

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA ĐIỆN – ĐIỆN TỬ



ĐỒ ÁN TỐT NGHIỆP
NGHIÊN CỨU THIẾT KẾ MÔ HÌNH
ROBOT LAU NHÀ THÔNG MINH

Giảng viên hướng dẫn : **TS TRẦN VĂN HÙNG**

Sinh viên thực hiện: **NGUYỄN CÔNG TÍNH**

Lớp : **Kỹ thuật điện tử và tin học công nghiệp**

Khoá : **55**

Hà Nội, tháng 12 năm 2018

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA ĐIỆN – ĐIỆN TỬ



ĐỒ ÁN TỐT NGHIỆP
NGHIÊN CỨU THIẾT KẾ MÔ HÌNH
ROBOT LAU NHÀ THÔNG MINH

Giảng viên hướng dẫn : **TS TRẦN VĂN HÙNG**

Sinh viên thực hiện: **NGUYỄN CÔNG TÍNH**

Lớp : **Kỹ thuật điện tử và tin học công nghiệp**

Khoá : **55**

Hà Nội, tháng 12 năm 2018

LỜI MỞ ĐẦU

Cùng với sự phát triển nhanh chóng của các ngành khoa học kỹ thuật, công nghệ chế tạo robot cũng phát triển theo. Robot ngày càng gia tăng về số lượng chủng loại, lẫn chất lượng nhằm đáp ứng nhu cầu tự động hoá trong các quy trình sản xuất, các nhu cầu thám hiểm ... cũng như phục vụ đời sống con người.

Trong xu thế sử dụng robot vào sản xuất và sinh hoạt ngày càng nhiều. Ở Việt Nam, ngành công nghiệp robot cũng đang từng bước phát triển. Tuy nhiên các robot này lại đa phần chỉ sản xuất dưới dạng người máy công nghiệp, tay máy hoặc những mô hình máy phục vụ cho nhu cầu giải trí. Trong khi đó, loại robot di động vẫn chưa được chú ý nhiều, mặc dù đây là loại robot có tiềm năng rất lớn, có thể dùng trong nhiều mục đích khác nhau như: giải trí trong gia đình, phục vụ trong các công sở, trong các mục đích quân sự, và dùng nhiều trong thám hiểm vũ trụ. Tiêu biểu trong nhóm này có thể kể đến như: chó robot Aibo, xe tự hành Sojourner thám hiểm sao hỏa và các loại robot dò mìn...

Robot di động là một lĩnh vực có nhiều tiềm năng, đặc biệt trong việc trợ giúp con người làm những công việc nhà. Nên em quyết định chọn đề tài “**Nghiên cứu thiết kế mô hình robot lau nhà thông minh**” làm đề tài nhiệm vụ thiết kế tốt nghiệp của em.

Nội dung đề tài gồm các phần như sau:

CHƯƠNG 1: KHÁI QUÁT VỀ ROBOT LAU NHÀ TỰ ĐỘNG.

CHƯƠNG 2: THIẾT KẾ MÔ HÌNH ROBOT LAU NHÀ THÔNG MINH

CHƯƠNG 3: XÂY DỰNG PHẦN MỀM ĐIỀU KHIỂN

KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

Em xin chân thành cảm ơn quý thầy cô trong Khoa **Điện-Điện Tử** đã giảng dạy và trang bị cho em từ những kiến thức cơ bản đến chuyên môn. Đặc biệt em xin thành cảm ơn Thầy TS.**Trần Văn Hưng**, đã tận tình hướng dẫn, giúp đỡ em trong quá trình nghiên cứu và thực hiện đề tài này.

Sinh viên

Nguyễn Công Tính

MỤC LỤC

LỜI MỞ ĐẦU	
CHƯƠNG 1: KHÁI QUÁT VỀ ROBOT LAU NHÀ TỰ ĐỘNG.....	1
1.1 Giới thiệu chung về robot	1
1.2 Tổng quan về Robot lau nhà	2
1.3 Các yêu cầu của hệ thống robot lau nhà thông minh.....	4
1.4 Lựa chọn phương án thiết kế robot lau nhà đơn giản	5
1.5 Kết luận chương	5
CHƯƠNG 2: THIẾT KẾ MÔ HÌNH ROBOT LAU NHÀ THÔNG MINH	6
2.1 Sơ đồ khối chung của hệ thống.....	6
2.2 Các phương pháp điều hướng cho robot.....	7
2.2.1 Phương pháp điều hướng có tính toán	7
2.2.2 Phương pháp điều hướng theo phản ứng.....	7
2.3 Thiết kế mô hình cơ khí cho robot	8
2.4 Phần mạch điện tử.....	12
2.4.1 Khối điều khiển và giới thiệu PIC16F877A	12
2.4.2 Khối nguồn	28
2.4.3 Hệ thống cảm biến vật cản.....	32
2.4.4 Khối cảm biến độ cao	33
2.4.5 Bàn phím.....	34
2.4.6 Còi báo	35
2.4.7 Khối ADC đo điện áp pin.....	35
2.4.8 Khối điều khiển motor chổi quét.....	36
2.4.9 Mạch điều khiển động cơ chính.....	37
2.4.10 Thiết kế mạch in – PCB (Printed Circuit Board) bằng Altium Designer.....	42

CHƯƠNG 3: XÂY DỰNG PHẦN MỀM ĐIỀU KHIỂN	46
3.1 Lập trình cho hệ thống thời gian thực	46
3.2 Sử dụng trình biên dịch CCS.....	47
3.3 Lưu đồ thuật toán điều khiển robot.....	49
3.3.1 Lưu đồ thuật toán chung.....	49
3.3.2 Lưu đồ thuật toán chế độ bật tắt bằng tay	50
3.3.3 Lưu đồ thuật toán chế độ hẹn giờ	51
3.4 Sử dụng ADC để đo điện áp pin	52
3.5 Cài đặt bộ hẹn giờ bằng Timer	52
3.6 Bài toán di chuyển của robot	53
3.7 Tránh vật cản và cầu thang	55
3.8 Xử lý bàn phím.....	56
3.9 Kết luận chương.....	57
KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN	59
A. Kiểm tra trước khi hoạt động	59
B. Chạy thử mô hình.....	61
C. Kết luận	64
Kết quả đạt được:	64
Các điểm hạn chế.....	65
D. Hướng phát triển cho đề tài.....	66
PHỤ LỤC.....	68
TÀI LIỆU THAM KHẢO	78

DANH MỤC BẢNG BIỂU

Bảng 2.1: Cấu hình Port A.....	15
Bảng 2.2: Cấu hình Port B.....	15
Bảng 2.3: Cấu hình Port C.....	16
Bảng 2.4: Cấu hình Port D.....	17
Bảng 2.5: Cấu hình Port E.....	18
Bảng 2.6: Cấu hình Prescale cho Timer0.....	20
Bảng 2.7: Bảng cấu hình khối CCP.....	24
Bảng 2.8: Bảng cấu hình xung nhịp cho ADC.....	25
Bảng 2.9: Bảng mô tả các tài nguyên vi xử lý được sử dụng.....	27
Bảng 2.10: Mô tả IC L298.....	40

DANH MỤC HÌNH ẢNH

Hình 1.1: Robot tự hành Rover thám hiểm sao Hoả.....	2
Hình 1.2: Robot dò thám Dragon Runner.....	2
Hình 1.3: Robot hút bụi Ilife A4s.....	3
Hình 1.4: Robot lau nhà iRobot Braava 380t.....	4
Hình 2.1: Sơ đồ khối các thành phần cơ bản của robot.....	6
Hình 2.2: Sơ đồ cấu trúc của phương pháp điều khiển có tính toán.....	7
Hình 2.3: Mô hình phương pháp điều hướng theo phản ứng.....	8
Hình 2.4: Thiết kế cơ khí của robot.....	9
Hình 2.5: Hình ảnh chổi quét của robot.....	10
Hình 2.6: Hình ảnh động cơ JGA25-370.....	11
Hình 2.7: Hình ảnh bánh xe được sử dụng trong robot.....	11
Hình 2.8: Sơ đồ chân của PIC16F877A loại cắm 40 chân.....	13
Hình 2.9: Sơ đồ khối của PIC16F877A.....	14
Hình 2.10: Sơ đồ khối Timer0.....	19
Hình 2.11: Sơ đồ khối Timer1.....	20
Hình 2.12: Sơ đồ khối Timer2.....	22
Hình 2.13: Sơ đồ khối PWM.....	23
Hình 2.14: Xung ra từ khối PWM.....	23
Hình 2.15: Sơ đồ khối ADC.....	25
Hình 2.16: Sơ đồ khối ổn áp 5v.....	28
Hình 2.17: Sơ đồ chân của LM2596S.....	29
Hình 2.18: Sơ đồ khối bên trong của LM2696S.....	29
Hình 2.19: Sơ đồ nguyên lý mạch nguồn dùng LM2596.....	30

Hình 2.20: Hình ảnh pin 18650.....	31
Hình 2.21: Sơ đồ nguyên lý của adapter.....	31
Hình 2.22: Công tắc hành trình.....	32
Hình 2.23: Sơ đồ bộ phận cảm nhận vật cản.....	33
Hình 2.24: Sơ đồ nguyên lý module thu phát hồng ngoại MH-IR01.....	34
Hình 2.25: Sơ đồ bàn kết nối bàn phím.....	34
Hình 2.26: Sơ đồ ghép nối loa.....	35
Hình 2.27: Sơ đồ mắc điện trở phân áp pin.....	35
Hình 2.28: Mô tả IRF540.....	36
Hình 2.29: Sơ đồ mạch điều khiển động cơ chổi quét.....	36
Hình 2.30: Mạch cầu H.....	37
Hình 2.31: Nguyên lý hoạt động của mạch cầu H.....	38
Hình 2.32: Sơ đồ chân IC L298.....	39
Hình 2.33: Sơ đồ khối IC L298.....	40
Hình 2.34: Module L298.....	41
Hình 2.35: Giao diện làm việc của Altium 17.....	43
Hình 2.36: Sơ đồ nguyên lý khối điều khiển.....	45
Hình 2.37: Sơ đồ mạch in khối điều khiển.....	45
Hình 3.1: Giao diện làm việc của CCS.....	48
Hình 3.2: Lưu đồ thuật toán của robot.....	49
Hình 3.3: Lưu đồ thuật toán chế độ Manual.....	50
Hình 3.4: Lưu đồ thuật toán chế độ Time.....	51
Hình 3.5: Lưu đồ thuật toán tránh vật cản và cầu thang.....	56
Hình 3.6: Lưu đồ chương trình kiểm tra nút.....	57

Hình 4.1: Kiểm tra nguồn Pin.....	59
Hình 4.2: Kiểm tra khối công suất.....	60
Hình 4.3: Kiểm tra phím nhấn.....	60
Hình 4.4: Vị trí bắt đầu chạy của robot.....	61
Hình 4.5: Robot gặp vật cản và dừng lại.....	61
Hình 4.6: Robot di chuyển sang hướng khác.....	62
Hình 4.7: Robot đi thẳng.....	62
Hình 4.8: Robot gặp cạnh bàn.....	63
Hình 4.9: Robot dừng và lùi.....	63
Hình 4.10: Rác robot thu được.....	64

KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT

STT	Từ viết tắt	Từ tiếng Anh	Nghĩa tiếng Việt
1	AC	Alternating Current	Dòng điện xoay chiều
2	ADC	Analog-to-digital converter	Bộ đổi tương tự ra số
3	BJT	Bipolar junction transistor	Tranzito lưỡng cực
4	CCP	Capture/Compare/PWM	
5	CMOS	Complementary Metal-Oxide-Semiconductor	một loại công nghệ dùng để chế tạo mạch tích hợp
6	CPU	Central Processing Unit	bộ xử lý trung tâm
7	DC	Direct Current	Dòng điện 1 chiều
8	GND	Ground	Tiếp địa
9	MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor	Tranzito trường
10	NC	Normal Close	Tiếp điểm thường đóng
11	NO	Normal Open	Tiếp điểm thường mở
12	PIC	Programmable Intelligent Computer	Máy tính khả trình thông minh
13	PVC	Polyvinylclorua	Một loại nhựa nhiệt dẻo
14	PWM	Pulse-width modulation	Điều chế độ rộng xung
15	RISC	Reduced Instructions Set Computer	Máy tính với tập lệnh đơn giản hóa
16	VCC	Voltage collector to collector	Chân cấp nguồn chính
17	VĐK		Vi điều khiển

CHƯƠNG 1: KHÁI QUÁT VỀ ROBOT LAU NHÀ TỰ ĐỘNG

1.1 Giới thiệu chung về robot

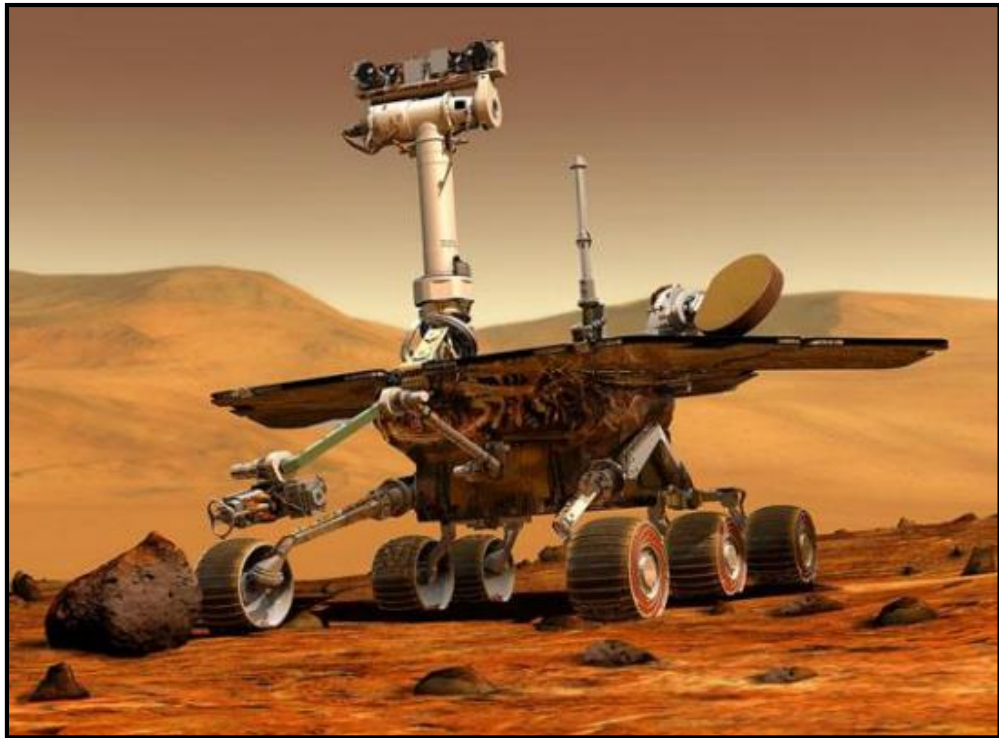
Robot hay người máy là một loại máy có thể thực hiện những công việc một cách tự động bằng sự điều khiển của máy tính hoặc các vi mạch điện tử được lập trình. Robot là một tác nhân cơ khí, nhân tạo, ảo, thường là một hệ thống cơ khí-điện tử. Với sự xuất hiện và chuyển động của mình, robot gây cho người ta cảm giác rằng nó giác quan giống như con người. Từ "robot" (người máy) thường được hiểu với hai nghĩa: robot cơ khí và phần mềm tự hoạt động. Về lĩnh vực người máy, Nhật Bản là nước đi đầu thế giới về lĩnh vực này.

