

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM



ĐỒ ÁN CHUYÊN NGÀNH
THIẾT KẾ XE TỰ HÀNH TRÁNH VẬT CẢN

LỚP 06DHLTDT 2 KHÓA 2016-2018

GVHD: TRẦN HOÀN

SINH VIÊN THỰC HIỆN: NGUYỄN TRƯỜNG GIANG

TP.HỒ CHÍ MINH, 11/2017

TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN : ĐỒ ÁN CHUYÊN NGÀNH 2



NGUYỄN TRƯỜNG GIANG

ĐỒ ÁN CHUYÊN NGÀNH

XE TỰ HÀNH TRÁNH VẬT CẢN

GVHD: TRẦN HOÀN

TP. HỒ CHÍ MINH, tháng 11 năm 2017

TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN : ĐỒ ÁN CHUYÊN NGÀNH 2



NGUYỄN TRƯỜNG GIANG

ĐỒ ÁN CHUYÊN NGÀNH

XE TỰ HÀNH TRÁNH VẬT CẢN

GVHD: TRẦN HOÀN

TP. HỒ CHÍ MINH, tháng 11 năm 2017

TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP.
HCM

KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN: ĐỒ ÁN CHUYÊN NGÀNH 2

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT
NAM

Độc lập - Tự do - Hạnh phúc

TP. HCM, ngày 20 tháng 11 năm 2017

NHẬN XÉT ĐỒ ÁN CHUYÊN NGÀNH CỦA GIẢNG VIÊN HƯỚNG DẪN

Tên đồ án:

ĐỒ ÁN CHUYÊN NGÀNH 2

Sinh viên thực hiện:

NGUYỄN TRƯỜNG MSSV: 2202162014
GIANG

Giảng viên hướng dẫn:

TRẦN HOÀN

Đánh giá Luận văn

1. Về cuốn báo cáo:

Số trang _____ Số chương _____

Số bảng số liệu _____ Số hình vẽ _____

Số tài liệu tham khảo _____ Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung đồ án:

3. Về tính ứng dụng:

4. Về thái độ làm việc của sinh viên:

Đánh giá chung:

Điểm từng sinh viên:

NGUYỄN TRƯỜNG GIANG:...../10

Người nhận xét

(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Để đồ án này đạt kết quả tốt đẹp, tôi đã nhận được sự hỗ trợ, giúp đỡ của nhiều cơ quan, tổ chức, cá nhân. Với tình cảm sâu sắc, chân thành, cho phép tôi được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân và cơ quan đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu đề tài.

Trước hết tôi xin gửi tới các thầy cô khoa Điện- Điện Tử Trường Đại Học Công Nghiệp Thực Phẩm TP Hồ Chí Minh lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay tôi đã có thể hoàn thành luận đề án: “xe tự hành tránh vật cản”.

Đặc biệt tôi xin gửi lời cảm ơn chân thành nhất tới thầy giáo Trần Hoàn đã quan tâm giúp đỡ, hướng dẫn tôi hoàn thành tốt luận văn này trong thời gian qua.

Tôi xin bày tỏ lòng biết ơn đến lãnh đạo Đại Học Công Nghiệp Thực Phẩm TP Hồ Chí Minh, các Khoa Phòng ban chức năng đã trực tiếp và gián tiếp giúp đỡ tôi trong suốt quá trình học tập và nghiên cứu đề tài.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một học viên, đồ án này không thể tránh được những thiếu sót. Tôi rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để tôi có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 20 tháng 11 năm 2017

Tác giả

Nguyễn Trường Giang

TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP.
HCM

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN: ĐỒ ÁN CHUYÊN NGÀNH 2

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

TP. HCM, ngày 20 tháng 11 năm 2017

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỒ ÁN: XE TỰ HÀNH TRÁNH VẬT CẢN	
Giảng viên hướng dẫn: TRẦN HOÀN	
Thời gian thực hiện: Từ ngày 15/09/2017 đến ngày 20/11/2017	
Sinh viên thực hiện: NGUYỄN TRƯỜNG GIANG	
Nội dung đề tài: <ul style="list-style-type: none">- Tổng quan về mô hình xe tự hành tránh vật cản.- Cơ sở lý thuyết.- Cơ sở thực hiện- Xây dựng mô hình trên cơ sở lý thuyết	
Kế hoạch thực hiện: <ul style="list-style-type: none">- Từ ngày 15/09/2017 đến ngày 10/10/2017: xây dựng mô hình.- Từ ngày 10/10/2017 đến ngày 20/10/2017: viết code.- Từ ngày 20/10/2017 đến ngày 1/11/2017: thử nghiệm mô hình.- Từ ngày 2/11/2017 đến ngày 25/11/2017: viết báo cáo.	
Xác nhận của giảng viên hướng dẫn	TP. HCM, ngày tháng năm Sinh viên

MỤC LỤC

DANH MỤC HÌNH ẢNH	3
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	5
1. Xu hướng phát triển của robot hiện đại:	5
2. Mục tiêu đề tài.....	8
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	10
1. Cảm biến siêu âm:	10
1.1. Hoạt động:.....	10
1.2. Cấu tạo:	11
2. Mạch Arduino UNO R3:.....	11
2.1. Một vài thông số của Arduino UNO R3	12
2.2. Vi điều khiển.....	13
2.3. Năng lượng	13
2.4. Các chân năng lượng	13
2.5. Bộ nhớ.....	15
2.6. Lập trình cho Arduino.....	16
2.7. Module L298N:.....	16
2.8. Sơ đồ kết nối chân:	18
CHƯƠNG 3: CƠ SỞ THỰC HIỆN	20
1. Nguyên lý điều khiển:	20
2. Điều khiển cảm biến siêu âm.	21
2.1 Flowchart cảm biến	21
2.2. Code điều khiển.	22
3. Điều khiển né vật cản.....	22

3.1.	Flowchart tránh vật cản.....	22
3.2.	Code điều khiển.	24
CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM.....		28
1.	Ảnh chụp các thành phần của mô hình.	28
2.	Mô hình hoàn thiện.	29
CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG ĐỀ TÀI.....		32
Kết quả đạt được:		32
1.	Hạn chế:	32
2.	Hướng phát triển của đề tài:	32
PHỤ LỤC.....		33
1.	Code chương trình:.....	33
2.	Giới thiệu phần mềm sử dụng:	40
2.1.	Hướng dẫn cài đặt phần mềm:	40
2.1.1.	Cài đặt Java Runtime Environment (JRE)	40
2.1.2.	Cài đặt Arduino IDE.....	41
2.1.3.	Cài đặt Driver	44
TÀI LIỆU THAM KHẢO.....		48

DANH MỤC HÌNH ẢNH

Hình 1.1: Robot thám hiểm sao hỏa.

Hình 1.2: Robot công nghiệp.

Hình 2.1: Cảm biến siêu âm.

Hình 2.2: Mạch arduino UNO R3

Hình 2.3: Sơ đồ cổng vào ra.

Hình 2.4: Module L298N

Hình 3.1: Sơ đồ khối nguyên lý hoạt động

Hình 3.2: Flowchart cảm biến siêu âm

Hình 3.4: Flowchart tránh vật cản

Hình 4.1: Board Arduino

Hình 4.2: Cảm biến siêu âm SRF05

Hình 4.3: Mạch cầu H (L298N)

Hình 4.4: Mô hình xe tự hành tránh vật cản

Hình 4.5: Mô hình xe tự hành tránh vật cản

Hình 4.6: Mô hình xe tự hành tránh vật cản

Hình 6.1: Đèn LED tích hợp với chân 13

Hình 6.2: Bản JRE mới nhất là 7u80

Hình 6.3: Download Arduino software

Hình 6.5: Download Arduino software

Hình 6.5: Giải nén chương trình

Hình 6.6: Khởi động Arduino

Hình 6.7: Cài đặt driver

Hình 6.8: Cài đặt driver

Hình 6.9: Cài đặt driver

Hình 6.10: Kết thúc cài đặt

Hình 6.11: Giao diện Arduino

Hình 6.12: Giao diện Arduino

Hình 6.13: Vùng thông báo

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1. Xu hướng phát triển của robot hiện đại:

Theo dự báo trong vòng 20 năm nữa mỗi người sẽ có nhu cầu sử dụng một rô bốt cá nhân như cần một máy tính PC hiện nay và Rô bốt sẽ là tâm điểm của một cuộc cách mạng công nghệ lớn sau Internet. Với xu thế này, cùng với các ứng dụng truyền thống khác của rô bốt trong công nghiệp, y tế, giáo dục đào tạo, giải trí và đặc biệt trong an ninh quốc phòng thì thị trường rô bốt và các dịch vụ ăn theo rô bốt sẽ vô cùng lớn.

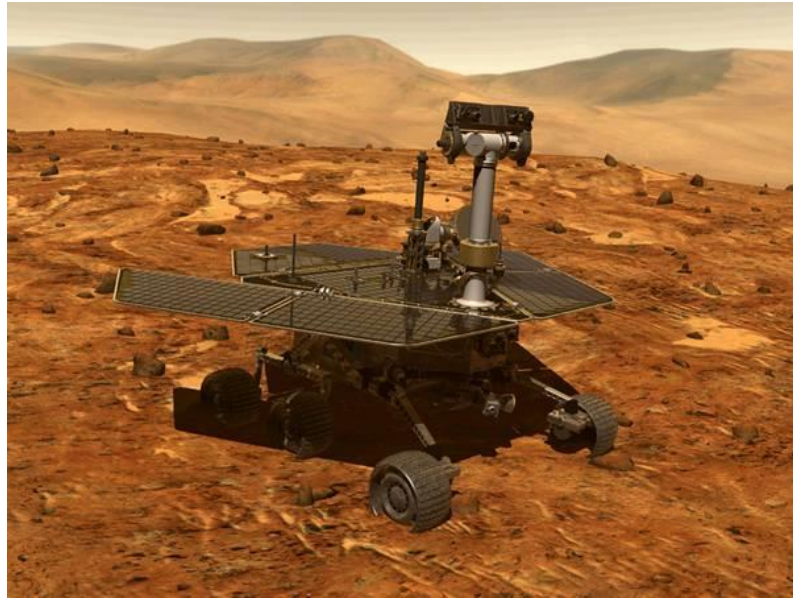
Robot tổng hợp trong nó cả khoa học và công nghệ. Để thiết kế và chế tạo được Robot, ta cần có các tri thức của toán học, cơ học, vật lý, điện tử, lý thuyết điều khiển, khoa học tính toán và nhiều tri thức khác. Để có thể ứng dụng được Robot, ta cần biết rõ về đối tượng ứng dụng. Robot là sản phẩm tích hợp cả khoa học và công nghệ với độ phức tạp cao.

