

LỜI MỞ ĐẦU

Cùng với sự phát triển không ngừng của các ngành khoa học kỹ thuật, các ngành công nghiệp cũng phát triển nhanh chóng. Việc áp dụng các máy móc hiện đại vào sản xuất là một yêu cầu không thể thiếu trong các nhà máy nhằm tăng năng suất, tăng chất lượng và giảm giá thành sản phẩm. Song song với sự phát triển đó, công nghệ chế tạo Robot cũng phát triển nhanh chóng đặc biệt là ở các nước phát triển nhằm đáp các nhu cầu về sản xuất, sinh hoạt, quốc phòng... Robot có thể thực hiện những công việc mà con người khó thực hiện và thậm chí không thực hiện được như: làm những công việc đòi hỏi độ chính xác cao, làm việc trong môi trường nguy hiểm (như lò phản ứng hạt nhân, dò phá mìn trong quân sự), thám hiểm không gian vũ trụ...

Trong các họ Robot, chúng ta không thể không nhắc tới Mobile Robot với những đặc thù riêng mà những loại Robot khác không có. Với khả năng di chuyển linh hoạt và vùng hoạt động rộng, thu hút nhiều sự đầu tư nghiên cứu hiện nay. Mobile Robot có thể phân chia thành nhiều loại theo cách vận hành (Robot di chuyển bằng chân, bánh xe, xích,...).

Để góp phần vào sự phát triển nền khoa học kỹ thuật của nước nhà, nhóm chúng em đã chọn Mobile Robot di chuyển đa hướng bằng bánh xe omni với hệ thống điều khiển thông qua Bluetooth và định vị GPS. Từ những suy nghĩ đó chúng em sử dụng những kiến thức còn hạn chế của mình để nghiên cứu chế tạo Mobile Robot trong phạm vi đồ án tốt nghiệp với ước muốn đóng góp vào công nghệ chế tạo Robot của nước nhà trong thời gian tới.

LỜI CẢM ƠN

Trước tiên, chúng em xin được gửi lời cảm ơn trân trọng và sâu sắc nhất thầy Nguyễn Mậu Tuấn Vương và thầy Nguyễn Ngọc Phi – người đã hết sức tạo điều kiện và tận tình hướng dẫn, động viên chúng em trong suốt quá trình nghiên cứu thực hiện đồ án này.

Xin trân trọng cảm ơn đến tất cả các quý thầy cô thuộc bộ môn Cơ Điện Tử trường Cao Đẳng Kỹ Thuật Cao Thắng, những người đã trang bị cho chúng em những kiến thức cơ bản, cũng như đã nhiệt tình hướng dẫn giúp đỡ chúng em trong suốt khóa học vừa qua.

Và sau cùng xin được cảm ơn đến tập thể lớp CĐ CĐT12A khóa 2012-2015, đã quan tâm giúp đỡ và tạo điều kiện thuận lợi để chúng tôi được học tập, nghiên cứu và hoàn thành đồ án tốt nghiệp này.

Một lần nữa xin cảm ơn.

Trân trọng!

TP.HCM, Ngày 10 Tháng 7 Năm 2015

Sinh viên thực hiện

Phan Minh Kha
Hò Văn Dương
Đặng Tấn Đạt
Nguyễn Hữu Chiến

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP.HCM, Ngày ... Tháng ... Năm 2015

Giảng viên hướng dẫn

(GV ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP.HCM, Ngày ... Tháng ... Năm 2015

Giảng viên phản biện

(GV ký tên và ghi rõ họ tên)

MỤC LỤC

MỤC LỤC	3
HÌNH ẢNH	3
LƯU ĐỒ.....	4
BẢN VẼ CƠ KHÍ	4
BẢNG SỐ LIỆU	4
CODE KCD2 MOBILE ROBOT.....	4
CHƯƠNG I: TỔNG QUAN.....	5
1. TỔNG QUAN VỀ MOBILE ROBOT:	5
1.1 GIỚI THIỆU:	5
1.2 LỊCH SỬ PHÁT TRIỂN:	5
2. PHÂN LOẠI MOBILE ROBOT:.....	8
3. MỘT SỐ DẠNG ĐIỀU KHIỂN:.....	8
3.1 ĐIỀU KHIỂN TỪ XA BẰNG TAY:	8
.....	9
3.2 THỰC THI THEO LỘ TRÌNH:	9
3.3 NGẪU NHIÊN HOẠT ĐỘNG ĐỘC LẬP:.....	9
.....	10
4. ỨNG DỤNG CỦA MOBILE ROBOT:.....	10
.....	10
5. GIỚI THIỆU KCD2 MOBILE ROBOT:	11
5.1 LÝ DO CHỌN ĐỀ TÀI:	11
5.2 ỨNG DỤNG CỦA ĐỀ TÀI VÀO THỰC TẾ:	12
6. HOẠT ĐỘNG CỦA KCD2 MOBILE ROBOT:.....	12
6.1 ĐIỀU KHIỂN BẰNG TAY HOẶC GIA TỐC:	12
6.2 ĐIỀU KHIỂN TỰ ĐỘNG:	13
7. ƯU NHƯỢC ĐIỂM KCD2 MOBILE ROBOT:.....	14
7.1 ƯU ĐIỂM:.....	14
7.2 NHƯỢC ĐIỂM:.....	14
CHƯƠNG II: THIẾT KẾ CƠ KHÍ	15
1. SƠ LƯỢC VỀ PHẦN MỀM THIẾT KẾ:.....	15
2. THIẾT KẾ CƠ KHÍ:	18
2.1 KHUNG CƠ KHÍ:	18
2.2 BÁNH XE ĐA HƯỚNG:	19

2.3	MỘT SỐ CHI TIẾT KHÁC:	19
CHƯƠNG III: HỆ THỐNG ĐIỆN		22
1.	THIẾT BỊ ĐIỆN:	22
1.1	ĐỘNG CƠ:	22
1.2	NGUỒN CUNG CẤP:	22
2.	THIẾT BỊ ĐIỀU KHIỂN:	23
2.1	ARDUINO MEGA 2560:	23
2.2	ARDUINO UNO R3:	25
2.3	MODULE GSM/GPS SIM908 EASY:	25
2.4	ARDUINO UNO EASY SHIELD:	26
2.5	MODULE BLUETOOTH:	28
2.6	CẢM BIẾN LA BÀN SỐ HMC5883L:	29
2.7	MẠCH ĐIỀU KHIỂN ĐỘNG CƠ L298:	30
2.8	MODULE ĐỊNH VỊ TOÀN CẦU UBLOX NEO-6M GPS:	31
CHƯƠNG IV: LẬP TRÌNH		33
1.	GIỚI THIỆU PHẦN MỀM LẬP TRÌNH:	33
1.1	NGÔN NGỮ LẬP TRÌNH CHO ARDUINO:	33
1.2	CÔNG CỤ TẠO ỨNG DỤNG TRÊN SMARTPHONE:	34
2.	LẬP TRÌNH:	35
2.1	CHƯƠNG TRÌNH KCD2 MOBILE ROBOT TRÊN APP INVENTOR:	35
2.2	KCD2 MOBILE ROBOT CODE:	38
CHƯƠNG V: KẾT LUẬN		41
1.	KẾT QUẢ ĐẠT ĐƯỢC:	41
1.1	VỀ MẶT LÝ THUYẾT:	41
1.2	VỀ MẶT THỰC NGHIỆM:	41
2.	HẠN CHẾ:	41
3.	HƯỚNG PHÁT TRIỂN:	41

MỤC LỤC

HÌNH ẢNH

Hình 1.1: iRobot's PackBot và Military-Robot	9
Hình 1.2: Máy hút bụi Roomba và Máy cắt cỏ Friendly Robotics	10
Hình 1.3: Robot Sojourner	10
Hình 1.4: Robot Pioneer và Robot MBARI's ALTEX AUV	11
Hình 1.5: Giao diện KCD2 MOBILE ROBOT	13
Hình 1.6: Hướng chạy của KCD2 MOBILE ROBOT.	14
Hình 2.1: Giao diện chính của solidworks 2013	18
Hình 2.2: Mica được thiết kế	19
Hình 2.3: Bánh xe omni	19
Hình 2.4: Part gá động cơ	20
Hình 2.5: Bulông đai ốc	20
Hình 2.6: Part bên	21
Hình 3.1: Sơ đồ chân của Arduino Mega 2560	24
Hình 3.2: Sơ đồ chân của Arduino Uno R3	25
Hình 3.3 Module GSM/GPS SIM908 Easy	25
Hình 3.4: Arduino UNO Easy Shield	27
Hình 3.5: Module bluetooth HC-05	29
Hình 3.6: Cảm biến la bàn HMC5883L	30
Hình 3.7: Mạch điều khiển động cơ L298	31
Hình 3.8: Module GPS NEO-6M	32
Hình 4.1: Giao diện Arduino IDE	33
Hình 4.2: Giao diện App Inventor	37
Hình 4.3: Giao diện phần Blocks (code)	37

LƯU ĐỒ

Hình 4.4: Lưu đồ giải thuật Robot có điều khiển	38
Hình 4.5: Lưu đồ giải thuật Robot tự động	39
Hình 4.6: Sơ đồ khối Module SIM908	40

BẢN VẼ CƠ KHÍ

Hình 1: Chi tiết đế Robot.....	42
Hình 2: Chi tiết khung Robot.....	43
Hình 3: Chi tiết Part gá động cơ.....	44
Hình 4: Chi tiết Part bên	45
Hình 5: Chi tiết trục đồng	46
Hình 6: Bản vẽ lắp chi tiết KCD2 Mobile Robot	47

BẢNG SỐ LIỆU

Bảng 1: Chú thích các chân của atmega 2560	65
--	----

CODE KCD2 MOBILE ROBOT

Robot có điều khiển.....	48
Robot tự động.....	54

CHƯƠNG I: TỔNG QUAN

1. TỔNG QUAN VỀ MOBILE ROBOT:

1.1 GIỚI THIỆU:

Thuật ngữ Robot xuất hiện lần đầu tiên vào năm 1922 trong tác phẩm “Rossum’s Universal Robot” của Karel Capek. Trong tác phẩm nhân vật Rossum và con trai đã tạo ra chiếc máy giống con người để phục vụ cho con người.

Mobile Robot là loại máy tự động có khả năng di chuyển trong một số môi trường nhất định. Mobile Robot hiện đang được các nhà nghiên cứu chú trọng và hầu như các trường đại học lớn đều có phòng thí nghiệm để tập trung nghiên cứu loại Robot này. Chúng cũng được sử dụng trong các ngành công nghiệp, quân sự và sinh hoạt..., như trong sinh hoạt có Robot hút bụi, lau nhà hay cắt cỏ....

1.2 LỊCH SỬ PHÁT TRIỂN:

Hình dạng Robot xuất hiện đầu tiên ở nước Hoa Kỳ, là loại tay máy chép hình dung trong phòng thí nghiệm vật liệu phóng xạ. Vào những năm 50 của thế kỷ trước, bên cạnh các loại tay máy chép hình cơ khí, các loại tay máy chép hình thủy lực điện tử đã xuất hiện. Tuy nhiên, các tay máy thương mại đều có chung nhược điểm là thiếu sự di động, các tay máy này hoạt động hạn chế quanh vị trí của nó. Ngược lại, Mobile Robot là loại Robot di động có thể di chuyển từ không gian này đến không gian khác một cách độc lập hay được điều khiển từ xa, do đó tạo không gian hoạt động lớn.

Từ năm 1939 đến năm 1945: Trong cuộc chiến thế giới lần thứ II, những con Robot di động đầu tiên xuất hiện. Nó là kết quả của những thành tựu công nghệ trong những lĩnh vực nghiên cứu mới có liên quan như khoa học máy tính và điều khiển học, hầu hết chúng là những quả bom bay, ví dụ như những quả bom chỉ nổ trong những dãy mục tiêu nhất định, sử dụng trong hệ thống hướng dẫn radar điều khiển. Tên lửa V₁ và V₂ có “phi công tự động” và hệ thống phát nổ, chúng là tiền thân của đầu đạn hạt nhân tự điều khiển hiện đại.

Từ năm 1948 đến 1948: W.Gray Walter tạo nên Elmer và Elsie, hai con Robot trông giống con đười ươi. Về mặt hành chính, chúng được gọi là Machina Speculatrix bởi vì những con Robot này hoạt động trong môi trường như những chú chim đười ươi. Elmer và Elsie được trang bị một bộ cảm biến sáng. Nếu chúng nhận ra một nguồn sáng, chúng sẽ di chuyển về phía nguồn sáng. Chúng có thể tránh hoặc chuyển hướng tránh ngại trên đường di chuyển. Những con Robot này chứng minh rằng những cử chỉ phức tạp có thể phát sinh từ một thiết kế đơn giản. Elmer và Elsie chỉ được thiết kế tương đương hai tế bào thần kinh.

Từ năm 1961 đến 1963: Trường đại học Johns Hopkins phát triển “Beast”. Beast sử dụng hệ thống định vị di chuyển xung quanh. Khi pin yếu nó sẽ tự tìm ổ cắm điện và cắm vào.

Năm 1969: Mowbot là con Robot đầu tiên cắt cỏ bãi cỏ một cách tự động. The Stanford Cart line follower là một con Robot di động có thể di chuyển thông qua nhận dạng đường kẻ trắng, sử dụng một camera để nhìn. Nó bao gồm một “kênh truyền thanh” gắn với hệ thống máy tính lớn để tạo ra những tính toán.

Năm 1970: Cùng thời điểm 1969-1972, viện nghiên cứu Stanford đang xây dựng và nghiên cứu ra Shakey. Shakey có một camera, một dây kính găng, một bộ cảm biến và một bộ truyền thanh. Shakey là con Robot đầu tiên lý giải về những chuyển động của nó. Điều này có nghĩa là Shakey có thể đưa nhiều mệnh lệnh chung và Robot này sẽ tính toán những bước cần thiết để hoàn thiện nhiệm vụ được giao.

Năm 1976: trong chương trình Vikiry, tổ chức NASA đã phóng hai tàu vũ trụ không người lái lên sao hỏa.

Năm 1977: Bộ phim “Chiến tranh giữa các vì sao” phần I, A new Hope mô tả R2D2, một con Robot di động hoạt động độc lập và C3P0, một con robot hình người. Họ đã khiến công chúng biết đến những con Robot.

Năm 1980: Thị hiếu của người tiêu dùng về Robot tăng, robot được bày bán và mua về để sử dụng trong nhà. Ví dụ RB5X vẫn tồn tại tới ngày nay và một loạt mẫu robot HERO. Robot The Stanford Cart được phát triển mạnh, nó đã có thể lái tàu biển vượt qua những trở ngại và tạo lên bản đồ những nơi nó đi qua.

Năm 1989: Mark Tinden phát minh ra BEAM robotics.

Năm 1990: Cha đẻ của nền robot công nghiệp Joseph Engelberger làm việc với các đồng nghiệp và đã phát minh ra những con robot tự động trong ngành y tế và được bán bởi Helpmate. Sở an ninh Mỹ gây quỹ cho dự án MDARS-I dựa vào robot bảo vệ trong nhà Cybermotion.

Năm 1993-1994: Dante-I và Dante-II được phát triển bởi trường đại học Carnegie Mellon, cả hai con Robot dùng để thám hiểm núi lửa đang hoạt động.

Năm 1995: Robot di động có thể lập trình Pioneer (người tiên phong) được bán sẵn ở một mức giá chấp nhận được, điều đó dẫn tới sự gia tăng rộng rãi về nghiên cứu Robot và các trường đại học nghiên cứu về Robot trong suốt các thập kỷ sau. Robot di động trở thành một phần không thể thiếu trong chương trình giảng dạy của các trường đại học.

Năm 1996-1997: NASA phóng con tàu Mars Pathfinder có 2 Robot Rover và Sojourner lên sao Hỏa. The Rover thám hiểm bề mặt sao hỏa được điều khiển từ mặt đất. Sojourner được trang bị với một hệ thống tránh rủi ro cao. Hệ thống này làm cho Sojourner có thể tìm thấy đường đi của nó một cách độc lập trên địa hình của sao Hỏa.

Năm 1999: Sony giới thiệu Aibo, một con robot có khả năng đi lại, quan sát và tác động qua lại tới môi trường. Robot điều khiển từ xa dùng cho quân sự PackBot cũng được giới thiệu.

Năm 2001: Dự án Swaim-Bots, Swaim-Bots giống những bầy côn trùng được khởi động. Chúng bao gồm một số lượng lớn các con Robot đơn lẻ, có thể tác động lẫn nhau và cùng nhau thực hiện những nhiệm vụ phức tạp.

Năm 2002: Roomaba, một con robot di động dùng trong gia đình, thực hiện công việc lau nhà xuất hiện. Tiếp tục phát triển hiện nay có rất nhiều loại Robot phục vụ cho con người dần xuất hiện ngày càng thân thiện hơn.

Năm 2004: Robosapien, một con robot đồ chơi, thiết kế bởi Mark Tilden được bán sẵn. Trong dự án “The Centibots Project” 100 con Robot cùng làm việc với nhau để tạo lên một bản đồ cho một vùng không xác định và tìm những vật thể

trong môi trường đó. Trong cuộc thi đầu tiên DARPA Grand Challenge, các con Robot tự động đã cùng nhau tranh tài cùng nhau trên sa mạc.