Khái niệm Robot xuất hiện lần đầu tiên ở New York vào ngày 9/10/1922 trong vở kịch "Rossum's Universal Robot" của nhà soạn kịch người Tiệp Khắc là Karen Chapek, còn từ Robot là một cách gọi khác của từ Robota-theo tiếng Tiệp có nghĩa là công việc lao dịch. Khi đó, Karen Chapek cho rằng Robot là những người máy có khả năng làm việc nhưng không có khả năng suy nghĩ. Gần một thế kỷ tiếp theo, khái niệm robot đã liên tục được phát triển, đóng góp thêm bởi nhiều nhà nghiên cứu, nhiều công ty chuyên về lĩnh vực robot.

Trước những năm 1970, người ta chỉ tập trung vào việc phát triển những robot tay máy hoạt động trong các nhà máy công nghiệp. Sau đó mới xuất hiện những khái niệm về robot thông minh, và các nghiên cứu bắt đầu tập trung hơn vào robot di động. Một trong những chuyên gia đầu ngành về robot di động là Hans P. Moravec (bắt đầu nghiên cứu từ năm 1964), và hiện nay, chuyên nghiên cứu về robot di động là Sebastien Thrun.

Các robot di động có người điều khiển đã được dùng cho các mục đích quân sự, các nhiệm vụ nguy hiểm như phá mìn, thăm dò đáy đại dương, hầm mỏ, kiểm tra các đường ống ngầm, hay thăm dò sao Hỏa... Sản phẩm robot di động được sản xuất đại trà và đưa vào thị trường lần đầu tiên là robot hút bụi Roomba và Trilobite của hãng Electrolux năm 2003.

Một số hình ảnh về các Robot di động và ứng dụng của nó:



Hình 1.1: Robot tự hành Rover thám hiểm sao Hoả



Hình 1.2: Robot do thám Dragon Runner

1.2 Tổng quan về Robot lau nhà

Hiện nay, robot lau nhà thông minh đang dần trở thành xu thế mới, thay thế các loại máy hút bụi truyền thống để hỗ trợ con người trong việc dọn dẹp nhà cửa. Tại Việt Nam, dòng sản phẩm này đã có mặt trên thị trường và hứa hẹn sẽ trở thành

một sản phẩm hot trong thời gian tới. Robot lau nhà tự động được xem là “người” giúp việc vô cùng thông minh và hiệu quả cho các gia đình, đặc biệt đối với những người nữ nội trợ phải đi làm suốt cả ngày, không có thời gian chăm sóc nhà cửa. Robot thiết kế thông minh, có thể tự động thực hiện công việc hút bụi, lau dọn nhà mà không cần có sự hướng dẫn của con người. Kích thước của robot hút bụi tự động nhỏ gọn, có thể đi đến tất cả các góc ngách như: gầm bàn, gầm giường, ghế sofa, góc tường... để làm sạch toàn diện. Hơn nữa, robot thông minh cũng có thể sử dụng được trên nhiều loại sàn nhà như: gỗ, thảm, sàn gạch, gạch ceramic...

Trên thị trường hiện nay có 3 loại robot làm sạch sàn nhà đó là sản phẩm Robot lau nhà và Robot hút bụi, với mỗi loại Robot lau nhà hay robot hút bụi thì đều có những ưu điểm và nhược điểm riêng chính vì vậy để tăng hiệu quả làm sạch một cách tối đa các nhà sản xuất đã kết hợp hai loại Robot lau nhà và hút bụi lại thành loại Robot vừa có khả năng lau nhà vừa có khả năng hút bụi. Tuy nhiên xét về sự phổ biến thì hiện nay thì robot hút bụi vẫn được sử dụng nhiều hơn vì có giá thành hợp lý nhưng vẫn giữ được nhiều ưu điểm lớn là có thể làm sạch trên nhiều bề mặt sàn khác nhau, có thể làm sạch nhiều loại bụi bẩn. Dưới đây Hình 1.3 là sản phẩm Robot lau nhà Ilife V4s, sản phẩm có thiết kế dạng khối trụ tròn khá thanh thoát.



Hình 1.3: Robot hút bụi Ilife A4s

Mặc dù Robot hút bụi có nhiều ưu điểm vượt trội và được sử dụng rộng rãi hơn nhưng cũng không thể phủ nhận những điểm tích cực của Robot lau nhà vẫn

chiếm được lòng tin của rất nhiều người dùng. Mặc dù không hoạt động tốt trên nhiều bề mặt nhưng hiệu quả làm sạch của Robot lau nhà có phần vượt trội so với Robot hút bụi, Robot lau nhà có thể làm sạch được cả những vết bẩn ướt, dính với bề mặt sàn. Bên cạnh đó Robot lau nhà hoạt động không gây nhiều tiếng ồn như Robot hút bụi, ít gây ảnh hưởng đến người sử dụng vào buổi đêm. Hình 1.4 là Robot lau nhà iRobot Braava 380t, robot có dạng hình vuông, cải thiện diện tích tiếp xúc của tấm rửa lau được gắn dưới đáy robot với mặt sàn giúp tăng hiệu quả làm sạch.



Hình 1.4: Robot lau nhà iRobot Braava 380t

Ngoài hai loại Robot kể trên hiện nay các nhà sản xuất đã tích hợp hai loại Robot trên vào cùng một sản phẩm vừa có khả năng lau nhà, vừa có khả năng hút bụi tuy nhiên giá thành của chúng rất cao, đó là một rào cản lớn với người dùng nếu muốn sở hữu chúng.

1.3 Các yêu cầu của hệ thống robot lau nhà thông minh

Yêu cầu cơ bản của một robot hút bụi là có thể tự động làm sạch trên nhiều chất liệu mặt sàn khác nhau một cách tối đa, có thể đi vào các góc ngách trong nhà, dễ dàng thao tác sử dụng, hoạt động với tiếng ồn nhỏ, không làm ảnh hưởng tới thú nuôi và vật dụng trong nhà.

Trong đồ án này em thiết kế Robot lau nhà đơn giản với chi phí thấp. Sản phẩm gồm các tính năng cơ bản sau:

- Đảm bảo có thể di chuyển gần hết diện tích mặt sàn
- Đảm bảo có thể lau, quét bụi bẩn, rác nhỏ trên mặt sàn
- Đảm bảo dễ dàng sử dụng.
- Đảm bảo robot hoạt động tốt, trơn tru, không làm ảnh hưởng, gây hỏng hóc cho các thiết bị khác trong nhà, văn phòng...
- Đảm bảo phần điện ổn định, không rò rỉ, gây ảnh hưởng tới con người, thú nuôi
- Có khả năng tránh vật cản trực diện.

1.4 Lựa chọn phương án thiết kế robot lau nhà đơn giản

Thiết kế một mô hình robot lau nhà có khả năng làm sạch mặt sàn, tự động tránh vật cản, tránh bị rơi khi gặp cầu thang, hoạt động theo thời gian định sẵn dựa trên những linh kiện phù hợp với túi tiền.

1.5 Kết luận chương

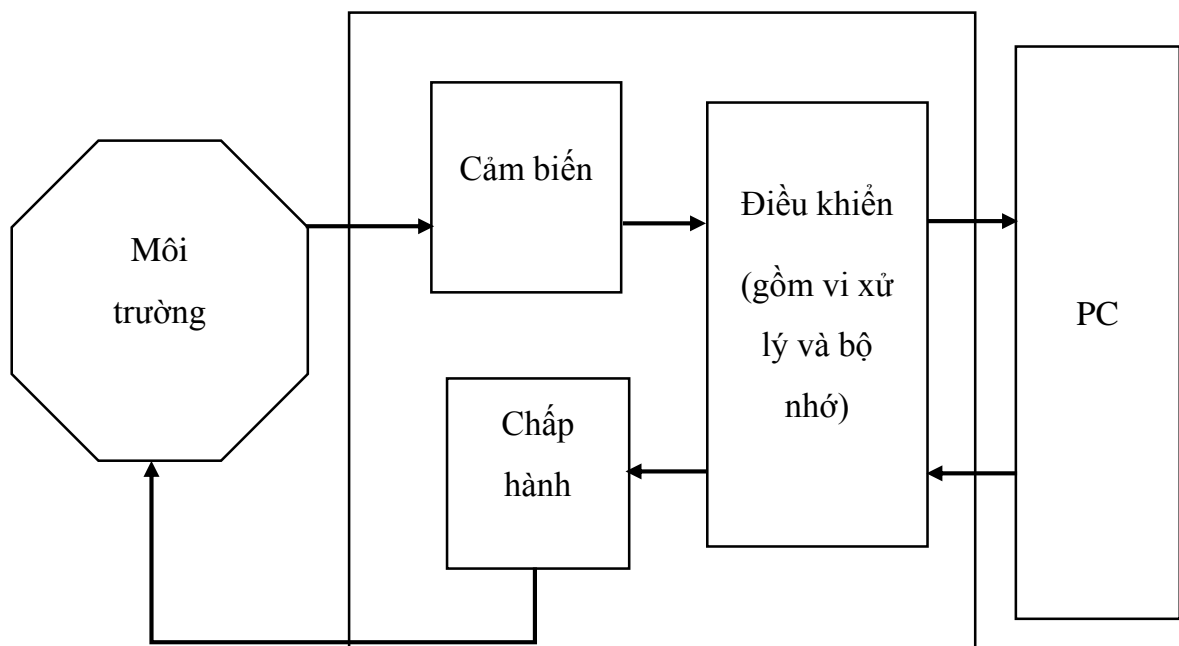
Như vậy, trong chương này đồ án đã giới thiệu một cách tổng quan về robot lau nhà tự động. Qua đó ta có thể hiểu một cách sơ bộ nhất về cấu tạo cũng như nguyên lý hoạt động của robot được thiết kế cho mục đích lau nhà tự động. Đồng thời cũng phân tích các yêu cầu của một robot lau nhà thông minh. Trên cơ sở đó cũng lựa chọn phương án thiết kế robot lau nhà đơn giản. Từ đó, đồ án có thể đưa ra yêu cầu, phương hướng triển khai và các thành phần bắt buộc trong việc sử dụng vi xử lý vào việc thiết kế mô hình robot lau nhà thông minh. Trong các chương tiếp theo đồ án sẽ tiếp tục giới thiệu rõ hơn về phần lý thuyết của từng modul cũng như nguyên lý hoạt động của chúng.

CHƯƠNG 2: THIẾT KẾ MÔ HÌNH ROBOT LAU NHÀ THÔNG MINH

2.1 Sơ đồ khối chung của hệ thống

Một robot thường gồm 3 bộ phận: bộ phận cảm biến, bộ phận điều khiển và bộ phận chấp hành.

- Bộ phận cảm biến: thu nhận thông tin từ môi trường. Bao gồm các sensor nội và sensor ngoại.
- Bộ phận điều khiển: nhận thông tin từ bộ phận cảm biến, từ đó đưa ra tín hiệu điều khiển bộ phận chấp hành. Bao gồm vi xử lý và bộ nhớ.
- Bộ phận chấp hành: nhận thông tin từ bộ điều khiển, thực hiện hành động của robot. Trong nhiều trường hợp, bộ phận chấp hành nhận thông tin trực tiếp từ bộ phận cảm biến.



Hình 2.1: Sơ đồ khối các thành phần cơ bản của robot

Trong hệ thống này, robot sẽ di chuyển liên tục và làm sạch rác trên mặt sàn. Trong quá trình di chuyển, bộ phận cảm biến sẽ nhận biết các chướng ngại vật hoặc cầu thang cản trở robot di chuyển và truyền thông tin về khối điều khiển. Sau khi nhận được thông tin có chướng ngại vật hoặc cầu thang từ khối cảm biến, khối điều khiển sẽ đưa ra phương án di chuyển tới khối chấp hành để giúp robot có thể tránh

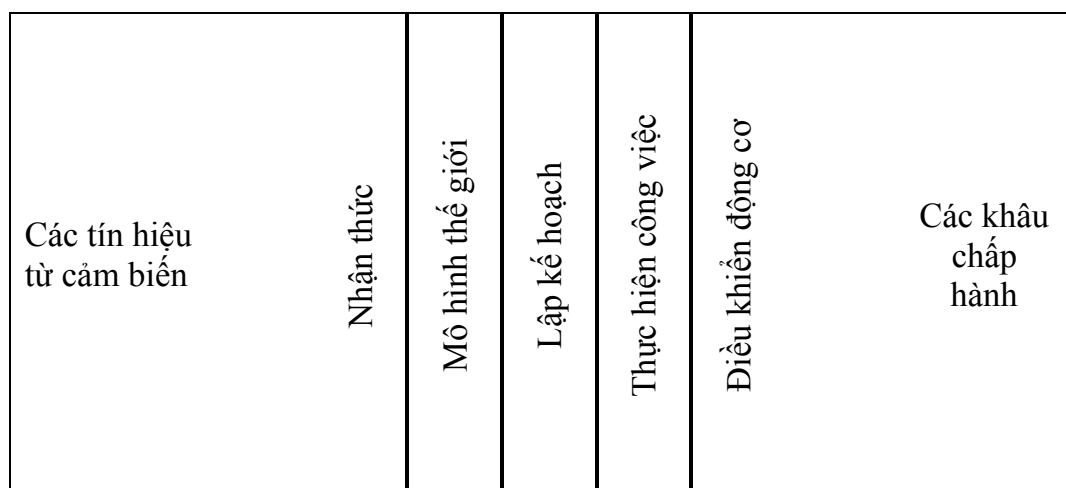
được chướng ngại vật hoặc cầu thang và tiếp tục dọn dẹp. Quá trình được lặp đi lặp lại

2.2 Các phương pháp điều hướng cho robot

2.2.1 Phương pháp điều hướng có tính toán

Phương pháp điều hướng có tính toán là phương pháp thực hiện theo trình tự: quan sát – lập kế hoạch – hành động. Thông thường một hệ thống có tính toán bao gồm 5 khâu: nhận thức (perception), mô hình thể giới (world modelling), lập kế hoạch (planning), thực hiện công việc (task execution) và điều khiển động cơ (motor control).

