác biết được đúng phương hướng để di chuyển.
- Trong lúc di chuyển, ta có kết hợp với 3 con cảm biến siêu âm để né vật cản, và 1 con siêu âm để tự động mở thùng rác hoạt động tương tự như phát hiện rác đầy
- Thay vì sử dụng Timer đã được tích hợp trong GPS, mạch sử dụng mạch riêng RTC DS1307 riêng, để đề phòng trường hợp GPS gặp sự cố về thời gian

## **Chương 4: THI CÔNG**

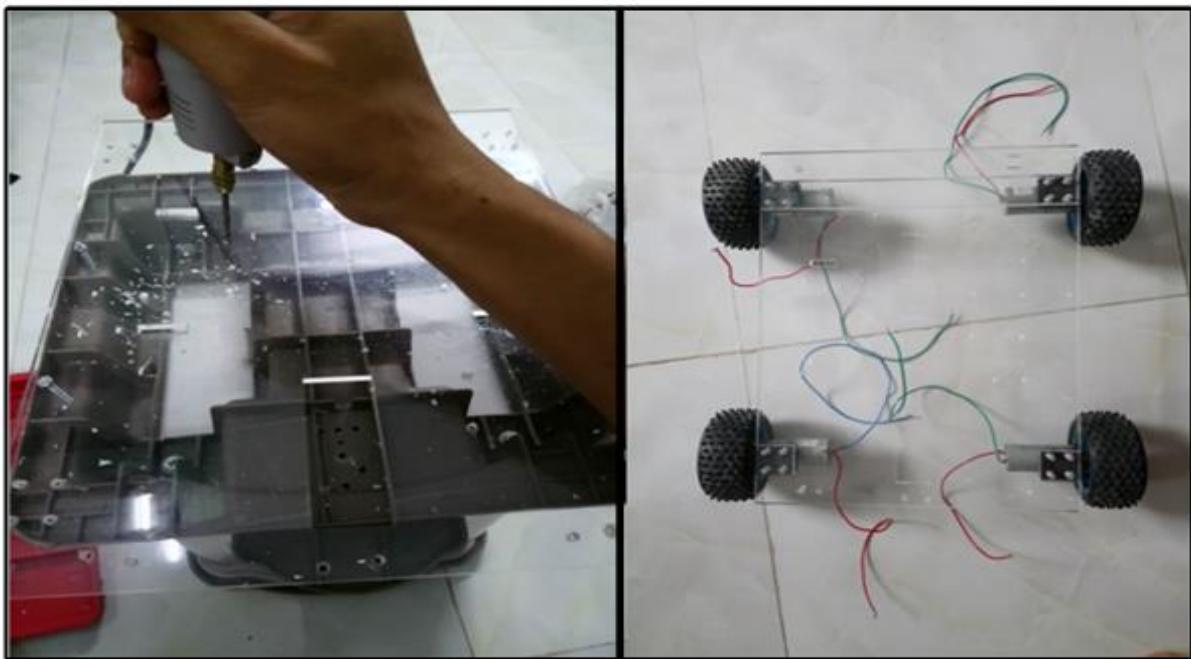
### **4.1 GIỚI THIỆU**

Trong chương này là quá trình thi công mô hình thùng rác, lập trình, lắp ráp phần cứng và test các module. Bên cạnh đó là hình vẽ được chụp từ mô hình thực của thùng rác bên ngoài, hình chụp các kết quả chạy mà mô hình có thể có tính tới thời điểm hiện tại.

### **4.2 THI CÔNG MÔ HÌNH XE RÁC**

- Chuẩn bị vật liệu: Thước, kéo, mica, khoan, ốc vít, 1 thùng rác có sẵn ở nhà làm mô hình luôn cho tiện.

- Ban đầu đo đạc và khoan lỗ để gắn ốc vít và cố định 4 bánh xe vào phần đế và cố định phần đế vào thùng rác:

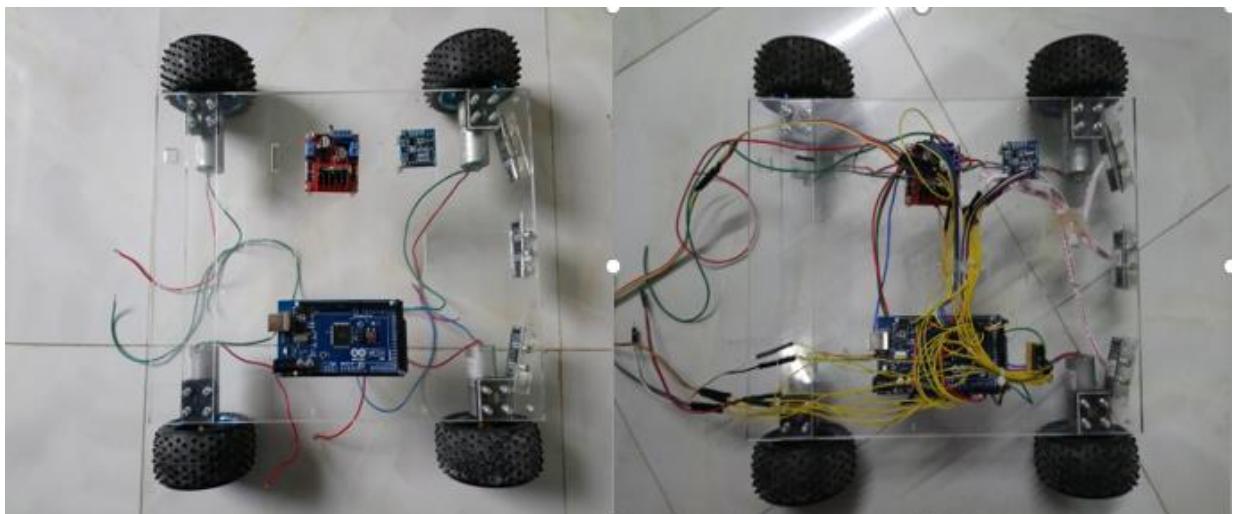


*Hình 4.1:Hình ảnh thực tế thiết kế mặt đế thùng rác.*

## CHƯƠNG 4. THI CÔNG

---

Bộ xử lý trung tâm, realtime RTC DS1307 và module L298 sẽ được đặt lên phía trên mặt của đế thùng rác và đi dây bên dưới để cho thùng rác được gọn gàng hơn sau khi ta ụp đáy của thùng rác lên những linh kiện điện tử đây.



*Hình 4.2: Hình ảnh thực tế thiết kế khung bánh xe rác*

Sau khi hoàn thành xong khung xe rác và phần đế ta sẽ tiến hành lắp ráp thùng rác vào và các module cảm biến.

- Để né vật cản theo 3 hướng, trái phải và thẳng thì cần trang bị ba module cảm biến khoảng cách lắp vào phía đầu khung thùng rác.
- Bình Ác quy được đặt phía sau thùng rác, và cố định chắc chắn bằng dây nhựa.
- Cảm biến siêu âm đo mức rác trong thùng được đặt phía trên thùng rác để dễ đi dây.
- Module đọc tọa độ GPS được đặt trên mặt thùng rác phía cao để anten dễ dàng nhận tín hiệu do GPS cập nhật liên tục.
- Module la bàn số đo góc HMC5883L đặt tách biệt ở phía sau so với các module khác để không bị nhiễu từ trường bởi các thanh kim loại cũng như của động cơ.

## CHƯƠNG 4. THI CÔNG

- Toàn bộ mạch sau này được ứng dụng thực tế sẽ được đóng gói cẩn thận để có thể dùng bền lâu.



Hình 4.3: Hình ảnh thực tế thiết kế thùng rác thông minh từ các hướng

- Tiến hành lắp ráp thêm Buzzer, led đơn và servo mở nắp vào mô hình. Vì là những linh kiện dễ thay đổi vị trí và cũng dễ sử dụng nên trong đây sẽ không trình bày kĩ.

### 4.3 LẬP TRÌNH CHO THÙNG RÁC.

#### 4.3.1. Giới thiệu phần mềm lập trình Arduino IDE:

Để lập trình được cho các board Arduino, cần phải có một công cụ gọi là **Integrated Development Environment (IDE)**. Công cụ này được đội ngũ kỹ sư của Arduino phát triển và có thể chạy trên Windows, MAC OS X và Linux.

Môi trường phát triển tích hợp IDE (Integrated Development Environment) của Arduino là một ứng dụng nền tảng được viết bằng Java, và từ IDE này sẽ được sử dụng cho Ngôn ngữ lập trình xử lý (Processing programming language) và project Wiring. Nó được thiết kế để dành cho những người mới tập làm quen với lĩnh vực phát triển phần mềm. Nó bao gồm một chương trình code editor với các chức năng như đánh dấu cú pháp, tự động brace matching, và tự động canh lề, cũng như compile (biên dịch) và upload chương trình lên board chỉ với 1 cú nhấp chuột. Một chương trình hoặc code viết cho Arduino được gọi là một sketch.

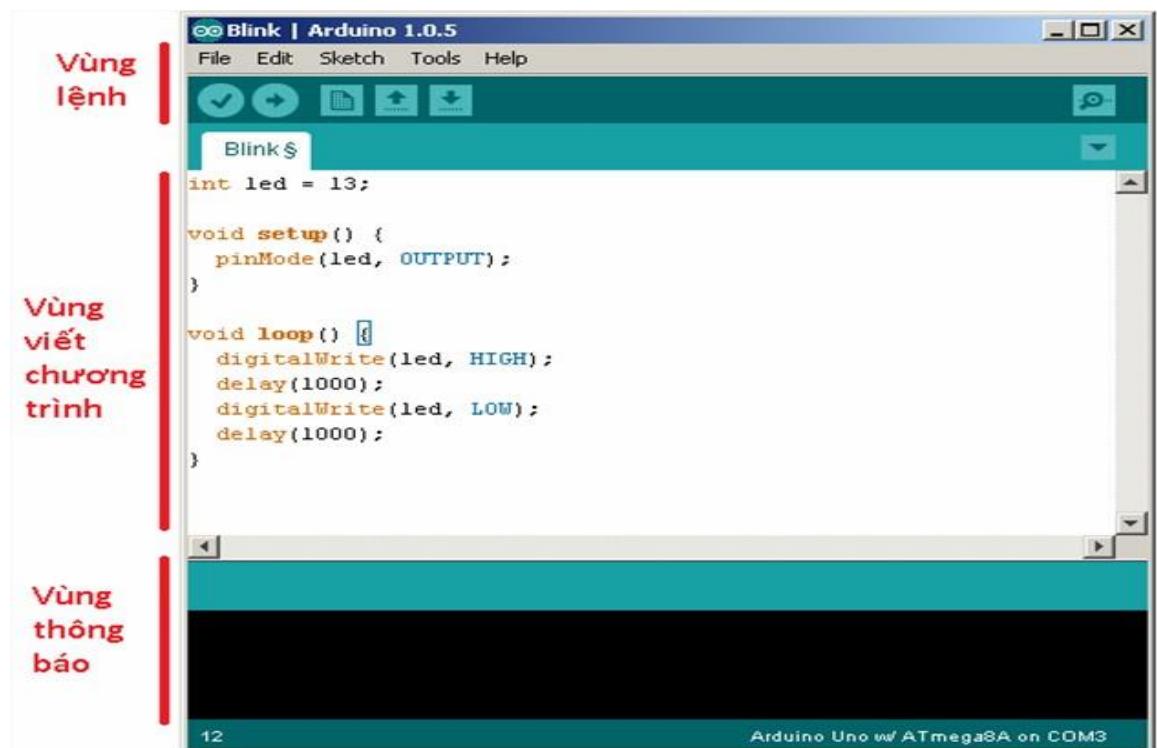
Các chương trình Arduino được viết bằng C hoặc C++. Arduino IDE đi kèm với một thư viện phần mềm được gọi là "Wiring", từ project Wiring gốc, có thể giúp các thao tác input/output được dễ dàng hơn. Người dùng chỉ cần định nghĩa 2 hàm để tạo ra một chương trình vòng thực thi (cyclic executive) có thể chạy được.