Rô bốt đã có những tiến bộ đáng kể trong hơn nửa thế kỷ qua. Rô bốt đầu tiên được ứng dụng trong công nghiệp vào những năm 60 để thay thế con người làm các công việc nặng nhọc, nguy hiểm trong môi trường độc hại. Do nhu cầu cần sử dụng ngày càng nhiều trong các quá trình sản xuất phức tạp nên rô bốt công nghiệp cần có những khả năng thích ứng linh hoạt và thông minh hơn. Ngày nay, ngoài ứng dụng sơ khai ban đầu của rô bốt trong chế tạo máy thì các ứng dụng khác như trong y tế, chăm sóc sức khỏe, nông nghiệp, đóng tàu, xây dựng, an ninh quốc phòng và gia đình đang có nhu cầu gia tăng đang là động lực cho các rô bốt địa hình và rô bốt dịch vụ phát triển.

Có thể kể đến một số loại rô bốt được quan tâm nhiều trong thời gian qua là: tay máy rô bốt (Robot Manipulators), rô bốt di động (Mobile Robots), rô bốt phỏng sinh học (Bio Inspired Robots) và rô bốt cá nhân (Personal Robots). Tay máy rô bốt bao gồm các loại rô bốt công nghiệp (Industrial Robot), rô bốt y tế (Medical Robot) và rô bốt trợ giúp người tàn tật (Rehabilitation robot). Rô bốt di động được nghiên cứu nhiều như xe tự hành trên mặt đất AGV (Autonomous Guided Vehicles), rô bốt tự hành dưới nước AUV (Autonomous Underwater Vehicles), rô bốt tự hành trên không UAV (Unmanned Aerial Vehicles) và rô bốt vũ trụ (Space robots). Với rô bốt phỏng sinh học, các nghiên cứu thời gian qua tập trung vào 2 loại chính là rô bốt đi (Walking robots) và rô bốt dáng người (Humanoid Robots). Bên cạnh đó, các loại rô bốt phỏng sinh học dưới nước như rô bốt cá, các cấu trúc chuyển động phỏng theo sinh vật biển cũng được nhiều nhóm nghiên cứu phát triển.

Mặc dù về cấu trúc các loại rô bốt có nhiều điểm khác nhau nhưng các nghiên cứu hiện nay đều hướng về các ứng dụng dịch vụ và hoạt động của rô bốt trong môi trường tự nhiên. Mỹ là nước tập trung đầu tư hàng trăm tỷ USD cho phát triển hệ thống tác chiến tương lai trong đó rô bốt quân sự hoạt động ở các địa hình trên cạn, dưới nước và trên không chiếm phần không nhỏ. Với sự phát triển

của xã hội và quá trình hiện đại hóa ở các nước G7 thì nhiều dịch vụ mới được hình thành làm thay đổi quan điểm về rô bốt, từ rô bốt phục vụ công nghiệp sang rô bốt phục vụ cho các nhu cầu xã hội và nhu cầu cá nhân của con người. Rô bốt được thiết kế phục vụ cho con người như những người giúp việc đắc lực, đa năng. Quá trình này dẫn đến các khái niệm mới và cả tay máy công nghiệp lẫn rô bốt di động truyền thống sẽ phải thay đổi một cách cơ bản mới đáp ứng được yêu cầu của thị trường dịch vụ con người này.



Hình 1.1: robot thám hiểm sao hỏa.

Tại Việt Nam, nghiên cứu phát triển rô bốt đã có những bước tiến đáng kể trong 25 năm vừa qua. Nhiều đơn vị trên toàn quốc thực hiện các nghiên cứu cơ bản và nghiên cứu ứng dụng về rô bốt như Trung tâm Tự động hoá, Đại học Bách Khoa Hà Nội, Viện Điện tử, Tin học, Tự động hoá thuộc Bộ Công nghiệp, Đại học Bách khoa TP.HCM, Viện Khoa học và Công nghệ quân sự, Học viện Kỹ thuật Quân sự, Viện Cơ học, Viện Công nghệ thông tin thuộc Viện KHCNVN. Bên cạnh đó còn phải kể đến Công ty Cổ phần Robot TOSY, doanh nghiệp thiết kế và chế tạo rô bốt Việt Nam có nhiều sản phẩm ấn tượng trên trường quốc tế.

Song song với chế tạo rô bốt thì các công trình nghiên cứu khoa học về rô bốt được công bố của các nhà khoa học Việt Nam rất đa dạng và theo sát được các hướng nghiên cứu của thế giới. Các nghiên cứu về rô bốt ở Việt Nam liên quan nhiều đến các vấn đề về động học, động lực học, thiết kế quỹ đạo, xử lý thông tin cảm biến, cơ cấu chấp hành, điều khiển và phát triển trí thông minh cho rô bốt. Các nghiên cứu về động học và động lực học rô bốt được các khoa cơ khí, chế tạo máy ở các trường đại học và các viện nghiên cứu về cơ học, chế tạo máy, cơ khí quan tâm cả trong dân sự và quân sự. Ngoài việc tìm các phương pháp giải các bài toán liên quan đến cơ học của các loại rô bốt nổi tiếp, song song, di động, thì các chương trình mô phỏng kết cấu và chuyển động 3D được

áp dụng và phát triển để minh họa cũng như phục vụ cho phân tích, thiết kế rô bốt. Các công bố liên quan về cơ học rô bốt thường được thực hiện bởi cán bộ của Viện Cơ học - Viện KH&CN Việt Nam,



Hình 1.2: robot công nghiệp.

Khoa Cơ khí Chế tạo máy thuộc đại học Bách Khoa Hà Nội và Đại học Bách Khoa Tp.HCM, các bộ môn Rô bốt và Cơ điện tử ở các trường đại học khác. Lĩnh vực điều khiển rô bốt rất phong phú, từ các phương pháp điều khiển truyền thống như PID, phương pháp tính mô men, phương pháp điều khiển trượt đến các phương pháp điều khiển thông minh như điều khiển sử dụng mạng nơ ron, logic mờ, thuật gen và các phương pháp điều khiển tự thích nghi, các phương pháp học cho rô bốt, các hệ visual servoing... Trong các công bố về điều khiển rô bốt phải kể đến các công trình của Viện Công Nghệ Thông tin Viện KH&CN Việt nam, Đại học Bách Khoa Hà Nội và Đại học Bách Khoa Tp.HCM. Gần đây, đội ngũ nghiên cứu và giảng dạy ở Đại học Quốc gia Hà Nội và Học viện Kỹ thuật Quân sự cũng có nhiều công bố liên quan đến lĩnh vực điều khiển rô bốt do đội ngũ giáo viên trẻ tốt nghiệp tiến sỹ ở nước ngoài về tiếp tục các nghiên cứu của mình. Lĩnh vực rô bốt di động với nhiều cảm biến dẫn đường và camera đang được nhiều đơn vị trong nước quan tâm nghiên cứu. Các vấn đề xử lý ảnh tốc độ cao, phối hợp đa cảm biến, định vị và lập bản đồ không gian, thiết kế quỹ đạo chuyển động cho rô bốt di động đã có nhiều công bố trong các Hội nghị cơ điện tử toàn quốc năm 2004, 2006, 2008 và 2010. Các nghiên cứu về thị giác rô bốt được quan tâm cả ở rô bốt công nghiệp và rô bốt di động, nhất là lĩnh vực nhận dạng và điều khiển rô bốt trên cơ sở thông tin hình ảnh. Các vấn đề về xử lý ngôn ngữ tự nhiên, nhận dạng và tổng hợp tiếng nói tiếng Việt bắt đầu được chú ý cho các loại rô bốt dịch vụ.

Các nghiên cứu cơ bản về rô bốt của Việt Nam đã được công bố nhiều trên các Hội Nghị và tạp chí quốc tế. Việc phối hợp với các nước như Nhật, Mỹ, Singapore, Đức tổ chức các hội nghị quốc tế tại Việt nam liên quan đến rô bốt

nếu RESCCE'98, RESCCE'00, RESCCE'02, ICMT2004, ICARCV 2008, ITOMM 2009 là một chuỗi hoạt động khoa học liên tục của cộng đồng Robotics Việt nam hòa nhập vào các hoạt động nghiên cứu khoa học với các nước khu vực và tiên tiến trên thế giới.

Với xu thế toàn cầu hóa, sự phân công lao động trong chuỗi cung sản phẩm và dịch vụ trên thế giới đã không còn giới hạn địa lý. Cơ hội cho mỗi cá nhân, tổ chức có thể tham gia vào các công việc trên thế giới là bình đẳng cho mọi người, mọi dân tộc và mọi quốc gia. Vì vậy, với bối cảnh rô bốt sẽ là trung tâm của cuộc cách mạng công nghệ kế tiếp sau PC- Internet của thế giới trong vòng 20 năm nữa, Việt Nam không thể bỏ lỡ cơ hội này như với máy vi tính PC 30 năm trước. Để đạt được điều này, chúng ta phải có những định hướng ngay từ bây giờ như:

- Về đào tạo: Tập trung phát triển đông đảo nguồn nhân lực có kiến thức toàn diện từ sử dụng đến nghiên cứu phát triển các rô bốt và các ứng dụng liên quan.
- Về nghiên cứu: Tập trung cho phát triển trí tuệ của rô bốt từ mức thấp với khả năng giải quyết một vài việc cụ thể đến mức cao với khả năng nhận thức, suy diễn và ra quyết định để rô bốt có thể thích ứng với môi trường tự nhiên và tham gia vào xã hội loài người.
- Về sản xuất: Tập trung thiết kế và chế tạo các loại rô bốt dịch vụ Việt Nam có tính thực dụng cao, giá rẻ, đơn giản và chuyên dụng cho các công việc cụ thể.
- Tham gia vào chuỗi cung toàn cầu của các sản phẩm, hệ thống rô bốt từ các công việc tay chân như lắp ráp, gia công đến các công việc trí óc như thiết kế, nghiên cứu và đào tạo. Chuẩn bị nguồn lực cho các nhu cầu của một xã hội có sự hiện hữu phổ biến của rô bốt trong đời sống hàng ngày.
- Viện Khoa học và Công nghệ Việt Nam cần có chiến lược dài hạn cho phát triển các nghiên cứu về rô bốt, phát huy vai trò đầu tàu về nghiên cứu cơ bản trong phát triển rô bốt ở Việt Nam, một lĩnh vực trung tâm của cuộc cách mạng công nghệ lớn tiếp theo, có tầm ảnh hưởng đến toàn xã hội loài người trong thời gian 15-20 năm tới.