Các khâu trên có thể được coi như là một chuỗi các “lát mỏng theo phương thẳng đứng” với các đầu vào là tín hiệu nhận được từ cảm biến ở phía bên trái và đầu ra tới các khâu chấp hành ở phía bên phải.



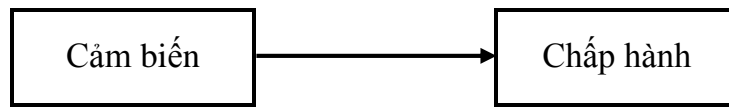
Hình 2.2: Sơ đồ cấu trúc của phương pháp điều khiển có tính toán

Trong cấu trúc này, mỗi một khâu như là một hệ thống con tương đối phức tạp và tất cả phải hoạt động một cách đồng bộ với hoạt động của robot tại mọi thời điểm.

2.2.2 Phương pháp điều hướng theo phản ứng

Phương pháp điều hướng theo phản ứng ra đời nhằm giải quyết các vấn đề có liên quan tới môi trường không biết trước hoặc môi trường thường xuyên thay đổi. Ở phương pháp điều hướng này không cần đến khối điều khiển, thông

tin từ khối cảm nhận được đưa thẳng tới khối chấp hành, từ đó cơ cấu chấp hành hoạt động.



Hình 2.3: Mô hình phương pháp điều hướng theo phản ứng

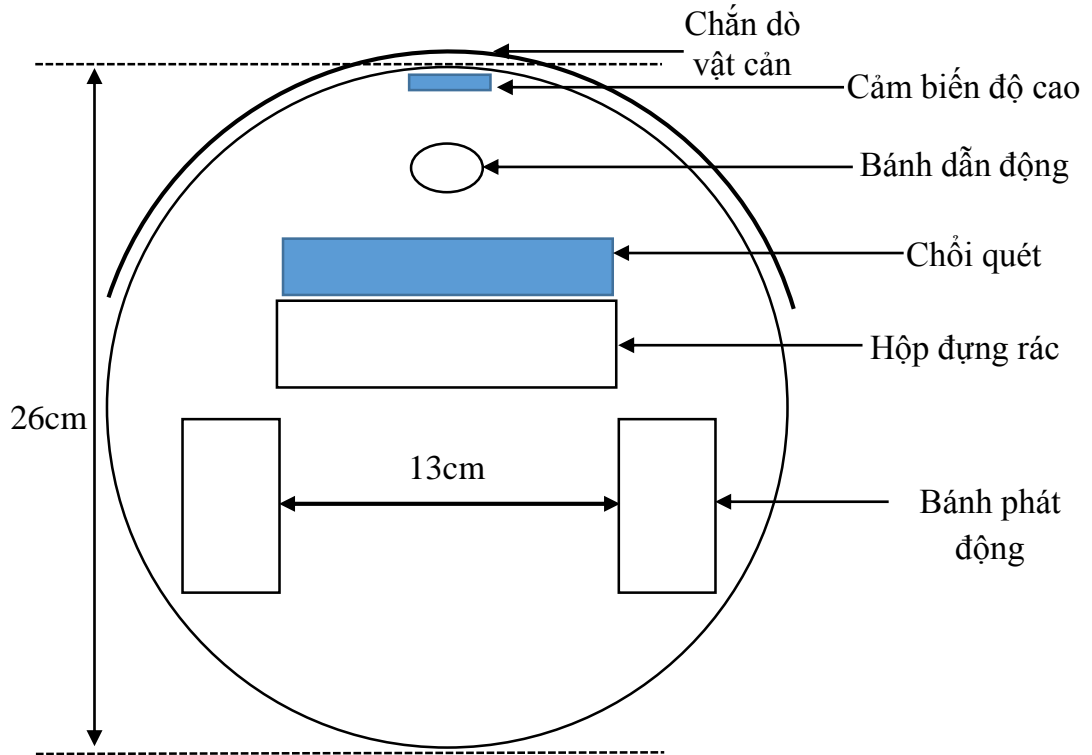
2.2.3 Phương pháp điều khiển kiểu lai

Điều hướng lai ghép là phương pháp kết hợp các ưu điểm của phương pháp điều hướng theo tính toán truyền thống với các hệ thống điều hướng dựa phản ứng. Mỗi phương pháp đều có những nhược điểm mà phương pháp kia có thể khắc phục được. Phương pháp điều hướng theo tính toán gặp phải khó khăn khi hoạt động trong các môi trường động, là nơi yêu cầu khả năng tính toán nhanh cũng như các kỹ năng tránh vật cản. Nếu phương pháp điều hướng theo phản ứng không kết hợp với bất cứ quá trình lập kế hoạch chuyển động nào thì có thể sẽ không đưa robot theo quỹ đạo tối ưu. Phương pháp điều khiển lai ra đời nhằm kết hợp các hoạt động có tính toán bậc cao với các hoạt động phản ứng bậc thấp. Các hoạt động phản ứng giúp robot an toàn và xử lý các tình trạng khẩn cấp trong khi phần điều khiển có tính toán sẽ giúp robot đạt được mục đích cuối cùng. Phương pháp điều khiển lai ghép có thể cho ta kết quả khả quan hơn khi chỉ sử dụng phương pháp điều hướng theo phản ứng hoặc điều hướng theo tính toán.

2.3 Thiết kế mô hình cơ khí cho robot

Mô hình Robot bao gồm các phần như:

- ❖ Phần cơ khí: đế, vỏ, bộ phận làm sạch, chấn dò vật cản, bộ phận cảm nhận độ cao và bộ phận cơ cấu chuyển động
- ❖ Phần mạch điện tử: khối nguồn, khối vi xử lý, khối điều khiển động cơ, các cảm biến, nút nhấn đèn và còi báo...



Hình 2.4: Thiết kế cơ khí của robot

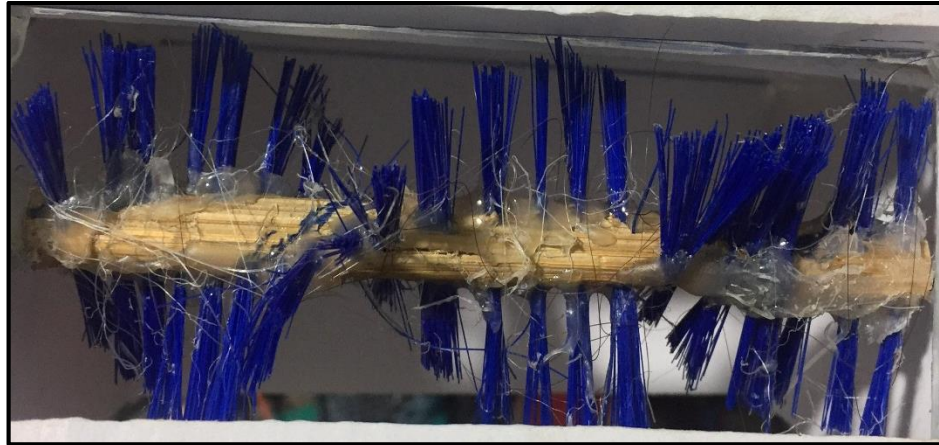
Chi tiết phần cơ khí của mô hình:

- Đế hình tròn bằng mica độ dày 2mm, đường kính 26cm.
- Vỏ được làm từ fomex

Tấm Formex (Viết chuẩn quốc tế là Fomex) là loại vật liệu được sản xuất theo công nghệ cao, chủ yếu sử dụng nhựa PVC (Poly Vinyl Clorua), với cộng tính, bọt và được nén thành hình dạng qua kỹ thuật hiện đại.

Đặc điểm của tấm formex (Fomex) là có trọng lượng nhẹ, dễ vận chuyển, chịu được lực tác động lớn, có khả năng cách âm, cách nhiệt tốt. Bên cạnh đó, gỗ còn có khả năng chống thấm nước vệ sinh dễ dàng, tuổi thọ cao bền lâu, không độc hại an toàn sử dụng. Đồng thời formex dễ dàng cắt, uốn nên thuận tiện trong tạo hình. Do đó em đã chọn formex để làm phần vỏ cho robot.

- Bộ phận làm sạch:
 - Chổi quét hoạt động bởi 1 động cơ chổi than 5v
 - Hộp đựng rác và khăn lau có thể tháo lắp.



Hình 2.5: Hình ảnh chổi quét của robot

- Chấn dò vật cản: là 2 miếng chấn cắt bằng fomex, lắp phía trước robot có gắn công tắc hành trình nhận biết vật cản bao gồm các vật dụng, tường trong gia đình, văn phòng.
- Cảm biến độ cao: sử dụng cảm biến hồng ngoại để nhận biết định hình giúp robot tránh rơi.
- Bộ phận phát động:

Do đặc điểm của Robot lau nhà là di chuyển với tốc độ chậm để tăng hiệu quả lau nhà nên trong đồ án này em lựa chọn sử dụng động cơ DC giảm tốc.

Robot sử dụng cặp động cơ giảm tốc JGA25-370 có:

- Điện áp hoạt động: 3-24v
- Điện áp hoạt động tốt nhất: 6-9v
- Công suất: 2W
- Trọng lượng: 100g
- Tốc độ hoạt động không tải:
 - 12V: 133rpm 60mA.
 - 6V: 77rpm 60mA.
- Tốc độ hoạt động có tải:
 - 12V: 113rpm 300mA 1kg.cm
 - 6V: 65rpm 300mA 2kg.cm



Hình 2.6: Hình ảnh động cơ JGA25-370

Cặp bánh xe có:

- Đường kính bánh xe: 65mm
- Độ dày lớp xe: 6.5mm
- Chiều rộng bánh: 27mm



Hình 2.7: Hình ảnh bánh xe được sử dụng trong robot

Với điện áp 9V và có tải, động cơ có thể quay với tốc độ tối đa 90 vòng/phút. Nếu sử dụng bánh đường kính 65mm động cơ có thể di chuyển được quãng đường 18m/phút.

➤ Bánh dẫn động dạng bi.

2.4 Phần mạch điện tử

Mạch điện tử của robot bao gồm các khối lớn sau:

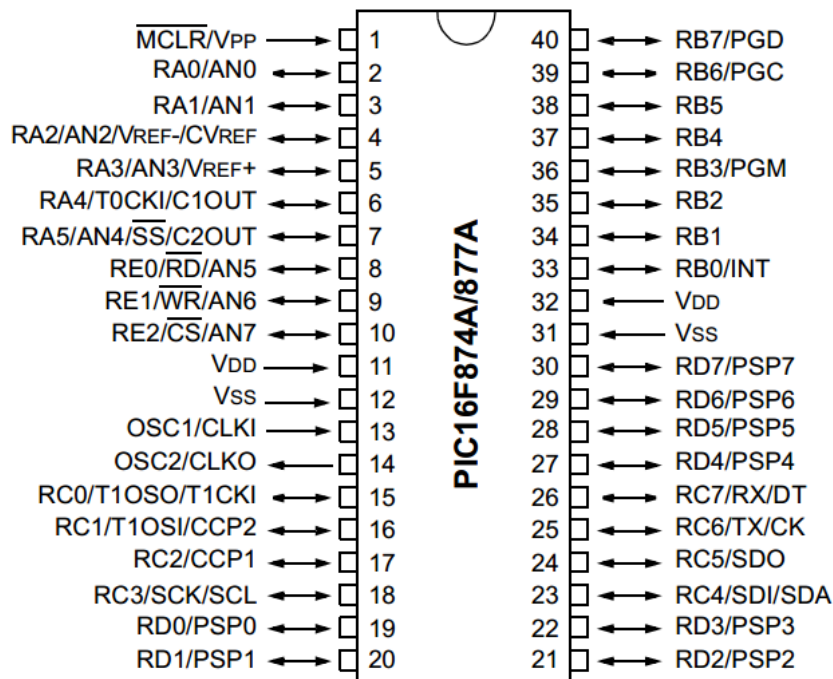
- Khối nguồn:
 - Nguồn lấy từ Pin khi hoạt động, điện lưới thông qua Adapter khi sạc Pin.
 - Khối nguồn cho động cơ.
 - Khối nguồn cho vi điều khiển, cảm biến.
- Khối giao tiếp:
 - Bàn phím.
 - Đèn hiển thị.
 - Loa.
- Khối cảm biến:
 - Công tắc hành trình dò vật cản phía trước.
 - Cảm biến hồng ngoại dò độ cao.
- Mạch công suất: Mạch điều khiển động cơ.
- Khối vi xử lý: sử dụng vi điều khiển PIC16F877A của hãng Microchip.

2.4.1 Khối điều khiển và giới thiệu PIC16F877A

PIC là viết tắt của “Programable Intelligent Computer”, có thể tạm dịch là “Máy tính khả trình thông minh” do hãng Genenal Instrument đặt tên cho vi điều khiển đầu tiên của họ là PIC1650. Lúc này, PIC1650 được dùng để giao tiếp với các thiết bị ngoại vi cho máy chủ 16bit CP1600. Vi điều khiển này sau đó được nghiên cứu phát triển thêm và từ đó hình thành nên dòng vi điều khiển PIC ngày. Các vi điều khiển thuộc dòng PIC tỏ ra rất ổn định, các khối chức năng hỗ trợ đầy đủ và giúp người dùng tiết kiệm thời gian thiết kế mạch cũng như lập trình. Hiện nay các loại vi điều khiển dòng PIC được sử dụng rất nhiều trong công nghiệp. Trong đồ án này em sử dụng một vi xử lý tiêu biểu thuộc dòng PIC là PIC16F877A do hãng Microchip sản xuất.