Arduino IDE sử dụng GNU toolchain và AVR Libc để biên dịch chương trình, và sử dụng AVRdude để upload chương trình lên board.

Vì nền tảng của Arduino là các vi điều khiển của Atmel, cho nên môi trường phát triển của Atmel, AVR Studio hoặc các phiên bản Atmel Studio mới hơn, cũng có thể được sử dụng để làm phần mềm phát triển cho Arduino.

### *Phần mềm Arduino IDE:*

#### *Giao diện:*



Hình 4.4: Giao diện phần mềm Arduino IDE.

*Lưu ý :* Tên chương trình của bạn được hiển thị ngay dưới dãy các Icon, ở đây nó tên là “Blink”. Để ý rằng phía sau tên chương trình có một dấu “§”. Điều đó có nghĩa là đoạn chương trình của bạn chưa được lưu lại.

#### *Vùng lệnh:*

Bao gồm các nút lệnh menu (File, Edit, Sketch, Tools, Help).

Phía dưới là bảng các icon cho phép sử dụng nhanh các chức năng thường dùng.

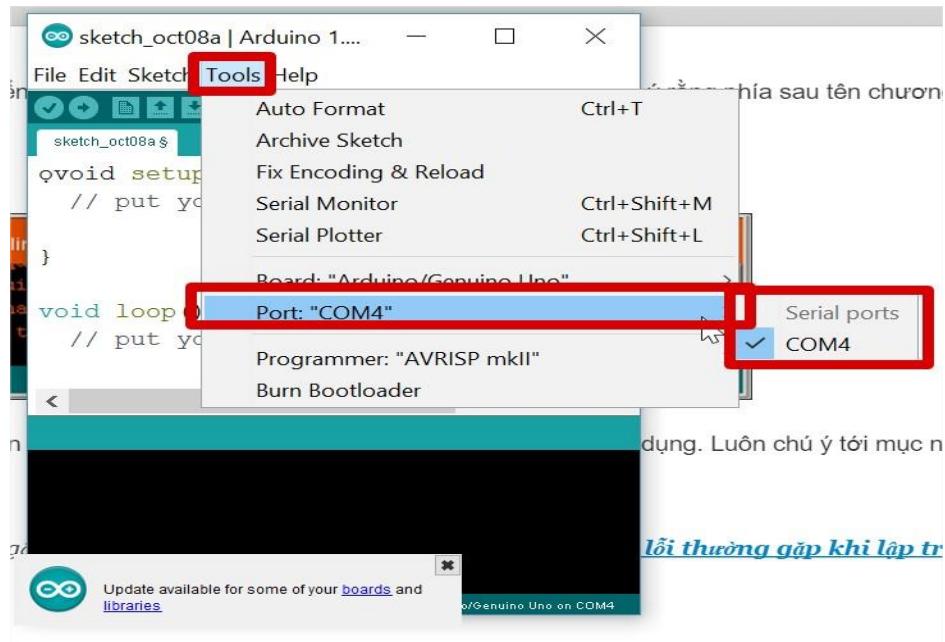
## CHƯƠNG 4. THI CÔNG

Icon	Chức năng
	Biên dịch chương trình đang soạn thảo để kiểm tra các lỗi lập trình.
	Biên dịch và upload chương trình đang soạn thảo.
	Mở một trang soạn thảo mới.
	Mở các chương trình đã lưu.
	Lưu chương trình đang soạn.
	Mở cửa sổ Serial Monitor để gửi và nhận dữ liệu giữa máy tính và board Arduino.

Hình 4.5: Chức năng các nút lệnh trong Arduino IDE.

### Một số lưu ý khi lập trình trên Arduino IDE:

Khi lập trình, các bạn cần chọn port (cổng kết nối khi gắn board vào) và board (tên board mà bạn sử dụng). Giả sử, bạn đang dùng mạch Arduino Uno, và khi gắn board này vào máy tính bằng cáp USB nó được nhận là COM4 thì cần chỉnh như thế này là có thể lập trình được.



Hình 4.6: Chọn Port kết nối.

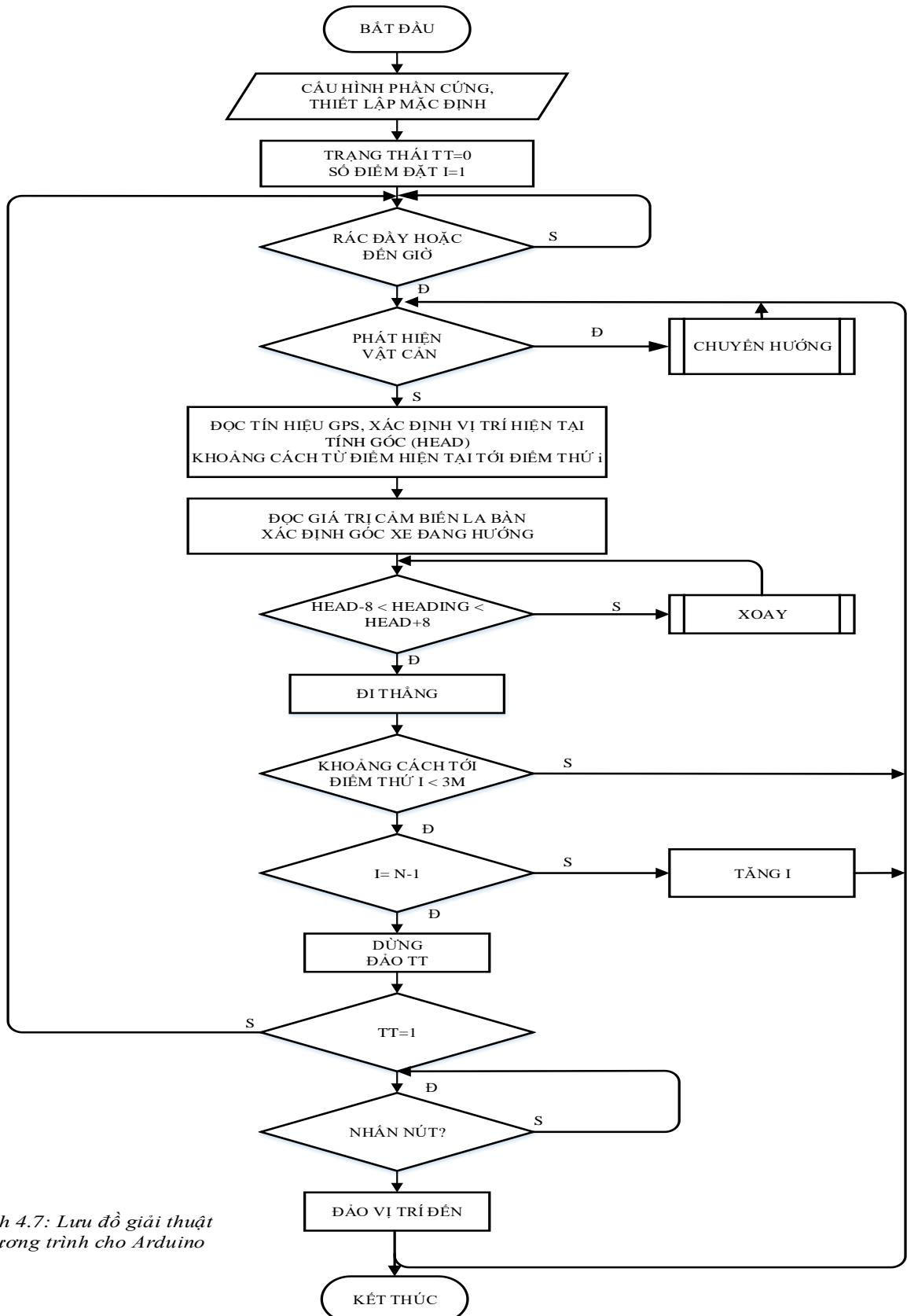
### 4.3.2. Lưu đồ giải thuật

- Yêu cầu ban đầu: Mô hình thùng rác thông minh này cần được bố trí ở ngoài trời, nơi thoáng đãng như trong khuôn viên trường Đh SPKT.
- Yêu cầu điều khiển:
  - + Khi có nguồn cung cấp, thùng rác sẽ ở trạng thái chờ.
  - + Khi phát hiện rác trong thùng đầy, thùng rác có thể tự di chuyển đến vị trí thùng rác tổng và có thể di chuyển về vị trí cũ điều khiển bằng một nút nhấn.
  - + Khi thùng rác đầy và trong lúc di chuyển ra thùng rác tổng, thùng rác sẽ phát tiếng kêu để báo động để mọi người né tránh.
  - + Trên đường đi, thùng rác có thể né được vật cản nhỏ
  - + Đến tối hoặc theo thời gian đã định sẵn, nếu rác chưa đầy thì thùng rác vẫn tự di chuyển tới vị trí thùng rác tổng và đứng sẵn ở đó tập trung cho đến khi có người bấm nút để quay lại vị trí cũ.