Năm 2006: Sony dừng việc sản xuất Aibo và Helpmate. PatrolBot trở lên phổ biến khi các Robot di động vẫn tiếp tục cạnh tranh nhau để trở thành mặt hàng độc quyền. Sở an ninh Mỹ đã bỏ dự án MDARS-I, nhưng lại gây quỹ cho dự án MDARS-E một loại Robot an ninh tự động khác. TALON-Sword, một loại Robot tự động dùng để bán sẵn với dàn phóng lựu đạn và những sự lựa chọn về vũ khí hợp thành khác đã ra đời. Asimo của Honda biết cách chạy và leo cầu thang chỉ với hai chân như con người.

Năm 2007: Hệ thống KiVa, Robot thông minh tăng nhanh về số lượng trong quy trình phân phối, những Robot thông minh này được phân loại theo mức độ phổ biến những nội dung của chúng. Robot Tug trở thành phương tiện phổ biến trong các bệnh viện dùng để vận chuyển đồ trong kho từ nơi này sang nơi khác. ARCSinside Speci-Minder mang máu và các vật mẫu từ trạm y tá tới phòng xét nghiệm. Seekur, Robot dịch vụ dùng ngoài trời với mục đích phi quân sự có thể kéo một xe qua một bãi đậu xe, lái một cách độc lập (tự động) vào trong nhà và bắt đầu học cách lái ra ngoài. Trong khi đó, PatrolBot học cách theo sau con người và nếu cửa mà mở thì đóng lại.

2. PHÂN LOẠI MOBILE ROBOT:

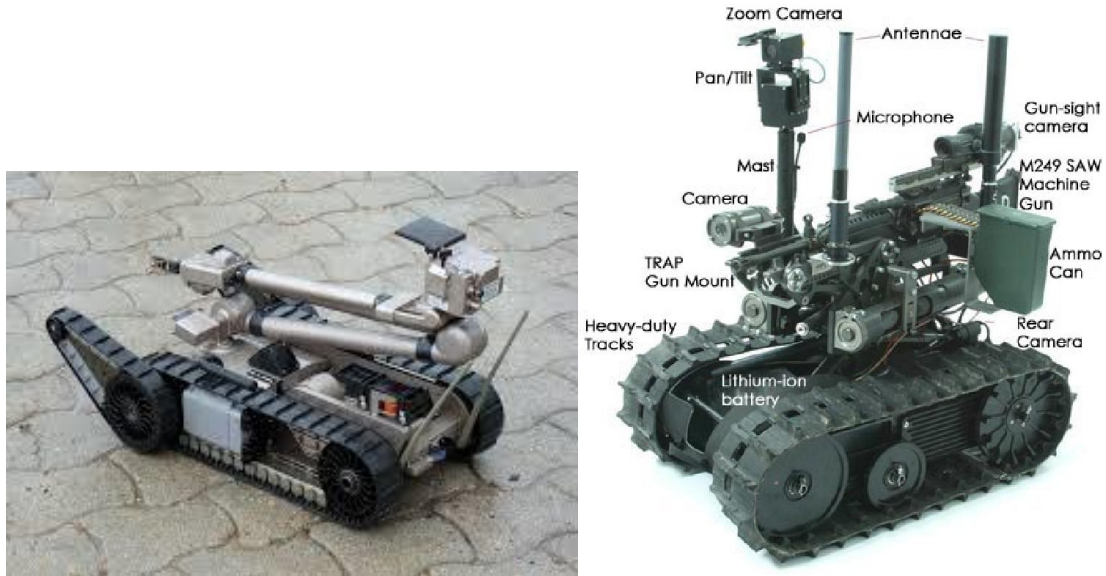
Có thể phân loại Mobile Robot theo phương pháp di chuyển:

- Robot có chân, chân giống người hay động vật.
- Robot di chuyển bằng xích, đai...
- Robot di chuyển bằng bánh xe.

3. MỘT SỐ DẠNG ĐIỀU KHIỂN:

3.1 ĐIỀU KHIỂN TỪ XA BẰNG TAY:

Robot điều khiển từ xa bằng tay được điều khiển thông qua sóng RF, wifi, hồng ngoại hay Bluetooth... Robot điều khiển từ xa giúp con người tránh khỏi những nguy hiểm. Ví dụ Robot điều khiển từ xa bằng tay gồm có: Military-Robot và iRobot's PackBot,...



Hình 1.1: iRobot's PackBot và Military-Robot

3.2 THỰC THI THEO LỘ TRÌNH:

Một vài Robot tự động đầu tiên là những con Robot theo lộ trình. Chúng có thể theo những đường được sơn khắc trên sàn trên trần nhà hay một dây điện. Đa số những Robots này hoạt động theo một thuật toán đơn giản là giữ lộ trình trong bộ cảm biến trung tâm, chúng không thể vòng qua các chướng ngại vật, chúng chỉ dừng lại khi có vật nào đó cản đường chúng. Rất nhiều mẫu của loại Robot này vẫn được bán bởi FMC, Egemin, HK sytem và một vài công ty khác.

3.3 NGẪU NHIÊN HOẠT ĐỘNG ĐỘC LẬP:

Robot hoạt động độc lập với những chuyển động ngẫu nhiên, về cơ bản đó là những chuyển động nhảy bật lên tường, những bức tường được cảm nhận do sự cản trở về mặt vật lý như máy hút bụi Roomba hoặc với bộ cảm biến điện tử của máy cắt cỏ Friendly Robotics.



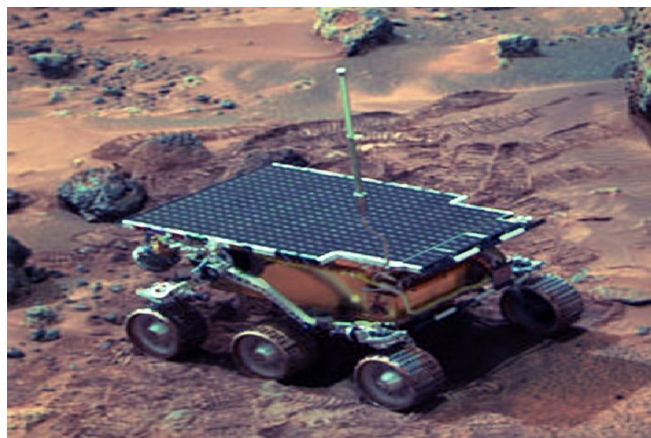
Hình 1.2: Máy hút bụi Roomba và Friendly

4. ỨNG DỤNG CỦA MOBILE ROBOT:

Mobile Robot được dùng phổ biến trong những môi trường độc hại, những nơi con người không thể đi tới hay đi tới một cách khó khăn và nguy hiểm, Mobile Robot cũng được dùng trong lĩnh vực giải trí và phục vụ đời sống.

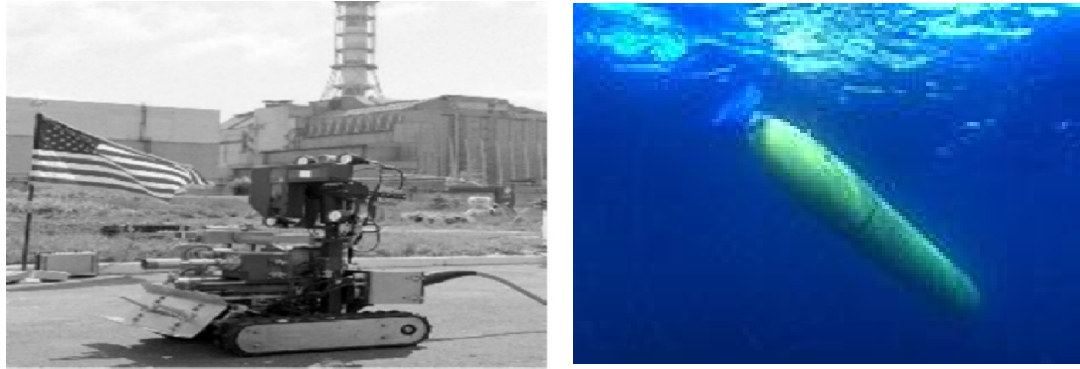
Các nơi con người không có khả năng đến được như sao Hỏa, đáy biển, núi lửa,...người ta phải sử dụng Robot tự hành với cấu trúc phù hợp với môi trường. Ví dụ như:

Robot Sojourner trong nhiệm vụ tìm kiếm sự sống trên sao hỏa năm 1997.



Hình 1.3: Robot Sojourner

Robot MBARI's ALTEX AUV là robot hoạt động dưới đáy biển với nhiệm vụ thăm dò độ phóng xạ, áp suất, độ sâu,... và Robot Pioneer được thiết kế để dò tìm và kiểm tra nồng độ trong thảm họa Chernobyl.



Hình 1.4: Robot Pioneer và Robot MBARI's ALTEX AUV

Robot sử dụng trong quân sự như: Military-Robot

Robot sử dụng trong sinh hoạt như: Robot lau nhà Robotking...

5. GIỚI THIỆU KCD2 MOBILE ROBOT:

5.1 LÝ DO CHỌN ĐỀ TÀI:

Ở nước ta cũng đã có rất nhiều đề tài nói về Robot di động như là Robot tránh vật cản, Robot leo tường, Robot leo cầu thang... Nhưng chúng em chưa tìm thấy đề tài nào nói về Robot di chuyển linh hoạt đa hướng cũng như là áp dụng những kỹ thuật tiên tiến như GPS, La bàn, Bluetooth... Nếu có về tài liệu thì cũng chỉ có một số ít tài liệu khái quát, chưa có tài liệu kỹ thuật chi tiết về Robot trên. Hiện tại, Robot chỉ được định hướng bằng công nghệ xử lý ảnh, đường dẫn (line), bằng la bàn, GPS, Bluetooth. Công nghệ xử lý ảnh cũng còn khá mới mẻ và công nghệ này chỉ là để phát hiện đối tượng chứ chưa phải là cách để định vị và định hướng cho robot. Dùng line thì Robot chỉ hoạt động trong một phạm vi nhỏ

hợp và chỉ thích hợp trong các cuộc thi Robot. La bàn điện tử là thiết bị dùng để định hướng cho Robot thông qua cảm biến từ trường Trái Đất, còn GPS thì chưa phổ biến.

Do đó, định vị và định hướng cho Robot là một việc rất cần thiết và cấp bách góp phần vào sự phát triển Robot của Việt Nam. Vì thế việc thiết kế một “KCD2 MOBILE ROBOT” sử dụng la bàn điện tử kết hợp với định vị GPS và Bluetooth là vấn đề mà chúng em cần phải nghiên cứu chế tạo trong đề tài này.

5.2 ỨNG DỤNG CỦA ĐỀ TÀI VÀO THỰC TẾ:

KCD2 MOBILE ROBOT thuộc họ Robot Mobile nên có những ứng dụng gần giống nhau vào thực tiễn, Robot được chế tạo và nghiên cứu nhằm vào các lĩnh vực quân sự, dân dụng...

- Ở lĩnh vực quân sự: Việc sử dụng Robot này để dò thám, thám hiểm, tháo bom hay đánh bom phòng vệ, xác định vị trí kẻ thù,...
- Ở lĩnh vực dân dụng: Có thể dùng Robot trong một số công việc như giám sát, định vị vị trí, dẫn đường, tìm thiết bị, làm việc nhà,...

6. HOẠT ĐỘNG CỦA KCD2 MOBILE ROBOT:

KCD2 MOBILE ROBOT hoạt động chủ yếu 2 chức năng: điều khiển bằng tay thông qua Bluetooth, chạy tự động thông qua hệ thống định vị GPS.

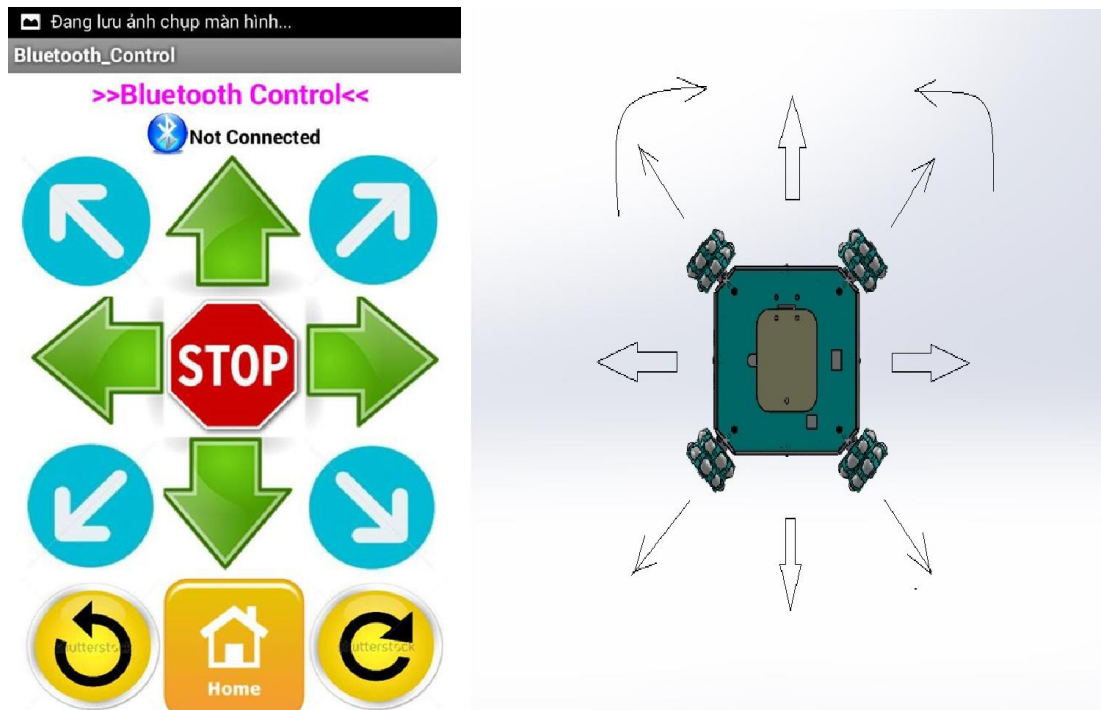
6.1 ĐIỀU KHIỂN BẰNG TAY HOẶC GIA TỐC:

KCD2 MOBILE ROBOT được trang bị bởi 4 bánh omni nên có thể chạy theo mọi hướng mà ta điều khiển gồm:

- Chạy tới, chạy lui, chạy trái, chạy phải.
- Chạy lệch góc 45^0 sang trái hoặc sang phải.
- Chạy xoay vòng trái, chạy xoay vòng phải.

Điều khiển phương hướng chạy bởi 1 chiếc Smartphone Android được cài đặt chương trình KCD2 MOBILE ROBOT kết nối thông qua Bluetooth. Khi ta mở Smartphone Android và chạy chương trình KCD2 MOBILE ROBOT lên giao diện của chương

trình như hình bên dưới. Tiếp theo chúng ta bật Bluetooth kết nối chúng và tiến hành điều khiển KCD2 MOBILE ROBOT. Điều khiển Robot rất đơn giản nếu muốn Robot chạy tới, lùi, rẽ trái, rẽ phải hoặc xoay.



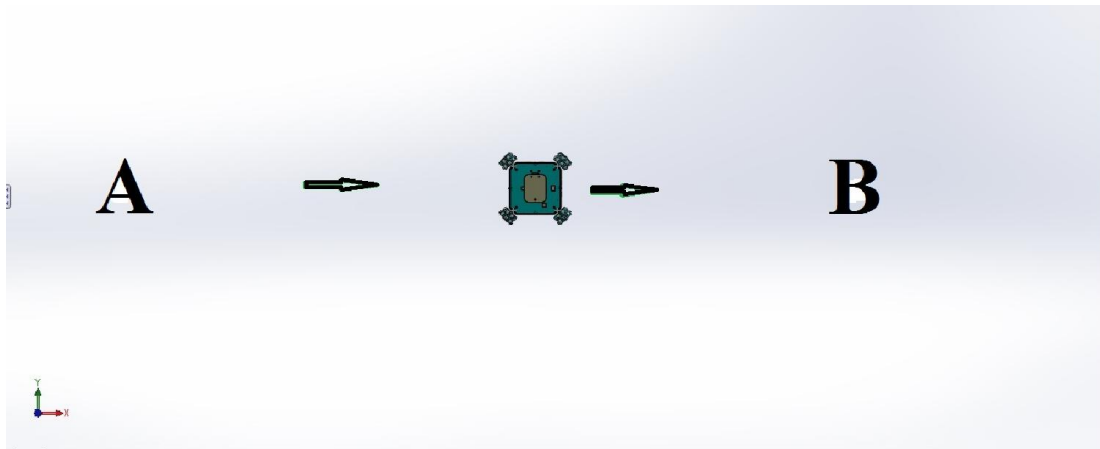
Hình 1.5: Giao diện KCD2 MOBILE ROBOT.

Đối với điều khiển bằng tay chỉ cần bấm vào những mũi tên ở trên giao diện thì Robot sẽ chạy theo ý mình muốn.