2.4.1.1 Một số đặc điểm của PIC16F877A:

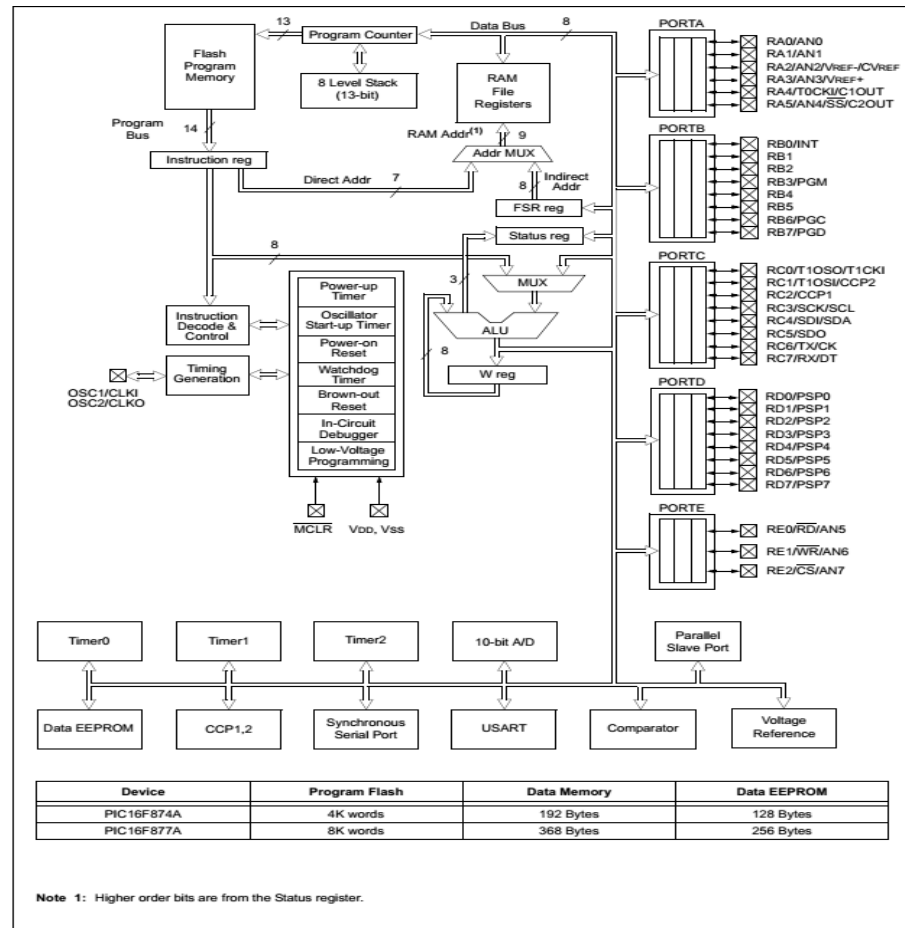
40-Pin PDIP



Hình 2.8: Sơ đồ chân của PIC16F877A loại cắm 40 chân

Hiện nay có khá nhiều dòng PIC và có rất nhiều khác biệt về phần cứng, nhưng chúng ta có thể điểm qua một vài nét như sau :

- Cấu trúc lệnh đơn giản: 35 lệnh đơn.
- CPU cấu trúc theo cấu trúc RISC (Reduced Instructions Set Computer - Máy tính với tập lệnh đơn giản hóa).
- Tần số hoạt động lớn nhất: DC - 20MHz.
- Flash Program Memory: 8K x 14 words.
- Data memory (RAM) : 368 x 8 bytes.
- EEPROM Data memory: 256 x 8bytes.
- Nạp chương trình và sửa lỗi onbroad.
- Chế tạo theo công nghệ CMOS (Complementary Metal-Oxide-Semiconductor).



Hình 2.9: Sơ đồ khối của PIC16F877A

2.4.1.2 Các cổng vào ra:

Vi điều khiển PIC16F877A có 5 cổng vào ra bao gồm: A, B, C, D, E. Trong đó:

○ Port A:

- Vào ra 6 bit, có định hướng. Khai báo định hướng vào ra ở thanh ghi TRISA. Nếu $TRISA_x=1$ thì chân Rax sẽ được định nghĩa là lối vào và ngược lại.
- Thanh ghi lưu trạng thái các chân của cổng A là thanh ghi PORTA.
- Chân RA4 của Port A có thể được sử dụng để đưa dao động vào cho Timer0 (T0CKI)
- Các chân còn lại của port A có thể được khai báo để được dùng như lối vào tương tự. Các thanh ghi liên quan ADCON0, ADCON1.

Bảng 2.1: Cấu hình Port A

Name	Bit#	Buffer	Function
RA0/AN0	Bit0	TTL	Input/output or analog input
RA1/AN1	Bit1	TTL	Input/output or analog input.
RA2/AN2/VREF- /CVREF	Bit2	TTL	Input/output or analog input or VREF- or CVREF
RA3/AN3/VREF+	Bit3	TTL	Input/output or analog input or VREF+.
RA4/T0CKI/C1OUT	Bit4	ST	Input/output or external clock input for Timer0 or comparator output. Output is open-drain type
RA5/AN4/SS/C2OUT	Bit5	TTL	Input/output or analog input or slave select input for synchronous serial port or comparator output.

TTL = TTL input, ST = Schmitt Trigger input

○ Port B:

- Vào ra số có độ rộng 8bit, có định hướng. Khai báo định hướng vào ra ở thanh ghi TRISB. Nếu $TRISB_x=1$ thì chân Rax sẽ được định nghĩa là lối vào và ngược lại.
- Thanh ghi lưu trạng thái các chân của cổng B là thanh ghi PORTB.
- 3 chân của port B là: RB3, RB6, RB7 còn được sử dụng làm chân lập trình PGM, PGC, PGD.
- 4 chân từ RB7:RB4 có ngắt on-change

Bảng 2.2: Cấu hình Port B

Name	Bit#	Buffer	Function
RB0/INT	Bit0	TTL/ST	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	Bit1	TTL	Input/output pin. Internal software programmable weak pull-up

RB2	Bit2	TTL	Input/output pin. Internal software programmable weak pull-up
RB3/PGM	Bit3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up
RB4	Bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up
RB5	Bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up
RB6/PGC	Bit6	TTL/ST	Input/output pin (with interrupt-on-change) or in-circuit debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	Bit7	TTL/ST	Input/output pin (with interrupt-on-change) or in-circuit debugger pin. Internal software programmable weak pull-up. Serial programming data.

○ Port C:

- Vào ra số có độ rộng 8bit, có định hướng. Khai báo định hướng vào ra ở thanh ghi TRISC. Nếu TRISCx=1 thì chân Rax sẽ được định nghĩa là lối vào và ngược lại.
- Thanh ghi lưu trạng thái các chân của cổng B là thanh ghi PORTC.
- Các chân port C đều có bộ đệm trigger smitt.
- Port C còn được sử dụng cho 1 số chức năng khác.

Bảng 2.3: Cấu hình Port C

Name	Bit#	Buffer	Function
RC0/T1OSO/T1CKI	Bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	Bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output

RC2/CCP1	Bit2	ST	Input/output port pin or Capture1 input/Compare1 output/ PWM1 output
RC3/SCK/SCL	Bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I2C modes
RC4/SDI/SDA	Bit4	ST	RC4 can also be the SPI data in (SPI mode) or data I/O (I2C mode).
RC5/SDO	Bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	Bit6	ST	Input/output port pin or USART asynchronous transmit or synchronous clock
RC7/RX/DT	Bit7	ST	Input/output port pin or USART asynchronous receive or synchronous data.

○ Port D:

- Vào ra số có độ rộng 8bit, có định hướng. Khai báo định hướng vào ra ở thanh ghi TRISC. Nếu TRISCx=1 thì chân Rax sẽ được định nghĩa là lối vào và ngược lại.
- Thanh ghi lưu trạng thái các chân của cổng B là thanh ghi PORTC.

Bảng 2.4: Cấu hình Port D

Name	Bit#	Buffer	Function
RD0/PSP0	Bit0	ST/TTL	Input/output port pin or Parallel Slave Port bit 0.
RD1/PSP1	Bit1	ST/TTL	Input/output port pin or Parallel Slave Port bit 1.
RD2/PSP2	Bit2	ST/TTL	Input/output port pin or Parallel Slave Port bit 2.
RD3/PSP3	Bit3	ST/TTL	Input/output port pin or Parallel Slave Port bit 3.
RD4/PSP4	Bit4	ST/TTL	Input/output port pin or Parallel Slave Port bit 4.

RD5/PSP5	Bit5	ST/TTL	Input/output port pin or Parallel Slave Port bit 5.
RD6/PSP6	Bit6	ST/TTL	Input/output port pin or Parallel Slave Port bit 6.
RD7/PSP7	Bit7	ST/TTL	Input/output port pin or Parallel Slave Port bit 7.

○ Port E:

- Port E chỉ có 3 chân vào ra, có bộ đệm trigger smitt.
- Cổng E có thể được dùng như lối vào ra số hoặc tương tự. Thiết lập tại các thanh ghi TRISE, ADCON1.

Bảng 2.5: Cấu hình Port E

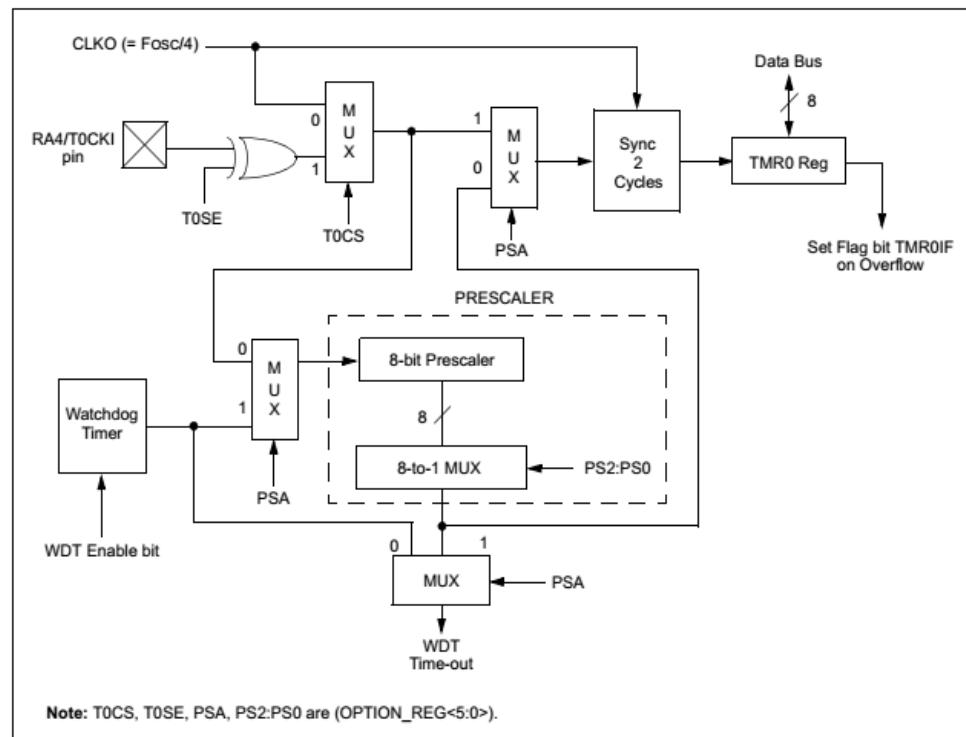
Name	Bit#	Buffer	Function
RE0/RD/AN5	Bit0	ST/TTL	I/O port pin or read control input in Parallel Slave Port mode or analog input: RD 1 = Idle 0 = Read operation. Contents of PORTD register are output to PORTD I/O pins (if chip selected).
RE1/WR/AN6	Bit1	ST/TTL	I/O port pin or write control input in Parallel Slave Port mode or analog input: WR 1 = Idle 0 = Write operation. Value of PORTD I/O pins is latched into PORTD register (if chip selected).
RE2/CS/AN7	Bit2	ST/TTL	I/O port pin or chip select control input in Parallel Slave Port mode or analog input: CS 1 = Device is not selected 0 = Device is selected

2.4.1.3 Các khối Timer

Vì điều khiển PIC16F877A có 3 bộ timer:

- Timer0:

- Là bộ đếm 8bit, có thể ghi đọc.
- Timer0 có 2 chế độ hoạt động là: định thời hoặc đếm.
- Khi hoạt động Timer đếm tiến từ giá trị được ghi trong thanh ghi TMR0 đến 256. Khi đếm đến 256 cờ tràn TMR0IF sẽ được bật lên.
- $\text{Time} = (\text{FOSC}/4) \times \text{Prescale} \times (256 - \text{TMR0})$



Hình 2.10: Sơ đồ khối Timer0

OPTION_REG REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

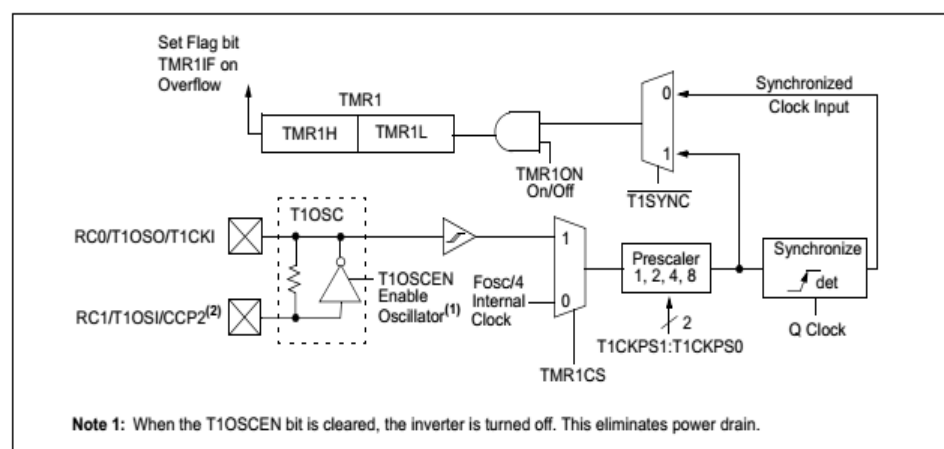
- Bit T0CS: chọn nguồn dao động cho Timer0.
- =1 nguồn dao động được đưa vào từ chân T0CKI (chế độ đếm).
- =0 nguồn dao động nội hệ thống.