Để có thể hội nhập và phát triển trong xu thế toàn cầu hóa hiện nay Việt Nam cần có một cộng đồng rộng lớn các chuyên gia tâm huyết, có môi trường học tập và nghiên cứu lành mạnh và một chính sách vĩ mô hỗ trợ tri thức phát triển thích nghi được với quá trình “phẳng” hóa thế giới hiện nay.

2. Mục tiêu đề tài

Mục đích của đề tài là xây dựng mô hình mobile robot có khả năng tự hành hóa tránh được các loại vật cản trên đường hành tiến, mục đích sâu xa của đề tài là phát

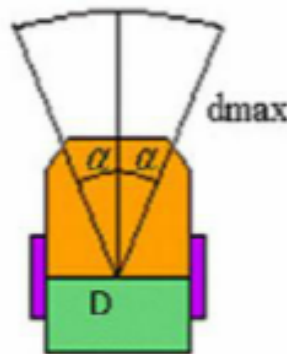
triển robot tự động thông minh, được ứng dụng phục vụ con người, giúp việc, thám hiểm, thực hiện các hoạt động nguy hiểm, trong không gian hạn chế

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

1. Cảm biến siêu âm:

1.1. Hoạt động:

Cảm biến sẽ phát ra sóng siêu âm với góc mở nhất định. Khi đó nếu trong tầm quét của nó phát hiện chướng ngại vật thì sóng siêu âm sẽ bị phản hồi lại. Ta có thể tính khoảng cách bằng cách tính thời gian từ lúc sóng siêu âm phát ra đến lúc sóng thu về, kết hợp với vận tốc sóng siêu âm (khoảng 343m/s) để biết được quãng đường mà sóng đi.



Hình 2.1: cảm biến siêu âm

Các cảm biến được đặt lệch một góc α , khoảng cách lớn nhất (tính từ D) mà các cảm biến có thể nhận được là d_{max} , d_{max} và α phải đảm bảo sao cho cảm biến có vùng kiểm tra đủ rộng để khi tiến thẳng robot có thể nhận diện được vật cản.

Ưu điểm:

Xử lý nhanh, kết quả tương đối chính xác.

Khuyết điểm:

Chỉ nhận biết được vật cản khi mặt phẳng quét của cảm biến cắt ngang vật cản, do đó sẽ không phát hiện được những vật cản nhỏ, thấp và nằm sát mặt đất.

Do sử dụng sóng siêu âm và sự phản xạ của nó để tính toán khoảng cách và phát hiện vật cản nên giải thuật khá phức tạp và phát sinh một số sai số khó khắc phục: sai số lặp, hiện tượng Forecasting, hiện tượng đọc chéo (crosstalk).

Hiện tượng Forecasting: là hiện tượng phản xạ góc sai lệch của cảm biến. Theo nguyên lý TOF, để có khoảng cách đúng cảm biến siêu âm phải hướng vuông góc với chướng ngại vật, tuy nhiên chướng ngại không bao giờ là bằng phẳng, nên phản xạ có thể không tới được góc tới. Các chùm tia này có năng lượng phản xạ thấp hơn. Tuy vậy, ở một khoảng cách nào đó, cảm biến siêu âm vẫn có thể nhận được những tín hiệu này, kết quả là thông số đọc về cảm biến bị lệch do góc mở cam biến lớn.

Sai số lặp: là sai số luôn xảy ra với mọi thiết bị đo lường, kể cả cảm biến siêu âm.

Hiện tượng crosstalk: là hiện tượng mà cảm biến siêu âm này ghi nhận tín hiệu phản xạ hoặc trực tiếp từ cảm biến khác, hoặc sau khi quá trình sóng siêu âm truyền đi và phản xạ qua các bề mặt nó quay lại cảm biến theo một cách không mong muốn.

1.2. Cấu tạo:

Cấu tạo gồm 3 phần:

- **Phần phát tín hiệu:** Các đầu phát và đầu thu siêu âm là các loa gồm được chế tạo đặc biệt, hoạt động phát siêu âm có cường độ cao nhất ở một tần số nào đó (thường là 40kHz cho các ứng dụng đo khoảng cách). Các loa này cần có nguồn tín hiệu điều khiển có điện áp cao mới phát tốt được (theo datasheet thì là ~ 30V). Chính vì vậy trong phần phát, phần đệm công suất sử dụng một con MAX232 làm nhiệm vụ đệm. Nó sẽ lấy tín hiệu từ bộ điều khiển, khuếch đại biên độ lên +/-30V cung cấp cho loa gồm.
Để tiết kiệm nguồn cho module cảm biến, phần cấp điện cho MAX232 được điều khiển thông qua một transistor PNP, khi không hoạt động, bộ điều khiển sẽ làm cho transistor này ngưng dẫn, hạn chế tiêu thụ dòng.
- **Phần thu tín hiệu:** Khi loa gồm làm đầu thu (loa này được chế tạo chỉ nhạy với một tần số nào đó - 40kHz) thu được sóng siêu âm, nó sẽ phát ra một điện thế giữa hai cực. Điện thế này là rất nhỏ, vì vậy nó được đưa qua một OPAM, ở đây là TL072 (Một số module sử dụng LM324,...). Tín hiệu này liên tục được khuếch đại biên độ và cuối cùng là đưa qua một bộ so sánh, kết hợp với tín hiệu từ bộ điều khiển để đưa về bộ điều khiển thông qua một transistor NPN.
- **Phần xử lý, điều khiển:** Phần xử lý, điều khiển thường sử dụng một vi điều khiển (PIC16F688, STC11,...) làm nhiệm vụ phát xung, xử lý tính toán thời gian từ khi phát đến khi thu được sóng siêu âm do nó phát ra nếu nhận được tín hiệu TRIG. Đến đây thì nguyên lý hoạt động thông thường của cảm biến này thì ai cũng biết rồi nhé (cấp xung TRIG, chờ đo độ rộng xung ECHO để tính toán thời gian,...).

2. Mạch Arduino UNO R3:

Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà người ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3). Bạn sẽ bắt đầu đến với Arduino qua thứ này. Bạn có thể dùng Arduino Nano cũng được nhưng mình khuyên bạn nên dùng cái này.



Hình 2.2: mạch arduino UNO R3

2.1. Một vài thông số của Arduino UNO R3

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	khoảng 30mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA

Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

2.2. Vi điều khiển.

Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lý những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ - độ ẩm và hiển thị lên màn hình LCD,... hay những ứng dụng khác mà bạn đã được xem.

Thiết kế tiêu chuẩn của Arduino UNO sử dụng vi điều khiển ATmega328 với giá khoảng 90.000đ. Tuy nhiên nếu yêu cầu phần cứng của bạn không cao hoặc túi tiền không cho phép, bạn có thể sử dụng các loại vi điều khiển khác có chức năng tương đương nhưng rẻ hơn như ATmega8 (bộ nhớ flash 8KB) với giá khoảng 45.000đ hoặc ATmega168 (bộ nhớ flash 16KB) với giá khoảng 65.000đ.

Ngoài việc dùng cho board Arduino UNO, bạn có thể sử dụng những IC điều khiển này cho các mạch tự chế. Vì sao ? Vì bạn chỉ cần board Arduino UNO để lập trình cho vi điều khiển.

2.3. Năng lượng

Arduino UNO có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyến dùng là 7-12V DC và giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lý nhất nếu bạn không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, bạn sẽ làm hỏng Arduino UNO.

2.4. Các chân năng lượng

- **GND (Ground):** cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.

- **5V:** cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
- **3.3V:** cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
- **Vin (Voltage Input):** để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
- **IOREF:** điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
- **RESET:** việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

Lưu ý:

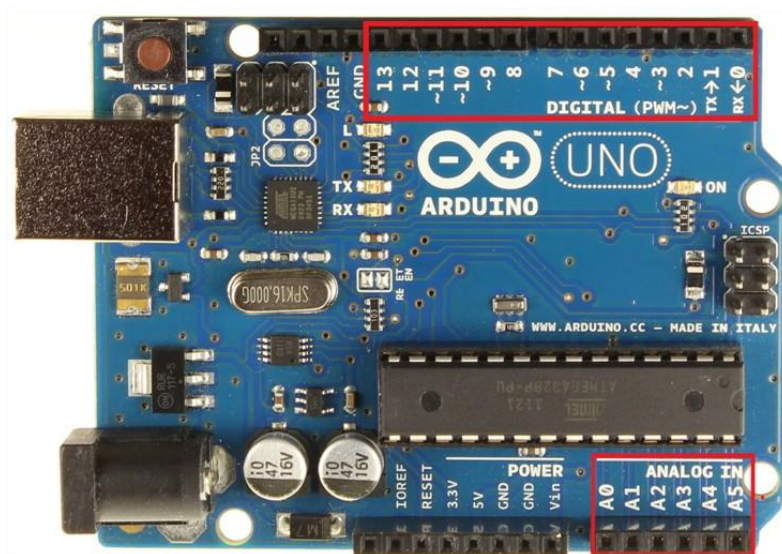
- Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó bạn phải hết sức cẩn thận, kiểm tra các cực âm – dương của nguồn trước khi cấp cho Arduino UNO. Việc làm chập mạch nguồn vào của Arduino UNO sẽ biến nó thành một miếng nhựa chặn giấy. mình khuyên bạn nên dùng nguồn từ cổng USB nếu có thể.
- Các chân 3.3V và 5V trên Arduino là các chân dùng để cấp nguồn ra cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng board. Điều này không được nhà sản xuất khuyến khích.
- Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng board.
- Cấp điện áp trên 13V vào chân RESET trên board có thể làm hỏng vi điều khiển ATmega328.
- Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.
- Cấp điện áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển.
- Cường độ dòng điện qua một chân Digital hoặc Analog bất kỳ của Arduino UNO vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó nếu không dùng để truyền nhận dữ liệu, bạn phải mắc một điện trở hạn dòng.