Đối với điều khiển bằng gia tốc thì chỉ cần nghiêng Smartphone qua trái, phải, tới, lùi hoặc là nghiêng xéo thì Robot sẽ chạy như các hướng mình nghiêng.

6.2 ĐIỀU KHIỂN TỰ ĐỘNG:

Việc đầu tiên thực hiện ở điều khiển tự động là lấy tọa độ của điểm cần cài đặt bằng cách gửi tin nhắn SMS đến Module GSM/GPS SIM908 Easy. Sau khi đã lấy được tọa độ, ta sẽ nhúng tọa độ vào code trên phần mềm Arduino IDE rồi đổ code vào vi điều khiển Arduino Mega 2560. KCD2 MOBILE ROBOT sẽ chạy tự động từ điểm A tới điểm B mà ta đã lấy tọa độ thông qua hệ thống GPS.



Hình 1.4: Hướng chạy của KCD2 MOBILE ROBOT.

7. ƯU NHƯỢC ĐIỂM KCD2 MOBILE ROBOT:

7.1 ƯU ĐIỂM:

- Di chuyển linh hoạt.
- Dễ điều khiển.
- Thiết kế đơn giản.
- Có khả năng ứng dụng ngay vào đời sống.

7.2 NHƯỢC ĐIỂM:

- Di chuyển trong một số môi trường nhất định.
 - Hướng di chuyển còn sai lệch.
 - Sai lệch vị trí khi định vị trên hệ thống GPS.
-

CHƯƠNG II: THIẾT KẾ CƠ KHÍ

1. SƠ LƯỢC VỀ PHẦN MỀM THIẾT KẾ:

Mobile robot được thiết kế trên phần mềm solidworks 2013.

Solidworks là phần mềm thiết kế cơ khí 3D, chạy trên hệ điều hành Windows. Solidworks được phát triển bởi Dassault Systèmes Solidworks Corp, một công ty con của Dassault Systèmes, SA (Vélizy, Pháp). Solidworks hiện đang được sử dụng bởi hơn 2 triệu kỹ sư và nhà thiết kế ở mức hơn 178.000 công ty trên toàn thế giới.

Solidworks đang là đối thủ cạnh tranh trực tiếp với Autodesk Inventor và Solid Edge. Về lịch sử: Tổng công ty Solidworks được thành lập vào tháng 12/1993 bởi Viện Công nghệ Massachusetts của Jon Hirschtick. Hirschtick sử dụng số tiền \$1.000.000 của ông kiếm được khi là thành viên của nhóm MIT Blackjack để thành lập công ty. Ban đầu công ty có trụ sở tại Waltham, Massachusetts, Hoa Kỳ, Hirschtick tuyển dụng một đội ngũ kỹ sư với mục tiêu xây dựng phần mềm CAD 3D mà dễ sử dụng, giá cả phải chăng, và chạy trên nền Windows. Hoạt động sau từ Concord, Massachusetts, Solidworks phát hành sản phẩm đầu tiên của mình Solidworks 95 vào năm 1995. Năm 1997, Dassault nổi tiếng với phần mềm CATIA CAD, mua lại Solidworks với giá \$ 310.000.000.

Solidworks đang tiếp thị một số phiên bản của phần mềm Solidworks CAD ngoài eDrawings-một công cụ cộng tác và DraftSight-một sản phẩm 2D CAD.

Solidworks đã được lãnh đạo bởi John McEleney từ năm 2001 đến tháng 7 năm 2007 và Jeff Ray từ năm 2007 đến tháng 1 năm 2011. Giám đốc điều hành hiện nay là Bertrand Sicot.

Các phiên bản đã phát hành như: SolidWorks 95, SolidWorks 96, SolidWorks 97, SolidWorks 97 Plus, SolidWorks 98, SolidWorks 98 Plus,..., SolidWorks 2013,...

Các sản phẩm solidworks cung cấp:

- Ứng dụng thiết kế cơ khí 3D: SolidWorks Standar, SolidWorks Professional, SolidWorks Premium cung cấp một bộ công cụ phát
-

triển sản phẩm thiết kế, thẩm tra thiết kế, quản lý dữ liệu, thông tin liên lạc và các công cụ cơ khí. SolidWorks Premium bao gồm tất cả các khả năng của SolidWorks chuyên nghiệp cũng như định tuyến và các công cụ phân tích, bao gồm SolidWorks Routing, SolidWorks Simulation, và SolidWorks Motion. SolidWorks Education Edition cung cấp các chức năng thiết kế tương tự nhưng được cấu hình và đóng gói cho kỹ thuật và sinh viên thiết kế công nghiệp.

- Thiết kế các công cụ chứng nhận: SolidWorks Simulation là một công cụ xác nhận thiết kế cho thấy các kỹ sư cách thiết kế của họ sẽ cư xử như các đối tượng vật lý. SolidWorks Motion là một công cụ tạo mẫu ảo cung cấp khả năng mô phỏng chuyển động để đảm bảo chức năng thiết kế đúng cách. SolidWorks Flow Simulation là một công cụ kiểm tra mô phỏng chất lỏng dòng chảy bên trong và bên ngoài và phân tích nhiệt để các nhà thiết kế có thể tiến hành thử nghiệm trên nguyên mẫu ảo. SolidWorks Simulation Premium là một phân tích phần tử hữu hạn (FEA) công cụ xác nhận thiết kế có thể xử lý một số mô phỏng đa môi trường cũng như các vật liệu phi tuyến. SolidWorks phát triển bền vững là một sản phẩm mà các biện pháp tác động môi trường của thiết kế trong khi họ được mô phỏng trong SolidWorks.
 - Sản phẩm công cụ quản lý dữ liệu: SolidWorks Workgroup PDM là một công cụ PDM cho phép người dùng SolidWorks hoạt động trong nhóm 10 thành viên hoặc ít hơn để làm việc trên các thiết kế đồng thời. Với SolidWorks PDM nhóm làm việc, các nhà thiết kế có thể tìm kiếm, chỉnh sửa, và dữ liệu ngăn CAD trong khi duy trì một lịch sử thiết kế chính xác. SolidWorks Enterprise PDM là một công cụ PDM cho phép người dùng SolidWorks hoạt động theo nhóm tại các cơ sở riêng biệt khác nhau để làm việc trên các thiết kế đồng thời. Với SolidWorks Doanh nghiệp PDM, các nhà thiết kế có thể tìm kiếm,
-

chỉnh sửa, và dữ liệu ngăn CAD trong khi duy trì một lịch sử thiết kế chính xác. Doanh nghiệp PDM duy trì một đường mòn kiểm toán, tương thích với một loạt các gói CAE (Autodesk, Siemens, PTC, Catia, vv) để duy trì quan hệ interfile, và sẽ quản lý các phiên bản của bất kỳ tài liệu lưu trong khu vực lưu trữ chung. PDM doanh nghiệp cũng sử dụng một sơ đồ quy trình làm việc để tự động thông báo cho thành viên trong nhóm khi một dự án di chuyển từ một giai đoạn đến, cũng như ý kiến theo dõi tiếp theo. PDM doanh nghiệp có khả năng giao tiếp với các hệ thống MRP / ERP và có thể được sử dụng trực tuyến để giao tiếp với khách hàng và chuỗi cung ứng.

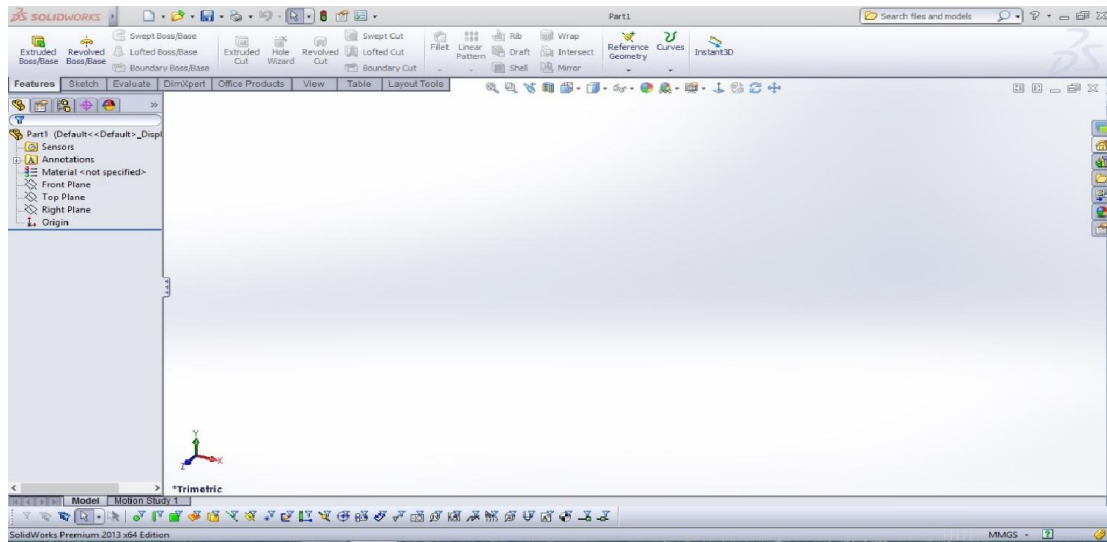
- Công cụ thiết kế đặc biệt: SolidWorks Routing, SolidWorks Electrical, SolidWorks Plastics là một công cụ xác nhận thiết kế khuôn mẫu đã được xây dựng thành một môi trường mô hình rắn. Nó cho phép các nhà thiết kế khuôn để nhanh chóng và dễ dàng xác minh xem một phần đúc nhựa có thể được lắp đầy. SolidWorks MoldBase là một danh mục các hội đồng cơ sở khuôn tiêu chuẩn và các thành phần. Các gói phần mềm cho phép các nhà thiết kế để tạo ra một cơ sở khuôn lắp ráp hoàn chỉnh. Print3D là một tính năng in 3D cho phép người dùng chuyển đổi mô hình CAD 3D của họ vào một tập tin STL. Và sau đó có nó được gửi đến các nhà sản xuất đặc biệt cho báo giá. Các tập tin STL. Có thể được sử dụng để tạo ra một liên kết ngay lập tức trích dẫn bằng cách sử dụng công nghệ Quickquote.

Solidworks với tính mở và tính tương thích của mình, Solidworks cho phép nhiều phần mềm ứng dụng khác chạy trực tiếp trên môi trường của nó, Solidworks cũng kết xuất ra các tập tin dữ liệu định dạng chuẩn để người sử dụng có thể khai thác trong môi trường các phần mềm khác.

Những nhà sản xuất, họ không những sử dụng máy CNC mà còn sử dụng nó với các phần mềm CAD/CAM như Pro-Engineer, Cimatron, MasterCam... là những phần mềm khá nổi tiếng đang được sử dụng ở nước ta. Những phần mềm như

Cimatron, MasterCam có tính năng CAM rất tốt nhưng lại có tính năng thiết kế 3D không mạnh như Pro-Engineer.

Nhưng theo phản ánh của các kỹ sư thì giao diện của Pro-Engineer làm chậm quá trình thiết kế và thậm chí làm rối các nhà thiết kế vì sự xuất hiện quá nhiều cửa sổ. Chính vì vậy các nhà sản xuất trên thế giới đang dần chuyển qua phần mềm Solidworks (Edited by congtd).



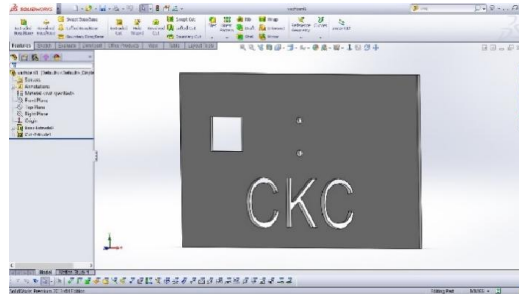
Hình 2.1: Giao diện chính của solidworks 2013

2. THIẾT KẾ CƠ KHÍ:

2.1 KHUNG CƠ KHÍ:

Hoạt động của Robot chủ yếu ở những nơi có điều kiện nhiệt độ, độ ẩm, địa hình thích hợp nên chọn mica làm phần khung Robot. Mica là một loại nhựa PPC tổng hợp, do đặc tính bền, dẻo, dễ sử dụng của mica mà ta có thể dùng dao cắt hay các máy cắt cnc (máy tự động có đầu cắt là hợp kim cứng) hoặc các máy laser (đầu

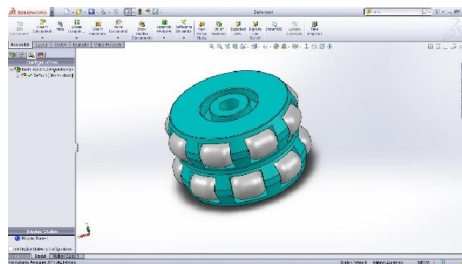
cắt sử dụng tia laser) để cắt mica theo thiết kế. Thông số của mica chọn làm khung như sau: bề dày 3mm đến 5mm, màu sắc mica trắng đục và xanh dương.



Hình 2.2: Mica được thiết kế

2.2 BÁNH XE ĐA HƯỚNG:

Để đạt được tính linh hoạt trong di chuyển thì chúng em chọn bánh xe đa hướng với kích thước màu sắc đã chọn trước. Bánh xe omni với đường kính ngoài là 49,2 mm, được làm từ nhựa PA, có khối lượng 50 gam, thể tích 106 cm³, có thể chịu được tải trọng lớn và di chuyển linh hoạt trong các môi trường khác nhau.

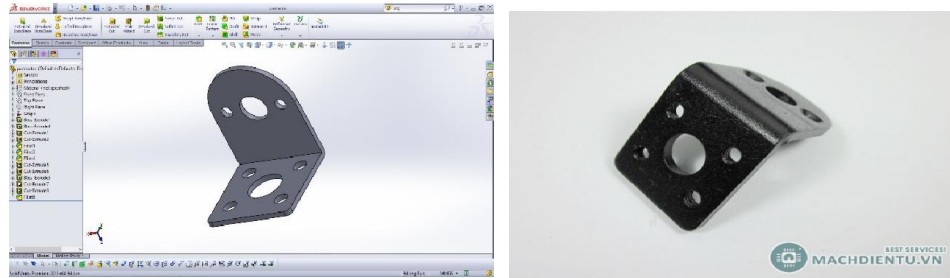


Hình 2.3: Bánh xe omni

2.3 MỘT SỐ CHI TIẾT KHÁC:

- **Part gá động cơ đường kính 25mm:**

Part gá động cơ được gia công với độ chính xác cao, khó biến dạng. Với chất liệu hợp kim nhôm cùng với độ dày 3mm được xử lý bề mặt, chống ăn mòn giúp cho việc gá đặt động cơ trở nên cứng cáp hơn.



Hình 2.4: Part gá động cơ

- **Bulông đai ốc:**

Bulông và đai ốc M3 x 12mm, M4 x 26mm sử dụng chủ yếu ở Robot có thông số như sau: Chất liệu nhôm hợp kim, đầu tròn,



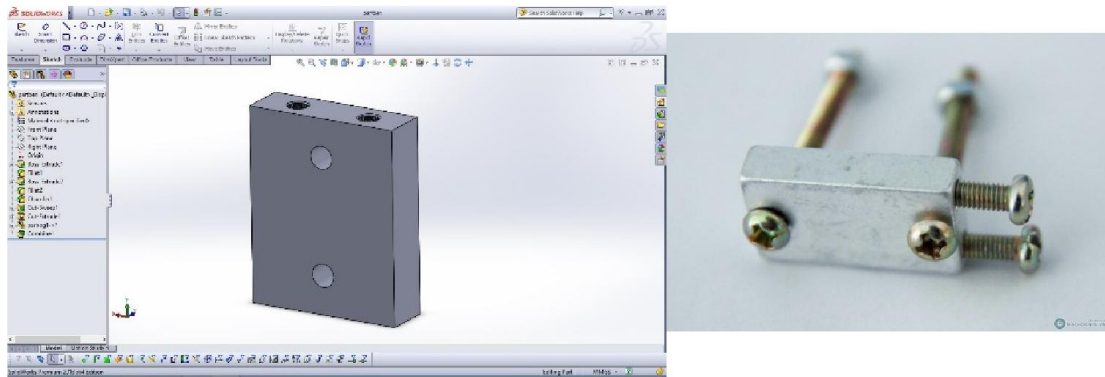
Hình 2.5: Bulông đai ốc

- **Trục bánh omni:**

Chất liệu được dùng để gia công trục bánh xe là đồng pha, có tính mềm dẻo gia công. Trục có kích thước 12 x 30mm, đường kính ren 3mm, đường kính trục 8mm...

- **Part bên:**

Được gia công bằng hợp kim nhôm có kích thước 20 x 25mm dày 6mm, giúp các kết cấu liên kết chặc vào nhau, có tính bền vững.



Hình 2.6: Part bên

CHƯƠNG III: HỆ THỐNG ĐIỆN

1. THIẾT BỊ ĐIỆN:

1.1 ĐỘNG CƠ:

Động cơ DC giảm tốc GA25 thích hợp với các ứng dụng xe mô hình, robot, ổ khóa điện tử, thiết bị thông minh,...