- Bit T0SE: ở chế độ đếm chọn sườn xung.
=1 sườn xuống.
=0 sườn lên.
- Bit PSA: chọn Prescale
=1 được dùng cho watchdog.
=0 được dùng cho Timer0.
- Bit PS2:PS0 : chọn tỉ lệ chia cho Prescale

Bảng 2.6: Cấu hình Prescale cho Timer0

Bit value	TMR0 Rate	WDT Rate
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

• Timer1



Hình 2.11: Sơ đồ khối Timer1

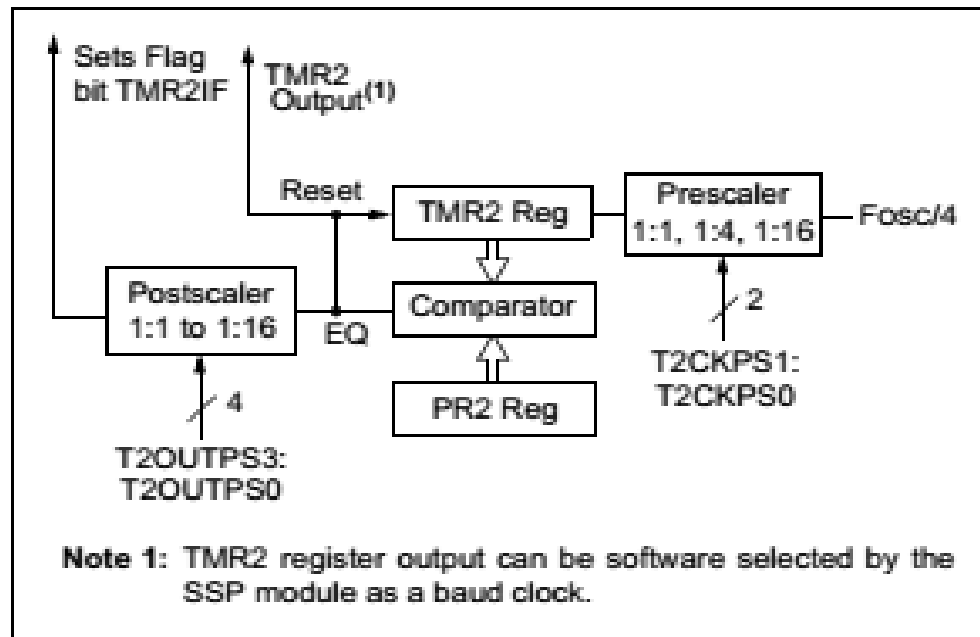
- Là bộ đếm 16bit, có thể ghi đọc. Giá trị timer được lưu trong 2 thanh ghi là TMR1L và TMR1H.
- Khác với Timer0, Timer1 có bit bật tắt timer.
- Timer1 có thể được dùng định thời cho khối CCP.
- Khi Timer1 được bật, timer sẽ đếm từ giá trị được ghi trong 2 thanh ghi TMR1L và TMR1H tới 0xFFFF, khi đó cờ tràn TMR1IF được bật.
- $\text{Time} = (\text{FOSC}/4) \times \text{Prescale} \times (0xFFFF - ((\text{TMR1H} \times 0xFFFF) + \text{TMR1L}))$

T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

- Bit T1OSCEN: chọn nguồn dao động
 =1: sử dụng dao động ngoaif.
 =0: sử dụng dao động nội.
- Bit TMR1CS=1: timer hoạt động như bộ đếm, xung vào từ chân RC1.
- Bit TMR1ON: bật, tắt timer1.
 =1: bật.
 =0: tắt.
- Bit T1CKPS1: T1CKPS0: chọn số chia cho Prescale.
 0:0 chia 1.
 0:1 chia 2.
 1:0 chia 4.
 1:1 chia 8.

- Timer2:



Hình 2.12: Sơ đồ khối Timer2

- Là bộ đếm 16bit, có thể ghi đọc. Giá trị timer được lưu trong 2 thanh ghi là TMR2L và TMR2H.
- Timer2 có bit bật tắt timer.
- Timer1 có thể được dùng định thời cho khối PWM.
- Khi Timer2 được bật, timer sẽ đếm từ giá trị được ghi trong 2 thanh ghi TMR2L và TMR2H tới 0xFFFF, khi đó cờ tràn TMR2IF được bật.
- Timer2 có cả Prescale và Postscale.

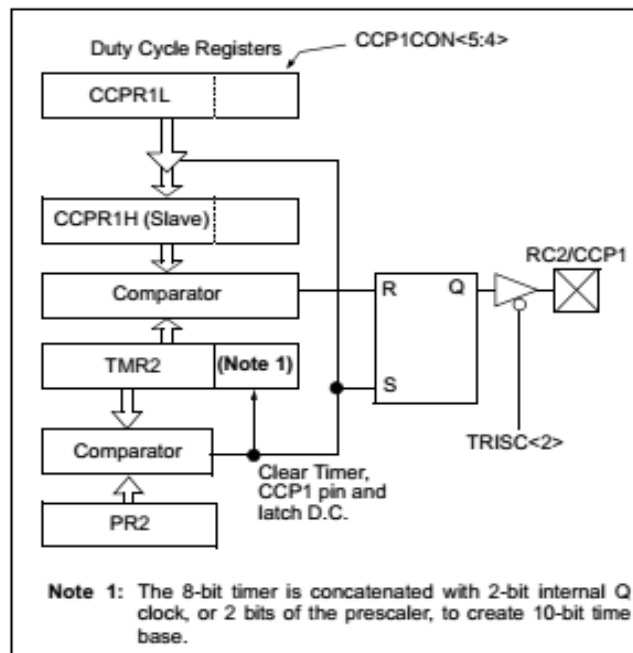
T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

- TMR2ON: bit bật tắt Timer2.
- Các bit TOUTPS3:TOUTPS0: chọn tỉ lệ chia cho postscale.
- T2CKPS1:T2CKPS0: chọn tỉ lệ chia cho prescale.

2.4.1.4 Khối PWM

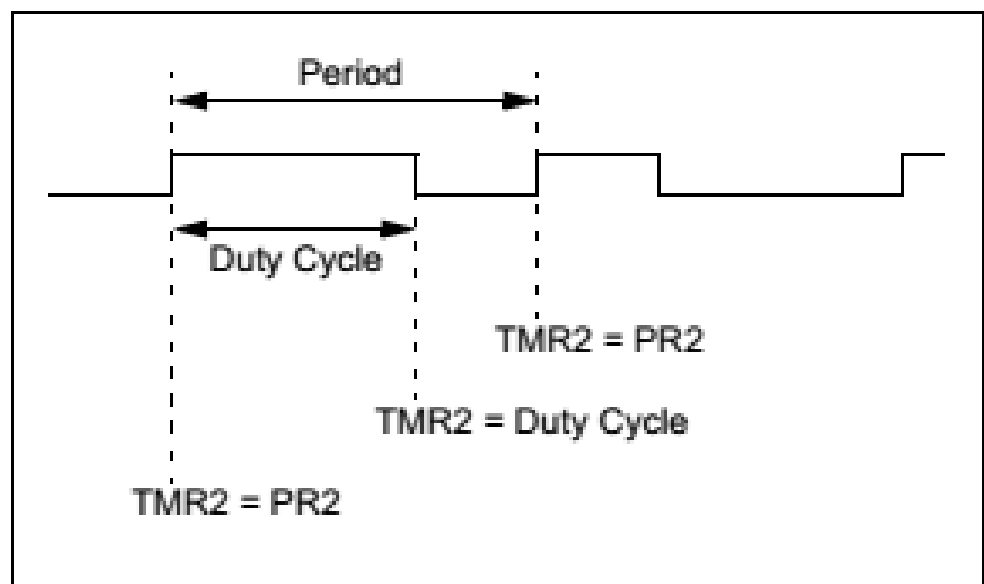
PIC16F877A có 2 khối PWM nằm trong 2 khối CCP.



Hình 2.13: Sơ đồ khối PWM.

- Độ rộng xung là dữ liệu 10 bit được lưu trong thanh ghi CCPR1L và các bit CCP1CON<5:4>.
- Chu kì được quyết định bởi giá trị trong thanh ghi PR2 và Timer2 theo công thức:

$$\text{PWM period} = [(PR2 + 1)] \times 4 \times \text{TOSC} \times (\text{TMR2 prescale value})$$



Hình 2.14: Xung ra từ khối PWM.

- Các bước cài đặt cho PWM:
 - Ghi chu kì vào thanh ghi PR2.
 - Ghi độ rộng xung vào thanh ghi CCPR1L và các bit CCP1CON<5:4>.
 - Chọn chân ra là chân RC1/2 bằng cách xoá bit tương ứng trong thanh ghi TRISC.
 - Thiết lập giá trị bộ chia tần số prescaler của Timer2 và cho phép Timer2 hoạt động bằng cách đưa giá trị thích hợp vào thanh ghi T2CON.
 - Cho phép CCP hoạt động ở chế độ PWM.

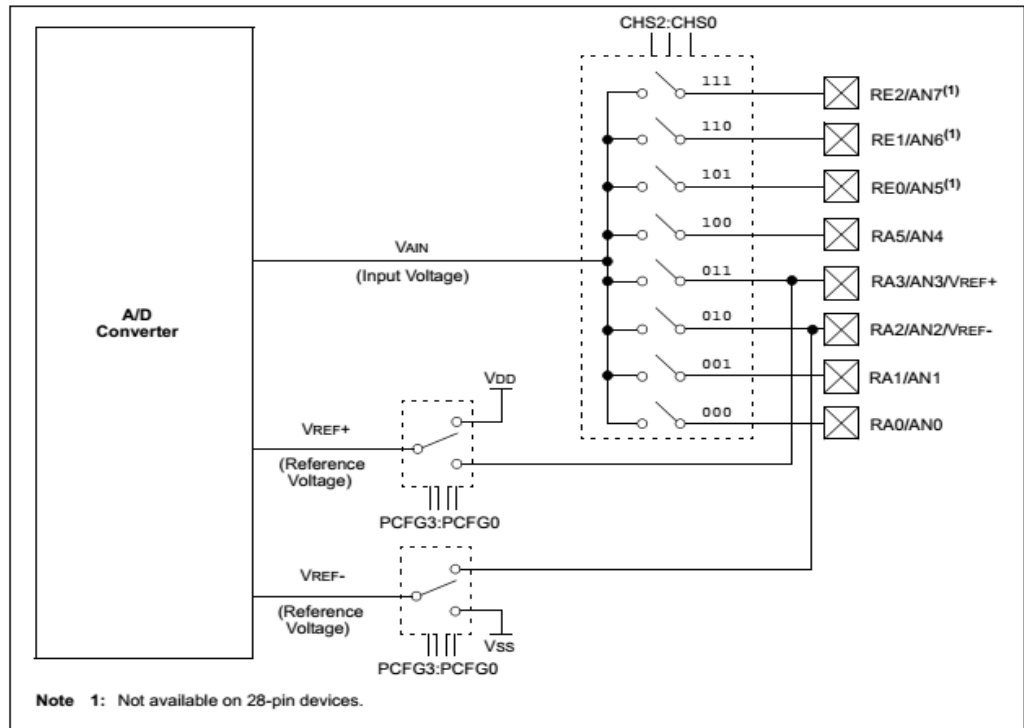
Bảng 2.7: Bảng cấu hình khối CCP.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN \bar{C}	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

2.4.1.5 Khối ADC

Khối ADC của PIC16F877A là loại ADC xấp xỉ 10 bit. Điện áp tham chiếu được lựa chọn bằng phần mềm, có thể là điện áp hệ thống, hoặc điện áp tham chiếu ngoài được đưa vào từ chân RA2 và RA3.

Khi việc chuyển đổi A/D hoàn tất cờ ADIF sẽ được bật.



Hình 2.15: Sơ đồ khối ADC

Khối chuyển đổi A/D có 4 thanh ghi là :

- Byte cao của kết quả chuyển đổi: ADRESH.
- Byte thấp của kết quả chuyển đổi: ADRESL.
- Hai thanh ghi điều khiển: ADCON0 và ADCON1.

○ **Thanh ghi ADCON0:**

ADCON0 REGISTER (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

- Các bit ADCS1:ADCS0 và ADCS2: chọn xung nhịp cho bộ ADC.

Bảng 2.8: Bảng cấu hình xung nhịp cho ADC.

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	FOSC/2
0	01	FOSC/8

0	10	FOSC/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	FOSC/4
1	01	FOSC/16
1	10	FOSC/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

- Các bit CHS2:CHS0: chọn kênh cho ADC:

000 : kênh 0 (AN0).

001 : kênh 1 (AN1).

010 : kênh 2 (AN2).

011 : kênh 3 (AN3).

100 : kênh 4 (AN4).

101 : kênh 5 (AN5).

110 : kênh 6 (AN6).

111 : kênh 7 (AN7).

- Bit GO/DONE: khi ADGO = 1, ADC bắt đầu chuyển đổi.
- Bit ADON: bit bật tắt ADC.

2.4.1.6 Các tài nguyên của PIC16f877A được sử dụng trong việc xây dựng khối điều khiển cho robot:

- I/O port.
- PWM.
- Timer0.
- ADC.

Cụ thể các tài nguyên được dùng trình bày trong bảng sau:

Bảng 2.9: Bảng mô tả các tài nguyên vi xử lý được sử dụng

Tên chân	Chân#	Kiểu	Mô tả
OSC1	13	I	Lối vào từ bộ dao động thạch anh
OSC2	14	O	Lối ra từ bộ dao động thạch anh
MCLR/VPP MCLR VPP	1	I P	Chân RESET của PIC. Chân này hoạt động ở mức thấp. Chân cấp nguồn nạp chương trình.
PORTA			
RA0	2	I	Kênh 0 của bộ A/D đo điện áp Pin.
RA1	3	I	Cảm nhận vật cản.
RA2	4	I	Cảm nhận vật cản.
RA3	5	I	Nút nhấn chọn chế độ bằng tay.
RA4	6	I	Nút nhấn chọn chế độ hẹn giờ.
RA5	7	I	Cảm nhận độ cao.
PORTB			
RB0	33	O	Điều khiển LED đơn.
RB1	34	O	Điều khiển LED đơn.
RB2	35	O	Điều khiển motor chổi quét.
RB3	36	O	Điều khiển còi báo.
PORTC			
RC0:RC5	15:23	O	Tín hiệu điều khiển bánh xe phát động.
VDD	11, 33	P	Nguồn 5v
VSS	12, 31	P	Chân 0v.

I = Input

O = Output

P = Power

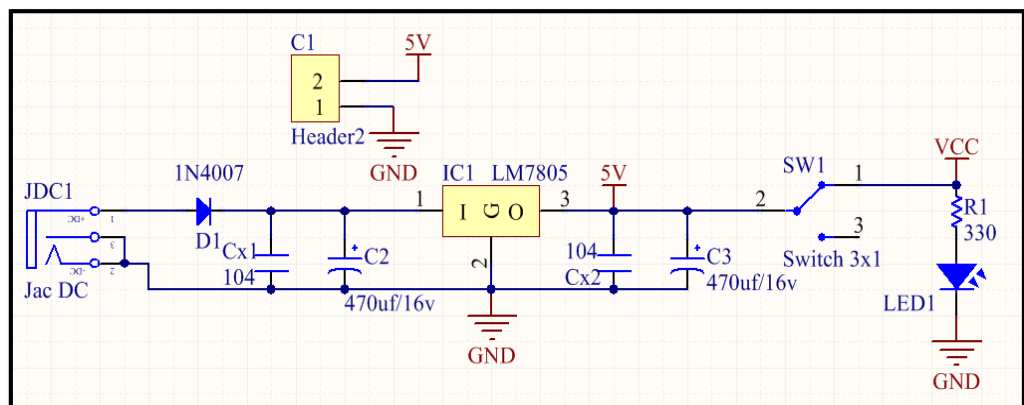
2.4.2 Khối nguồn

2.4.2.1 Khối ổn áp 5v

Tạo điện áp 5v:

Sử dụng IC7805.

Điện áp 5v được dùng để nuôi vi xử lý, các khối cảm biến, motor quét, loa và các led hiển thị.