Khi mình nói rằng bạn “*có thể làm hỏng*”, điều đó có nghĩa là chưa chắc sẽ hỏng ngay bởi các thông số kỹ thuật của linh kiện điện tử luôn có một sự tương đối nhất định. Do đó hãy cứ tuân thủ theo những thông số kỹ thuật của nhà sản xuất nếu bạn không muốn phải mua một board Arduino UNO thứ 2. Khi mình nói rằng bạn “*có thể làm hỏng*”, điều đó có nghĩa là chưa chắc sẽ hỏng ngay bởi các thông số kỹ thuật của linh kiện điện tử luôn có một sự tương đối nhất định. Do đó hãy cứ tuân thủ theo những thông số kỹ thuật của nhà sản xuất nếu bạn không muốn phải mua một board Arduino UNO thứ 2.

2.5. Bộ nhớ

Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

- **32KB bộ nhớ Flash:** những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng đừng lo, bạn hiếm khi nào cần quá 20KB bộ nhớ này đâu.
- **2KB cho SRAM (Static Random Access Memory):** giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ mà bạn phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
- **1KBEEPROM (Electrically Erasable Programmable Read Only Memory):** đây giống như một chiếc ổ cứng mini – nơi bạn có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.



Hình 2.3: sơ đồ cổng vào ra

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

Một số chân digital có các chức năng đặc biệt như sau:

- **2 chân Serial:** 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết
- **Chân PWM (~): 3, 5, 6, 9, 10, và 11:** cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ $0 \rightarrow 2^8-1$ tương ứng với $0V \rightarrow 5V$) bằng hàm `analogWrite()`. Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
- **Chân giao tiếp SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.
- **LED 13:** trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

Arduino UNO có 6 chân analog (A0 \rightarrow A5) cung cấp độ phân giải tín hiệu 10bit ($0 \rightarrow 2^{10}-1$) để đọc giá trị điện áp trong khoảng $0V \rightarrow 5V$. Với chân **AREF** trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ $0V \rightarrow 2.5V$ với độ phân giải vẫn là 10bit.

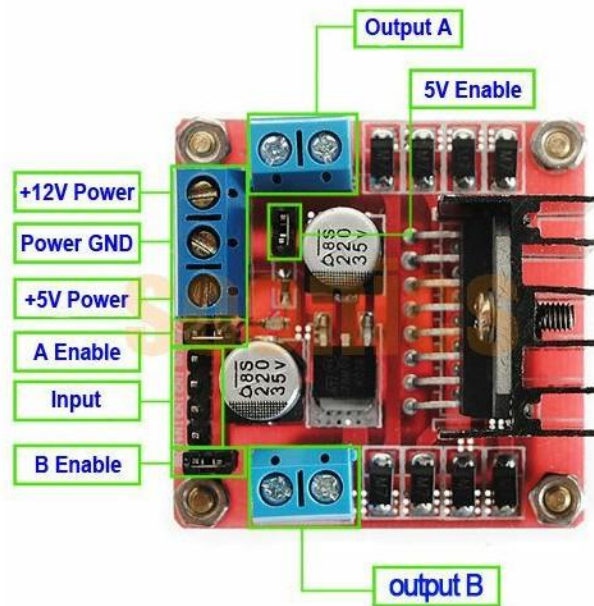
Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

2.6. Lập trình cho Arduino

Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++. Riêng mình thì gọi nó là “*ngôn ngữ Arduino*”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu.

2.7. Module L298N:

Module L298N cũng là một module thông dụng với chức năng thông dụng và giá thành cực kỳ rẻ là lựa chọn của các bạn học sinh, sinh viên.



Hình 2.4: module L298N

Thông số kỹ thuật:

- Driver: L298N tích hợp hai mạch cầu H.
- Điện áp điều khiển: +5 V ~ +12 V
- Dòng tối đa cho mỗi cầu H là: 2A (\Rightarrow 2A cho mỗi motor)
- Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
- Dòng của tín hiệu điều khiển: 0 ~ 36mA (Arduino có thể chơi đến 40mA nên khỏe re nhé các bạn)
- Công suất hao phí: 20W (khi nhiệt độ $T = 75^\circ\text{C}$)
- Nhiệt độ bảo quản: $-25^\circ\text{C} \sim +130^\circ\text{C}$

L298 gồm các chân:

- 12V power, 5V power. Đây là 2 chân cấp nguồn trực tiếp đến động cơ.
 - Bạn có thể cấp nguồn 9-12V ở 12V.
 - Bên cạnh đó có jumper 5V, nếu bạn để như hình ở trên thì sẽ có nguồn 5V ra ở cổng 5V power, ngược lại thì không. Bạn để như hình thì ta chỉ cần cấp nguồn 12V vô ở 12V power là có 5V ở 5V power, từ đó cấp cho Arduino
- Power GND chân này là GND của nguồn cấp cho Động cơ.
 - Nếu chơi Arduino thì nhớ nối với GND của Arduino
- 2 Jump A enable và B enable, để như hình.
- Gồm có 4 chân Input. IN1, IN2, IN3, IN4.

- Output A: nối với động cơ A. bạn chú ý chân +, -. Nếu nối ngược thì động cơ sẽ chạy ngược. Và chú ý nếu nối động cơ bước, bạn phải đấu nối các pha cho phù hợp.

2.8. Sơ đồ kết nối chân:

Arduino	Module L298	Servo	SRF-05	Đế Pin	Bánh Xe 1	Bánh Xe 2
Nguồn VCC	VCC			RED -C.TẮC		
Nguồn GND	GND			BLACK		
5V		RED				
GND		BROWN				
5V			VCC			
GND			GND			
~5			ECHO			
			TRIG			

~6						
~9		ORANGE				
~10	IN-A					
~11	IN-C					
12	IN-D					
13	IN-B					
	OUT-A				RED	
	OUT-B				BLACK	
	OUT-C					RED
	OUT-D					BLACK

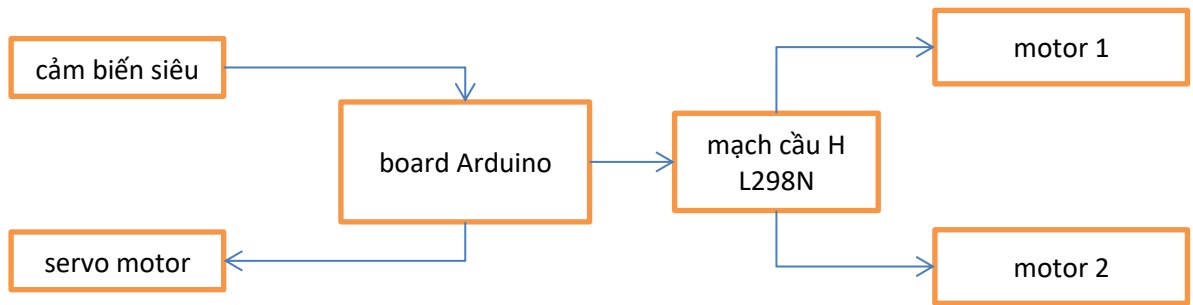
CHƯƠNG 3: CƠ SỞ THỰC HIỆN

1. Nguyên lý điều khiển:

Robot tự hành sử dụng cảm biến siêu âm SRF05 để đo khoảng cách. Môi trường làm việc của robot là một mặt phẳng, được giới hạn bởi các bức tường, các vật cản được xem là vật cản hai chiều tĩnh hoàn toàn. Trong bài toán robot được trang bị sensor cảm biến siêu âm SRF05 để đo khoảng cách tới vật cản.

Đầu tiên, sensor sẽ quét bên trái trước, sau đó gửi kết quả đo đạt về vi điều khiển (nguyên lý đo khoảng cách của cảm biến siêu âm là sử dụng nguyên lý phản xạ của sóng để đo khoảng cách). Khi muốn đo khoảng cách, SRF05 sẽ phát ra một bộ 8 xung với tần số 40KHz, sau đó sẽ chờ đợi xung phản xạ về. Thời gian tính từ lúc xung đi cho tới xung về có thể tính để dàng khoảng cách từ SRF05 tới vật cản. Khi phát ra xung và chờ phản xạ về, chân Echo của SRF05 được kéo lên mức cao (echo=1) khi có xung phản xạ về chân Echo sẽ được kéo xuống mức thấp hoặc sau 30 μ s nếu không có xung phản xạ về. Vi điều khiển căn cứ vào kết quả đo được của cảm biến SRF05 tiến hành xử lý (căn cứ vào các tình huống có thể xảy ra khi sensor phát hiện vật cản, mỗi tình huống cần có cách xử lý riêng) và đưa ra quyết định điều khiển động cơ 1 hay động cơ 2 hay cả 2 động cơ cùng lúc.