Sử dụng động cơ DC giảm tốc GA25 với thông số kỹ thuật sau:

- Điện áp cung cấp: 3 ~ 12V.
- Tốc độ không hộp số: 5600rpm.
- Dòng trung bình: 150 mA đến 450 mA.
- Tỷ số truyền: 1/45.
- Tốc độ qua hộp số: 3V ~ 30rpm.
- 6V ~ 60rpm.
- 12V ~ 130rpm.



1.2 NGUỒN CUNG CẤP:

Wild Scorpion 11.1V 1500mAh 35C.

Thông số kỹ thuật:

- Pin Wild Scorpion 11.1V 1500mAh 35C Li-Po 3 Cell RC
- Điện áp: 11.1V
- Dung lượng: 1500mAh
- Tỷ lệ xả: 35C
- Tỷ lệ sạc: 5C
- Số cell: 3Cell
- Kích thước: 22 x 35 x 75mm
- Trọng lượng: 115g.



tomtop_uk

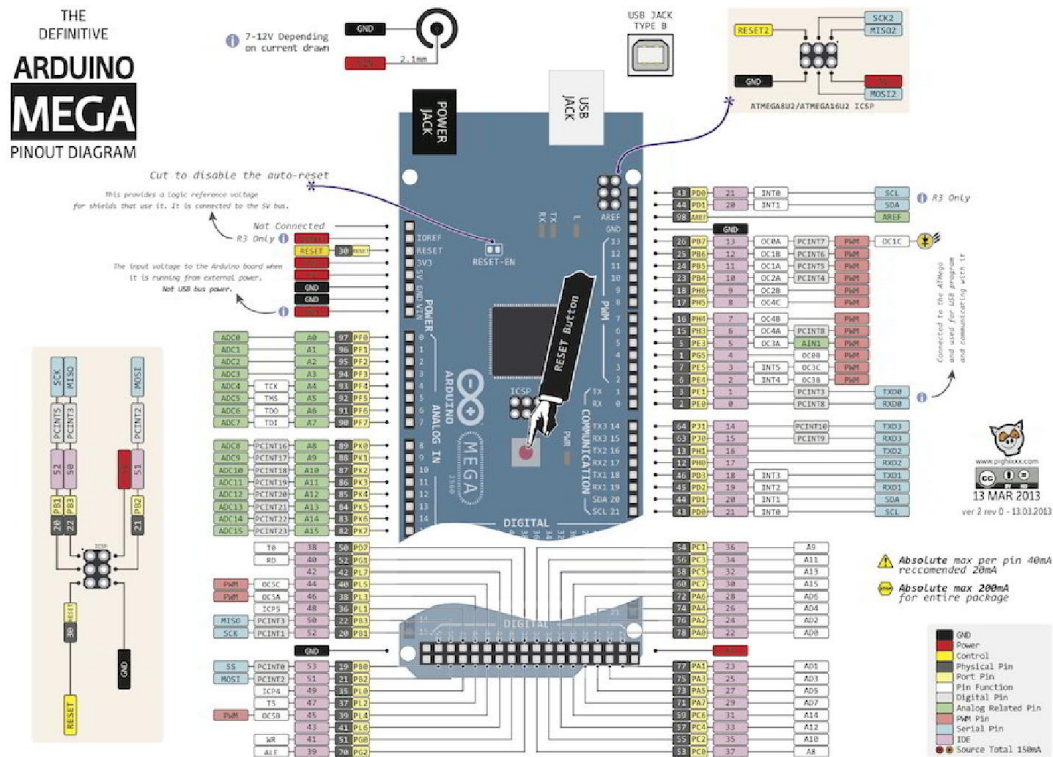
2. THIẾT BỊ ĐIỀU KHIỂN:

2.1 ARDUINO MEGA 2560:

Arduino Mega 2560 là một bo mạch thiết kế với bộ xử lý trung tâm là vi điều khiển AVR Atmega2560. Cấu tạo chính của Arduino Mega 2560 bao gồm các phần sau:

- Cổng USB: đây là loại cổng giao tiếp để ta upload code từ PC lên vi điều khiển. Đồng thời nó cũng là giao tiếp serial để truyền dữ liệu giữa vi điều khiển và máy tính.
 - Jack nguồn: để chạy Arduino thì có thể lấy nguồn từ cổng USB ở trên, nhưng không phải lúc nào cũng cắm với máy tính được. Lúc đó ta cần nguồn từ 9v đến 12v cắm vào jack nguồn.
 - Có 54 chân ra/vào số đánh số thứ tự từ 0 đến 13, ngoài ra có một chân nối đất (GND) và một chân điện áp tham chiếu (AREF).
 - Vi điều khiển AVR: đây là bộ xử lý trung tâm của bo mạch. Với mỗi mẫu Arduino khác nhau thì chip khác nhau. Ở con Arduino Mega 2560 này thì sử dụng ATMega2560.
 - Có một nút nhấn reset và một đầu ICSP
 - Các thông số chi tiết của Arduino Mega 2560:
 - Vi xử lý: 5V
 - Điện áp hoạt động: 7-12V
 - Điện áp đầu vào: 6-20V
 - Chân vào/ra (I/O) số: 54 chân (15 chân là đầu ra PWM)
 - Chân vào tương tự: 16
 - Dòng điện trong mỗi chân I/O: 40mA
 - Dòng điện chân nguồn 3.3V: 50mA
 - Bộ nhớ trong: 256KB
 - SRAM: 8 KB
 - EEPROM: 4 KB
 - Xung nhịp: 16 MHz
-

- Atmega 2560:
 - Các Mega 2560 có 16 chân vào tương tự, mỗi ngõ vào tương tự đều có độ phân giải 10 bit (tức là 1024 giá trị khác nhau). Theo mặc định đo từ 0 đến 5 volts, mặc dù là nó có thể thay đổi phần trên của phạm vi bằng cách sử dụng chân Aref và analogReference chức năng.
 - Các Atmega 2560 có 256 KB bộ nhớ flash để lưu trữ mã (trong đó có 8 KB được sử dụng cho bộ nạp khởi động), 8 KB SRAM và 4 KB EEPROM.

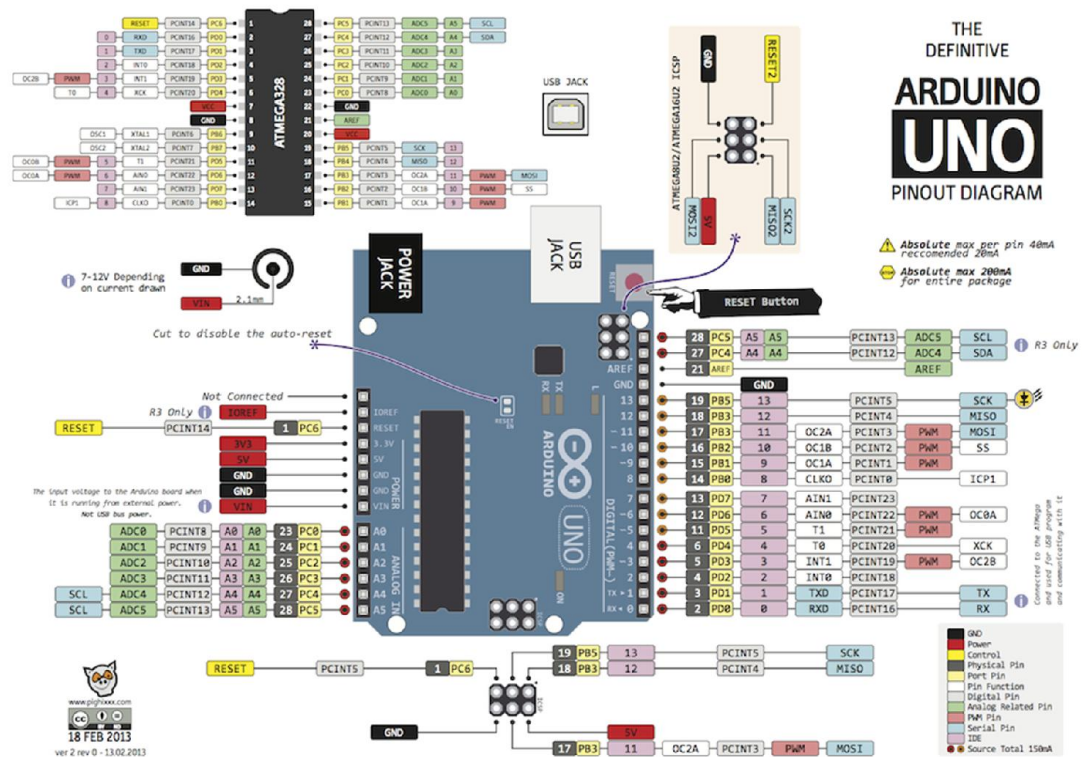


Hình 3.1: Sơ đồ chân của Arduino Mega 2560

2.2 ARDUINO UNO R3:

Arduino UNO R3 có chức năng và bộ xử lý trung tâm của Arduino UNO R3 cũng gần giống với Arduino Mega 2560 với các chân điều khiển ít hơn.

Arduino UNO R3 với vi xử lý trung tâm là Atmega328 có 14 chân I/O tín hiệu số, trong đó 6 chân có thể được sử dụng làm bộ điều chế độ rộng xung PWM, 6 ngõ vào tín hiệu tương tự, sử dụng thạch anh dao động 16MHz, kết nối USB, có ICSP Header...

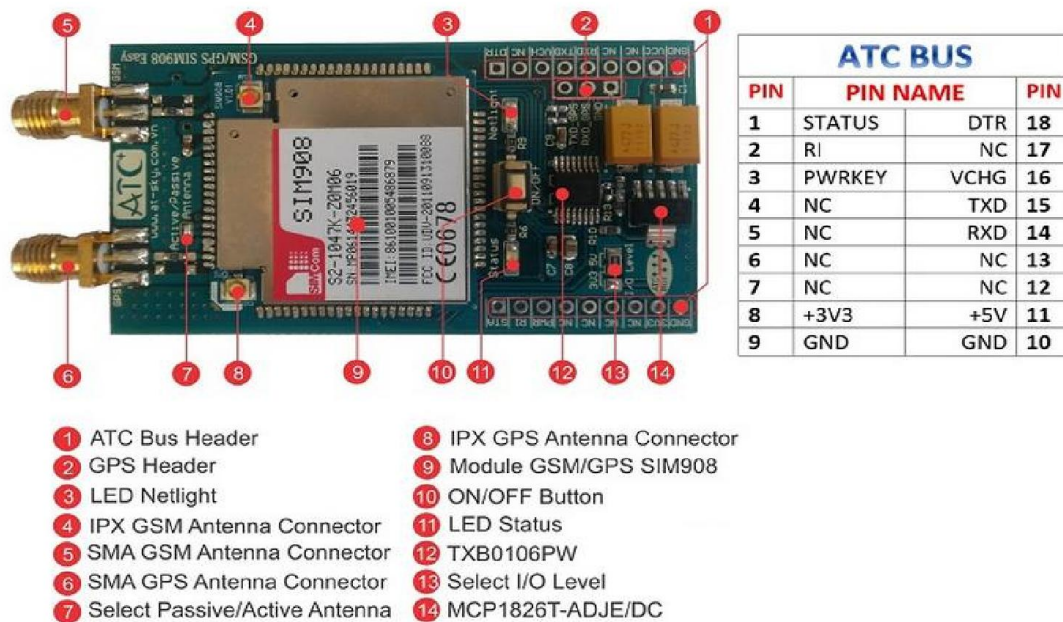


Hình 3.2: Sơ đồ chân của Arduino Uno R3

2.3 MODULE GSM/GPS SIM908 EASY:

Được thiết kế cho thị trường quốc tế, module SIM908 được tích hợp một bộ GSM/GPRS và một bộ GPS mạnh mẽ. Bộ GSM/GPRS hoạt động ở 4 băng tần là GSM 850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. Module SIM908 hỗ trợ GPRS multi-slot class 10/ class 8 và các mã hóa chương trình CS1, CS2, CS3 và CS4. Bộ GPS của SIM908 cho thời gian khởi động, độ chính xác và độ nhạy tốt nhất. Với việc được đóng gói nhỏ gọn, SIM908 có thể đáp ứng được không gian tốt trong các ứng dụng như M2M, Smart phone, PDA, các thiết bị định vị và di động khác.

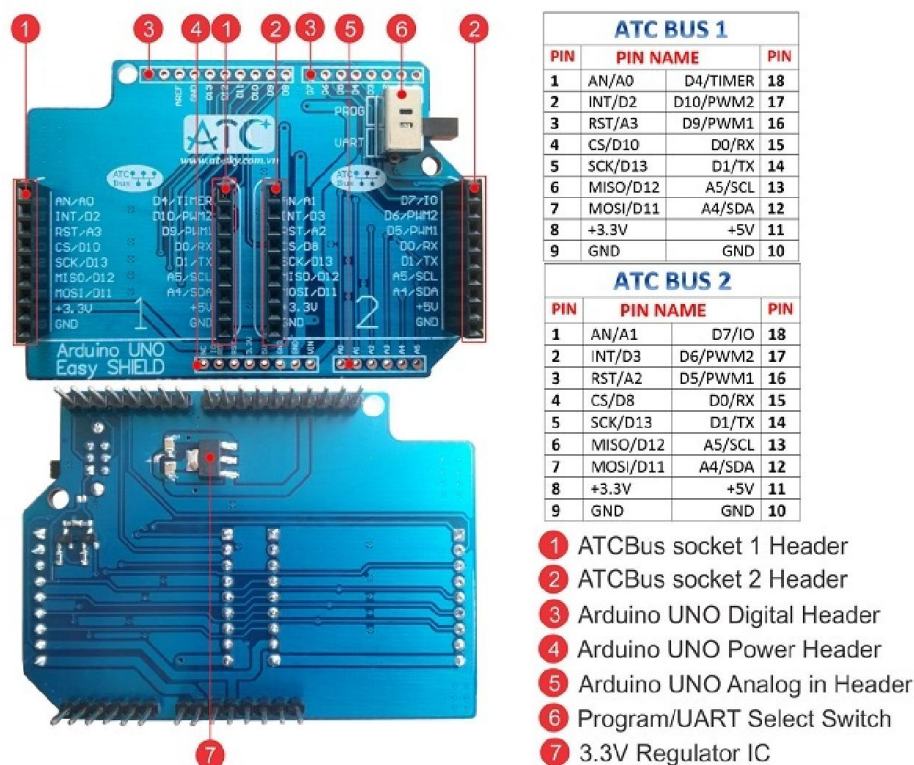
GSM/GPS SIM908 Easy là một sản phẩm do AT-COM phát triển dựa trên các tính năng của module SIM908. GSM/GPS SIM908 Easy được thiết kế giúp người dùng dễ dàng nghiên cứu và triển khai các ứng dụng liên quan đến GSM, GPRS như điều khiển, giám sát, truyền nhận dữ liệu, và ứng dụng GPS như : định vị tọa độ hiển thị lên bản đồ Google, đo tốc độ , thời gian...với độ chính xác cao. Với kích thước nhỏ gọn, ngõ ra dữ liệu tiện dụng, GSM/GPS SIM908 Easy sẽ mang đến những cảm hứng thiết kế hiện đại và tinh tế nhất.



Hình 3.3 Module GSM/GPS SIM908 Easy

2.4 ARDUINO UNO EASY SHIELD:

Arduino UNO Easy Shield là một board mở rộng cho Arduino UNO và những board tương thích với Arduino. Sản phẩm hỗ trợ 2 ATC Bus socket cho phép người dùng kết nối những Easy Board với Arduino UNO một cách nhanh chóng và dễ dàng để thực hiện các ứng dụng như: GSM, GPS, Wifi, Bluetooth,... Ngoài ra, Shield có hỗ trợ Switch cho phép nạp chương trình cho Arduino mà không cần phải tháo board thiết bị đang lắp trên Shield. Bên cạnh đó, với kích thước tương đương với Arduino UNO, ngõ ra dữ liệu tiện dụng, Arduino UNO Easy Shield mang lại những cảm hứng tinh tế khi kết hợp Arduino và Easy Board.



Hình 3.4: Arduino UNO Easy Shield

2.5 MODULE BLUETOOTH:

Sử dụng module bluetooth HC-05

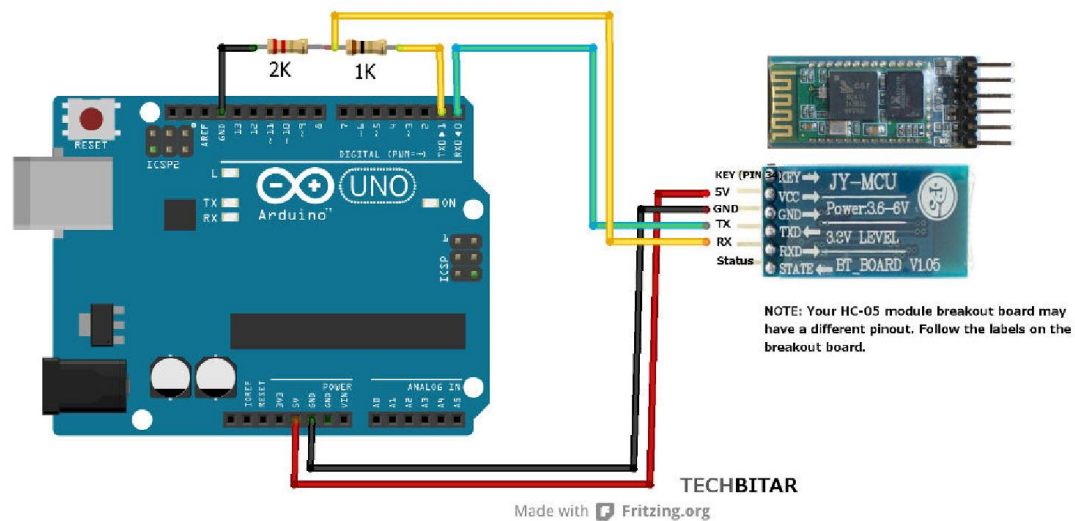
Module bluetooth HC05 master/slave dùng để thiết lập kết nối Serial giữa 2 thiết bị bằng sóng bluetooth. Điểm đặc biệt của module bluetooth HC-05 là module có thể hoạt động được ở 2 chế độ: MASTER hoặc SLAVE. Trong khi đó, bluetooth module HC-06 chỉ hoạt động ở chế độ SLAVE.