Với những yêu cầu như trên, chúng ta sẽ có những lưu đồ giải thuật như sau:

a) **Lưu đồ chương trình chính:**

## CHƯƠNG 4. THI CÔNG



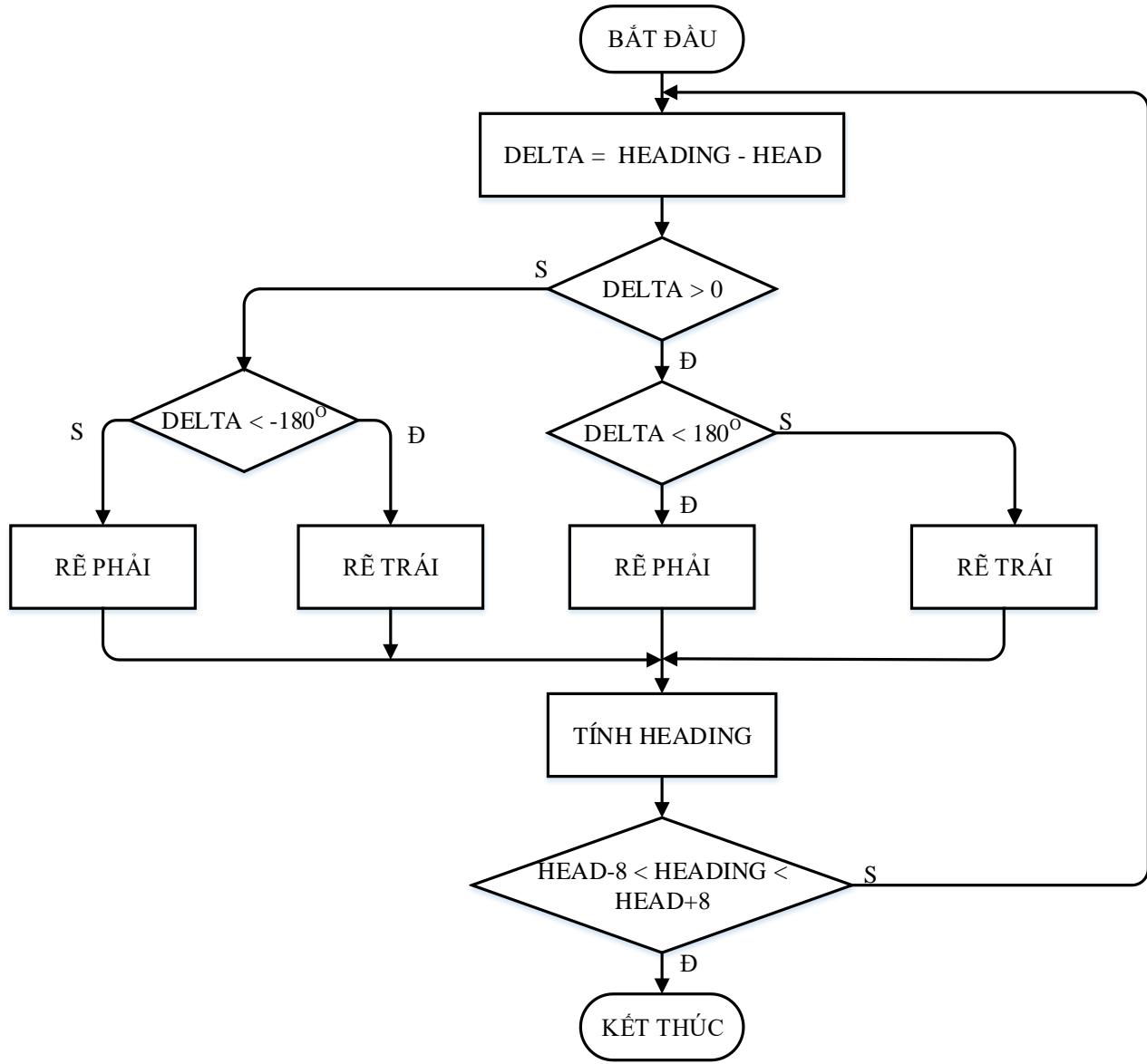
## CHƯƠNG 4. THI CÔNG

---

### ❖ Giải thích lưu đồ chương trình chính:

- Khi bắt đầu, chương trình sẽ tiến hành khởi tạo, định chức năng cho các chân cần sử dụng, các cấu thiết lập mặc định, thiết lập độ rộng xung, tốc độ cho 2 động cơ, khai báo mảng kinh độ, vĩ độ của các điểm đã định vị sẵn, khởi tạo các biến lưu dữ liệu và thiết lập tốc độ baud cho adruino.
- Khi xác định được rác đã đầy hoặc đến thời gian đặt trước ta cho thùng rác di chuyển, gấp vật cản, chương trình con chuyển hướng sẽ được thực thi, còn nếu chưa đầy, thùng rác vẫn đứng im và ở trạng thái chờ.
- Khi thùng rác di chuyển và không nhận biết được vật cản, thì xe rác sẽ xác định vị trí hiện tại, đồng thời xác định khoảng cách từ vị trí đó tới vị trí định vị thứ nhất và kết hợp cùng với la bàn số để đo góc xe đang hướng để xác định xem thùng rác có di chuyển đúng hướng hay không.
- Nếu góc đo đo được bị sai lệch nhỏ hơn  $8^0$  thì thùng rác sẽ tiếp tục đi thẳng, nếu sai lệch quá  $8^0$  thì xe thùng rác sẽ thực hiện chương trình con xoay.
- Ta có mảng biến để lưu trữ vị trí các điểm, nếu khoảng cách của xe đến điểm thứ nhất nhỏ hơn 3m, ta cho xe đến điểm thứ 2, tương tự xe đến điểm cuối cùng thì dừng. Tới vị trí định vị cuối cùng, nếu ta nhấn nút, các vị trí áy sẽ bị đảo ngược lại, làm như vậy xe thùng rác sẽ di chuyển lại các điểm ban đầu.

b) Lưu đồ chương trình xoay:

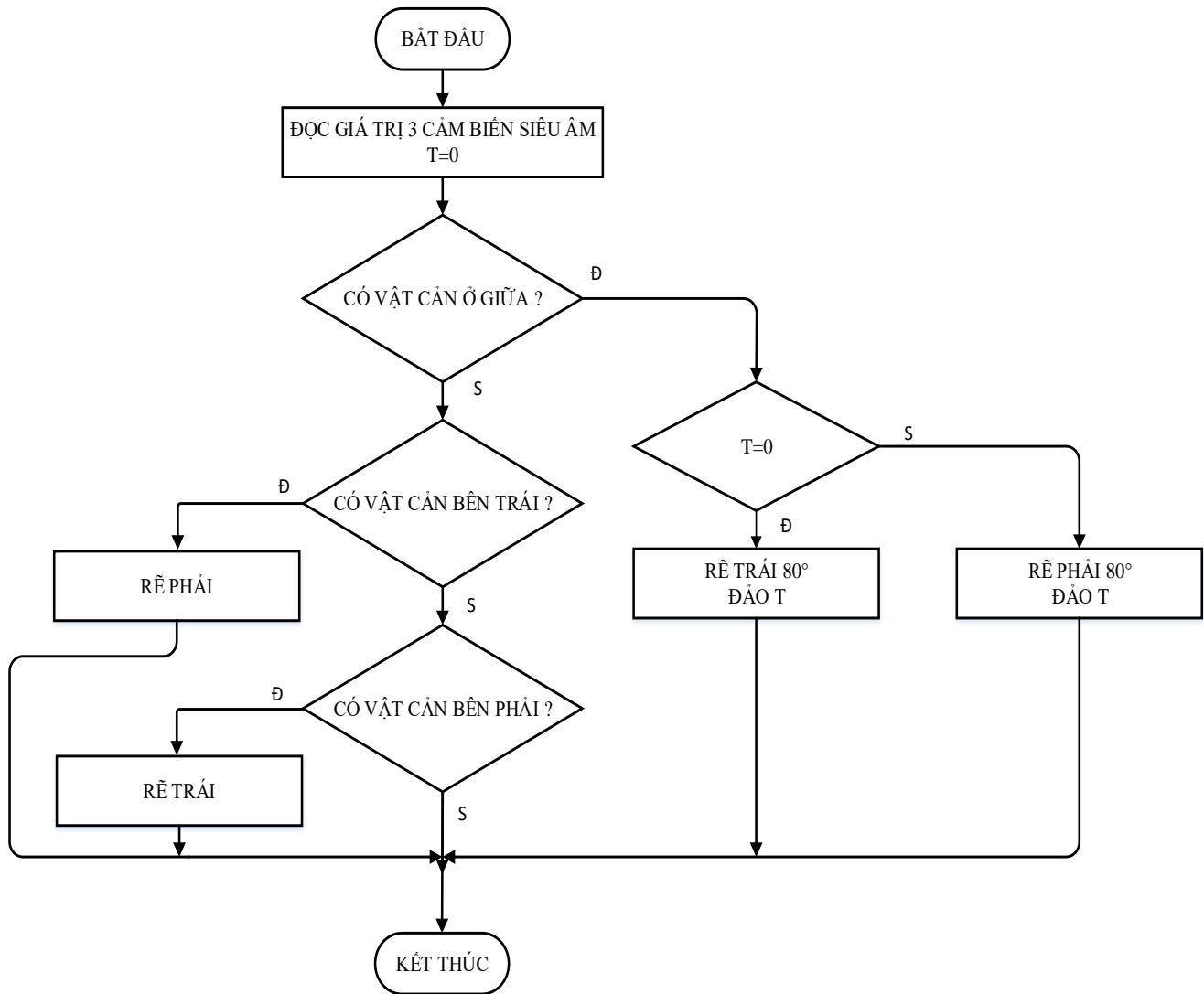


Hình 4.8: Lưu đồ chương trình con Xoay

❖ Giải thích lưu đồ chương trình con Xoay:

- Delta là độ chênh lệch giữa góc hiện tại (heading) và góc muốn hướng đến (head). Ta cần chia trường hợp cho delta để góc quay của thùng rác là nhỏ nhất.

### c) Lưu đồ chương trình chuyển hướng:



Hình 4.9: Lưu đồ chương trình con Chuyển hướng

#### ❖ Giải thích lưu đồ chương trình con Chuyển hướng:

- Chương trình con chuyển hướng được đảm nhận bởi ba module cảm biến siêu âm đặt trước thùng rác có tác dụng để né tránh vật cản, điều hướng di chuyển tốt hơn cho thùng rác.

- Ứng với ba trường hợp vật cản ở giữa, trái hay phải thì sẽ lập trình theo sơ đồ trên để điều khiển hướng di chuyển cho thùng rác cụ thể: khi nhận thấy vật cản ở

## **CHƯƠNG 4. THI CÔNG**

---

bên trái, thùng rác sẽ né bên phải; khi nhận thấy vật cản ở phải, thùng rác sẽ rẽ trái; khi vật cản ở chính giữa, ban đầu thùng rác sẽ rẽ trái, lần thứ hai gấp vật cản ở giữa thì rẽ phải và sẽ lặp lại tương tự như vậy ở các lần gấp tiếp theo.