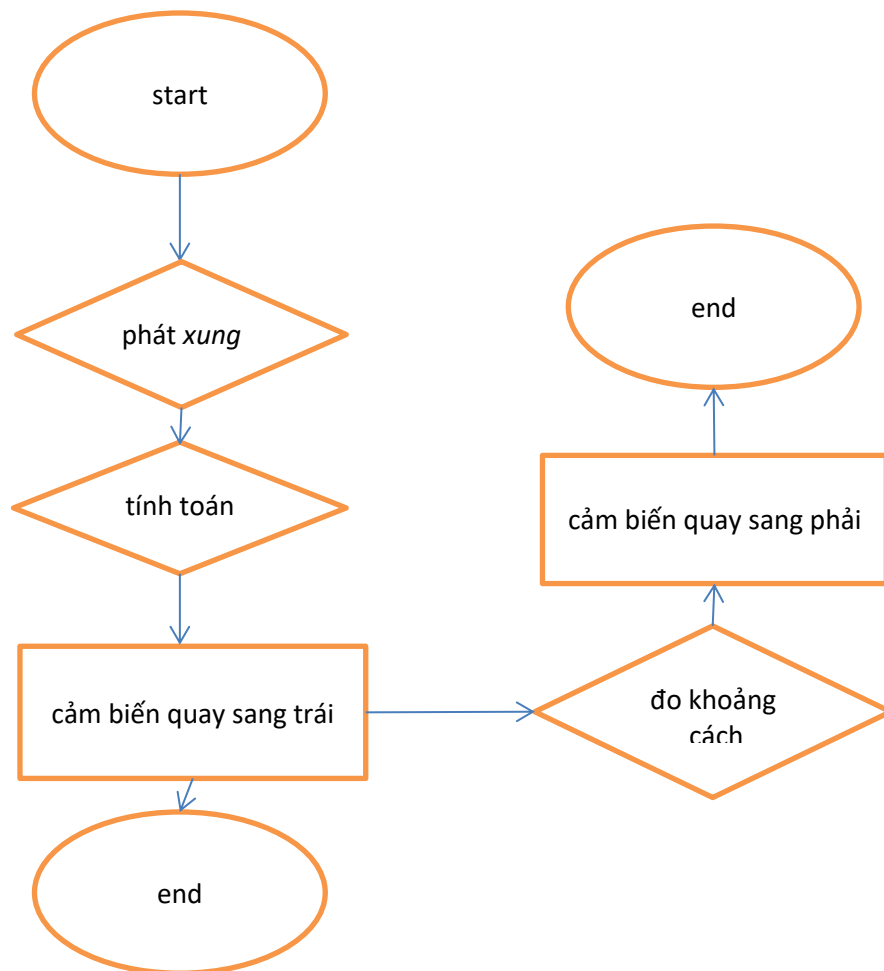
Khởi đầu cảm biến sẽ quét, truyền tín hiệu về Arduino, Arduino gửi tín hiệu về servo motor, quay servo motor sang trái trước để cảm biến siêu âm quét vật cản. Code sử dụng biến giới hạn là 25 cm để xác định khoảng cách cần dừng. Khi ta muốn thay đổi khoảng cách, ta thay đổi tham số "gioihan" đang để mặc định là 25. (đơn vị cm). Xe ưu tiên rẽ trái trong các trường hợp phía trước có vật cản. Ví dụ: khi khoảng cách phía trước < 25, cảm biến đo bên trái trước, nếu không có vật cản thì xe rẽ trái. nếu bên trái cũng có vật cản lúc đó xe mới kiểm tra bên phải. Khi cả phía trước, trái, phải đều có chướng ngại vật. xe đi lùi 1 đoạn, sau đó đo lại khoảng cách.



Hình 3.1: Sơ đồ khối nguyên lý hoạt động

2. Điều khiển cảm biến siêu âm.

2.1 Flowchart cảm biến



Hình 3.2: Flowchart cảm biến siêu âm

2.2. Code điều khiển.

```
void dokhoangcach()
{
    /* Phát xung từ chân trig */

    digitalWrite(trig,0); // tắt chân trig
    delayMicroseconds(2);

    digitalWrite(trig,1); // phát xung từ chân trig
    delayMicroseconds(10); // xung có độ dài 5 microSeconds
    digitalWrite(trig,0); // tắt chân trig

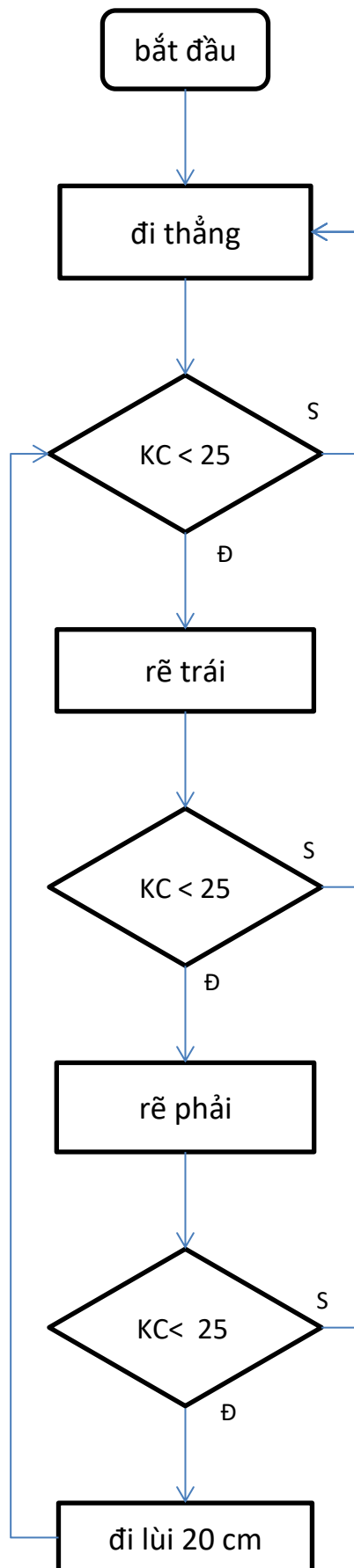

    /* Tính toán thời gian */

    // Đo độ rộng xung HIGH ở chân echo.
    thoigian = pulseIn(echo,HIGH);

    // Tính khoảng cách đến vật.
    khoangcach = int(thoigian/2/29.412);
}
```

3. Điều khiển né vật cản.

3.1. Flowchart tránh vật cản.



3.2. Code điều khiển.

```
void loop()
{
    khoangcach=0;
    dokhoangcach();
    if(khoangcach>gioihan||khoangcach==0)
    {
        dokhoangcach();
        if(khoangcach>gioihan||khoangcach==0)
        {
            dithang();
        }
    }
    else
    {

        resetdongco();
        quaycbsangtrai();

        khoangcachtrai=khoangcach;

        quaycbsangphai();
        khoangcachphai=khoangcach;
        if(khoangcachphai<10&&khoangcachtrai<10){
            dilui();
        }

        else
        {
```

```
        if(khoangcachphai>khoangcachtrai)
        {
            disangtrai();
            delay(500);
        }
        if(khoangcachphai<khoangcachtrai)
        {
            disangphai();
            delay(500);
        }
    }
}
```

```
//Serial.println(khoangcach);
}
```

```
void dithang()
{
```

```
    digitalWrite(tien1,HIGH);
    digitalWrite(tien2,HIGH);
    //delay(2);

}
```

```
void disangtrai()
{
    resetdongco();
    digitalWrite(lui1,HIGH);
    delay(250);
    digitalWrite(lui1,LOW);
```



```
    }  
    void disangphai()  
    {  
        resetdongco();  
        digitalWrite(lui2,HIGH);  
        delay(250);  
        digitalWrite(lui2,LOW);  
    }  
  
    void dilui()  
    {  
        resetdongco();  
        for(i=0;i<20;i++)  
  
            digitalWrite(lui1,HIGH);  
            digitalWrite(lui2,HIGH);  
            delay(700);  
  
            digitalWrite(lui1,LOW);  
            digitalWrite(lui2,LOW);  
    }  
  
    void resetdongco()  
    {  
        digitalWrite(tien1,LOW);
```

```
digitalWrite(tien2,LOW);  
digitalWrite(lui1,LOW);  
digitalWrite(lui2,LOW);  
}
```

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

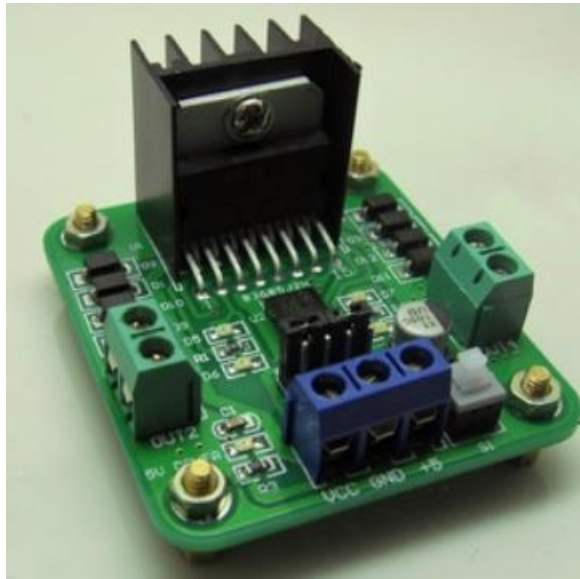
1. Ảnh chụp các thành phần của mô hình.