Hình 2.16: Sơ đồ khối ổn áp 5v

2.4.2.2 Khối nguồn cho động cơ chính.

Để đảm bảo cho robot di chuyển một cách đồng đều khi có sự thăng giáng của điện áp trên nguồn hoặc khi sử dụng một nguồn điện khác, đồng thời làm tăng độ bền của động cơ chính, ta sử dụng thêm một mạch ổn áp cung cấp nguồn cho motor chính.

Các yêu cầu cho bộ nguồn motor:

- Điện áp ra 9v (điện áp khuyến nghị cho motor).
- Dòng cung cấp có thể đạt tới 3A.
- Đáp ứng các tiêu chí trên và một vài yếu tố khác, ở đây ta sử dụng mạch ổn áp theo kiểu Switching Step-Down dùng IC LM2596S.

Lý do lựa chọn nguồn Switching thay cho nguồn tuyến tính (Linear):

- Hiệu suất làm việc cao.

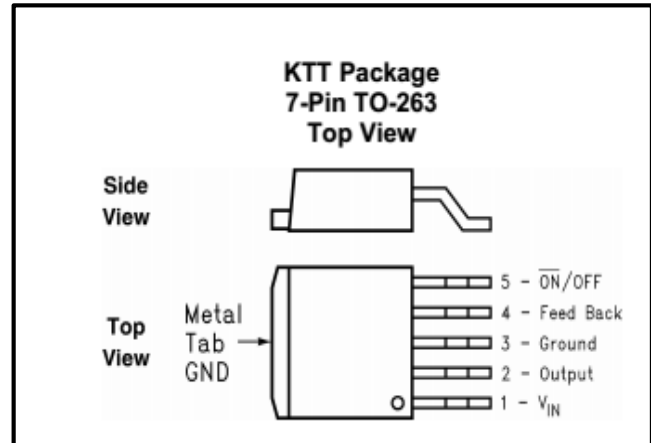
- Có thể cung cấp dòng ra lớn (điều mà hiếm IC nguồn Linear nào có thể đạt được hoặc phải dùng tản nhiệt lớn, ví dụ như LM138 – 5A, LM150 – 3A, LM317 – 1.5A).
- Công suất tiêu tán nhỏ.

Mô tả về LM2596S

Các đặc tính:

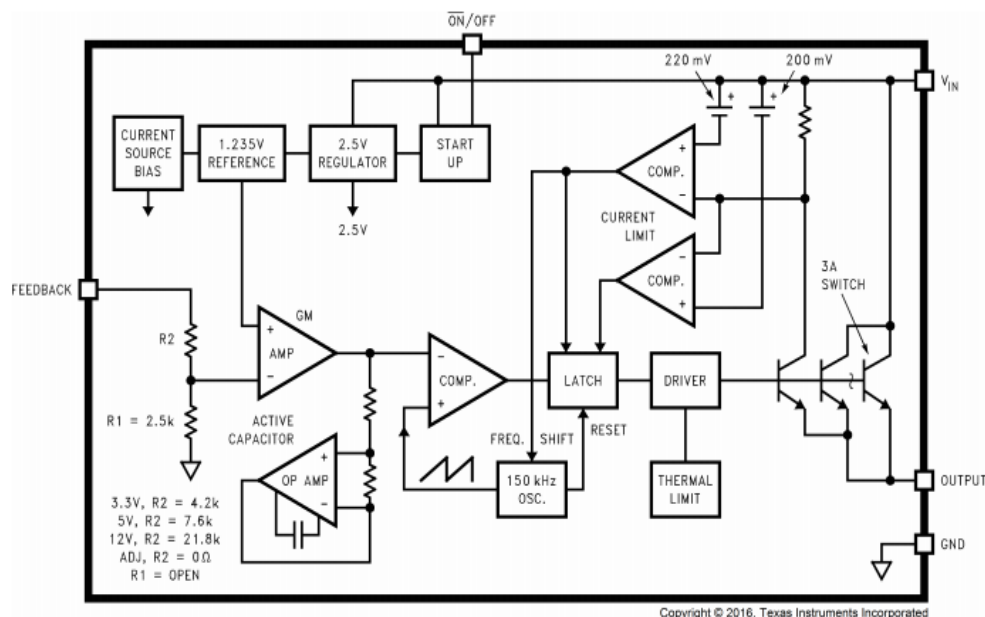
- Dải điện áp vào:
4.5 – 40VDC.
- Điện áp ra:
1.7V – 37VDC.
- Dòng điện ra:

0.2A – 3A.

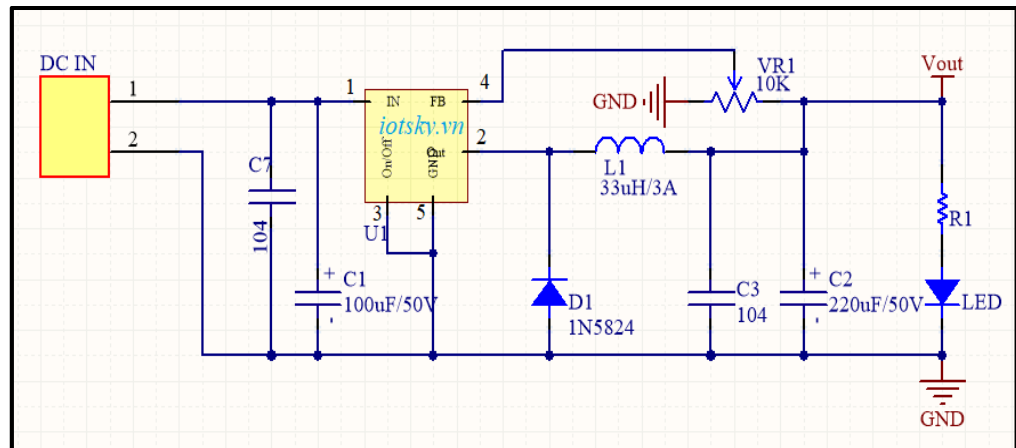


Hình 2.17: Sơ đồ chân của LM2596S

- Dải nhiệt độ hoạt động: -45 – 125°C.
- Có khả năng ngắt ngoài khi dòng chờ 50uA và ngắt tự động khi quá nhiệt.



Hình 2.18: Sơ đồ khối bên trong của LM2596S



Hình 2.19: Sơ đồ nguyên lý mạch nguồn dùng LM2596

Điện áp ra của mạch được tính theo công thức:

$$V_{out} = V_{REF} \left(1 + \frac{VR_{out}}{VR_{GND}} \right) \quad (2.1)$$

Trong đó $V_{REF} = 1.23V$.

Điện áp ra V_{out} được quyết định thông qua điện áp phản hồi về chân 4 (FB- Feedback). Điện áp này sau đó được so sánh với một điện áp chuẩn V_{REF} để điều khiển độ dẫn của Transistor trong LM2596 sao cho điện áp ra luôn ổn định theo mong muốn.

Lựa chọn linh kiện:

- Diode Schottky D1: 1N5824 (SS54) 3A.
- Cuộn cảm L1: sử dụng cuộn cảm 47uH (470) dạng chân dán.
- Biến trở VR1: chọn điện trở 10k (104) được xoay chỉnh sao cho điện áp trên 2 cực động cơ là 9V.

2.4.2.3 Pin và mạch nạp pin

Pin 18650 là loại pin sạc sử dụng công nghệ Li-ion, viên pin có điện áp 3.7V, đường kính 1.8cm và chiều dài 6.5cm. Khi sạc đầy, điện pin có thể đạt 4.2V và đạt dưới 3V khi pin yếu. Tuổi thọ pin có thể đạt tới là 500 xả và sạc đầy.



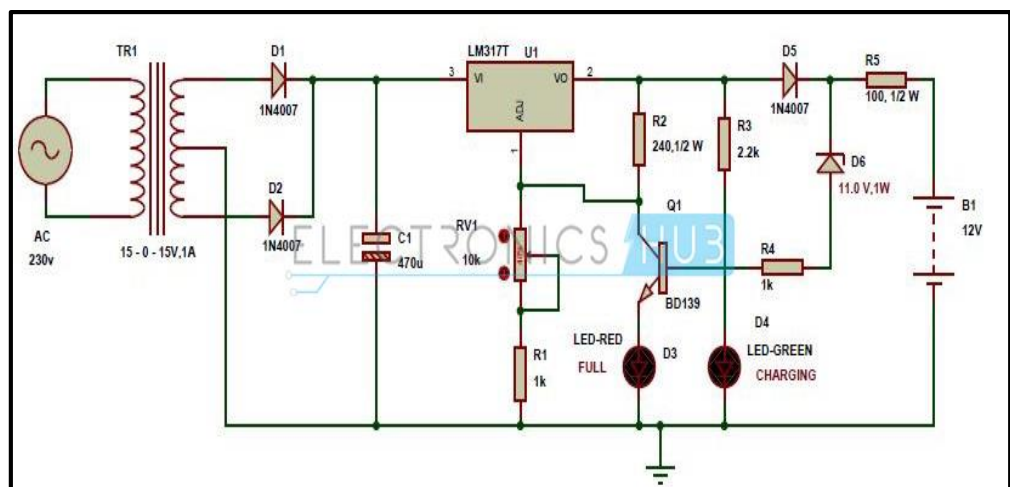
Hình 2.20: Hình ảnh pin 18650

Robot trong đề tài sử dụng 3 cell pin 18650, mỗi viên có dung lượng 4000mAh, tạo ra nguồn điện DC khoảng 12V cung cấp cho robot.

Mạch nạp pin:

Sử dụng adapter chuyên dụng để sạc pin cho robot.

- Điện thế AC vào dải rộng từ 100-240v.
- Điện thế ra DC 12,6v.
- Dòng sạc ổn định 2A.
- Có bảo vệ quá điện áp, bảo vệ ngắn mạch và bảo vệ phân cực ngược.



Hình 2.21: Sơ đồ nguyên lý của adapter

Nguyên lý hoạt động:

Khi kết nối bộ nạp, nếu điện áp pin dưới 11V, dòng điện từ LM317 chảy qua diode D5 và điện trở R5 vào pin. Tại thời điểm này diode zener D6 phân cực thuận, BD139 không dẫn. Diode D5 làm nhiệm vụ chống phân cực ngược khi kết nối pin với bộ nạp.

Khi điện áp pin tăng lên 12.6V, quá trình nạp dừng lại và diode zener D6 phân cực ngược làm cho BD139 dẫn, chân ADJ của LM317 ở mức thấp, không có điện áp ra từ LM317, quá trình nạp bị ngắt.

2.4.3 Hệ thống cảm biến vật cản

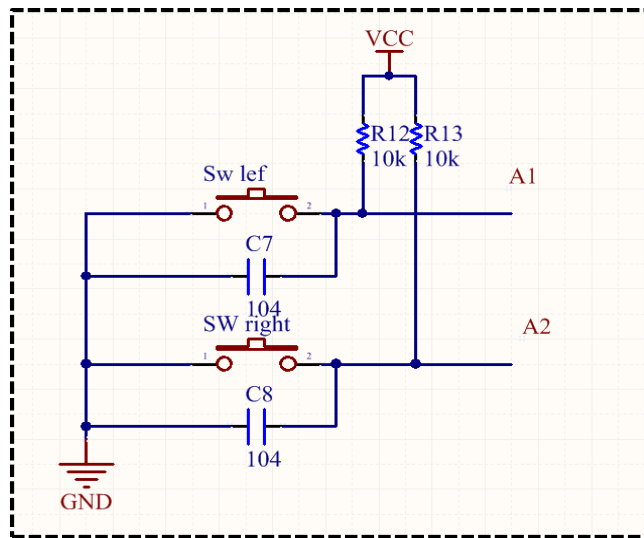
Khối này sử dụng 2 công tắc hành trình để có thể cảm nhận va chạm với vật cản từ bên trái hoặc phải. Khi robot chạm vào vật cản thì hệ thống cơ khí sẽ đóng tiếp điểm thường mở (NO) nối chân vi xử lý xuống GND, làm chân vi xử lý chuyển trạng thái từ mức 1 xuống mức 0. Từ sự thay đổi trạng thái chân vi xử lý ta có thể biết có vật cản bên phải hay bên trái để đưa ra hướng xử lý giúp robot tránh được vật cản.



Hình 2.22: Công tắc hành trình

Các điện trở R12 và R13 là các điện trở kéo lên. Khi không có vật cản, công tắc không được nhấn, tiếp điểm NO ở trạng thái hở, chân vi xử lý được nối

với VCC thông qua điện trở 10K làm chân vi xử lý ở mức 1. Các tụ C7 và C8 là tụ gốm 104 giúp chống dội phím.



Hình 2.23: Sơ đồ bộ phận cảm nhận vật cản.

2.4.4 Khối cảm biến độ cao

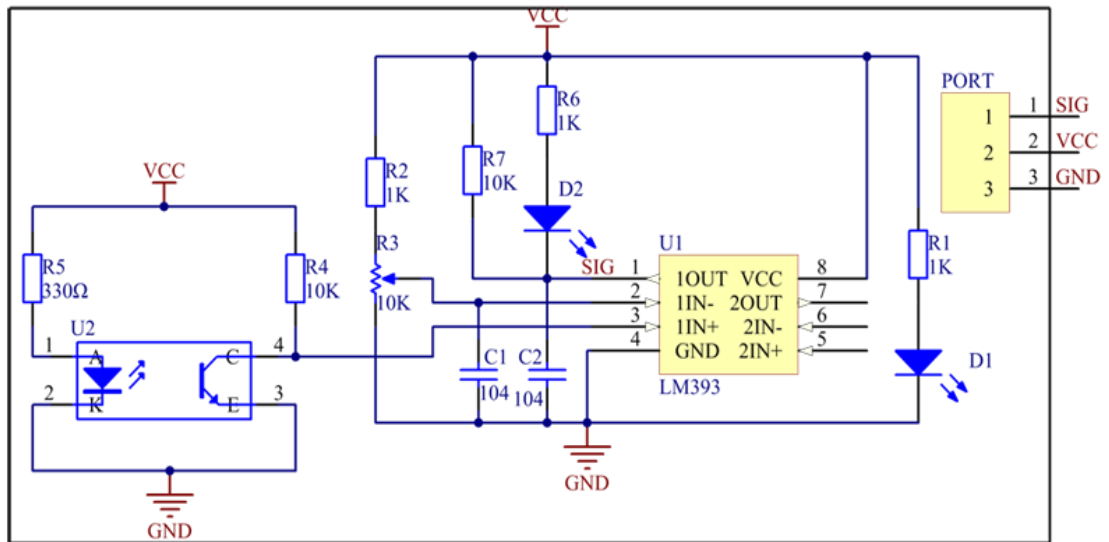
Khi robot làm việc ở những vùng có bậc thang, để không bị rơi, robot phải có bộ phận cảm nhận độ cao.