Ở chế độ SLAVE: bạn cần thiết lập kết nối từ smartphone, laptop, usb bluetooth để dò tìm module sau đó pair với mã PIN là 1234. Sau khi pair thành công, bạn đã có 1 cổng serial từ xa hoạt động ở baud rate 9600.

Ở chế độ MASTER: module sẽ tự động dò tìm thiết bị bluetooth khác (1 module bluetooth HC-06, usb bluetooth, bluetooth của laptop...) và tiến hành pair chủ động mà không cần thiết lập gì từ máy tính hoặc smartphone.

Module bluetooth HC05 được điều khiển bằng tập lệnh AT để thực hiện các tác vụ mong muốn. Để bluetooth module chuyển từ chế độ thông thường qua điều khiển bằng AT, ta có 2 cách như sau:

- Cấp nguồn cho module bluetooth (Vcc và Gnd) đồng thời cấp mức điện áp cao (=Vcc) cho chân KEY của module bluetooth. Khi đó giao tiếp bằng tập lệnh AT với module bằng cổng Serial (Tx và Rx) với baud rate là 38400 (khuyến dùng).
- Cấp nguồn cho module bluetooth trước, sau đó cấp mức điện áp cao cho chân KEY của module bluetooth. Lúc này bạn có thể giao tiếp với module bằng tập lệnh AT với baud rate là 9600. Sau khi pair thành công với thiết bị bluetooth khác, đèn trên module bluetooth HC05 sẽ nhấp nháy chậm cho thấy kết nối Serial đã được thiết lập. Nguồn cung cấp cho module bluetooth là nguồn từ 3.6V đến 6V. Quá áp sẽ gây cháy module. Ngoài ra module tương thích với các vi điều khiển 5V mà không cần chuyển đổi mức giao tiếp 5V về 3.3V như nhiều loại module bluetooth khác.



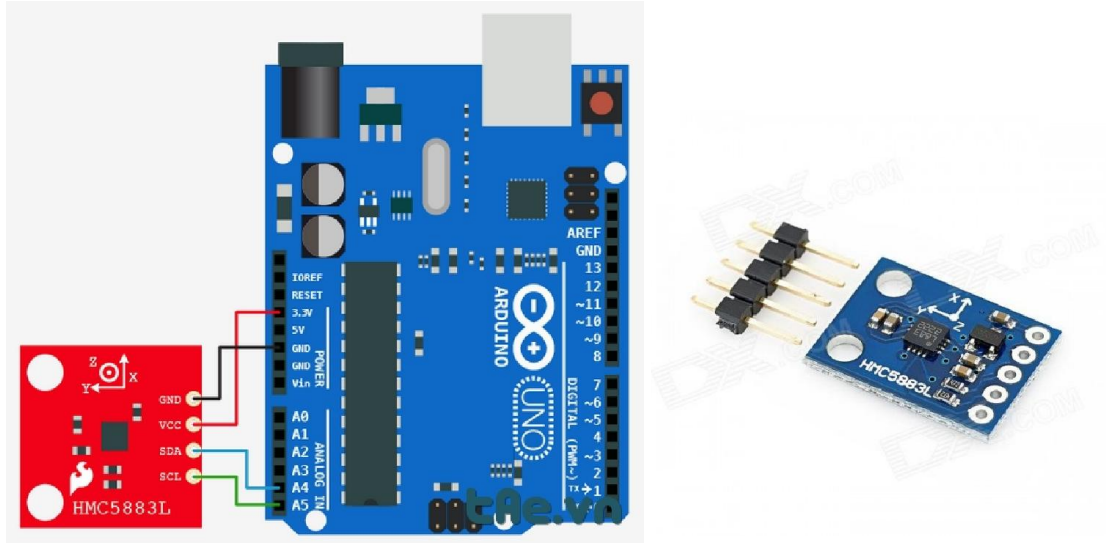
Hình 3.5: Module bluetooth HC-05

2.6 CẢM BIẾN LA BÀN SỐ HMC5883L:

Cảm biến la bàn số HMC5883L được dùng để đo từ trường của trái đất nhằm xác định phương hướng của vị trí hiện tại với độ chính xác lên đến 1 hoặc 2 độ. Nó cung cấp các cách đo riêng biệt cho từng trục và có thể kết hợp lại để tính toán 3D. Có thể dùng để đo từ trường thô hoặc các nguồn từ trường mạnh hơn gần nó. Mạch 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.

- Thông số kỹ thuật:
 - Điện áp cung cấp: 3 ~ 5VDC.
 - Chuẩn giao tiếp: I2C (3 ~ 5 TTL).
 - Kích thước: 14 x 13mm.
- Kết nối đến Arduino:
 - Arduino GND -> HMC5883L GND
 - Arduino 3.3V -> HMC5883L VCC

- Arduino A4 (SDA) -> HMC5883L SDA
- Arduino A5 (SCL) -> HMC5883L SCL



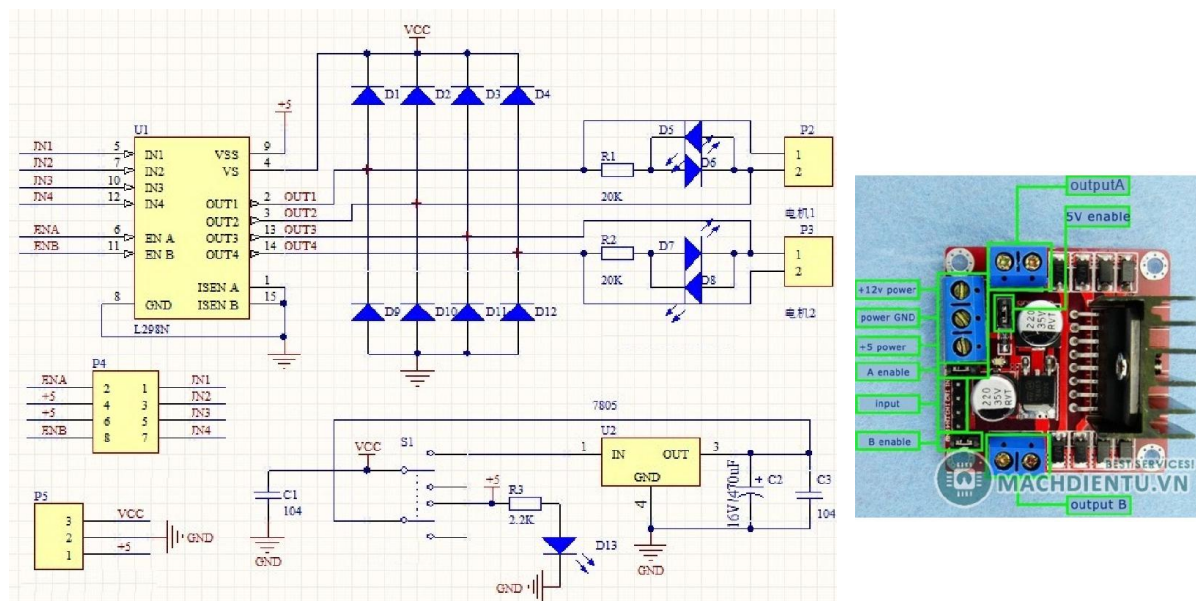
Hình 3.6: Cảm biến la bàn HMC5883L

2.7 MẠCH ĐIỀU KHIỂN ĐỘNG CƠ L298:

Module L298 có thể điều khiển 2 động cơ DC hoặc 1 động cơ bước, có 4 lỗ nằm ở 4 góc thuận tiện cho người sử dụng cố định vị trí của module.

Thông số kỹ thuật:

- Driver: L298N tích hợp hai mạch cầu H.
- Điện áp điều khiển: +5 V ~ +35 V
- Dòng tối đa cho mỗi cầu H là: 2A
- Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
- Dòng của tín hiệu điều khiển: 0 ~ 36mA
- Công suất hao phí: 20W (khi nhiệt độ $T = 75\text{ }^{\circ}\text{C}$)
- Nhiệt độ bảo quản: $-25\text{ }^{\circ}\text{C} \sim +130\text{ }^{\circ}\text{C}$



Hình 3.7: Cảm biến la bàn HMC5883L

2.8 MODULE ĐỊNH VỊ TOÀN CẦU UBLOX NEO-6M GPS:

GPS NEO-6M v2 là module định vị toàn cầu sử dụng hệ thống vệ tinh GPS của Mỹ. Module GPS NEO-6M cho tốc độ xác định vị trí nhanh và chính xác, có nhiều mức năng lượng hoạt động, phù hợp với các ứng dụng chạy pin.

Module GPS sử dụng board điều khiển kết nối của hãng U-BLOX đến từ Thụy Sĩ có rất nhiều năm kinh nghiệm trong lĩnh vực sản xuất module định vị toàn cầu.

Module GPS NEO-6M v2 có thể được cấp nguồn từ 3.3V - 5V nhưng chỉ giao tiếp ở mức 3.3V. Trên module có sẵn port serial để bạn kết nối.

Việc cấu hình thông số kết nối GPS, thời gian nhấp nháy LED, mức năng lượng hoạt động... được thiết lập thông qua phần mềm u-Center.

Thiết lập sẵn một số tính năng quan trọng của module GPS NEO-6M:

- Serial hoạt động ở baudrate 9600

- Mức năng lượng kết nối: Max performance (0), với mức này bạn có tọa độ chính xác nhất nhưng đòi hỏi module phải để ngoài trời, nếu hoạt động trong nhà sẽ khó bắt sóng. Bạn có thể cấu hình cho module hoạt động ở mức Eco power (4) để tiết kiệm năng lượng, ở chế độ (4) module chỉ cần 3-4 vệ tinh là có thể bắt sóng được.
- LED nhấp nháy khi chưa kết nối theo tần số 1Hz, duty cycle: 50%. Sau khi module fix được tín hiệu GPS, LED nhấp nháy nhanh hơn theo tần số 5Hz, duty cycle 20%.

Để cấu hình module GPS NEO-6M bạn cần 1 cổng Serial giao tiếp ở mức 3.3V. Chúng tôi hiện có 2 loại USB to TTL Serial có thể đáp ứng tốt: PL2303HX và CP2102. Bạn cũng có thể gắn antenna bên ngoài hoặc hàn trực tiếp vào module. Khi cho module hoạt động, hướng antenna lên trên để module bắt sóng tốt.



Hình 3.8: Module GPS NEO-6M

CHƯƠNG IV: LẬP TRÌNH

1. GIỚI THIỆU PHẦN MỀM LẬP TRÌNH:

1.1 NGÔN NGỮ LẬP TRÌNH CHO ARDUINO:

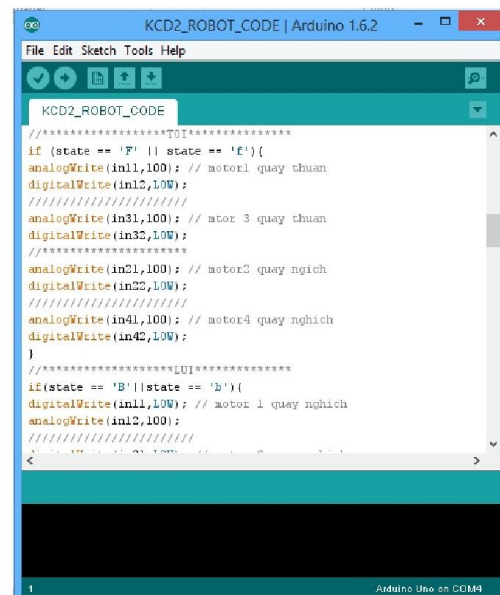
Phần mềm Arduino IDE

Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho Arduino, tuy nhiên sức mạnh thực sự của Arduino nằm ở phần mềm. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.

Arduino IDE là phần mềm dùng để lập trình cho Arduino. Môi trường lập trình Arduino IDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh osx và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

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

Hình 4.1: Giao diện Arduino IDE



Hiện tại, Arduino IDE có thể download từ trang chủ <http://arduino.cc/> bao gồm các phiên bản sau:

- Arduino 1.0.5
- Arduino 1.5.5 BETA (Hỗ trợ cho 2 board Arduino mới nhất là: Arduino Yun và Arduino Due).
- Arduino IDE cho Intel Galileo.

1.2 CÔNG CỤ TẠO ỨNG DỤNG TRÊN SMARTPHONE:

Để tạo một ứng dụng đơn giản không nhất thiết là một lập trình viên, có kiến thức về lập trình, mọi thứ đều đơn giản đối với App Inventor.

Ngày 12/7/2010, Google chính thức giới thiệu công cụ lập trình trực quan App Inventor dùng để phát triển phần mềm ứng dụng trên hệ điều hành Android. App Inventor là công cụ lập trình dành cho mọi người, kể cả trẻ em.

Tại hội thảo Activate 2010 (1/7/2010), Eric Schmidt - giám đốc điều hành Google khẳng định: "Công nghệ di động hiện là lĩnh vực phát triển nóng nhất. Những lập trình viên giỏi nhất đang ưu tiên tạo ra phần mềm ứng dụng cho thiết bị di động, thay vì phần mềm cho máy tính cá nhân trên Windows hoặc trên Apple Mac. Một trong những nguyên nhân của xu hướng này là các thiết bị di động có tính cá nhân sâu sắc hơn nhiều so với máy tính cá nhân".

Với công cụ App Inventor, Google tạo điều kiện để mọi người có thể tự xây dựng phần mềm ứng dụng cho thiết bị di động dùng hệ điều hành Android. Trang web của dự án App Inventor nêu rõ: "Với App Inventor, bạn có thể xây dựng phần mềm ứng dụng bất kỳ theo ý tưởng của mình. Nhiều người bắt đầu với việc tạo ra trò chơi đơn giản như trò "bắt chuột chũi" (bắt một chú chuột chũi nhảy ngẫu nhiên trên màn hình) hoặc phần mềm vẽ "râu ria" trên hình chân dung của bạn bè. Dựa vào các bộ cảm ứng có sẵn trong điện thoại, bạn còn có thể tạo ra trò chơi di chuyển quả bóng trong "mê cung" bằng cách nghiêng điện thoại.

Việc lập trình ứng dụng không chỉ giới hạn ở những trò chơi đơn giản. Bạn cũng có thể tạo ra phần mềm để cung cấp thông tin hoặc phục vụ cho việc dạy và học,

chẳng hạn phần mềm giúp bạn cùng các bạn học của mình ôn tập kiến thức để chuẩn bị cho kỳ thi nào đó. Do Android có sẵn chức năng đọc từ thành tiếng, phần mềm của bạn còn có thể đọc to câu hỏi ôn tập.

App Inventor thực chất là một ứng dụng web, chạy bởi trình duyệt trên máy tính cá nhân. Tuy nhiên, người dùng vẫn phải cài đặt một phần mềm Java mang tên App Inventor Extras, có nhiệm vụ điều khiển điện thoại Android (kết nối với máy tính thông qua cổng USB). Nhờ vậy, người dùng có thể nhanh chóng chuyển ứng dụng từ máy tính cá nhân qua điện thoại Android để chạy thử. Nhóm dự án App Inventor tại Google dự định bổ sung những bộ mô phỏng điện thoại Android để có thể sử dụng App Inventor mà không cần có điện thoại Android thực sự gắn vào máy tính.

2. LẬP TRÌNH:

2.1 CHƯƠNG TRÌNH KCD2 MOBILE ROBOT TRÊN APP INVENTOR:

❖ Cách tạo ứng dụng trên App Inventor:

Đầu tiên phải truy cập vào địa chỉ sau <http://ai2.appinventor.mit.edu>

Trước tiên click vào New để tiến hành tạo ứng dụng. Sau đó đặt tên cho ứng dụng mà bạn sẽ tạo (tạo chương trình phát nhạc).

Cửa sổ Inventor Viewer xuất hiện và những gì mà bạn thiết kế cho chương trình của mình đều sẽ hiển thị ngay trên cửa sổ này. Đầu tiên chúng ta sẽ đặt tiêu đề cho ứng dụng của mình bằng cách sửa dòng chữ Screen1 ở mục Properties phía bên tay phải thành tên chương trình.