Hình 4.1: Board Arduino

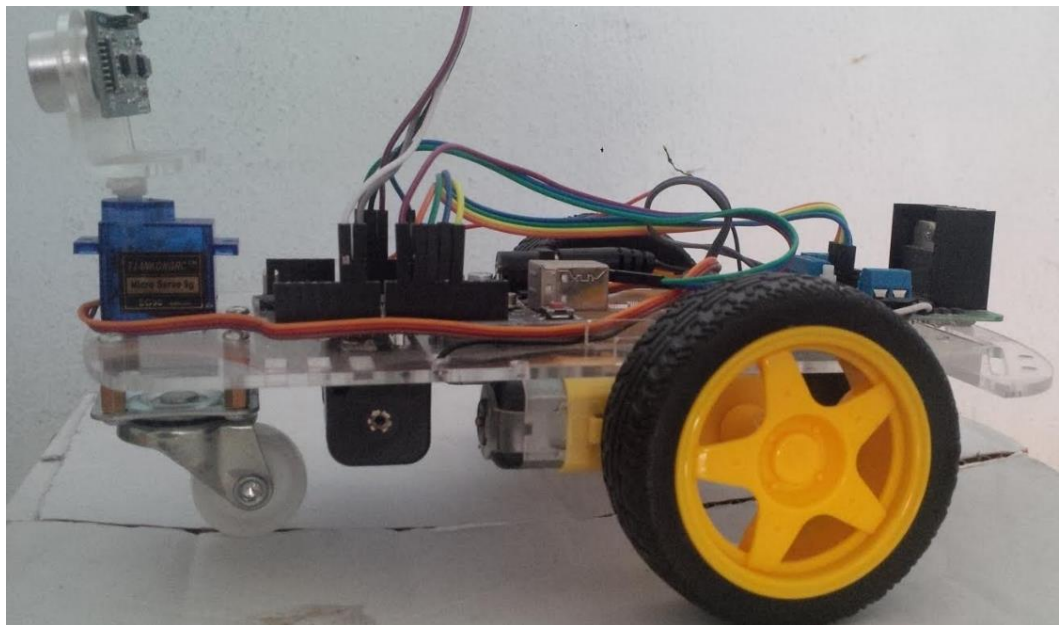


Hình 4.2: Cảm biến siêu âm SRF05

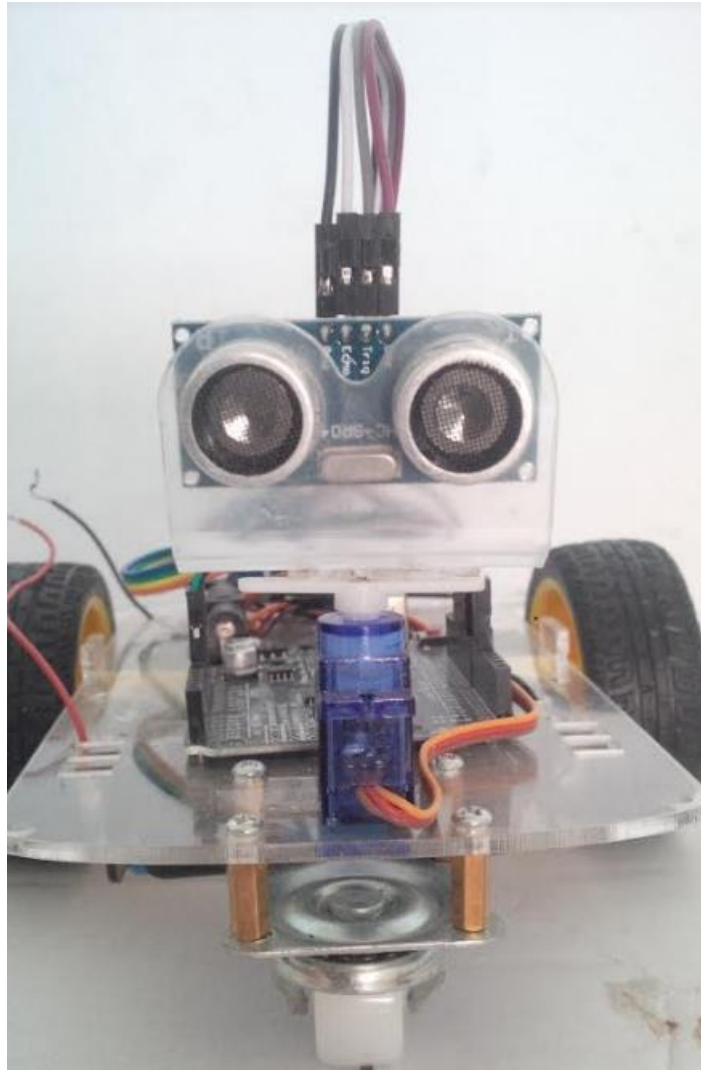


Hình 4.3: Mạch cầu H (L298N)

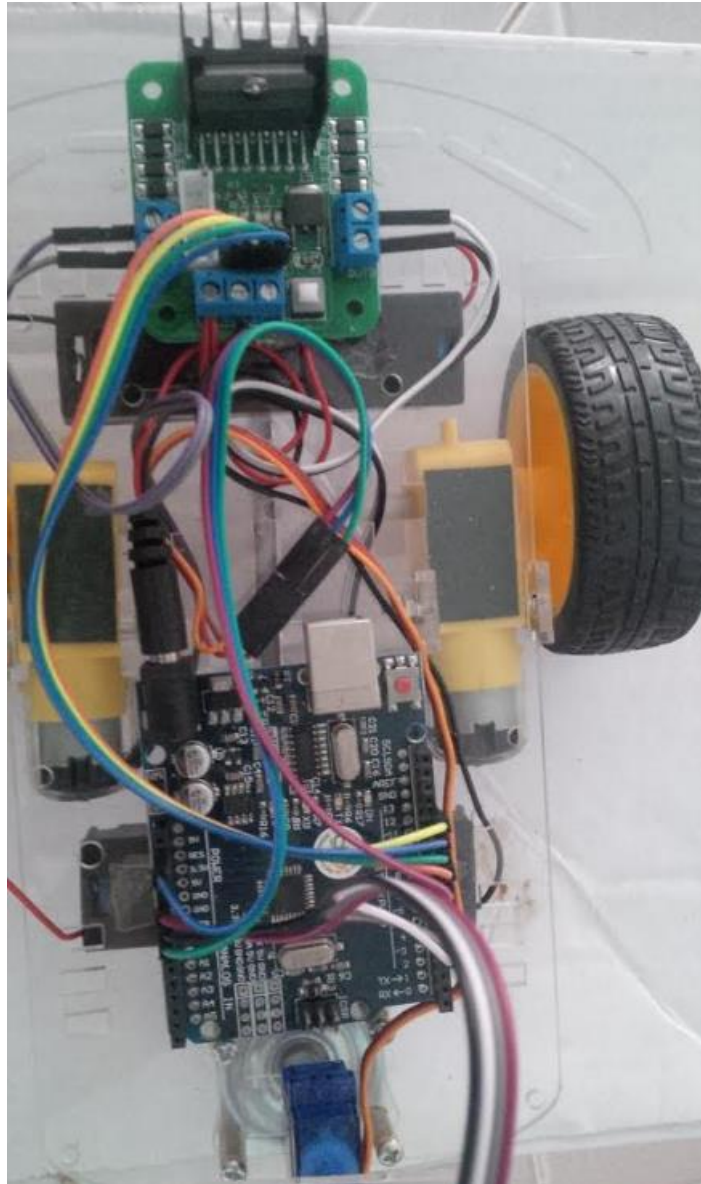
2. Mô hình hoàn thiện.



Hình 4.4: Mô hình xe tự hành tránh vật cản



Hình 4.5: Mô hình xe tự hành tránh vật cản



Hình 4.6: Mô hình xe tự hành tránh vật cản

CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG ĐỀ TÀI

Kết quả đạt được:

Do trình độ có hạn và thời gian đầu tư cho đề tài chưa nhiều nên đề tài đã được những kết quả sau:

- Khảo sát được quy trình phát triển của robot.
- Nắm bắt được các khái niệm cơ bản phục vụ viết báo cáo.
- Phân tích được vai trò của robot trong cuộc sống hiện đại.
- Thiết kế, chế tạo mô hình robot tránh vật cản.

Hệ thống có khả năng điều hướng đưa robot tránh được các vật cản trên đường hành tiến. Tuy nhiên, thực tế gắn cảm biến siêu âm lên robot thì trong quá trình duy chuyển xuất hiện các nhiễu làm robot không phát hiện được vật cản.

1. Hạn chế:

- Phân thị giác của robot chỉ quan sát được không gian phía trước robot nên robot có thể sẽ đụng vật cản bên hông Hay phía sau khi di chuyển lùi. Vấn đề này có thể được giải quyết bằng cách trang bị thêm các cảm biến xung quanh robot.
- Tích hợp thêm các cảm biến định vị cần thiết để tăng sự chính xác.

2. Hướng phát triển của đề tài:

- Cải tiến, hoàn thiện một số tính năng chưa hoàn chỉnh của mô hình.
- Mục đích của đề tài là tạo ra nền tảng cho việc xây dựng một mô hình robot dịch vụ hoàn chỉnh, một robot có khả năng làm các công việc thay cho con người như: bưng bê đồ đạc, lau nhà, hướng dẫn khách hàng.

PHỤ LỤC

1. Code chương trình:

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo

int pos = 0; // variable to store the servo position

/***** khai báo chân input/output*****/

const int trig = 6; // chân trig của SRF-05.

const int echo = 5; // chân echo của SRF-05.

int tien1=10; // chân IN - A của Module L298.

int tien2=11; // chân IN - C của Module L298.

int lui1=12; // chân IN - D của Module L298.

int lui2=13; // chân IN - B của Module L298.

int dongcoservo=9; // chân Orange của Servo.

int gioihan = 25;

int i;

int x=0;

unsigned long thoigian; // biến đo thời gian

int khoangcach; // biến lưu khoảng cách

int khoangcachtrai,khoangcachphai;

void dokhoangcach();
```



```
void dithang(int duongdi);

void disangtrai();

void disangphai();

void dilui();

void resetdongco();

void quaycbsangphai();

void quaycbsangtrai();

void setup() {

    // put your setup code here, to run once:

    myservo.attach(9); // attaches the servo on pin 9 to the servo object

    pinMode(trig,OUTPUT); // chân trig sẽ phát tín hiệu

    pinMode(echo,INPUT); // chân echo sẽ nhận tín hiệu


    pinMode(tien1,OUTPUT);

    pinMode(tien2,OUTPUT);

    pinMode(lui1,OUTPUT);

    pinMode(lui2,OUTPUT);

    digitalWrite(tien1,LOW);

    digitalWrite(tien2,LOW);

    digitalWrite(lui1,LOW);

    digitalWrite(lui1,LOW);

    // Serial.begin(9600);

    myservo.write(90);

    delay(500);
```

```
}
```

```
void loop()
```

```
{
```

```
    khoangcach=0;
```

```
    dokhoangcach();
```

```
    if(khoangcach>gioihan||khoangcach==0)
```

```
    {
```

```
        dokhoangcach();
```

```
        if(khoangcach>gioihan||khoangcach==0)
```

```
        {
```

```
            dithang();
```

```
        }
```

```
    }
```

```
    else
```

```
    {
```

```
        resetdongco();
```

```
        quaycbsangtrai();
```

```
        khoangcachtrai=khoangcach;
```

```
    quaycbsangphai();  
    khoangcachphai=khoangcach;  
    if(khoangcachphai<10&&khoangcachtrai<10){  
        dilui();  
    }  
    else  
    {  
        if(khoangcachphai>khoangcachtrai)  
        {  
            disangtrai();  
            delay(500);  
        }  
        if(khoangcachphai<khoangcachtrai)  
        {  
            disangphai();  
            delay(500);  
        }  
    }  
    }  
    //Serial.println(khoangcach);  
}  
  
void dithang()  
{
```

```
        digitalWrite(tien1,HIGH);  
        digitalWrite(tien2,HIGH);  
        //delay(2);  
    }
```