Nguyên lý cảm nhận: sử dụng module thu phát hồng ngoại MH-IR01 được lắp đặt ở phía trước của robot. Bình thường, tín hiệu phát từ led phát hồng ngoại gặp sàn nhà sẽ phản xạ từ mặt sàn lên led thu hồng ngoại, nhưng nếu gặp phải độ cao (tức là khoảng cách khoảng cách giữa mặt sàn và bộ thu phát hồng ngoại lớn hơn ngưỡng đặt trước) thì tín hiệu phản xạ sẽ rất yếu hoặc không có. Tín hiệu thu được trên led thu sẽ đi qua bộ lọc, khuếch đại rồi so sánh để chuyển thành tín dạng hiệu số.

Thông số kỹ thuật module thu phát hồng ngoại MH-IR01:

- VCC: 3.3 – 5V (có thể dùng nguồn trực tiếp từ vi điều khiển).
- Đầu ra: logic 0/1.
- Khoảng cách phát hiện vật cản: 2 – 30cm (có thể điều chỉnh bằng biến trở).
- Góc phát hiện vật cản: 35°

- Đầu ra có thể nối trực tiếp với các chân I/O của vi điều khiển hoặc dùng để điều khiển relay.

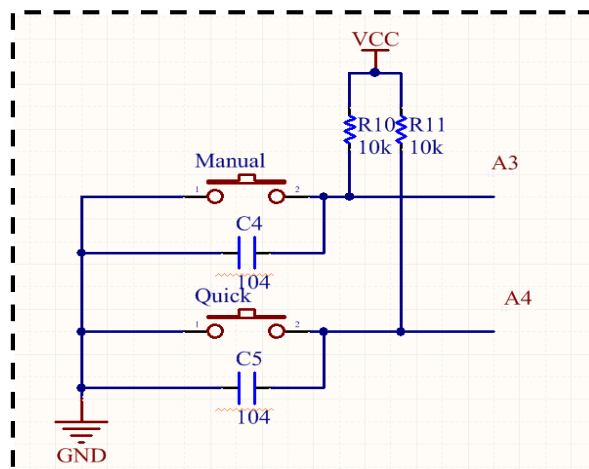


Hình 2.24: Sơ đồ nguyên lý module thu phát hồng ngoại MH-IR01

2.4.5 Bàn phím

Robot sử dụng 2 phím nhấn để chọn chế độ hoạt động:

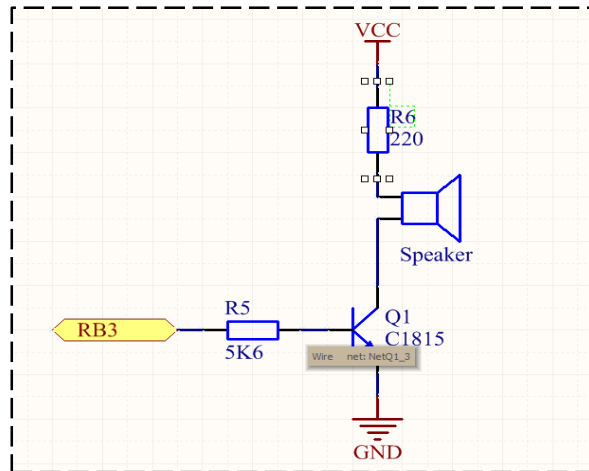
- Manual (dùng để bật tắt robot chế độ bằng tay – robot hoạt động tới khi tắt hoặc hết pin).
- Quick (dùng để bật tắt robot chế độ làm sạch nhanh – robot hoạt động tới khi hết thời gian đặt trước hoặc tắt đi).



Hình 2.25: Sơ đồ bàn kết nối bàn phím

2.4.6 Còi báo

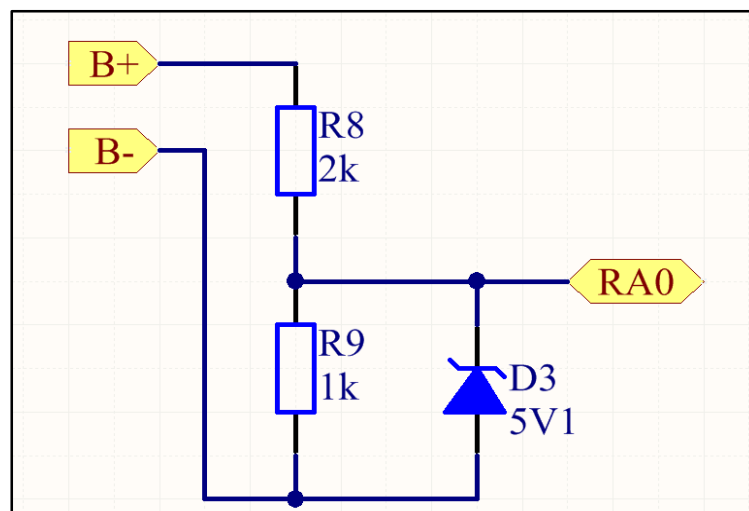
Khi điện áp pin dưới mức cho phép, còi báo sẽ kêu để báo hiệu cho người sử dụng tiến hành sạc pin để đảm bảo hoạt động của robot, và tăng độ bền của pin. Tín hiệu điều khiển còi báo được cấp từ vi điều khiển.



Hình 2.26: Sơ đồ ghép nối loa.

2.4.7 Khối ADC đo điện áp pin

Do điện áp lớn nhất PIC16F877A có thể đo trực tiếp là khoảng 5V, mà điện áp pin lớn nhất lại là 12.6V nên muốn đo được điện áp pin ta phải dùng cầu điện trở phân áp để không làm hỏng vi điều khiển.

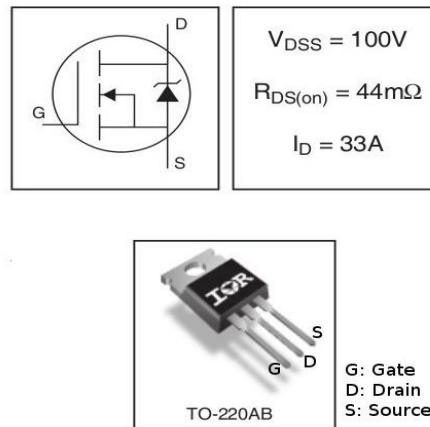


Hình 2.27: Sơ đồ mắc điện trở phân áp pin.

Ở đây ta chọn $R9:R8 = 1:2$, khi điện áp trên Pin là 12.6V thì điện áp được đưa vào vi xử lý là $12.6/3 \times 1 = 4.2V$ (thoả mãn nhỏ hơn 5V).

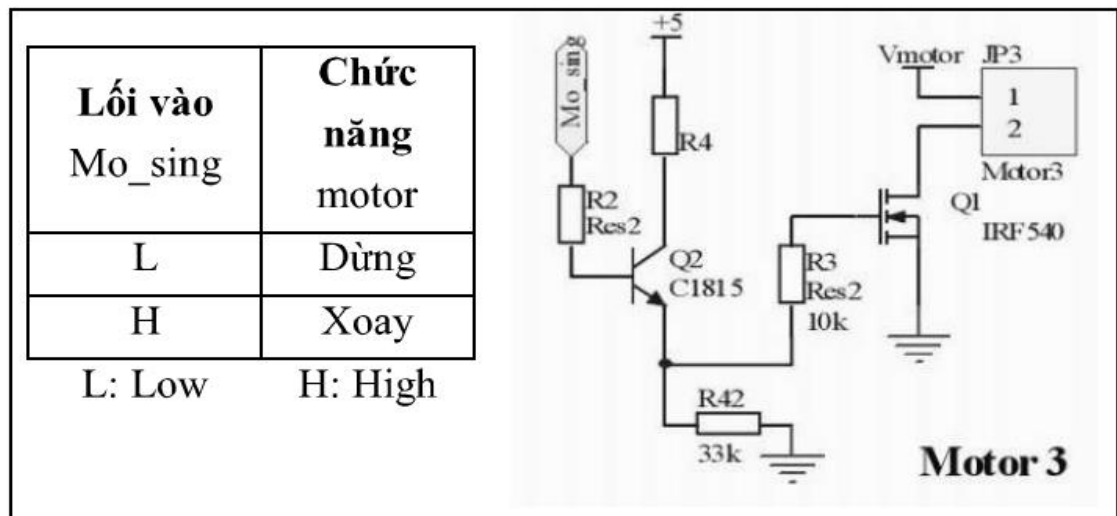
2.4.8 Khối điều khiển motor chổi quét.

Chổi quét sử dụng một động cơ có dòng tiêu thụ tối đa là 0.5A. Do chổi quét chỉ quay theo một chiều nên mạch điều khiển không cần mạch cầu H mà chỉ sử dụng một MOSTFET-N là IRF540 làm nhiệm vụ đóng mở thông thường.



Hình 2.28: Mô tả IRF540

Như vậy, IRF540 có thể tải được dòng khá lớn, có thể lên tới 33A - đủ để đáp ứng cho động cơ chổi quét, điện trở cực nguồn (S) và cực máng (D) khi mở rất nhỏ 4mΩ nên sụt áp trên hai cực DS cũng nhỏ. Thêm vào đó, trong IRF540 có sẵn một diode Schottky để bảo vệ dòng ngược từ S sang D mà có thể làm hỏng cấu trúc của MOSFET.



Hình 2.29: Sơ đồ mạch điều khiển động cơ chổi quét

Trong sơ đồ mạch điều khiển này, để đóng mở IRF540 sử dụng thêm một tầng phụ dùng transistor C1815. Điều này có thể không cần thiết nhưng sẽ đảm bảo an toàn nếu có sự cố làm hỏng MOSFET cũng không làm ảnh hưởng đến vi điều khiển. Hai cực động cơ được nối thông qua jack JP3.

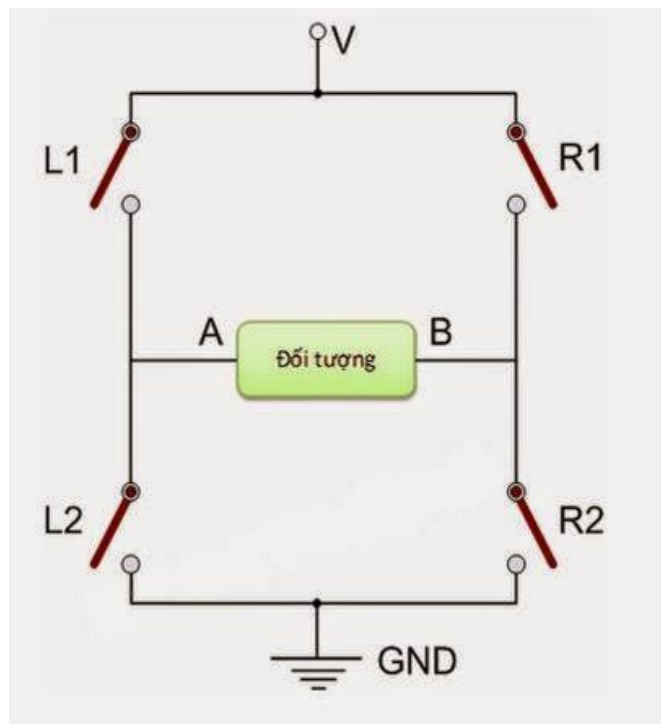
Khi tín hiệu điều khiển Mo_sing từ vi điều khiển ở mức thấp (Low) thì C1815 sẽ ngắt, cực G của IRF540 nối đất thông qua điện trở R42, do vậy IRF540 không dẫn và làm động cơ ở trạng thái dừng. Ngược lại, khi Mo_sing ở mức cao (High), C1815 dẫn, điện áp cực E của C1815 sẽ ở mức cao do R42 có giá trị rất lớn làm IRF540 mở, kết quả là có dòng chạy qua làm động cơ quay.

Nhược điểm của mạch: vì IRF540 mở hoàn toàn ở 10V nhưng với mạch trên ta lấy tín hiệu điều khiển trực tiếp từ cổng I/O của vi điều khiển nên điện áp điều khiển chỉ ở ngưỡng 5V. Vì vậy, trong quá trình hoạt động transistor rất nóng làm ảnh hưởng tới việc hoạt động lâu dài của robot.

2.4.9 Mạch điều khiển động cơ chính

Robot sử dụng hai động cơ DC chổi than làm động cơ chính, cho nên để điều khiển hướng di chuyển của robot ta sử dụng mạch cầu H để điều khiển đóng mở và chiều động cơ.

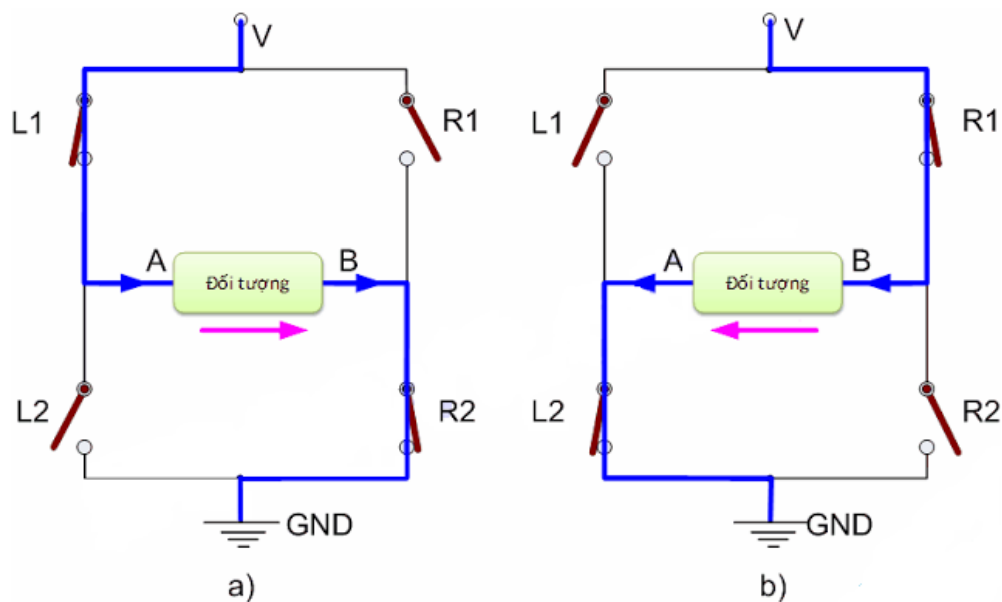
2.4.9.1 Giới thiệu về mạch cầu H



Hình 2.30: Mạch cầu H

Mạch cầu H là một mạch thông dụng và đơn giản dùng trong điều khiển động cơ DC. Xét một cách tổng quát, mạch cầu H là một mạch gồm 4 "công tắc" được mắc theo hình chữ H như Hình 2.30. Thành phần chính tạo nên mạch cầu H của là 4 "khóa" L1, L2, R1 và R2 (L: Left, R:Right). Ở điều kiện bình thường 4 khóa này "mở", mạch cầu H không hoạt động. Việc khảo sát hoạt động của mạch cầu H sẽ được minh họa trên Hình 2.31.