Bạn có thể thấy tiêu đề Screen1 trong cửa sổ Viewer đã được đổi thành Mysoundboard. Tiếp đến ta cần tạo một khung để chứa các nút của chương trình. Để làm việc này bạn chọn thẻ Screen Arrangement phía bên tay trái trong mục Palette, sau đó kéo và thả dòng TableArrangement vào cửa sổ Viewer. Trong mục Properties của Table Arrangement bạn sửa thông số Column và Row thành 3 để khung có kích thước 3 hàng 3 cột.

Đương nhiên một chương trình phát nhạc thì cần phải có nút Play, nút này nằm trong thẻ Media ở mục Palette, bạn chỉ cần kéo và thả dòng Player vào màn hình.

Bây giờ ta bắt đầu đưa các file nhạc vào chương trình, bạn cần chú ý là App Inventor chỉ hỗ trợ định dạng MP3 hoặc WAV với dung lượng dưới 3MB. Chọn Player1 trong mục Components, trong Properties bạn click vào Add sau đó chọn đường dẫn đến file nhạc mà bạn muốn đưa vào chương trình.

Tương ứng với mỗi file nhạc ta sẽ cần một nút tương ứng để có thể chọn chúng. Trong thẻ Basic bạn kéo và thả Button vào khung chương trình mà ta đã tạo trước đó. Sau đó sửa tên các nút tương ứng với các file nhạc trong ô Text.

Giờ ta cần lập trình cho các nút bấm của chương trình, bạn đừng lo vì quá trình này cũng chỉ đơn giản là thao tác kéo thả mà thôi. Để bắt đầu bạn Click vào Blocks Editor. Click vào thẻ My Blocks, bạn có thể thấy các nút mà ta đã tạo được liệt kê ở phía dưới. Trong Button1 bạn kéo và thả “miếng ghép” Button1. Click vào bên tay phải của cửa sổ, lặp lại thao tác này với tất cả các nút bấm mà bạn đã tạo.

Tiếp đến bạn chọn Player1 ở dưới sau đó kéo và thả Player1.Source vào phần trống của tất cả các mảnh Button. Click ở trên.

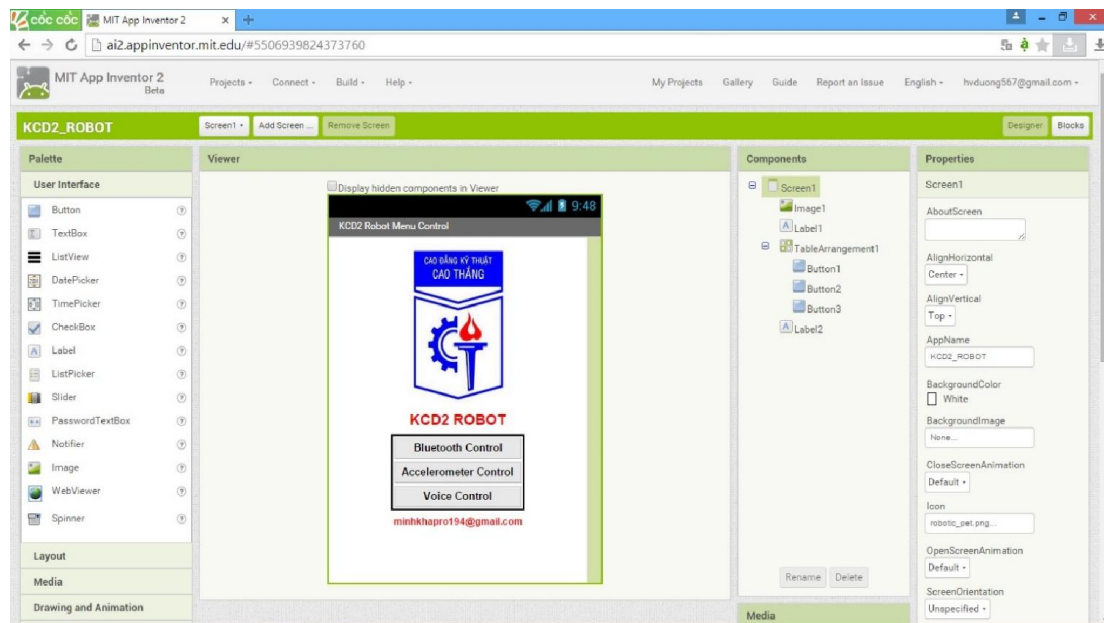
Chọn thẻ Built-In sau đó chọn Text. Tương tự bạn kéo và thả miếng ghép text vào phía sau của Player1.Source. Ô text này sẽ cho chương trình biết file nào sẽ được chạy khi nút được nhấn nên bạn cần đổi tên dòng text thành tên file nhạc tương ứng. Lặp lại quá trình này đối với tất cả các nút khác.

Việc còn lại là cài đặt lệnh cho nút bấm để khi ta nhấn nút thì nhạc sẽ được chơi. Đơn giản bạn chỉ cần kéo miếng Player1.Start trong thẻ My Blocks và ghép vào Button.Click của từng nút.

Vậy là chương trình Soundboard của chúng ta đã hoàn thiện, bạn có thể chạy thử chương trình trên trình giả lập của Inventor hoặc tải vào điện thoại của mình.

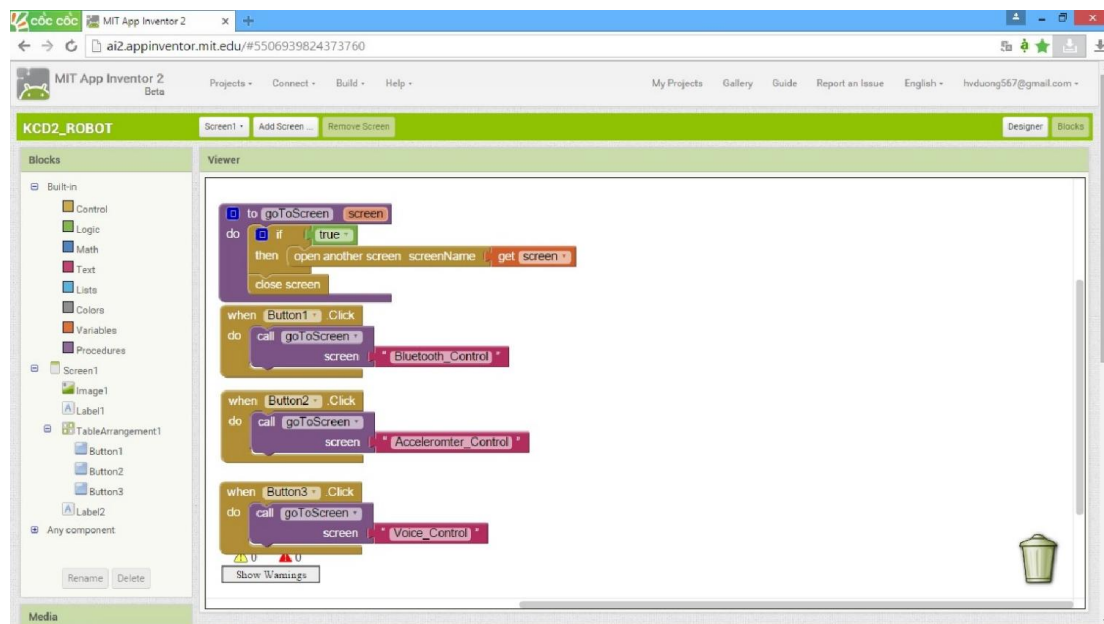
❖ Tạo ứng dụng cho KCD2 MOBILE ROBOT:

Việc đầu tiên là đăng nhập vào App Inventor, sau đó tạo một Projects. Ở phần designer chúng ta tạo một giao diện theo thiết kế ban đầu bằng việc kéo thả những khối công cụ Palette bên góc trái màn hình như ví dụ trên.



Hình 4.2: Giao diện App Inventor

Tiếp theo vào phần Blocks để ghép các câu lệnh lại với nhau theo tính toán từ trước. Các câu lệnh của KCD2 MOBILE ROBOT như sau:

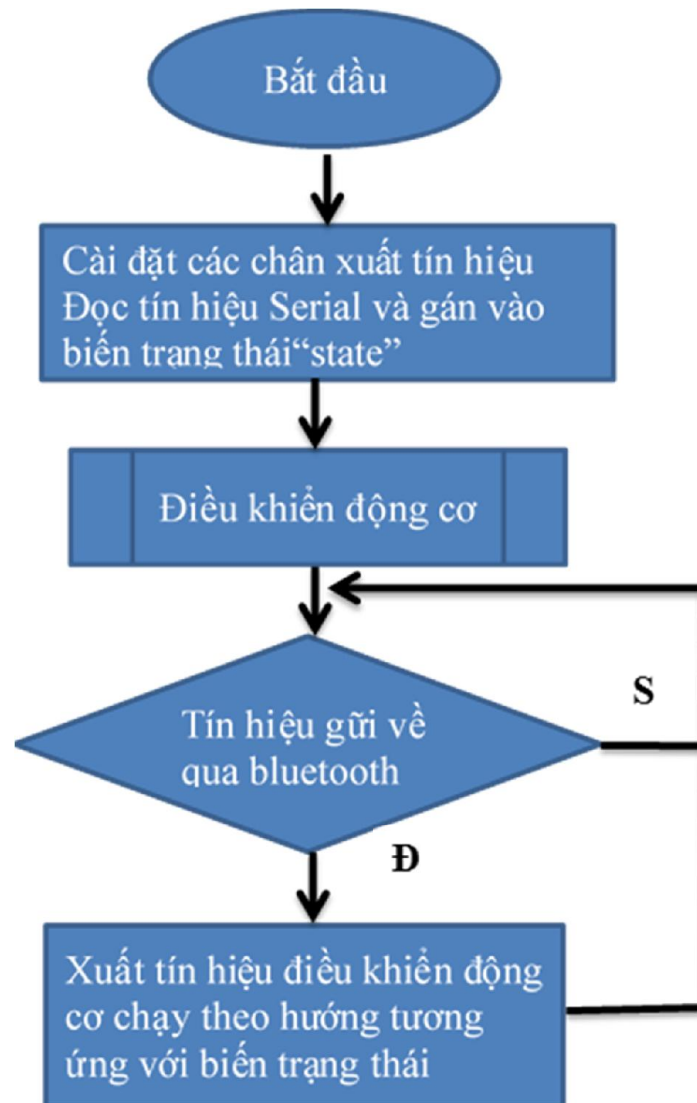


Hình 4.3: Giao diện phần Blocks (code)

2.2 KCD2 MOBILE ROBOT CODE:

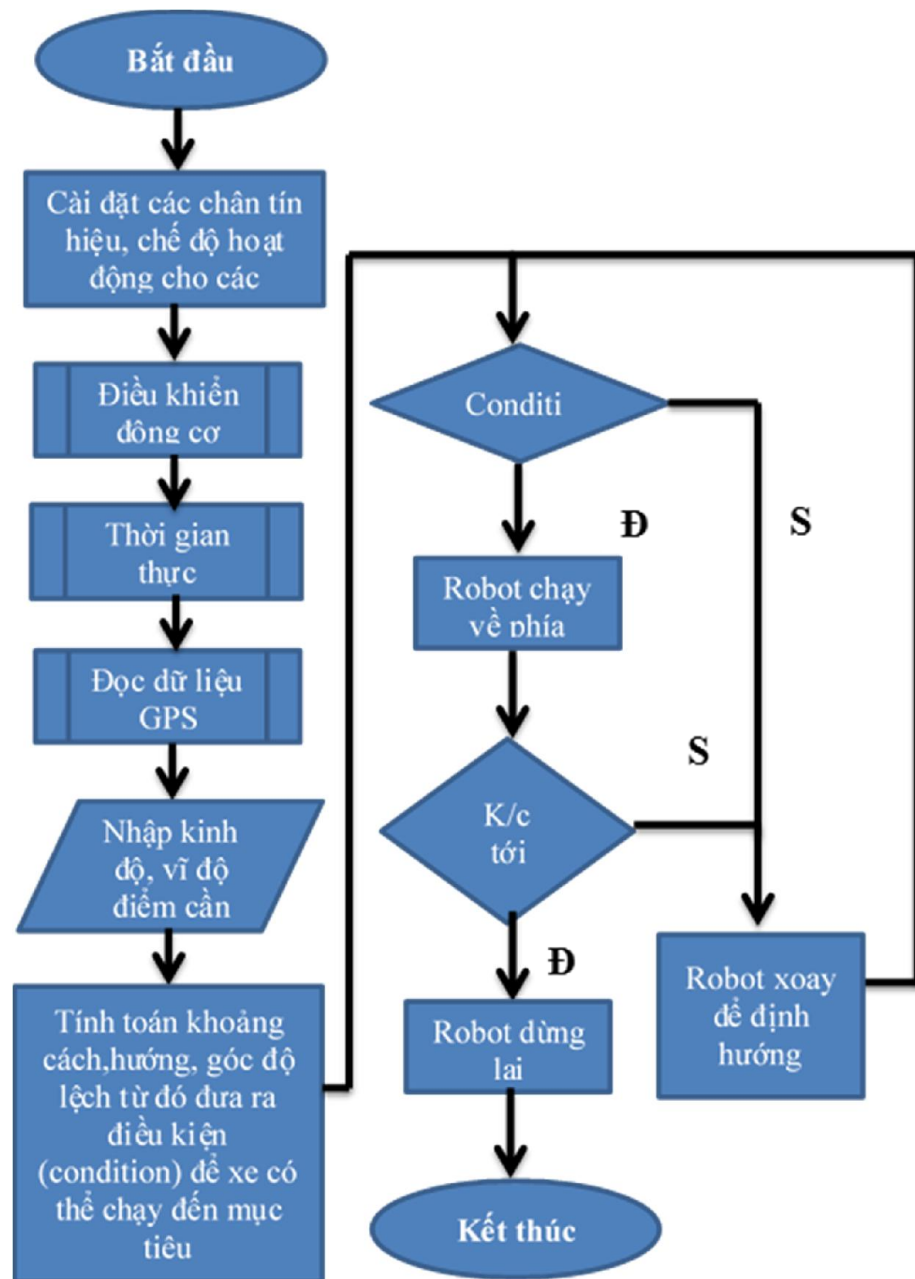
❖ Lưu đồ giải thuật của KCD2 MOBILE ROBOT:

Chức năng chạy có điều khiển

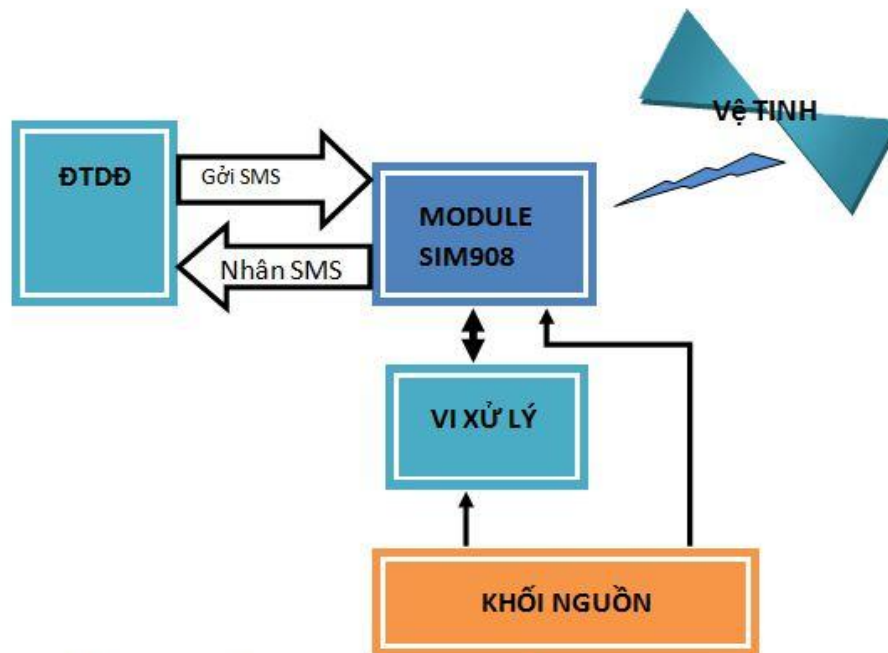


Hình 4.4: Lưu đồ giải thuật Robot có điều khiển

Chức năng chạy tự động



Hình 4.5: Lưu đồ giải thuật Robot tự động



Hình 4.6: Sơ đồ khối Module SIM908

❖ **Phần code điều khiển:**

Tham khóa code CKD2 MOBILE ROBOT ở phần phụ lục code.

CHƯƠNG V: KẾT LUẬN**1. KẾT QUẢ ĐẠT ĐƯỢC:****1.1 VỀ MẶT LÝ THUYẾT:**

- Thiết kế và lựa chọn kết cấu hợp lý.
- Xây dựng phần mềm điều khiển từ Smartphone android.
- Tính toán phép toán cho Robot.

1.2 VỀ MẶT THỰC NGHIỆM:

- Thiết kế chế tạo thành công KCD2 MOBILE ROBOT
- Thêm kinh nghiệm về lập trình android tạo app.
- Thêm kinh nghiệm về thiết kế và thực thi phần cơ khí.
- Đạt được mục tiêu đề ra lúc đầu.