```
void disangtrai()  
{  
    resetdongco();  
    digitalWrite(lui1,HIGH);  
    delay(250);  
    digitalWrite(lui1,LOW);  
  
}
```

```
void disangphai()  
{  
    resetdongco();  
    digitalWrite(lui2,HIGH);  
    delay(250);  
    digitalWrite(lui2,LOW);  
}
```

```
}
```

```
void dilui()
```

```
{
```

```
    resetdongco();
```

```
    for(i=0;i<20;i++)
```

```
        digitalWrite(lui1,HIGH);
```

```
        digitalWrite(lui2,HIGH);
```

```
        delay(700);
```

```
        digitalWrite(lui1,LOW);
```

```
        digitalWrite(lui2,LOW);
```

```
}
```

```
void resetdongco()
```

```
{
```

```
    digitalWrite(tien1,LOW);
```

```
    digitalWrite(tien2,LOW);
```

```
    digitalWrite(lui1,LOW);
```

```
    digitalWrite(lui2,LOW);
```

```
}
```

```
/****** chương trình đo khoảng cách SRF04 *****/
```

```
void dokhoangcach()
```

```
{  
/* Phát xung từ chân trig */  
  
digitalWrite(trig,0); // tắt chân trig  
  
delayMicroseconds(2);  
  
digitalWrite(trig,1); // phát xung từ chân trig  
  
delayMicroseconds(10); // xung có độ dài 5 microSeconds  
  
digitalWrite(trig,0); // tắt chân trig  
  
  
/* Tính toán thời gian */  
  
// Đo độ rộng xung HIGH ở chân echo.  
  
thoigian = pulseIn(echo,HIGH);  
  
// Tính khoảng cách đến vật.  
  
khoangcach = int(thoigian/2/29.412);  
  
  
}  
  
  
/***** chương trình quay cảm biến xang trái *****/  
  
void quaycbsangtrai()  
{  
  
myservo.write(180); // tell servo to go to position in variable 'pos'  
  
delay(1000);  
  
dokhoangcach();  
  