# **Chương 5: KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

## **5.1 GIỚI THIỆU**

Chương này trình bày kết quả của cả quá trình nghiên cứu làm đề tài trong thời gian 12 tuần. Bên cạnh đó là nhận xét, đánh giá và đề xuất hướng phát triển của sản phẩm mô hình để hoàn thiện và có thể đi vào thực tế.

## **5.2 KẾT QUẢ ĐẠT ĐƯỢC**

Qua quá trình nghiên cứu và thực hiện đề tài, nhóm sinh viên nhận thấy được đề tài có khả năng ứng dụng vào thực tiễn cao, đáp ứng được nhu cầu cuộc sống hiện đại ngày nay, nhất là trong các trường đại học- cao đẳng dạy về kỹ thuật, việc tự động hóa là rất cần thiết. Đồng thời đề tài này cũng là một nguồn tài liệu có giá trị cho các bạn sinh viên những khóa tiếp theo có thể tham khảo khi nghiên cứu những đề tài có liên quan hoặc phát triển thêm nhiều khía cạnh mới của đề tài. Bên cạnh đó, nhóm sinh viên cũng bổ sung cho mình những kiến thức hay và bổ ích:

### **5.2.1 Biết cách sử dụng các cảm biến**

Các cảm biến sử dụng trong đề tài này đa phần là cảm biến siêu âm. Tuy vậy, chỉ cần nhìn vào mức độ xuất hiện của cảm biến siêu âm cũng như tính đa năng của nó kết hợp được với nhiều yêu cầu đo khoảng cách khác nhau cũng đã chỉ cho chúng ta thấy được tính thông dụng của nó như thế nào trong cuộc sống hiện đại ngày nay.

Ngoài ra còn là cảm biến la bàn số, cũng là một cảm biến rất cần thiết đối với cuộc sống con người.

Có thể nói, các cảm biến ngày nay được sản xuất ngày một đơn giản và nhỏ gọn hơn, nhằm đáp ứng nhu cầu của người tiêu dùng khó tính như hiện nay. Quá trình nghiên cứu đề tài này đã giúp nhóm thực hiện nâng cao kỹ năng đọc – hiểu cảm biến, trau dồi thêm kiến thức về các loại cảm biến. Đặc biệt là, tăng khả năng

vận dụng lý thuyết vào thực tế, có khả năng lựa chọn được loại cảm biến phù hợp với từng yêu cầu của thực tiễn đời sống.

### **5.2.2 Biết cách sử dụng Arduino Mega 2560**

Arduino Mega 2560 là một trong những board mạch thông minh được sử dụng rộng rãi. Tuy nhiên, để sử dụng thành thạo và đầy đủ các chức năng của nó thì không phải dễ. Và đó là kết quả cả quá trình nghiên cứu đề tài này mà nhóm thực hiện có được. Ngoài khả năng điều khiển các ngõ ra để điều khiển động cơ, tốc độ động cơ theo ý muốn; khả năng đọc giá trị của các loại cảm biến khác nhau. Trong đề tài này, việc ứng dụng Arduino Mega để truy xuất dữ liệu từ cảm biến cũng như dùng để điều khiển động cơ hay nhận dữ liệu từ module GPS đã cho thấy điều đó.

### **5.2.3 Biết cách sử dụng module GPS các loại**

Việc sử dụng module GPS lần đầu là một khó khăn đối với nhóm, cơ bản vì module GPS nhận tín hiệu GPS khá sai số tận 5-10m khi chạy thử nghiệm trên thực tế. Tuy nhiên, bằng sự cố gắng tìm tòi và tìm cách khắc phục, hiện tại module GPS đã đọc chính xác hơn và chỉ còn sai số <3m, khi chạy trong môi trường thực tế, việc kết hợp cùng với các cảm biến siêu âm để né vật cản đã phần nào giúp cho thùng rác chạy tốt và ổn định hơn. Sau nhiều lần đem vào trường thí nghiệm các điểm đặt thì hiện tại, xe rác đã chạy rất trơn tru, không đi lệch và đọc sai hướng như những lần đầu. Đây thực sự là một module rất hay và hữu ích, nhóm thực sự cảm thấy rất may mắn vì đây cũng là dịp và cơ hội để nghiên cứu thêm về hệ thống định vị toàn cầu và cách thức hoạt động của chúng ra sao, đồng thời từ đó, có nhiều ý tưởng mới xuất phát từ chính module đặc biệt này. Đây cũng chính là module quan trọng nhất đề tài vì xác định được vị trí và cho xe chạy đúng vị trí đó trong môi trường ngoài trời thực tế không phải là chuyện đơn giản.

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

---

### 5.3 KẾT QUẢ THỰC NGHIỆM

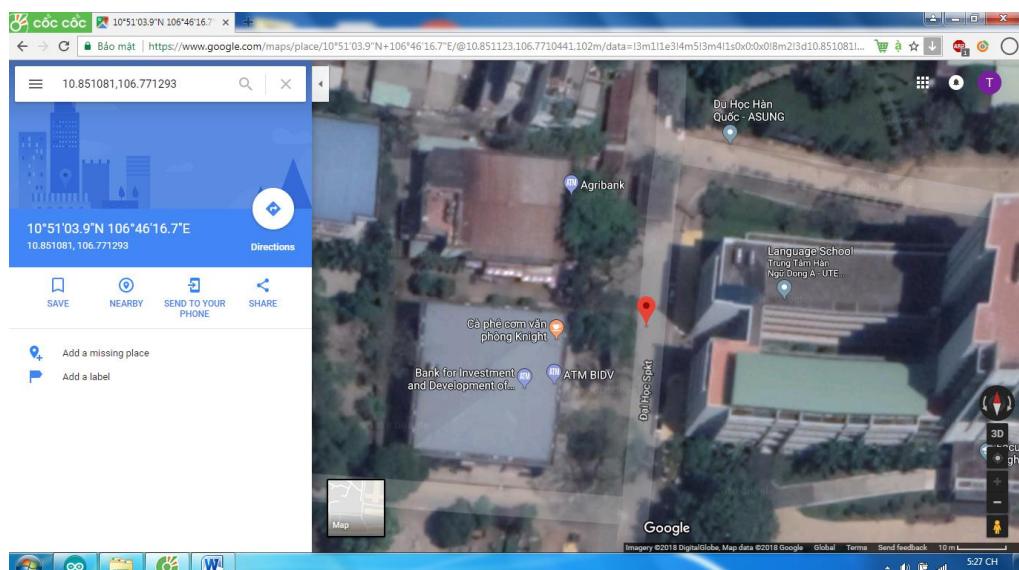
#### 5.3.1. Định vị tọa độ trên Google Maps

Để sử dụng được thùng rác thông minh này, chúng ta cần phải định vị những điểm cố định mà sẽ dự định thùng rác sẽ đi qua.

Bởi vì thùng rác thông thường luôn đặt ở một vị trí sẵn nhất định, nên những điểm tọa độ này sẽ được gán sẵn vào code trong vi xử lý.

Ta cho thùng rác chạy thử nghiệm trên đoạn đường 153m, gồm 7 điểm.