2. HẠN CHẾ:

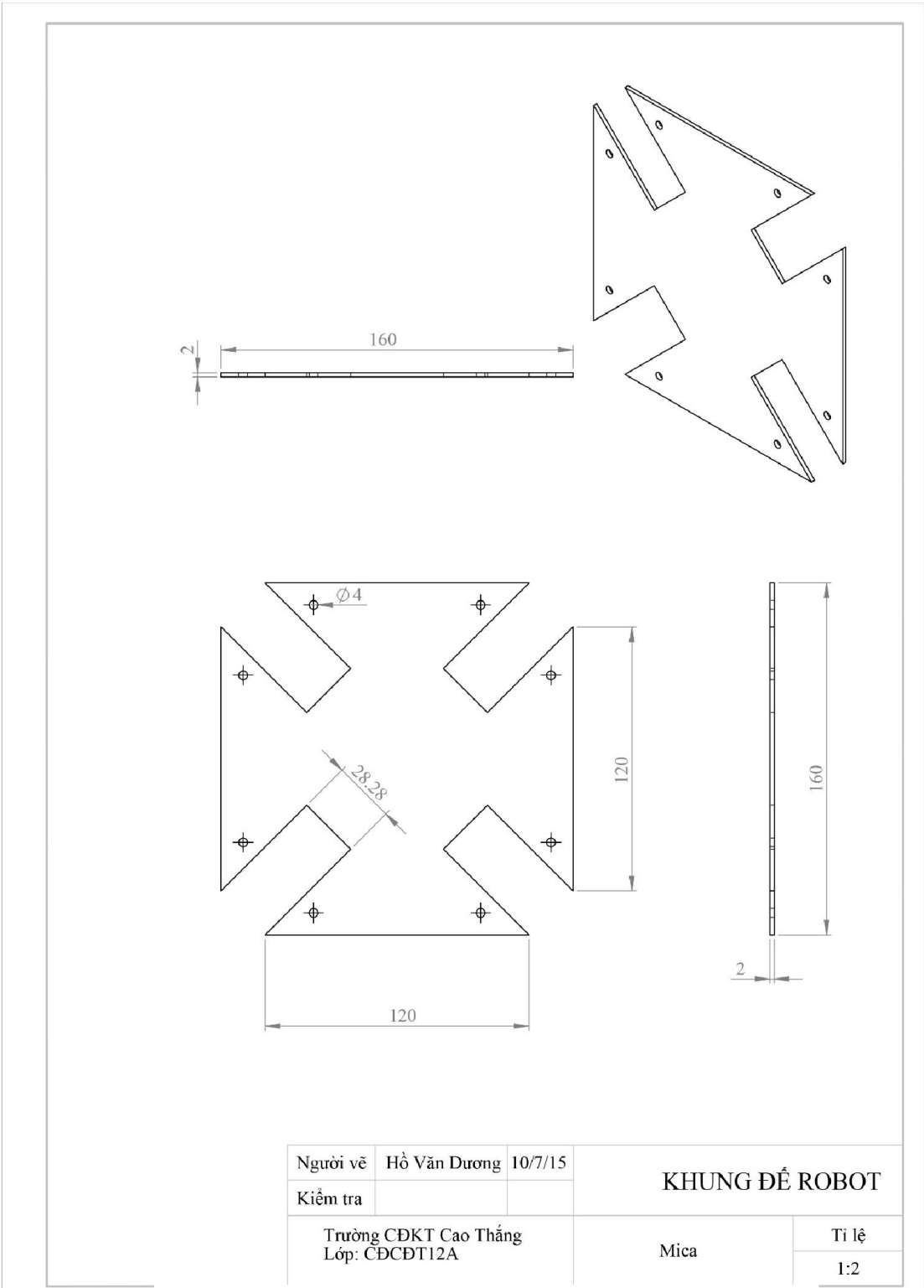
- Kết cấu, hình dạng Robot chưa thực sự đạt tính vững chắc và tính thẩm mỹ.
- Hoạt động còn hạn chế trong một số môi trường (địa hình gồ ghề hay trơn).
- Robot hoạt động chưa thật sự chính xác hoàn toàn so với mặt lý thuyết.

3. HƯỚNG PHÁT TRIỂN:

- Đề tài có tính mở, có rất nhiều vấn đề cần tiếp tục nghiên cứu phát triển để đáp ứng vào nhu cầu thực tế.
- Có thể thiết kế Robot gọn hơn mức có thể.
- Hoàn thiện tối ưu hơn về phần lập trình điều khiển.
- Trang bị thêm cho Robot hệ thống quan sát, sensor đo nhiệt độ, độ ẩm, v.v.
- Có thể gắn thêm tay máy cho thuận tiện vào việc cầm nắm tránh vật cản.

Trong suốt quá trình thực thi đồ án chúng em gặp rất nhiều khó khăn về tài chính, kiến thức chuyên môn và thời gian có hạn nên chúng em chưa hoàn thiện những vấn đề trên. Đây là một đề tài hay và mới, có rất nhiều hướng phát triển, có tính thực tế cao. Chúng em hy vọng mô hình sẽ được các khóa sau hoàn thiện phát triển hơn để góp phần làm cho nền khoa học công nghệ của nước nhà phát triển mạnh.

BẢN VẼ CƠ KHÍ

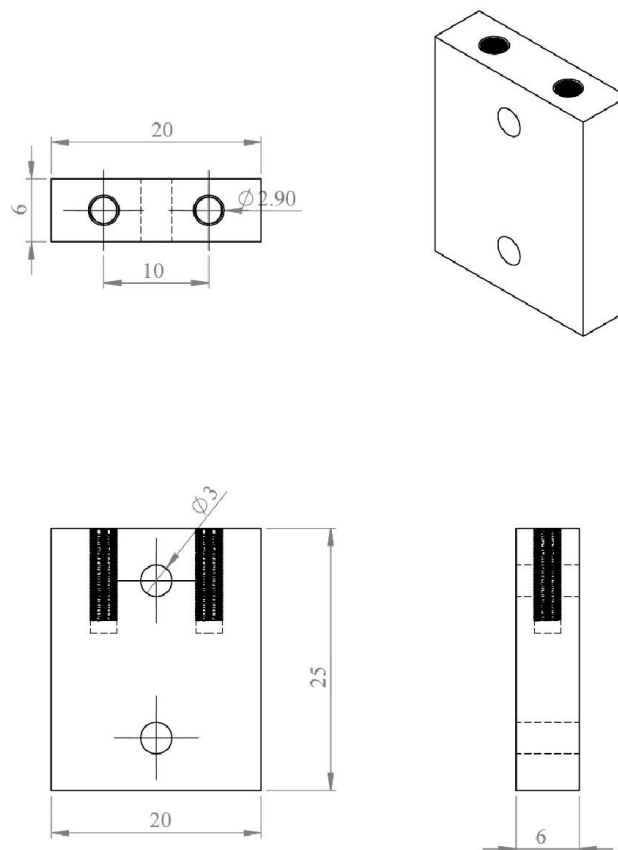


Hình 1: Chi tiết đế Robot

Hình 2: Chi tiết khung Robot

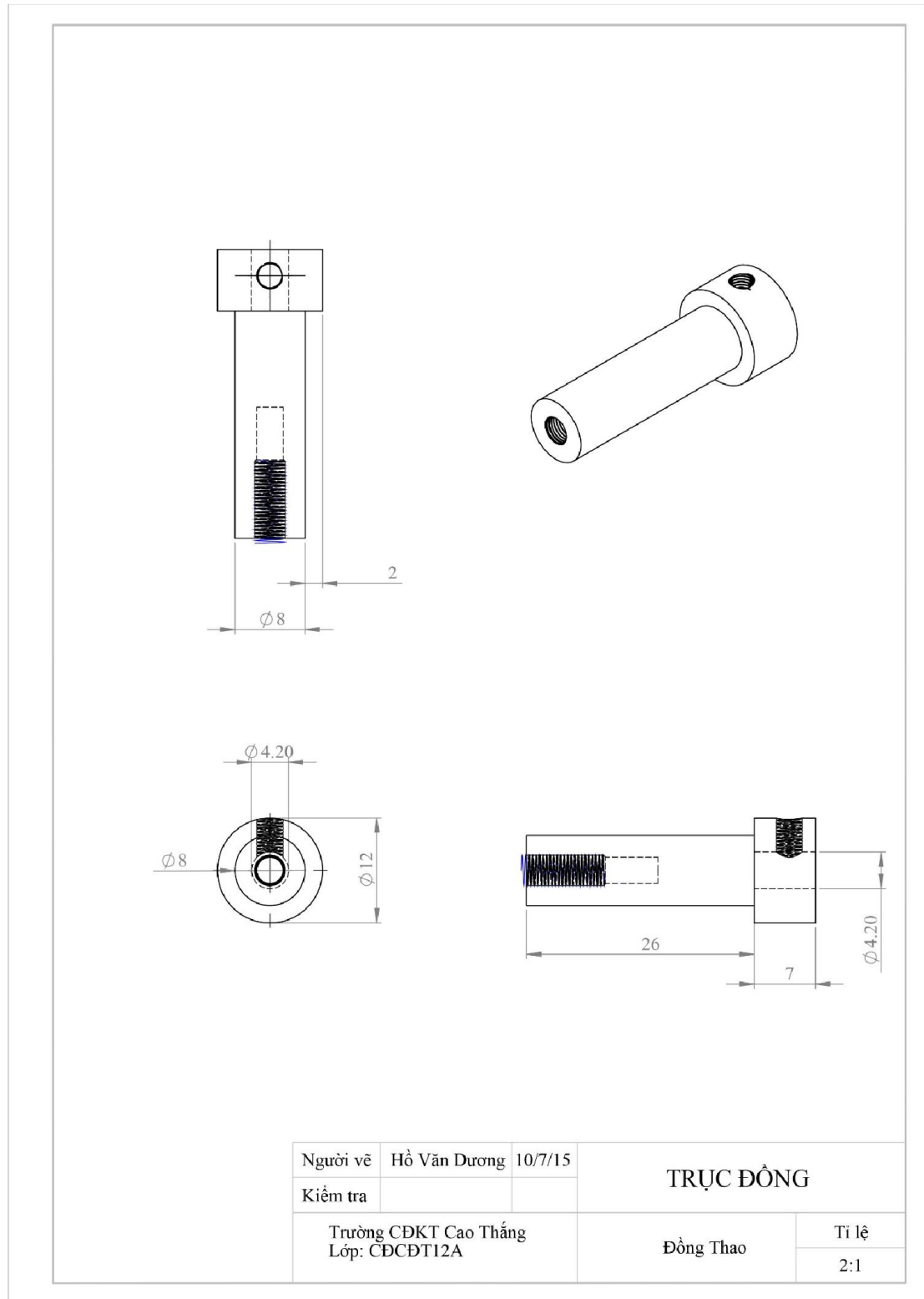


Hình 3: Chi tiết Part gá động cơ

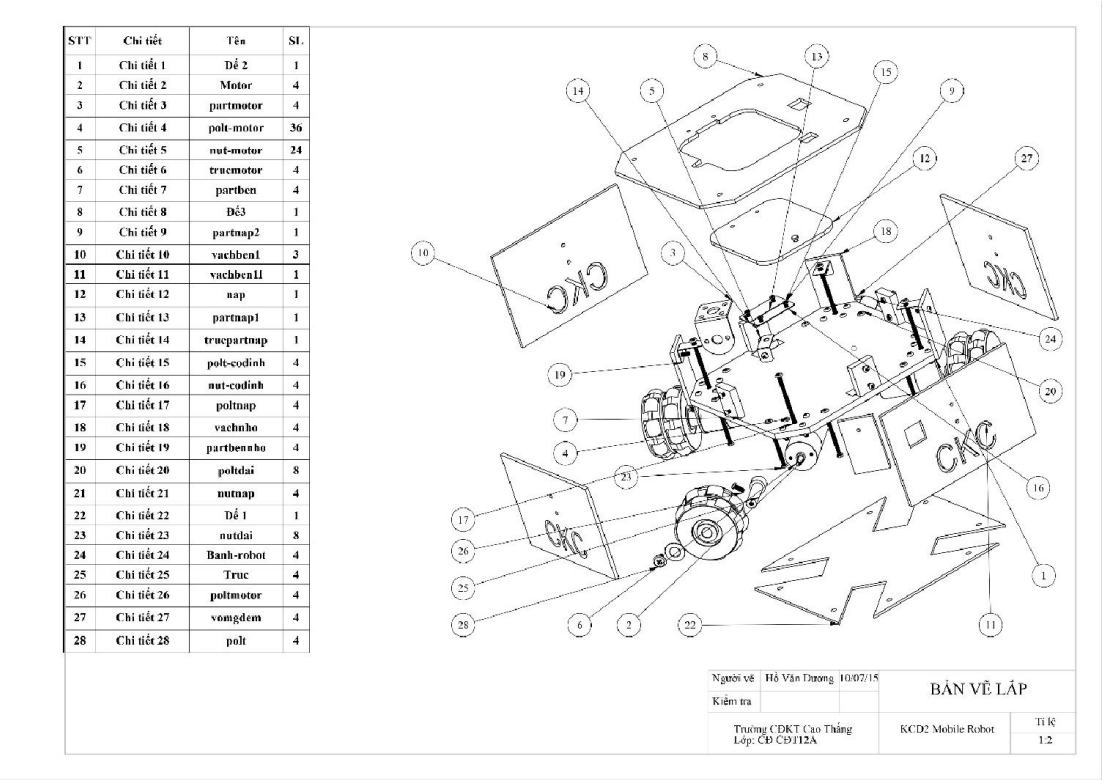


Người vẽ	Hồ Văn Dương	10/7/15	PART BÊN	
Kiểm tra				
Trường CĐKT Cao Thắng Lớp: CĐCĐT12A			Nhóm	Tỉ lệ
				2:1

Hình 4: Chi tiết Part bên



Hình 5: Chi tiết trục đồng



Hình 6: Bản vẽ lắp chi tiết KCD2 Mobile Robot

MỤC LỤC CODE CỦA CKD2 MOBILE ROBOT**❖ ROBOT CÓ ĐIỀU KHIỂN:**

```
#include <SoftwareSerial.h>
char state;
// motor1
int in11 =8;
int in12 =9;
//*****

//motor2
int in21 =2;//2
int in22 =3;//3
//*****

//motor3
int in31=4;
int in32=5;
//*****

//motor4
int in41=6;
int in42=7;
//*****

void setup() {
  Serial.begin(9600);
  pinMode(in11, OUTPUT);
  pinMode(in12, OUTPUT);
  pinMode(in21, OUTPUT);
  pinMode(in22, OUTPUT);
  pinMode(in31, OUTPUT);
```

```
pinMode(in32, OUTPUT);
pinMode(in41, OUTPUT);
pinMode(in42, OUTPUT);
}

void dung() {
digitalWrite(in11, LOW);
digitalWrite(in12, LOW);
digitalWrite(in21, LOW);
digitalWrite(in22, LOW);
digitalWrite(in31, LOW);
digitalWrite(in32, LOW);
digitalWrite(in41, LOW);
digitalWrite(in42, LOW);
}

void loop() {
if (Serial.available()) {
state = Serial.read();
//*****TÔI*****
if (state == 'F' || state == 'f') {
analogWrite(in11, 100); // motor1 quay thuận
digitalWrite(in12, LOW);
//////////
analogWrite(in31, 100); // mtor 3 quay thuận
digitalWrite(in32, LOW);
//*****

analogWrite(in21, 100); // motor2 quay nghịch
digitalWrite(in22, LOW);
//////////
analogWrite(in41, 100); // motor4 quay nghịch
```

```
digitalWrite(in42,LOW);
}
//*****LUI*****
if(state == 'B' || state == 'b'){
digitalWrite(in11,LOW); // motor 1 quay nghịch
analogWrite(in12,100);
//////////
digitalWrite(in31,LOW); // motor 3 quay nghịch
analogWrite(in32,100);
//*****
digitalWrite(in21,LOW); // motor 2 quay thuận
analogWrite(in22,100);
//////////
digitalWrite(in41,LOW);
analogWrite(in42,100); // motor 4 quay thuận
//////////
}
//*****TRÁI*****
if(state == 'L' || state == 'l'){

analogWrite(in21,100);
digitalWrite(in22,LOW);
//////////
analogWrite(in41,100);
digitalWrite(in42,LOW);
//*****
digitalWrite(in11,LOW);
analogWrite(in12,100);
//////////
```

```
digitalWrite(in31,LOW);
analogWrite(in32,100);
}
//*****PHẢI*****
if(state == 'R' || state == 'r'){
digitalWrite(in21,LOW);
analogWrite(in22,100);
//////////
digitalWrite(in41,LOW);
analogWrite(in42,100);
//////////
analogWrite(in11,100);
digitalWrite(in12,LOW);
//////////
analogWrite(in31,100);
digitalWrite(in32,LOW);
}
//*****STOP*****
if(state == 'S' || state == 's'){
dung();
}
// *****XOAY TRÁI*****
if(state == 'A' || state == 'a'){
analogWrite(in11,50); // motor1 quay thuận
digitalWrite(in12,LOW);
//////////
digitalWrite(in21,LOW); // motor 2 quay thuan
analogWrite(in22,50);
//////////
```

```
digitalWrite(in31,LOW); // motor 3 quay nghịch
analogWrite(in32,50);
//////////
analogWrite(in41,50); // motor4 quay nghịch
digitalWrite(in42,LOW);
}
//*****XOAY PHẢI*****
if(state =='I'||state =='i'){
digitalWrite(in11,LOW); // motor 1 quay nghịch
analogWrite(in12,50);
//////////
analogWrite(in21,50); // motor2 quay nghịch
digitalWrite(in22,LOW);
//////////
analogWrite(in31,50); // mtor 3 quay thuận
digitalWrite(in32,LOW);
//////////
digitalWrite(in41,LOW);
analogWrite(in42,50); // motor 4 quay thuận
}
//*****QUA TRÁI 45° *****
if(state == 'T' || state =='t'){
analogWrite(in21,100); // motor2 quay nghịch
digitalWrite(in22,LOW);
//////////
analogWrite(in41,100); // motor4 quay nghịch
digitalWrite(in42,LOW);
}
```

```
//*****QUA PHẢI 450 *****
if(state == 'P' || state == 'p'){
  analogWrite(in11,100); // motor1 quay thuận
  digitalWrite(in12,LOW);
  ///////////
  analogWrite(in31,100); // mtor 3 quay thuận
  digitalWrite(in32,LOW);
}
//*****LUI PHẢI 450 *****
if(state == 'H' || state == 'h'){

  ///////////
  analogWrite(in22,100); // motor2 quay nghịch
  digitalWrite(in21,LOW);
  ///////////
  analogWrite(in42,100); // motor4 quay nghịch
  digitalWrite(in41,LOW);
}
//*****LUI TRÁI 450 *****
if(state == 'K' || state == 'k'){
  analogWrite(in12,100); // motor1 quay thuận
  digitalWrite(in11,LOW);
  ///////////
  analogWrite(in32,100); // mtor 3 quay thuận
  digitalWrite(in31,LOW);
}
}
delay(10);
}
```