myservo.write(90); // tell servo to go to position in variable 'pos'
```

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

2. Giới thiệu phần mềm sử dụng:

2.1. Hướng dẫn cài đặt phần mềm:

Để lập trình được cho các board Arduino, các bạn cần phải có một công cụ gọi là **Intergrated 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. Ở đây mình sẽ hướng dẫn các bạn cài đặt nó trên Windows. Các bạn cũng làm tương tự các bước này cho các nền tảng khác

2.1.1. Cài đặt Java Runtime Environment (JRE)

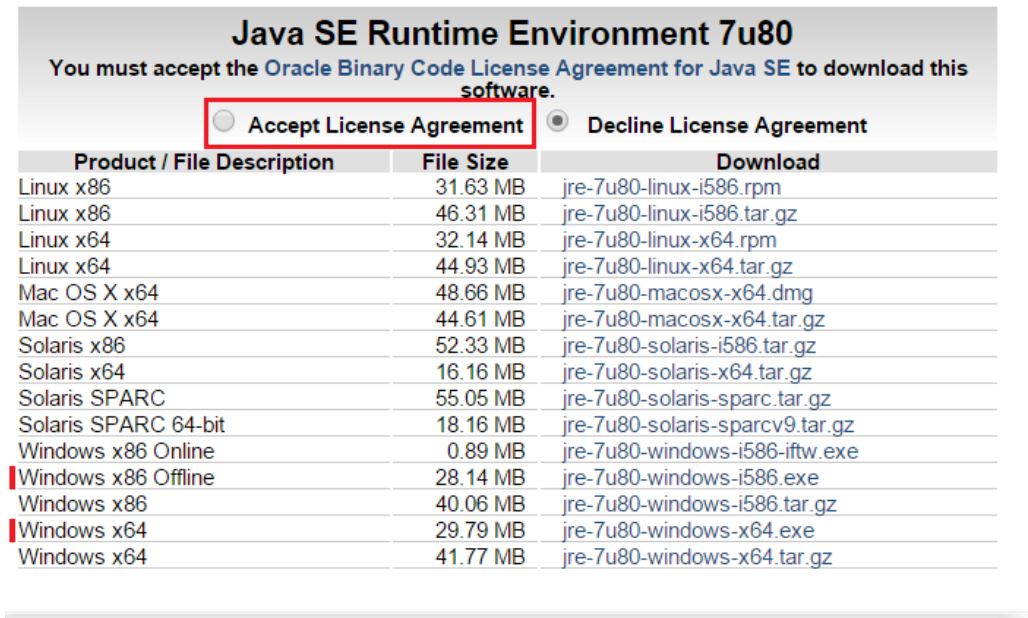
Vì **Arduino IDE** được viết trên **Java** nên bạn cần phải cài đặt **JRE** trước **Arduino IDE**.

Link tải: <http://www.oracle.com/technetwork/ja...>

Chú ý:

Nhiều bạn do không cài **JRE** trên máy nên thường hay gặp phải tình trạng không chạy được **Arduino IDE**.

2 bản **JRE** phổ biến nhất là bản dành cho Windows 32bit (x86) và Windows 64bit (x64) mình đã đánh dấu trong hình. Nhớ chọn "**Accept License Agreement**".



Hình 6.2: Bản JRE mới nhất là 7u80

2.1.2. Cài đặt Arduino IDE

Bước 1: Truy cập địa chỉ [http://arduino.cc/en/Main/Software/...](http://arduino.cc/en/Main/Software/) . Đây là nơi lưu trữ cũng như cập nhật các bản IDE của Arduino. Bấm vào mục WindowZIP file for non admin install như hình minh họa.



Hình 6.3: Download Arduino software

Bạn sẽ được chuyển đến một trang mời quyền góp tiền để phát triển phần mềm cho Arduino, tiếp tục bấm JUSTDOWNLOAD để bắt đầu tải.

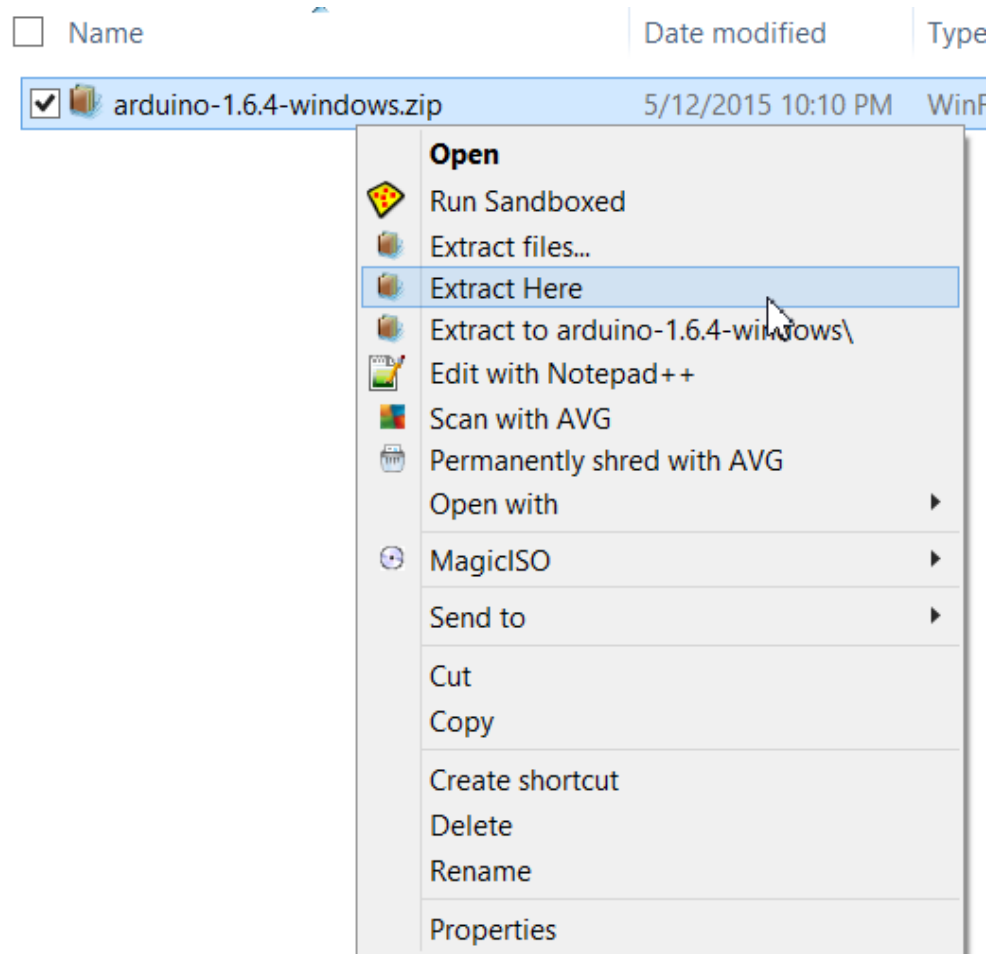
Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)




Hình 6.4: Download Arduino software

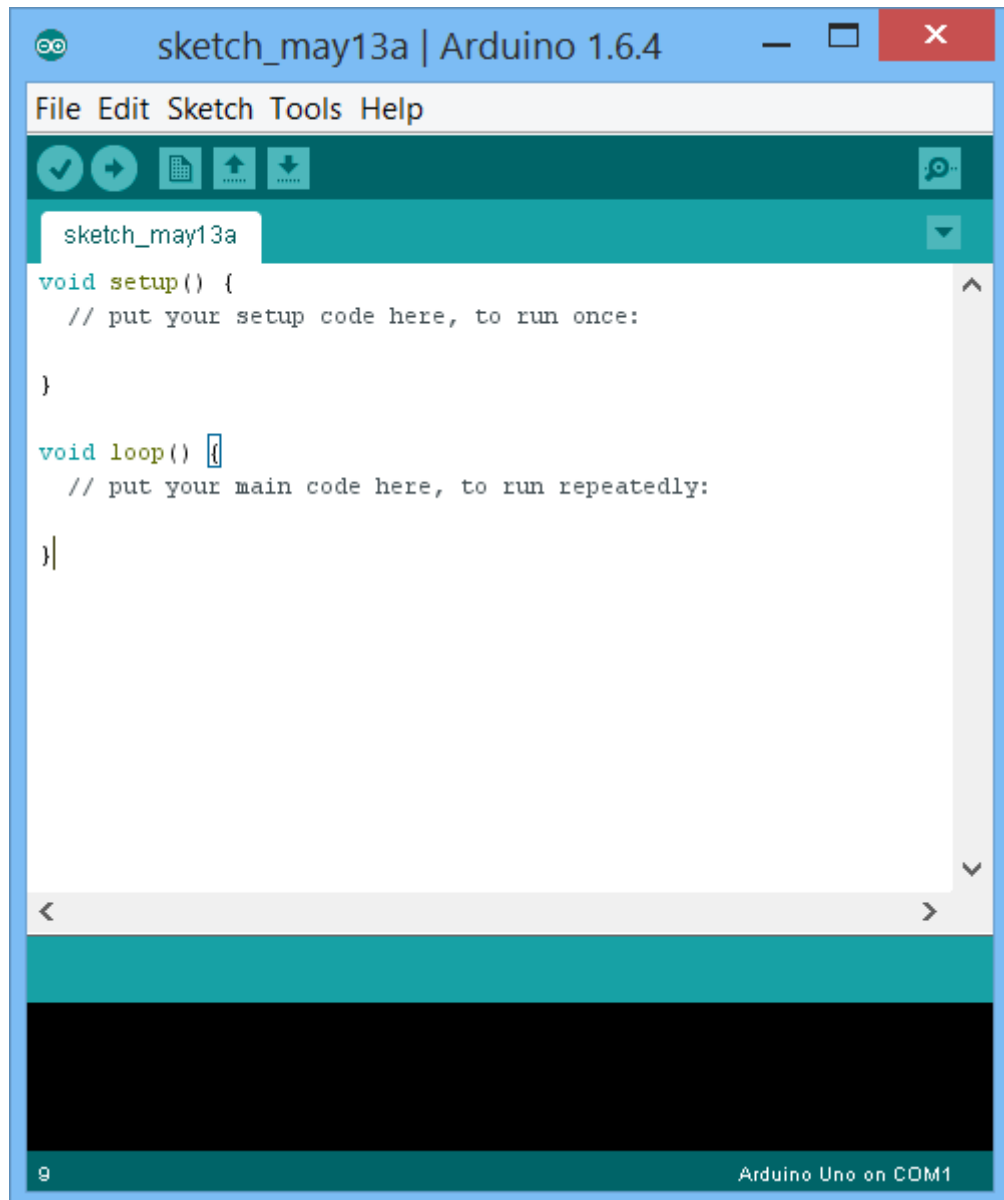
Bước 2: Sau khi download xong, các bạn bấm chuột phải vào file vừa download **arduino-1.6.4-windows.zip** và chọn “**Extract here**” để giải nén.



Hình 6.5: Giải nén chương trình

Bước 3: Copy thư mục **arduino-1.6.4** vừa giải nén đến nơi lưu trữ.

Bước 4: Chạy file  **arduino.exe** trong thư mục **arduino-1.6.4** để khởi động **Arduino IDE**



Hình 6.6: Khởi động Arduino

2.1.3. Cài đặt Driver

Để máy tính của bạn và board Arduino giao tiếp được với nhau, chúng ta cần phải cài đặt driver trước tiên.

Nếu bạn dùng Windows 8, trong một số trường hợp Windows **không cho phép** bạn cài Arduino driver (do driver không được kí bằng chữ kí số hợp lệ). Do vậy bạn cần vào Windows ở chế độ **Disable driver signature enforcement** thì mới cài được driver

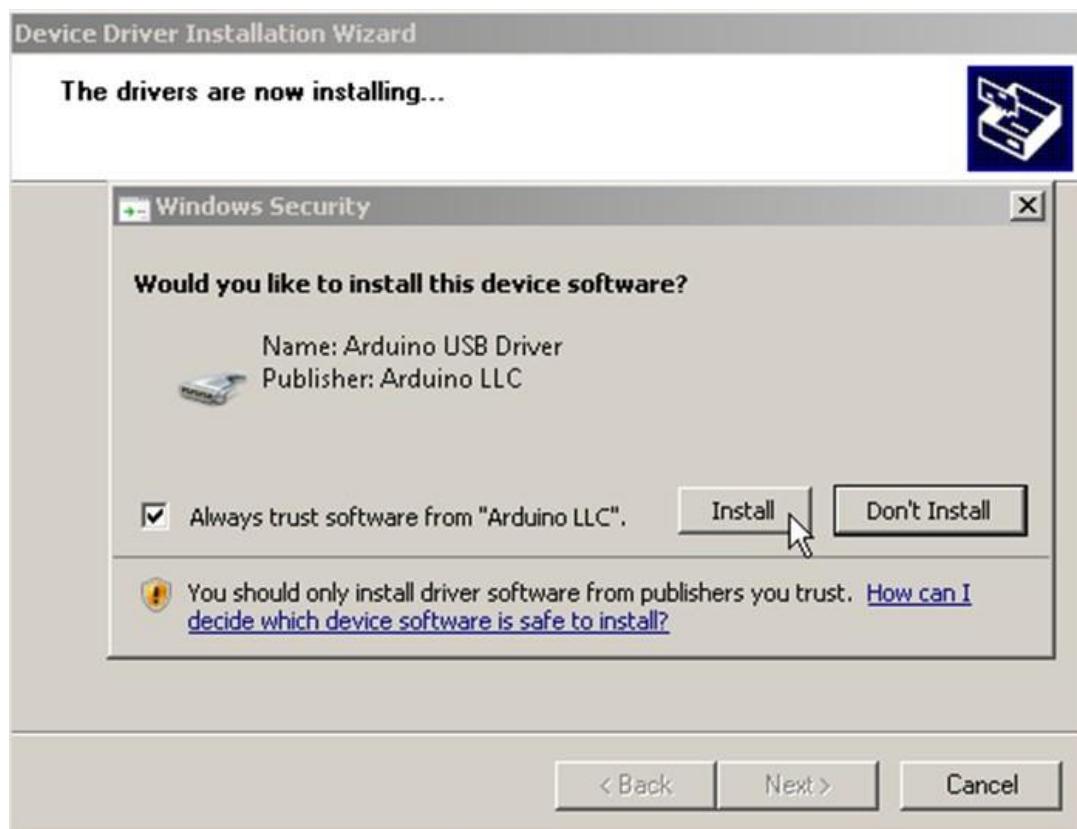
Xem hướng dẫn thực hiện tại bài viết **Disabling Driver Signature on Windows 8** của **SparkFun**.

Đầu tiên, các bạn chạy file **arduino-1.6.4\drivers\dpinst-x86.exe** (Windows x86) hoặc **arduino-1.6.4\drivers\dpinst-amd64.exe** (Windows x64). Cửa sổ “**Device Driver Installation Wizard**” hiện ra, các bạn chọn **Next** để tiếp tục.



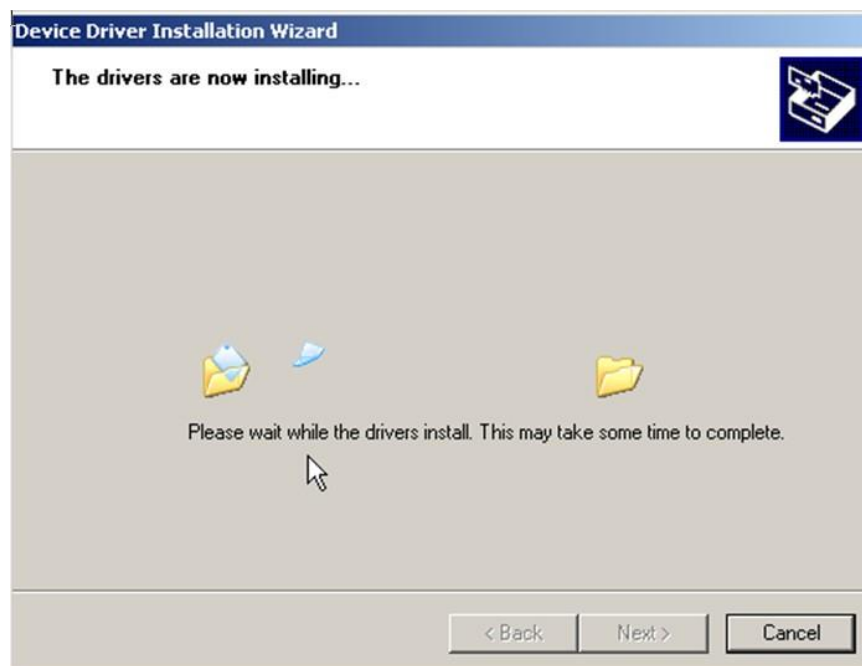
Hình 6.7: Cài đặt driver

Khi có yêu cầu xác nhận cài đặt driver, chọn “**Install**”



Hình 6.8: Cài đặt driver

Đợi khoảng 10 giây trong lúc quá trình cài đặt diễn ra ...



Hình 6.9: Cài đặt driver

Quá trình cài đặt đã hoàn tất. Bấm “Finish” để thoát.



Hình 6.10: Kết thúc cài đặt

TÀI LIỆU THAM KHẢO

[1] arduino.vn.

[2] wikipedia.

[3] Trần Thị Thúy Hà “ Nghiên cứu cấu trúc và phát triển chương trình điều khiển thông minh của các robot nhiều bậc tự do”.

[4] Phan Phú Hữu “Nghiên cứu thử nghiệm một mô hình thị giác máy tính dùng cho bám đối tượng và dẫn đường của robot di động thông minh”.