#### Điểm thứ 0 (điểm xuất phát)

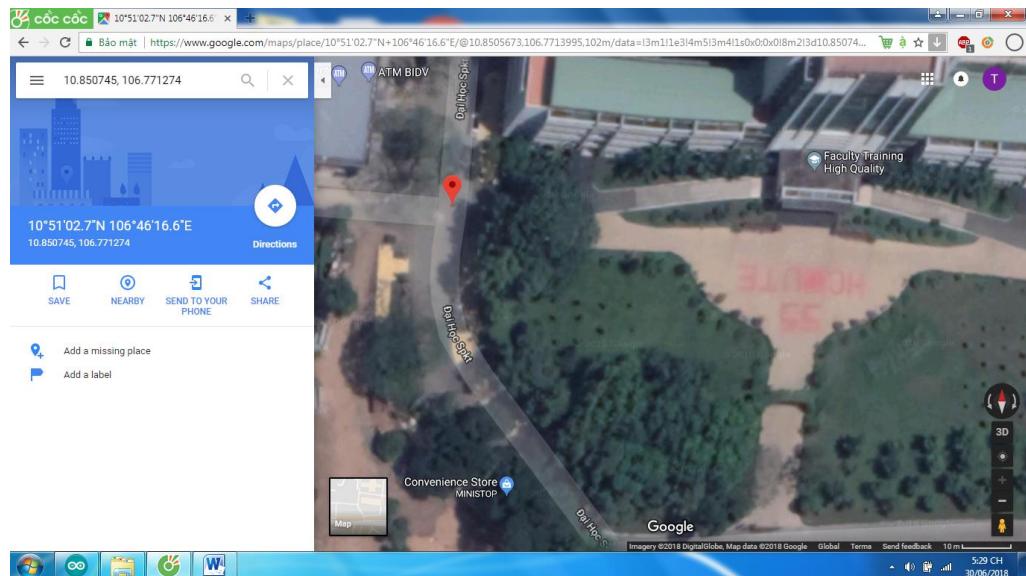


Hình 5.1:Lấy tọa độ điểm xuất phát trên Google map

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

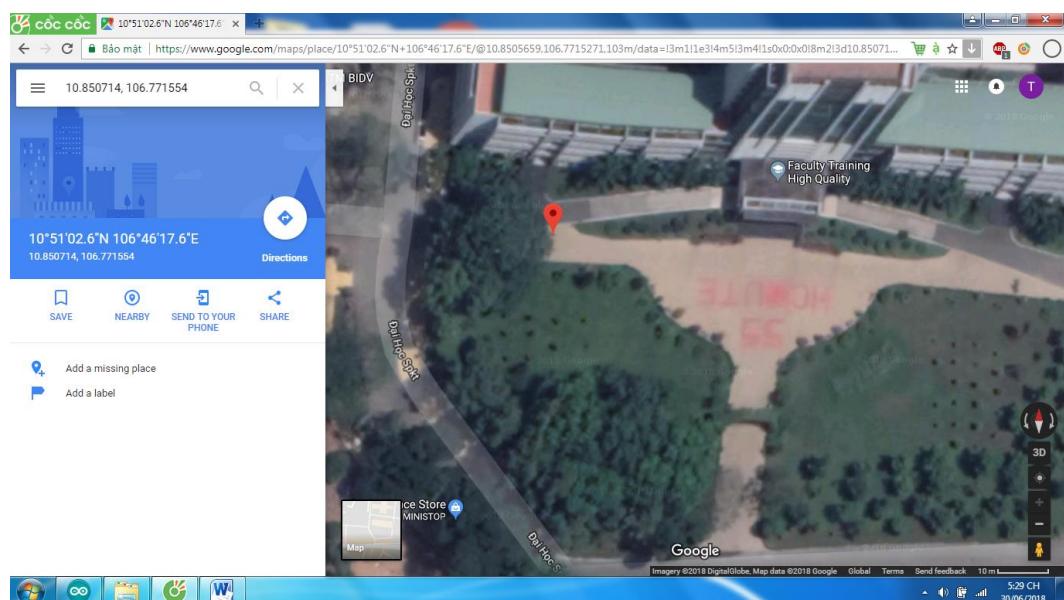
---

### Điểm thứ 1



Hình 5.2: Lấy tọa độ điểm thứ nhất trên Google map

### Điểm thứ 2

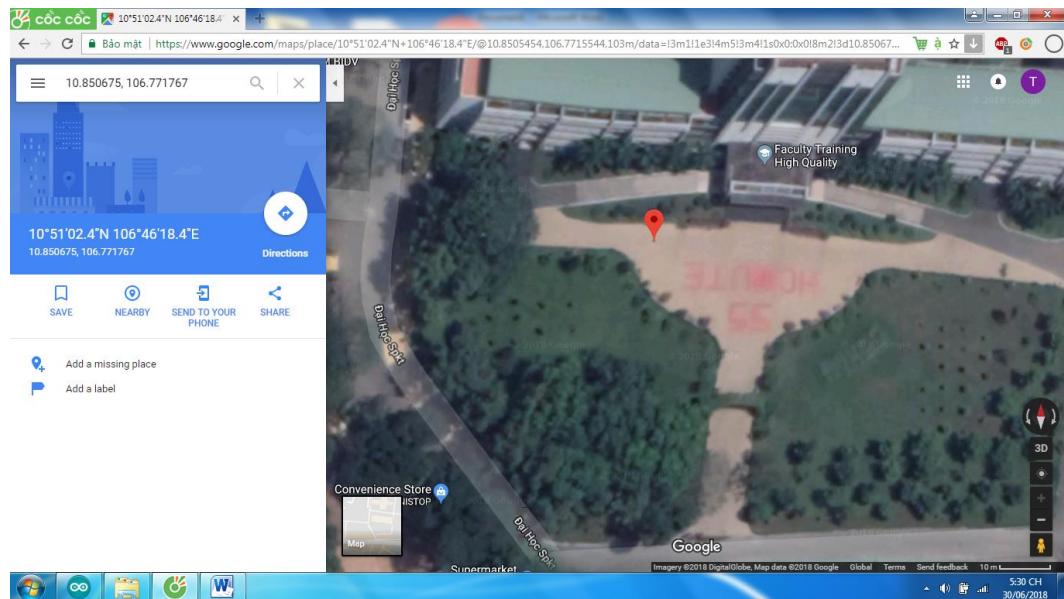


Hình 5.3: Lấy tọa độ điểm thứ hai trên Google map

## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

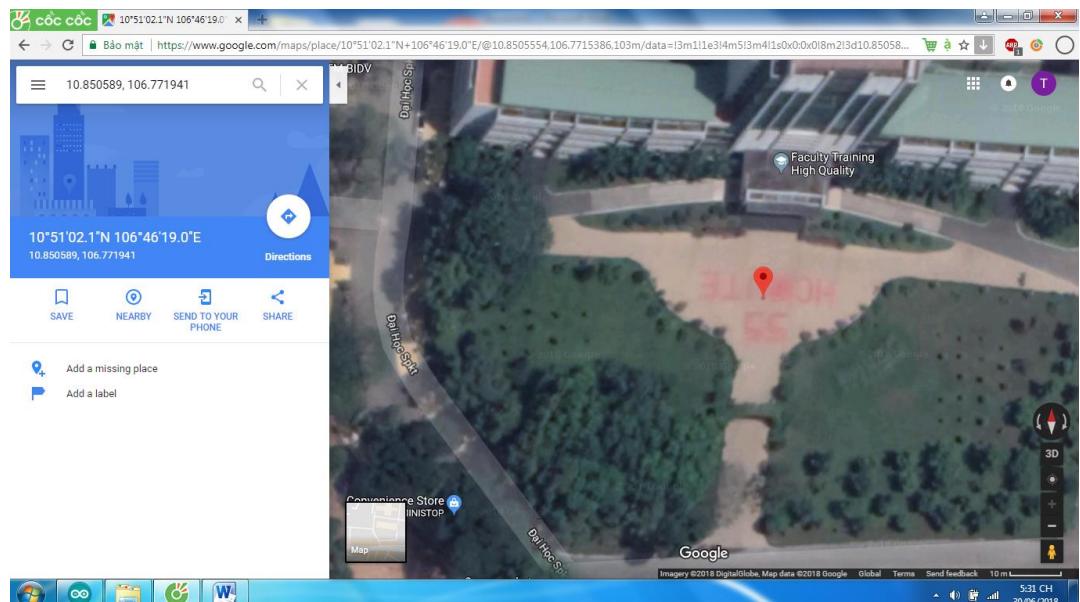
---

### Điểm thứ 3



Hình 5.4: Lấy tọa độ điểm thứ ba trên Google map

### Điểm thứ 4

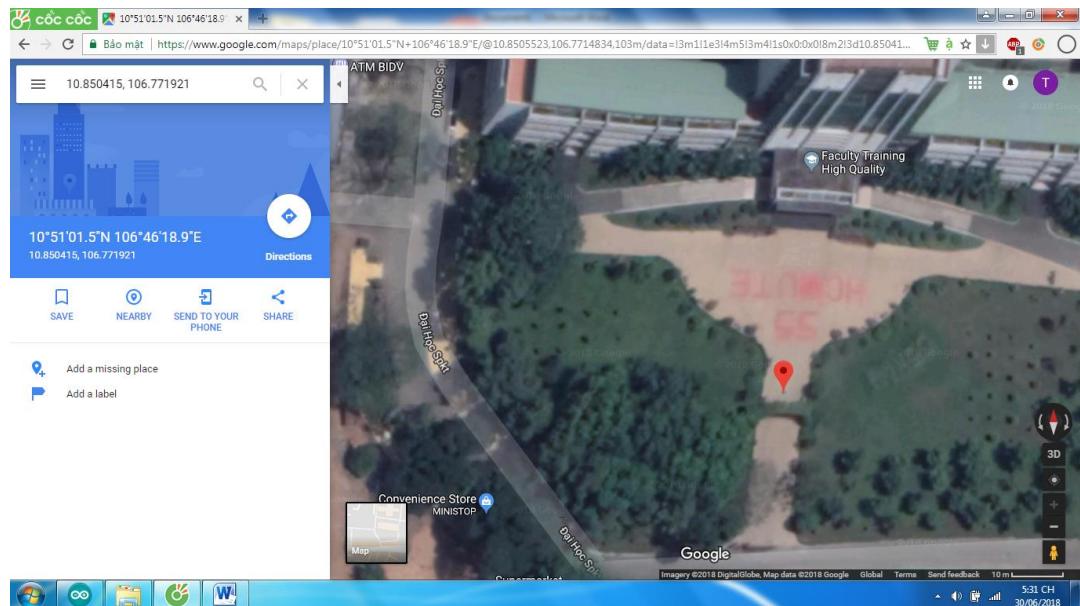


Hình 5.5: Lấy tọa độ điểm thứ tư trên Google map

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

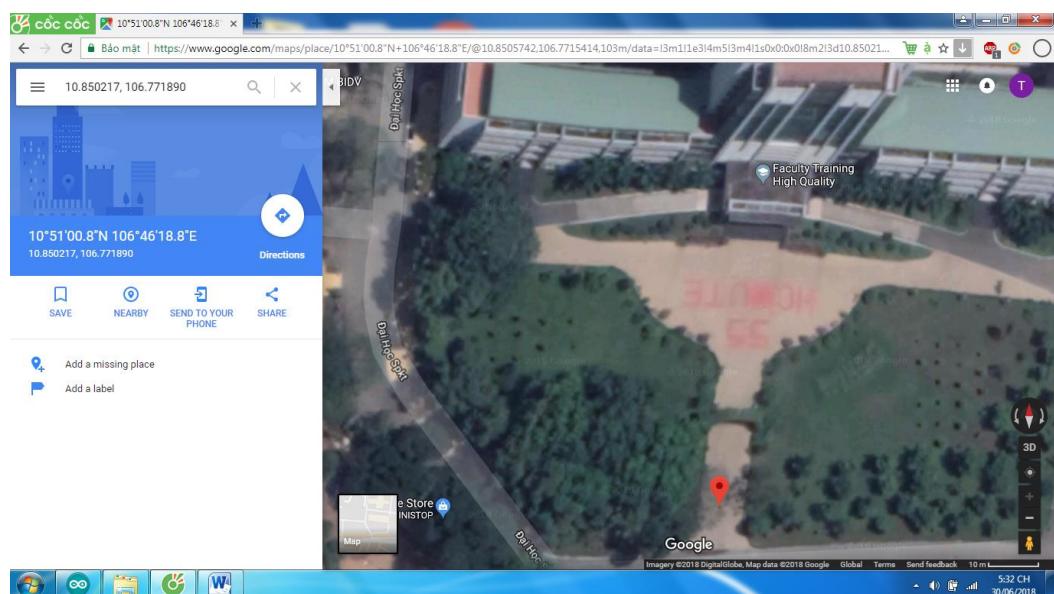
---

### **Điểm thứ 5**



*Hình 5.6: Lấy tọa độ điểm thứ năm trên Google map*

### **Điểm thứ 6 (điểm cuối)**



*Hình 5.7: Lấy tọa độ điểm cuối cùng trên Google map*

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---

Đây là con đường được lấy thử nghiệm trong khuôn viên trường Đại học Sư Phạm Kỹ Thuật Tphcm, đoạn đường được lấy từ tước phòng bảo vệ cạnh căn tin đầu tòa nhà trung tâm.

Vị trí được lấy làm thử nghiêm thoáng mát, ít bị che khuất, tiện lối cho mô hình “Thùng rác thông minh” di chuyển

- **Giải thích hình ảnh thực nghiệm**

Khi rác đầy thùng rác sẽ chạy từ điểm xuất phát (điểm thứ 0), đến điểm thứ 1, thứ 2... và đến điểm cuối cùng (điểm thứ 6).