❖ **ROBOT TỰ ĐỘNG:**

```
// #include <LiquidCrystal.h>
#include <TinyGPS.h>
#include <Wire.h>
//wire pins are in my board( Mega2560) 20 (SDA), 21 (SCL)
#define address 0x1E //0011110b, I2C 7bit address of
HMC5883
TinyGPS gps;
#define rxPin 19 // to yellow wire in my gps
#define txPin 18
bool feedgps();
//int echoPin = 40; // pingpin to Digital 8
//int trigPin = 42; // inpin to Digital 9
//int SAFE_ZONE = 10;
//Motor direction control pin def
#define in1 8 // motor 1 RIGHT
#define in2 9
#define in4 6// motor 4 RIGHT
#define in5 7
#define in21 2// motor 2 LEFT
#define in22 3
#define in31 4// motor 3 LEFT
#define in32 5
// LiquidCrystal lcd(6 ,7 ,2 ,3, 4, 5); // define our LCD and
which pins to user
void setup()
```

```
{
  // lcd.begin(20, 4); // need to specify how many columns and
rows are in the LCD unit
  Serial.begin(9600);
  delay(100);          // Power up delay
  Serial.println("THULANA                      VIMUKTHI
ABEYWARDANA");
  Serial.println("Kotelawala Defence University");
  Serial.println("Department of Electrical and Electronic
Engineering");
  Serial.println("=====
=====");
  Serial.println("                      ");
  Serial1.begin(9600);
  Wire.begin();
  // Set operating mode to continuous in compass
  Wire.beginTransmission(address);
  Wire.write(byte(0x02));
  Wire.write(byte(0x00));
  Wire.endTransmission();
  pinMode(in11, OUTPUT); //R
  pinMode(in12, OUTPUT); //R
  pinMode(in41, OUTPUT); //R
  pinMode(in42, OUTPUT); //R
  pinMode(in21, OUTPUT); //L
  pinMode(in22, OUTPUT); //L
  pinMode(in31, OUTPUT); //L
  pinMode(in32, OUTPUT); //L
  // pinMode(echoPin, OUTPUT);
```

```
//pinMode(trigPin, INPUT);

}

/***** MOTOR 1 *****/
void motor1_thuan(){
    analogWrite(in11,100); // motor1 quay thuận
    digitalWrite(in12,LOW);
}
void motor1_nghich(){
    digitalWrite(in11,LOW); // motor 1 quay nghịch
    analogWrite(in12,100);
}

/***** MOTOR 2 *****/
void motor2_thuan(){
    digitalWrite(in21,LOW); // motor 2 quay thuận
    analogWrite(in22,100);
}
void motor2_nghich(){
    analogWrite(in21,100); // motor2 quay nghịch
    digitalWrite(in22,LOW);
}

*****/
void motor3_thuan(){
    analogWrite(in31,100); // mtor 3 quay thuận
    digitalWrite(in32,LOW);
}
void motor3_nghich(){
    digitalWrite(in31,LOW); // motor 3 quay nghịch
    analogWrite(in32,100);}
```

```

/***** MOTOR 4 *****/
void motor4_thuan(){
    digitalWrite(in41,LOW);
    analogWrite(in42,100); // motor 4 quay thuận
}
void motor4_nghich(){
    analogWrite(in41,100); // motor4 quay nghịch
    digitalWrite(in42,LOW);
}
/*****/

void chaytoi(){
    motor1_thuan();
    motor3_thuan();
    motor2_nghich();
    motor4_nghich();
}

void chaylui(){
    motor1_nghich();
    motor3_nghich();
    motor2_thuan();
    motor4_thuan();
}

void xoay_trai(){
    analogWrite(in11,250); // motor1 quay thuận
    digitalWrite(in12,LOW);
    digitalWrite(in21,LOW); // motor 2 quay thuận
    analogWrite(in22,250);
    digitalWrite(in31,LOW); // motor 3 quay nghịch
    analogWrite(in32,250);

```

```
analogWrite(in41,250); // motor4 quay ngược
digitalWrite(in42,LOW);
}
void xoay_phai(){
digitalWrite(in11,LOW); // motor 1 quay ngược
analogWrite(in12,40);
analogWrite(in21,40); // motor2 quay ngược
digitalWrite(in22,LOW);
analogWrite(in31,40); // motor 3 quay thuận
digitalWrite(in32,LOW);
digitalWrite(in41,LOW);
analogWrite(in42,40); // motor 4 quay thuận
}
/*****/
void forward()
{
    chaytoi();
}
void Backward()
{
    chaylui();
}
void search()
{
    xoay_trai();
}
void Stop()
{
    digitalWrite(in11,LOW);
```

```
digitalWrite(in12,LOW);
digitalWrite(in21,LOW);
digitalWrite(in22,LOW);
digitalWrite(in31,LOW);
digitalWrite(in32,LOW);
digitalWrite(in41,LOW);
digitalWrite(in42,LOW);
}
void right()
{
    analogWrite(in21,50); // motor2 quay nghịch
    digitalWrite(in22,LOW);
    analogWrite(in31,50); // mtor 3 quay thuận
    digitalWrite(in32,LOW);
}
void left()
{
    analogWrite(in11,50); // motor1 quay thuận
    digitalWrite(in12,LOW);
    analogWrite(in41,50); // motor4 quay nghịch
    digitalWrite(in42,LOW);
}
/*****/
void loop(){
    // long cm=getDistance();
    feedgps();
    // getDistance();
    unsigned char cc = Serial1.read();
    float flat, flon,x2lat,x2lon;
```

```

    unsigned long age;
    gps.f_get_position(&flat, &flon, &age);
    feedgps();
    //getDistance();

    float flat1=flat;    // flat1 = our current latitude. flat is from
the gps data.
    float flon1=flon; // flon1 = our current longitude. flon is from
the fps data.
    float dist_calc=0;
    float angle_calc=0;
    float dist_calc2=0;
    float diflat=0;
    float diflon=0;
    //10.810834,                      106.675220//10.810859,
106.674847//10.810843,                106.675230//10.810352,
106.674885//10.810489, 106.675200//10.810636, 106.675223
    x2lat= 10.810636; //enter a latitude point here  this is going
to be your waypoint(here KURUNEGALA)
    x2lon=106.675223; // enter a longitude point here  this is
going to be your waypoint
    //----- distance formula below.
Calculates distance from current location to waypoint
    dist_calc=sqrt((((flon1)-(x2lon))*((flon1)-(x2lon)))+(((x2lat-
flat1)*(x2lat-flat1))));
    dist_calc*=110567 ; //Converting to meters
    //=====angle=====//
    angle_calc=atan2((x2lon-flon1),(x2lat-flat1)); // tính góc

    float declinationAngle2 = 57.29577951; // góc đo lệch

```

```
angle_calc*= declinationAngle2;
feedgps();
//getDistance();
if(angle_calc < 0){ // goc do lech <0
    angle_calc = 360 + angle_calc;
    feedgps();
    //getDistance();
}
// Check for wrap due to addition of declination.
if(angle_calc >0){
    angle_calc= angle_calc;
    feedgps();
    //getDistance();
}
float angleDegrees = angle_calc; //goc do
feedgps();
int x, y, z;
//getDistance();
// Bắt đầu truyền thông với loa bàn
Wire.beginTransmission(address);
Wire.write(byte(0x03));    // Send request to X MSB register
Wire.endTransmission();
Wire.requestFrom(address, 6); // Request 6 bytes; 2 bytes per
axis
if(6<=Wire.available()) { // If 6 bytes available
    x = Wire.read()<<8; //X msb
    x |= Wire.read(); //X lsb
    z = Wire.read()<<8; //Z msb
    z |= Wire.read(); //Z lsb
```

```
y = Wire.read() << 8; //Y msb
y |= Wire.read(); //Y lsb
feedgps();
}
float heading = atan2(y,x); // Hướng
float declinationAngle = 0.0457;
heading += declinationAngle; // hướng = hướng + độ lệch
hướng
feedgps();
//getDistance();
// Correct for when signs are reversed.
if(heading < 0){
    heading += 2*PI;
    feedgps();
    //getDistance();
}
// Check for wrap due to addition of declination.
if(heading > 2*PI){
    heading -= 2*PI;
    feedgps();
    //getDistance();
}
float headingDegrees = heading*180/M_PI;
feedgps();
//getDistance();
Serial.print("Lat: "); // vĩ độ
Serial.print(flat,8);
Serial.print(", Long: "); // kinh độ
Serial.println(flon,8);
```

```
Serial.println("distance");
Serial.println(dist_calc,8); //print the distance in meters

//compass=====
=====
Serial.print("Heading:\t"); // hướng
Serial.print(headingDegrees);
Serial.println(" Degrees \t"); // độ
Serial.println("AngleDegrees");
Serial.print(angleDegrees);
Serial.println(" Degrees \t");
// Serial.print(cm);
//Serial.print("cm");
Serial.println(" ");
delay(500);
*****

// Bắt đầu hoạt động//
bool condition = ((angleDegrees -100) < headingDegrees)
&& ((angleDegrees +10) > headingDegrees);
if(condition){ //condition == true?
    Serial.println("MOVE");
    feedgps();
Serial.println("move");
    forward();
    long stTime = millis();
    while(millis()-stTime < 4000){
        feedgps();
        feedgps();
    }
}
```

```
    }
    if(dist_calc<4)
    {
        feedgps();
        Stop();
        Serial.println("    stop    "); }
    else {
        search();
        feedgps();
        delay(20);
        Stop();
        feedgps();
        delay(30);
    }
}

void myDelay(long duration)
{
    long stTime = millis();
    while(millis() - stTime <duration){
        feedgps();
        delay(1);}
}

static bool feedgps(){
    while (Serial1.available()) {
        if (gps.encode(Serial1.read()))
            return true;  }
    return false;
}
```

*****KẾT THÚC*****

BẢNG CHÚ THÍCH CÁC CHÂN CỦA ATMEGE 2560

Số Pin	Tên Pin	Ảnh xạ Tên Pin
1	PG5 (OC0B)	Kỹ thuật số pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Pin kỹ thuật số 0 (RX0)
3	PE1 (TXD0)	Pin kỹ thuật số 1 (TX0)
4	PE2 (XCK0 / AIN0)	
5	PE3 (OC3A / AIN1)	Pin kỹ thuật số 5 (PWM)
6	PE4 (OC3B / INT4)	Kỹ thuật số pin 2 (PWM)
7	PE5 (OC3C / INT5)	Pin Digital 3 (PWM)
8	PE6 (Q3 / INT6)	
9	PE7 (CLKOUT / ICP3 / INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Kỹ thuật số pin 17 (RX2)
13	PH1 (TXD2)	Kỹ thuật số pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Pin kỹ thuật số 6 (PWM)
16	PH4 (OC4B)	Kỹ thuật số pin 7 (PWM)
17	PH5 (OC4C)	Kỹ thuật số pin 8 (PWM)
18	PH6 (OC2B)	Kỹ thuật số pin 9 (PWM)
19	PB0 (SS / PCINT0)	Kỹ thuật số pin 53 (SS)
20	PB1 (SCK / PCINT1)	Kỹ thuật số pin 52 (SCK)
21	PB2 (MOSI / PCINT2)	Kỹ thuật số pin 51 (MOSI)
22	PB3 (MISO / PCINT3)	Kỹ thuật số pin 50 (MISO)
23	PB4 (OC2A / PCINT4)	Pin kỹ thuật số 10 (PWM)
24	PB5 (OC1A / PCINT5)	Kỹ thuật số pin 11 (PWM)
25	PB6 (OC1B / PCINT6)	Kỹ thuật số pin 12 (PWM)
26	PB7 (OC0A / OC1C / PCINT7)	Kỹ thuật số pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	Thiết lập lại	Thiết lập lại
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2

34	XTAL1	XTAL1
35	PL0 (ICP4)	Kỹ thuật số pin 49
36	PL1 (ICP5)	Kỹ thuật số pin 48
37	PL2 (T5)	Kỹ thuật số pin 47
38	PL3 (OC5A)	Kỹ thuật số pin 46 (PWM)
39	PL4 (OC5B)	Kỹ thuật số pin 45 (PWM)
40	PL5 (OC5C)	Kỹ thuật số pin 44 (PWM)
41	PL6	Kỹ thuật số pin 43
42	PL7	Kỹ thuật số pin 42
43	PD0 (SCL / INT0)	Kỹ thuật số pin 21 (SCL)
44	PD1 (SDA / INT1)	Kỹ thuật số pin 20 (SDA)
45	PD2 (RXDI / INT2)	Kỹ thuật số pin 19 (RX1)
46	PD3 (TXD1 / INT3)	Kỹ thuật số pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Kỹ thuật số pin 38
51	PG0 (WR)	Kỹ thuật số pin 41
52	PG1 (RD)	Kỹ thuật số pin 40
53	PC0 (A8)	Kỹ thuật số pin 37
54	PC1 (A9)	Kỹ thuật số pin 36
55	PC2 (A10)	Kỹ thuật số pin 35
56	PC3 (A11)	Kỹ thuật số pin 34
57	PC4 (A12)	Kỹ thuật số pin 33
58	PC5 (A13)	Kỹ thuật số pin 32
59	PC6 (A14)	Kỹ thuật số pin 31
60	PC7 (A15)	Kỹ thuật số pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Kỹ thuật số pin 15 (RX3)
64	PJ1 (TXD3 / PCINT10)	Kỹ thuật số pin 14 (TX3)
65	PJ2 (XCK3 / PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Kỹ thuật số pin 39

71	PA7 (AD7)	Kỹ thuật số pin 29
72	PA6 (AD6)	Kỹ thuật số pin 28
73	PA5 (AD 5)	Kỹ thuật số pin 27
74	PA4 (AD4)	Kỹ thuật số pin 26
75	PA3 (AD3)	Kỹ thuật số pin 25
76	PA2 (tuyến AD2)	Kỹ thuật số pin 24
77	PA1 (AD1)	Kỹ thuật số pin 23
78	PA0 (AD0)	Kỹ thuật số pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15 / PCINT23)	Analog pin 15
83	PK6 (ADC14 / PCINT22)	Analog pin 14
84	PK5 (ADC13 / PCINT21)	Analog pin 13
85	PK4 (ADC12 / PCINT20)	Analog pin 12
86	PK3 (ADC11 / PCINT19)	Analog pin 11
87	PK2 (ADC10 / PCINT18)	Analog pin 10
88	PK1 (ADC9 / PCINT17)	Analog pin 9
89	PK0 (ADC8 / PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5 / TMS)	Analog pin 5
93	PF4 (ADC4 / TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	Aref	Analog Reference
99	GND	GND
100	AVCC	VCC

TÀI LIỆU THAM KHẢO

- [1] Tài liệu “ động học mobile robot di chuyển trên mặt phẳng”
- [2] Các diễn đàn trên mạng: <http://doc.edu.vn/>, <http://doantotnghiep.vn/>,
<http://codientu.org/>, <http://arduino360.com/>, ...
- [3] Mua linh kiện điện tử, động cơ ở <http://hshop.vn/>, <http://roboconshop.com/>,
<http://machtudong.vn/>, ...
- [4] Giáo trình Solidworks 2013 và giáo trình Arduino tham khảo trên
<http://codientu.org/>