Khi đến điểm cuối, thùng rác dừng lại. Ta nhấn nút nhấn trên thùng rác thì nó sẽ quay lại điểm xuất phát. Lúc này điểm thứ 6 sẽ trở thành điểm thứ 0, điểm thứ 5 trở thành điểm thứ 1, tương tự điểm thứ 0 sẽ trở thành điểm thứ 6 (điểm cuối - thùng rác dừng lại).

### **5.3.2. Kết quả di chuyển thực tế**

Úng với những điểm tọa độ trên, thùng rác đã được thử nghiệm nhiều lần ở các cung đường khác nhau trong khuôn viên trường Đh SPKT.

Một trong số đó là tuyến đường được định vị trên Google maps ở trên, tương ứng với những vị trí định vị ở những hình trên, ta có những hình ảnh tương ứng thực tế như sau:



Hình 5.8: Hình ảnh thực tế thùng rác thông minh chạy ngoài trời

### 5.4 NHẬN XÉT – ĐÁNH GIÁ:

#### 5.4.1 Nhận xét

Sau thời gian 12 tuần nghiên cứu và thực hiện đề tài, mạch cơ bản đáp ứng được yêu cầu thiết kế ban đầu. Dưới đây là một số nhận xét:

##### a. Ưu điểm

-Thùng rác có thể sử dụng trong môi trường thông thoáng như khuôn viên trường học đại học- cao đẳng, các khu kí túc xá, các nơi công cộng

-Thùng rác tự di chuyển ra nơi chứa thùng rác tổng tạo thiện cảm và sự chú ý, tò mò thích thú của mọi người, đặc biệt là những người đam mê về công nghệ, những em sinh viên năm nhất, năm hai.

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---

- Pin Ác quy dung lượng lớn, theo tính toán thì dùng 1 tuần mới cần sạc lại một lần.

### **b. Hạn chế**

-Do thùng rác tự di chuyển nên có thể gây cảm giác ngáng đường ngáng lối

-Di chuyển còn chưa có độ chính xác cao, có thể gây ra tình trạng không kiểm soát được

-Chưa có sự hỗ trợ của nhiều module để xe thùng rác có thể hoàn thiện, tự hành độc lập được tốt hơn

-Chưa có thiết bị giám sát, dễ gây tình trạng bị mất hoặc thùng rác di chuyển lạc đường sai lối.

- Mỗi thùng rác chỉ được cài đặt một tọa độ nhất định, khi cần thay vị trí khác, phải thay thế lại tọa độ bằng việc nạp lại code mới, gây khó khăn.

- Khó có thể dùng trong điều kiện thời tiết mưa bão, hoặc những nơi mặt đường quá gồ ghề.

### **5.4.2 Đánh giá**

Sau quá trình vận hành thử chạy thùng rác loanh quanh trường, nhóm thực hiện đánh giá thùng rác thông minh đã hoàn thành những yêu cầu ban đầu đã đề ra. Tuy rằng khi chạy trong môi trường ngoài trời, mạch chạy cũng chưa thực sự tối ưu, tuy nhiên, với khả năng tự động hóa, cùng tính thẩm mỹ cảnh quan nhà trường với giá thành không quá mắc thì đây có thể được coi là một sản phẩm có thể chấp nhận được.

Với kết quả như trên, nhóm đã thu thập được một vài số liệu như sau:

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---

**Bảng 5.1 Dung lượng điện năng tiêu thụ thực nghiệm**

Lần	Điện áp tiêu hao (V)	Thời gian (phút)
1	0,16	12
2	0,15	12
3	0,16	12
4	0,16	12
5	0,16	12

Thời gian chạy thực tế nhiều hơn 2 phút so với thời gian tính toán lý thuyết.

Dung lượng bình acquy khi đầy là 13,6 V, khi điện áp giảm xuống 10,8 V ta cần sạc lại. Vậy số lần chạy được là  $(13,6 - 10,8)/0,16 = 17$  lần.

Nếu thùng rác di chuyển 2 lần 1 ngày thì sau 8 ngày ta cần sạc lại acquy. Khá chênh lệch so với tính toán lý thuyết (26 ngày).

# *Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN*

## 6.1. KẾT LUẬN:

Sau khoảng thời 12 tuần nghiên cứu và tìm hiểu, nhóm thực hiện đã hoàn thành quyển đồ án và thi công mô hình theo những yêu cầu đã đặt ra ban đầu. Trong quá trình thực hiện, nhóm đã thu được những kết quả nhất định.

- Thiết kế thành công mô hình thùng rác khá cân đối, gọn gàng tuy nhiên chưa thực sự thẩm mỹ.
- Thùng rác có khả năng nhận biết được rác đầy trong thùng.
- Khi phát hiện rác trong thùng đầy hoặc đến thời gian được cài đặt sẵn, thùng rác có thể tự di chuyển đến vị trí thùng rác tổng đã được định vị sẵn.
- Tới vị trí định vị sẵn thì dừng lại, chờ nhấn nút thì có thể quay về vị trí cũ
- Có khả năng né những vật cản xung quanh đường đi của thùng rác.
- Trong lúc di chuyển, thùng rác có phát ra tiếng kêu để mọi người né tránh.
- Có kiến thức vững chắc về hệ thống định vị toàn cầu GPS, các chuẩn truyền UART, I2C, cách thực hoạt động các cảm biến la bàn, cảm biến siêu âm, module thời gian thực, cách lập trình Arduino Mega.

Tuy nhiên, đề tài còn một số điểm hạn chế như:

- Mô hình hiện tại sử dụng không phù hợp với môi trường sử dụng ngoài trời mà phù hợp sử dụng trong văn phòng, trong nhà hơn
- Thỉnh thoảng gặp trục trặc về vấn đề cảm biến siêu âm không do được khoảng cách mặc dù cảm biến siêu âm hoạt động bình thường
- Sai số vị trí vẫn còn tương đối lớn (bảng 5.3) do module GPS cập nhật vị trí vẫn còn chưa chính xác.
- Thời lượng pin sử dụng chưa được lâu (bảng 5.6)
- Khả năng bảo quản thùng rác khi ở điều kiện xấu như mưa bão sẽ là rào cản lớn khi thùng rác được sử dụng ở ngoài trời

### 6.2. HƯỚNG PHÁT TRIỂN:

Đè tài là “Thùng rác thông minh” sử dụng tính năng chính là định vị qua GPS, khả năng hoạt động không khác gì một chiếc xe tự hành, có khả năng tự động hóa cao và tự vận hành một cách độc lập, được sử dụng trong môi trường và điều kiện bên ngoài trời. Chính vì vậy để phát triển và nâng tầm chiếc thùng rác là rất cần thiết để thùng rác có thể hoạt động ổn định và trở thành sản phẩm thương mại. Một số hướng phát triển của đèn tài có thể là:

- Sử dụng nguồn điện bằng pin năng lượng mặt trời
- Ứng dụng xử lý ảnh để nhận biết vật cản cũng như các vật xung quanh một cách chính xác
- Ứng dụng Máy học (Machine Learning) để dạy cho thùng rác những con đường quen thuộc.
- Kết hợp thêm các cảm biến để thùng rác thực sự thông minh hơn
- Ứng dụng IOT để giao thức truyền nhận dữ liệu để giám sát, điều khiển hoạt động của thùng rác qua Webserver hoặc qua nền tảng Android, quản lý nhiều thùng rác cùng một lúc.

#### 6.2.1 Sử dụng nguồn điện bằng pin năng lượng mặt trời

Để sử dụng một hệ thống tự động có thể tự vận hành ngoài trời trong một thời gian dài thì yếu tố về năng lượng là một vấn đề quan trọng mà chúng ta cần quan tâm. Do đó, để tận dụng những điều kiện thuận lợi về tự nhiên thì năng lượng bằng ánh sáng mặt trời là một giải pháp rất tốt để chúng ta có thể thu thập được năng lượng. Chính vì vậy, chúng ta hoàn toàn yên tâm sử dụng bình ác quy như nguồn điện một cách thoải mái mà không sợ lo hết dung lượng pin trong bình vì những tấm pin năng lượng mặt trời này sẽ giúp sạc bình tự động vào mỗi khi trời sáng. Như vậy sẽ tiết kiệm được rất nhiều hiệu năng sử dụng.

Hiện nay, những tấm pin năng lượng mặt trời được bán trên thị trường rất nhiều, chúng ta có thể dễ dàng mua và sử dụng chúng để ứng dụng vào sản phẩm để cải tiến chất lượng, hiệu năng sử dụng của nguồn điện.

### 6.2.2 Úng dụng xử lý ảnh để nhận biết vật thể

Có thể nói, cùng với sự phát triển và tiến bộ không ngừng của khoa học kỹ thuật thì xử lý ảnh là một trong những vấn đề được quan tâm và phát triển hiện nay. Từ những nghiên cứu ban đầu về ảnh trắng đen, ảnh xám, ảnh màu, xử lý ảnh đã được nghiên cứu chuyên sâu và ứng dụng rất nhiều trong cuộc sống. Xử lý ảnh đã đạt được nhiều thành tựu và tiến bộ vượt bậc trong các lĩnh vực quan trọng như: quân sự, y học, an ninh quốc phòng, công nghiệp, nông nghiệp, giao thông vận tải... trong đó, công nghệ nhận dạng là một trong các công nghệ được quan tâm và phát triển. Nhận dạng có thể bao gồm nhận dạng âm thanh, hình ảnh, nhận dạng tĩnh hoặc nhận dạng trong thời gian thực. Các đối tượng nhận dạng có thể là tiếng nói, chữ viết, khuôn mặt, biển số xe, biển báo giao thông mã vạch, vân tay, cây cối... Chính vì thế, xử lý ảnh sẽ nâng cao được khả năng tương tác giữa người và máy hơn từ những giũ liệu mà nó đọc được

Để ứng dụng được xử lý ảnh trong đề tài “Thùng rác thông minh” này có thể sẽ rất cần thiết. Việc sử dụng các hàm có sẵn trong **OpenCV**, chúng ta hoàn toàn có thể giúp cho thùng rác có thể nhận biết được những vật cản khó có thể nhận biết bằng cảm biến siêu âm như những lùm cây, bám sát vỉa hè, nhận biết những vũng nước hay những con đường gồ ghề, khó di chuyển để tìm những đường khác dễ đi hơn. Tuy vậy, đây vẫn là một lĩnh vực khó, thực sự để đáp ứng được thì cần phải có rất nhiều thời gian nghiên cứu và tìm tòi.

### 6.2.3 Úng dụng Máy học (Machine learning) để cho thùng rác học

Machine Learning là hướng nghiên cứu trong Trí tuệ nhân tạo (AI), tập trung vào việc tạo ra các cỗ máy có khả năng học hỏi mà không cần phải được lập trình một cách cụ thể. Học hỏi là một kỹ năng quan trọng bậc nhất đối với sự hình thành trí tuệ con người; do đó nếu chúng ta muốn xây dựng các hệ thống AI có trí tuệ như con người thì chúng ta trước hết phải tạo ra các cỗ máy có khả năng tự học hỏi dựa trên các kinh nghiệm mà chúng thu thập được trong suốt quá trình hoạt động. Trong

lĩnh vực này, những nhà nghiên cứu đã đưa ra rất nhiều thuật toán và những cách phân loại cụ thể để áp dụng những thuật toán ấy để giải quyết vấn đề như học có giám sát, học không giám sát, học bán giám sát và học củng cố.

Giống với lĩnh vực xử lý ảnh thì trong máy học cũng sử dụng ngôn ngữ lập trình chính là Python và vì điều khiển để đáp ứng nhu cầu máy học này thì sử dụng kit *Raspberry Pi* sẽ ứng dụng tốt hơn.

Với sức mạnh của Máy học cùng sự kết hợp của xử lý ảnh, sẽ chẳng có gì khó hiểu khi nếu được áp dụng vào đề tài “Thùng rác thông minh” thì chắc chắn sẽ rất hợp lý và thành công. Những chiếc thùng rác này sau khi được học hỏi và di chuyển nhiều lần những con đường quen thuộc vào mỗi ngày sẽ dần dần di chuyển chính xác hơn, xảy ra ít lỗi và chậm hơn, định vị từ tín hiệu GPS sẽ chuẩn xác hơn so với những lần đầu thu thập tín hiệu GPS. Đó thực sự là giải pháp tốt cho những chiếc thùng rác luôn luôn di chuyển theo một con đường nhất định.

Đây cũng là hai ứng dụng chủ đạo để tạo nên thương hiệu nổi tiếng xe tự lái (Tesla) hiện nay

### 6.2.4 Kết hợp thêm những cảm biến khác

Như đã đề cập ở những phần trên, khi được cung cấp càng nhiều dữ liệu thì thùng rác càng di chuyển và vận hành một cách chính xác nhất. Một trong số cách để cung cấp thông tin dữ liệu cho bộ vi xử lý hiểu được là cải thiện những cảm biến cũ và thêm những cảm biến mới.

Ngoài những dữ liệu đã quá quen thuộc như GPS, sóng siêu âm phản xạ để đo khoảng cách hay cảm biến la bàn để tìm và dò hướng đúng hơn thì module đo quán tính IMU sẽ được áp dụng.

- Mô-đun IMU dùng để xác định vị trí, gia tốc tịnh tiến và góc quay của một vật chuyển động trong không gian.

- Là bộ não trung tâm điều khiển cân bằng và quỹ đạo của các phương tiện không người lái: UAV, robot tự hành, xe tự hành...

## **CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

---

Chính vì thế, đây thực sự là một mảnh ghép quan trọng trong việc giúp xe định hướng tốt hơn trong không gian.

Việc chế tạo và nghiên cứu “Thùng rác thông minh” này sẽ tương ứng với hướng nghiên cứu dành cho **Xe tự hành** đã được phát triển gần đây

### **6.2.5 Ứng dụng IOT- Thùng rác thông minh-thành phố thông minh**

Nhiệm vụ xây dựng thành phố thông minh là một sáng kiến đã được hình thành từ rất lâu rồi. Tuy vậy, nhiệm vụ này thực tế mới chỉ hình thành và được áp dụng ở một số nơi trên thế giới đối với một số thành phố lớn của những quốc gia phát triển. Và một trong điều kiện tiên quyết để hình thành nên một thành phố thông minh đó chính là liên quan tới việc quản lý rác thải.

Với sự phát triển vượt bậc của công nghệ kỹ thuật hiện nay và mạnh mẽ nhất phải kể đến IOT (Internet of things) thì chúng ta dễ dàng kết nối tất cả mọi thiết bị với nhau, từ smartphone đến máy tính bàn, tủ lạnh, máy lạnh... tới các thiết bị điện tử nhỏ nhất trong gia đình như đèn chiếu sáng, công tắc đều đã được áp dụng công nghệ này một cách rất thành công.

Bản chất của IOT là chúng ta cần một giao thức truyền thông không dây kết nối, thu thập các dữ liệu từ các cảm biến. Hiện tại, có một số công nghệ nổi trội được sử dụng như: Bluetooth, Zigbee, NFC, Wifi, Lora,...

Công nghệ truyền không dây LoRa cũng tương tự như truyền Zigbee và cho phép giao tiếp với nhiều thiết bị cùng lúc với khoảng cách xa (khoảng 3000m-5000m) với tốc độ nhanh hơn những thiết bị thu phát RF khác.

Từ sự nhìn nhận những giao thức truyền thông này, chúng ta có thể dễ dàng nhìn thấy LORA chính là sự lựa chọn phù hợp nhất khi có khả năng truyền trong môi trường bên ngoài với khoảng cách rất lớn và có nhiều node như vậy có thể giúp cho các thùng rác có thể giao tiếp được với nhau.

Để quan sát, giám sát hành trình thùng rác di chuyển, lượng rác trong mỗi thùng hay việc luân chuyển xe ô tô đi để lấy rác trong những thùng rác tổng thì tất nhiên là phải có một giao diện người dùng. Việc phát triển phần mềm giao diện thì

## **CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

---

đó lại là tùy theo nhu cầu và sức sáng tạo của mỗi người. Chúng ta có thể sử dụng những phần mềm đơn giản như app Inventer hay cao cấp hơn là lập trình Android Studio để lập trình phần mềm trên smart phone, giúp cho việc giám sát hoặc định vị lại vị trí của những chiếc thùng rác đó được tốt hơn. Hoặc là có thể viết một Webside chuyên dụng để giám sát những hoạt động này. Nếu thành công, đây chắc chắn sẽ là một dự án tiềm năng trên thị trường.

Nói tóm lại, chúng ta sẽ sử dụng một giao thức truyền nhận dữ liệu là LORA để đọc các thông tin từ những chiếc thùng rác thông minh của chúng ta và đồng thời có thể giám sát, điều khiển, định vị GPS thông qua nền tảng lập trình Android hoặc App Inventor cho những chiếc thùng rác ấy từ khoảng cách xa. Những dữ liệu đã cập nhật và giám sát ấy sẽ được lưu lại vô cơ sở dữ liệu và sẽ được trích xuất ra thống kê khi cần thiết.

# TÀI LIỆU THAM KHẢO

### 1. Sách tham khảo

- [1] TS. Trần Thu Hà, “Giáo trình Điện tử cơ bản”, Đại học SPKT Tp.HCM, 2013.
- [2] ThS Nguyễn Đình Phú, “Giáo trình Vi xử lý”, NXB Đại Học Quốc Gia, 2013.
- [3] Nguyễn Ngọc Bảo- Bùi Sỹ Nguyên, “GPS và ứng dụng định vị giám sát”, Đồ Án Tốt Nghiệp ĐH, Trường ĐH Sư Phạm Kỹ Thuật, 2010.
- [4] Ngô Phương Nam- Lê Văn Tài, “Xe điều khiển bằng giọng nói và hỗ trợ trẻ em học qua trò chơi điện thoại Android”, Đồ Án Tốt Nghiệp ĐH, Trường ĐH Sư Phạm Kỹ Thuật, 2017.
- [5] Shamlee Rashinkar- Sneha Ghatole- Swati Kadapatti-Varsha Yadave- Chaitanya Jambotkar, “IoT Based Smart Trash Bins – A Step Toward Smart City”, International Research Journal of Engineering and Technology (IRJET), Dec- 2017.

### 2. Trang web tham khảo

- [6] Cộng đồng Arduino Việt Nam, “Giới thiệu Arduino Mega2560”, 28/11/2015, <http://arduino.vn/bai-viet/542-gioi-thieu-arduino-mega2560/>
- [7] Project tham khảo: <http://www.instructables.com/id/Self-Driving-Car-Using-Arduinoautonomous-Guided-Ve/>
- [8] Cộng đồng Arduino Việt Nam, “Robot tránh vật cản”, 17/10/2016, <http://arduino.vn/result/1306-robot-tranh-vat-can/>
- [9] Diễn Đàn Vi Điều Khiển, “Bài 8: Giao Tiếp UART”, 17/09/2012, <http://mcu.banlinhkien.vn/threads/bai-8-giao-tiep-uart.28/>
- [10] Cộng đồng Arduino Việt Nam, “Giới thiệu cách sử dụng module GPS Neo 6 và Neo 7 của hãng Ublox”, 17/02/2017, <http://arduino.vn/tutorial/1482-gioi-thieu-cach-su-dung-module-gps-neo-6-va-neo-7-cua-hang-ublox/>
- [11] Movable Type Ltd, “Calculate distance, bearing and more between Latitude/Longitude points”, <https://www.movable-type.co.uk/scripts/latlong.html/>, 2017.
- [12] Bioenable, “Smart bins for smart city”, <https://www.bioenabletech.com/smart-bins-for-smart-city.html>, 27/6/2017.

## TÀI LIỆU THAM KHẢO

---

- [13] Thuật toán xác định vị trí chính xác,  
<https://en.wikipedia.org/wiki/Trilateration>, wikipedia, 2017.
- [14] Trang web chuyên về cách thức hoạt động vệ tinh,  
<https://spotlight.unavco.org/how-gps-works/gps-basics/satellites-controllers-users.html>
- [15] KhoaHoc.tv, “Google Maps hoạt động như thế nào”, 2017,  
<http://khoaahoc.tv/google-maps-hoat-dong-nhu-the-nao-88542>,

## **PHỤ LỤC: CODE ARDUINO**

```
//khai báo thư viện
#include <Wire.h>
#include <TinyGPS.h>
#include <NewPing.h>
//định nghĩa IO, địa chỉ cho các linh kiện
#define addr_hmc 0x1E
#define addr_DS1307 0x68
#define PI_314 3.1415926535
#define trig2 A0
#define echo2 A1
#define trig_t A2
#define echo_t A3
#define trig_g A4
#define echo_g A5
#define trig_p A6
#define echo_p A7
#define MAX_DISTANCE 200
#define buzzer A10
#define n 7
#define button 12

int m01 = 2, m02 = 3, m03 = 4, m04 = 5;
int second, minute, hour;
//điểm đặt trước chia thành 2 mảng, mảng chứa kinh độ và mảng chứa vĩ độ
const float lat_i[n] =
{10.851081,10.850745,10.850714,10.850675,10.850589,10.850415,10.850217};
const float lon_i[n] =
{106.771293,106.771274,106.771554,106.771767,106.771941,106.771921,106.771890};
;
long cm_t, cm_p, cm_g;
long cm2;
float heading;
byte dem = 1, chay = 0, TT = 0, BDN;
byte i;
float lat_i_tam[n], lon_i_tam[n];
int16_t x, y, z;
float head, distance = 0.0;

NewPing sonar2(trig2, echo2, 200);
NewPing sonar_t(trig_t, echo_t, 200);
NewPing sonar_p(trig_p, echo_p, 200);
NewPing sonar_g(trig_g, echo_g, 200);
TinyGPS gps;

void gpshead(float x2lat, float x2lon);
void stopp();
void rightturn();
void gostraight();
void leftturn();
void turn();

void setup() {
    Wire.begin();
```

## PHỤ LỤC

---

```
Serial1.begin(9600);
Serial.begin(9600);
setTime(2, 30, 45);
Wire.beginTransmission(addr_hmc); //start talking
Wire.write(0x02); // Set the Register
Wire.write(0x00); // Tell the HMC5883 to Continuously Measure
Wire.endTransmission();

cli();
TCCR1A = 0;
TCCR1B = 0;
TIMSK1 = 0;
TCCR1B |= (1 << CS11) | (1 << CS10);      // prescale = 64
TCNT1 = 40536;
TIMSK1 = (0 << TOIE1);
sei();
pinMode(button, INPUT_PULLUP);
pinMode(mo1, OUTPUT);
pinMode(mo2, OUTPUT);
pinMode(mo3, OUTPUT);
pinMode(mo4, OUTPUT);
pinMode(buzzer, OUTPUT);
stopp();
}

void loop() {
    if (chay == 0) //thùng rác ở điểm xuất phát, đợi rác đầy hoặc đến
    {               //thời gian đặt trước sẽ cho chép di chuyển
        TIMSK1 = (0 << TOIE1);
        readDS1307();
        cm2 = sonar2.ping_cm();
        delay(50);
        if (cm2 < 6 && cm2 > 0)
        {
            delay(2000);
            cm2 = sonar2.ping_cm(); delay(50);
            if (cm2 < 6 && cm2 > 0)
            {
                TIMSK1 = (1 << TOIE1);
                delay(3000);
                chay = 1;
                for (int j = 0; j < n; j++)
                {
                    lat_i_tam[j] = lat_i[j];
                    lon_i_tam[j] = lon_i[j];
                }
            }
        }
    else if (hour == 2 && minute == 33)
    {
        TIMSK1 = (1 << TOIE1);
        delay(5000);
        chay = 1;
        for (int j = 0; j < n; j++)
        {
            lat_i_tam[j] = lat_i[j];
            lon_i_tam[j] = lon_i[j];
        }
    }
}
```

## PHỤ LỤC

---

```
        }
    }
else if (chay == 1) //chạy đến điểm đặt trước
{
    cm_t = sonar_t.ping_cm();
    cm_p = sonar_p.ping_cm();
    cm_g = sonar_g.ping_cm();
    if (cm_g < 50 && cm_g > 0) //phát hiện vật cản chính giữa, cho xe chuyển
    hướng
    {
        doc_gt_hmc5883l();
        if(i%2==0)
        {
            head = heading - 90;
            if (head < 0)
                head = head + 360;
            i++;
        }
        else
        {
            head = heading + 90;
            if (head > 360)
                head = head - 360;
            i++;
        }
        while ((heading > head + 8) || (heading < head - 8))
        {
            turn();
            delay(5);
            doc_gt_hmc5883l();
        }
        gostraight();
    }
    else if (cm_t < 100 && cm_t > 0) //phát hiện vật cản bên trái, cho xe rẽ
    phải
    {
        rightturn();
        delay(200);
        gostraight();
    }
    else if (cm_p < 100 && cm_p > 0) //phát hiện vật cản bên phải, cho xe rẽ
    trái
    {
        leftturn();
        delay(200);
        gostraight();
    }
    else //nếu không có vật cản
    {
        do { //đọc vị trí hiện tại tính
            khoảng cách và góc giữa
            gpshead(lat_i_tam[dem], lon_i_tam[dem]); //điểm hiện tại và điểm thứ
            i đang hướng đến
        } while (distance == 0.0);
```

## PHỤ LỤC

---

```
doc_gt_hmc5883l(); //đọc góc hiện tại
while ((heading > head + 8) || (heading < head - 8)) //xoay xe đến khi
nào chêch lệch
{
    //giữa góc hiện
tại và góc hướng đến nhỏ hơn 8 độ
    turn();
    delay(5);
    doc_gt_hmc5883l();
}

gostraight(); //sau đó cho xe đi thẳng
if ((distance < 3) && (dem != (n - 1))) //nếu khoảng cách giữa xe và
điểm thứ i nhỏ hơn 3m
{
    dem++; //tăng i, cho thùng rác đến
điểm tiếp theo
    do {
        gpshead(lat_i_tam[dem], lon_i_tam[dem]);
    } while (distance < 3);
}
if ((distance < 3) && (dem == (n - 1)) && (TT == 0)) //đến điểm cuối
cùng, dừng lại
{
    stopp(); chay = 2; TT = 1; dem = 1;
}
if ((distance < 3) && (dem == (n - 1)) && (TT == 1)) {
    stopp();
    chay = 0;
    TT = 0;
    dem = 1;
}
}
else if (chay == 2) // đợi nhấn nút để di chuyển về
{
    TIMSK1 = (0 << TOIE1);
    BDN = 0;
    digitalWrite(buzzer, 0);
    stopp();
    while (digitalRead(button) == HIGH) {
        // Do nothing
    }
    TIMSK1 = (1 << TOIE1);
    chay = 1;
    dem = 1;
    for (int k = 0; k < n; k++) //đảo các điểm muốn đến
    {
        lat_i_tam[k] = lat_i[n - k - 1];
        lon_i_tam[k] = lon_i[n - k - 1];
    }
}
void turn()
{ float tur = heading - head;
  if (tur < 0.0)
  { if (tur > -180.0)
    rightturn();
```

## PHỤ LỤC

---

```
    else
        leftturn();
    }
    else
    { if (tur < 180.0)
        leftturn();
        else rightturn();
    }
}
void leftturn()
{ digitalWrite(mo1, 0);
  analogWrite(mo2, 0);
  digitalWrite(mo3, HIGH);
  analogWrite(mo4, 0);
  delay(10);
}
void stopp()
{
  digitalWrite(mo1, LOW);
  digitalWrite(mo2, LOW);
  digitalWrite(mo3, LOW);
  digitalWrite(mo4, LOW);
}

void rightturn()
{ digitalWrite(mo1, HIGH);
  analogWrite(mo2, 0);
  digitalWrite(mo3, 0);
  analogWrite(mo4, 0);
}
void gostraight()
{ digitalWrite(mo1, HIGH);
  analogWrite(mo2, 51);
  digitalWrite(mo3, HIGH);
  analogWrite(mo4, 51);
}
void gpshead(float x2lat, float x2lon)
{
  bool newData = false;
  for (unsigned long start = millis(); millis() - start < 1000;)
  {
    while (Serial1.available())
    {
      char c = Serial1.read();
      // Serial.write(c); // uncomment this line if you want to see the GPS
data flowing
      if (gps.encode(c)) // Did a new valid sentence come in?
        newData = true;
    }
  }
  if (newData)
  {
    x2lat = radians(x2lat);
    x2lon = radians(x2lon);
    float flat1, flon1;
    unsigned long age;
    gps.f_get_position(&flat1, &flon1, &age);
  }
}
```

## PHỤ LỤC

---

```
    flon1 = radians(flon1); //also must be done in radians
    flat1 = radians(flat1); //also must be done in radians
    head = atan2(sin(x2lon - flon1) * cos(x2lat), cos(flat1) * sin(x2lat) -
sin(flat1) * cos(x2lat) * cos(x2lon - flon1));
    head = head * 180 / 3.1415926535; // convert from radians to degrees
    float dist_calc = 0;
    float diflat = 0;
    float diflon = 0;
    diflat = x2lat - flat1; //notice it must be done in radians
    diflon = (x2lon) - (flon1); //subtract and convert longitudes to radians
    distance = (sin(diflat / 2.0) * sin(diflat / 2.0));
    dist_calc = cos(flat1);
    dist_calc *= cos(x2lat);
    dist_calc *= sin(diflon / 2.0);
    dist_calc *= sin(diflon / 2.0);
    distance += dist_calc;
    distance = (2 * atan2(sqrt(distance), sqrt(1.0 - distance)));
    distance *= 6371000.0; //Converting to meters
    if (head < 0) {
        head += 360; //if the heading is negative then add 360 to make it
positive
    }
}
void doc_gt_hmc5883l()
{
    Wire.beginTransmission(addr_hmc);
    Wire.write(0x03); //start with register 3.
    Wire.endTransmission();
    Wire.requestFrom(addr_hmc, 6);
    if (6 <= Wire.available()) {
        x = Wire.read() << 8; //MSB x
        x |= Wire.read(); //LSB x
        z = Wire.read() << 8; //MSB z
        z |= Wire.read(); //LSB z
        y = Wire.read() << 8; //MSB y
        y |= Wire.read(); //LSB y
    }
    heading = atan2(y, x);
    if (heading < 0)
        heading += 2 * PI_314;
    heading = heading * 180 / PI_314;
}
void readDS1307()
{
    Wire.beginTransmission(addr_DS1307);
    Wire.write((byte)0x00);
    Wire.endTransmission();
    Wire.requestFrom(addr_DS1307, 3);

    second = bcd2dec(Wire.read() & 0x7f);
    minute = bcd2dec(Wire.read());
    hour = bcd2dec(Wire.read() & 0x3f); // ch? d? 24h.
}
/* Chuyển từ format BCD (Binary-Coded Decimal) sang Decimal */
int bcd2dec(byte num)
{
```

## PHỤ LỤC

---

```
    return ((num / 16 * 10) + (num % 16));
}
/* Chuyển từ Decimal sang BCD */
int dec2bcd(byte num)
{
    return ((num / 10 * 16) + (num % 10));
}
/* cài đặt thời gian cho DS1307 */
void setTime(byte hr, byte minute, byte sec)
{
    Wire.beginTransmission(addr_DS1307);
    Wire.write(byte(0x00)); // đặt lại pointer
    Wire.write(dec2bcd(sec));
    Wire.write(dec2bcd(minute));
    Wire.write(dec2bcd(hr));
    Wire.endTransmission();
}
ISR (TIMER1_OVF_vect)
{
    TCNT1 = 40536;
    BDN++;
    if (BDN < 1)
    {
        digitalWrite(buzzer, 1);
    }
    else {
        digitalWrite(buzzer, 0);
    }
    if (BDN == 10) BDN = 0;
}
}
```