Khi 2 khóa L1 và R2 được "đóng lại" (L2 và R1 vẫn mở), ta dễ dàng thấy được dòng điện chạy từ V qua khóa L1 đến đầu A và xuyên qua đối tượng đến đầu B của nó trước khi qua khóa R2 và về GND (như Hình 2.31a). Ngược lại khi khóa R1 và L2 đóng trong khi L1 và R2 mở, dòng điện lại xuất hiện và lần này nó sẽ chạy qua R1, qua đối tượng theo chiều từ B đến A rồi cuối cùng xuống GND qua L2 (như trong Hình 2.31b). Vậy là đã rõ, chúng ta có thể dùng mạch cầu H để đảo chiều dòng điện qua một "đối tượng" (hay cụ thể là đảo chiều quay động cơ).



Hình 2.31: Nguyên lý hoạt động của mạch cầu H.

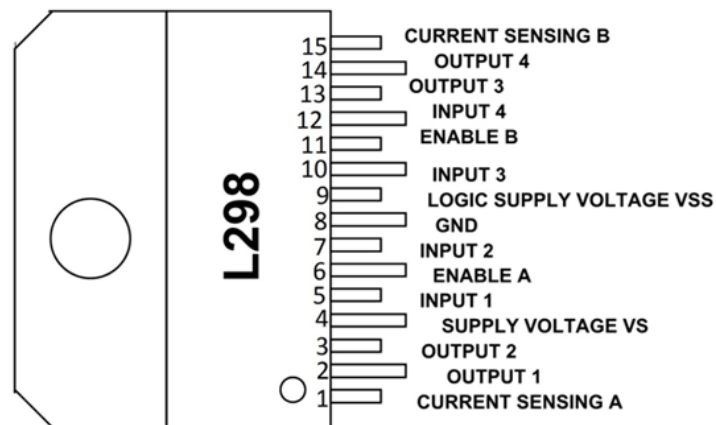
Do thành phần chính của mạch cầu H chính là các "khóa", việc chọn linh kiện để làm các khóa này phụ thuộc vào mục đích sử dụng mạch cầu, loại đối tượng cần điều khiển, công suất tiêu thụ của đối tượng. Nhìn chung, các khóa của mạch cầu H thường được chế tạo bằng rơ le (relay), BJT (Bipolar Junction Transistor) hay MOSFET (Metal Oxide Semiconductor Field-Effect Transistor).

Để thay đổi tốc độ động cơ ta chỉ cần chỉ cần thay đổi điện áp đặt vào 2 cực điều khiển của mạch cầu H. Hiệu điện thế giữa 2 cực điều khiển càng lớn thì động cơ chạy càng nhanh. Động cơ chạy theo chiều thuận khi điện áp ở A nhỏ hơn B và ngược lại.

Trong phạm đồ án này em sử dụng mạch cầu H trong IC L298.

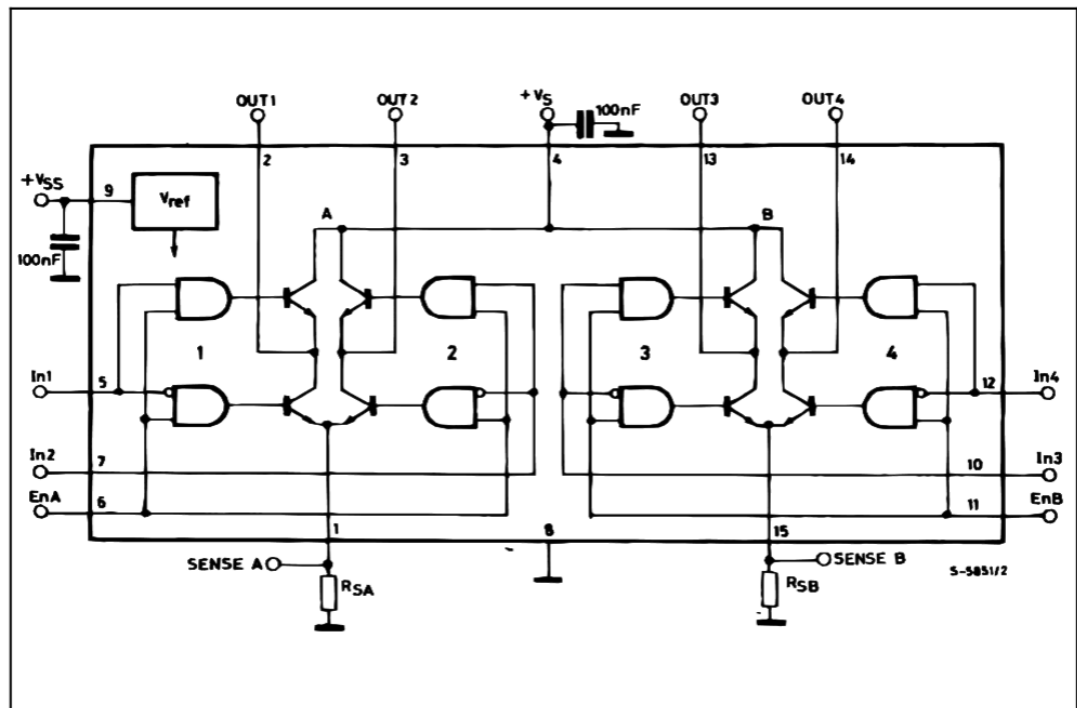
2.4.9.2 IC L298

IC L298 là IC có chứa 2 mạch cầu H hoàn toàn độ lập và có thể ghép song song với nhau, mỗi cầu có khả năng cung cấp dòng 2A – đáp ứng được yêu cầu cho mỗi motor có dòng tiêu thụ tối đa là 1.5A.



Hình 2.32: Sơ đồ chân IC L298

IC L298 là mạch tích hợp đơn chip có kiểu vỏ công suất 15 chân (multiwatt 15) và PowerSO20 (linh kiện dán công suất). Là IC mạch cầu đôi (dual full-bridge) có khả năng hoạt động ở điện thế cao, dòng cao. Nó được thiết kế tương thích chuẩn TTL và điều khiển tải cảm kháng như relay, cuộn solenoid, động cơ DC và động cơ bước. IC có 2 chân enable (cho phép) để cho phép/không cho phép IC hoạt động, độc lập với các chân tín hiệu vào. Cực phát (emitter) của transistor dưới của mỗi mạch cầu được nối với nhau và nối ra chân ngoài để nối với điện trở cảm ứng dòng khi cần. Nó có thêm một chân cấp nguồn giúp mạch logic có thể hoạt động ở điện thế thấp hơn.



Hình 2.33: Sơ đồ khối IC L298

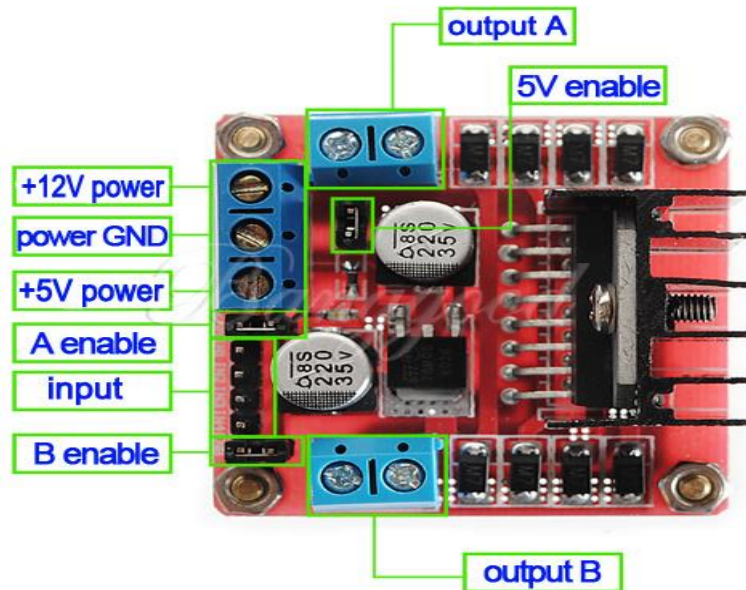
Bảng 2.10: Mô tả IC L298

Chức năng	Chân số	Tên	Chức năng
Nguồn	Chân 4	VS	Điện áp dương cho motor
	Chân 9	VSS	Điện áp 5V cho khối logic
	Chân 8	GND	Chân nối đất
Lối vào	Chân 5, 7	Input 1, Input 2	Lối vào TTL của cầu A
	Chân 10, 12	Input 3, Input 4	Lối vào TTL của cầu B
Lối ra	Chân 2, 3	Output 1, Output 2	Lối ra của cầu A
	Chân 13, 14	Output 3, Output 4	Lối ra của cầu B
Cảm nhận	Chân 1, 15	IsA, IsB	Dùng để điều khiển dòng tải

2.4.9.3 Sử dụng module L298 để điều khiển động cơ

Modul L298 là một mạch tích hợp sử dụng IC L298. IC L298 đã tích hợp sẵn hai mạch cầu H, ứng dụng trong việc điều khiển cùng lúc 2 động cơ

theo chiều quay bất kì, kết hợp với điều xung PWM có thể điều chỉnh tốc độ xoay của động cơ.



Hình 2.34: Module L298

Các thông số kỹ thuật cơ bản của mạch cầu H gồm:

- 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
- Công suất hao phí: 20W (khi nhiệt độ $T = 75^{\circ}\text{C}$)
- Nhiệt độ bảo quản: $-25^{\circ}\text{C} \sim +130^{\circ}\text{C}$

Các chân kết nối trên modul L298:

- +12V Power: Đây là chân cấp nguồn trực tiếp cho modul và toàn bộ động cơ.
 - Nguồn cấp cho modul có thể giao động từ 9-12V.
- Power GND: Chân mát của modul và toàn bộ hệ thống.
- +5V Power: Chân cấp nguồn cho vi điều khiển và các cảm biến.

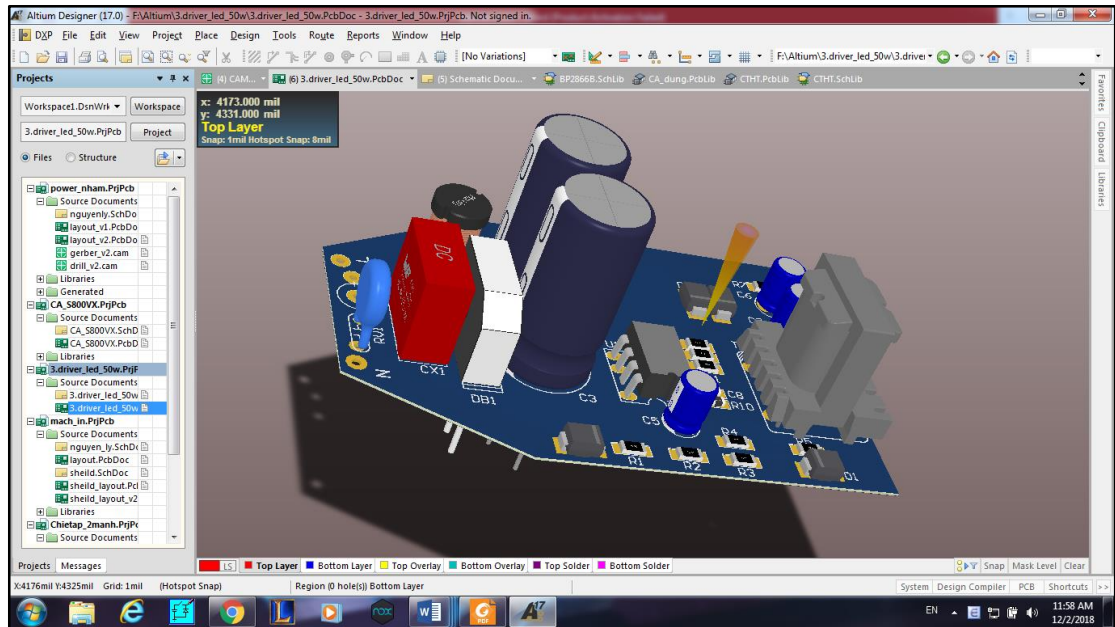
- 5V Enable: Chân Enable nguồn 5V. Nếu hai jum này được nối (thì sẽ có nguồn 5V tại chân +5V Power, ngược lại thì không.
- A Enable: Chân enable cổng output A. Nếu chân này được tích cực ở 5V cổng output A được enable. Nếu muốn điều khiển tốc độ động cơ sử dụng xung PWM thì xung PWM sẽ được cấp vào chân A Enable này để điều khiển tốc độ động cơ A.
- B Enable: Chân enable cổng output B. Nếu chân này được tích cực ở 5V cổng output B được enable. Nếu muốn điều khiển tốc độ động cơ sử dụng xung PWM thì xung PWM sẽ được cấp vào chân B Enable này để điều khiển tốc độ động cơ B.
- Input: Nhóm này gồm 4 chân INA, INB, INC, IND. Đây là 4 chân chính dùng để nhận tín hiệu điều khiển động cơ từ vi điều khiển.
 - $INA = 1$, $INB = 0$: Động cơ A sẽ quay theo chiều thuận.
 - $INA = 0$, $INB = 1$: Động cơ sẽ A quay theo chiều nghịch.
 - $INA = 0$, $INB = 0$: Dừng động cơ A.
 - $INC = 1$, $IND = 0$: Động cơ B quay theo chiều thuận.
 - $INC = 0$, $IND = 1$: Động cơ B quay theo chiều nghịch.
 - $INC = 0$, $IND = 0$: Dừng động cơ B

Nếu 4 chân input được thiết lập như trên thì động cơ sẽ quay với tốc độ tối đa. Tương tự như các chân A Enable và B Enable ta cũng có thể điều khiển tốc độ động cơ bằng cách cấp xung PWM vào các chân này, giả sử ta cấp một xung PWM có duty cycle = 50% vào chân INA và giữ chân INB ở mức 0 thì động cơ A sẽ quay với tốc độ 50%. Tương tự với các chân còn lại.

2.4.10 Thiết kế mạch in – PCB (Printed Circuit Board) bằng Altium Designer

Altium Designer trước kia có tên gọi quen thuộc là Protel DXP, là một trong những công cụ vẽ mạch điện tử mạnh nhất hiện nay. Được phát triển bởi hãng Altium Limited. Altium designer là một phần mềm chuyên ngành được sử dụng trong thiết kế mạch điện tử. Nó là một phần mềm mạnh với nhiều tính năng

thứ vị, và được nâng cấp, chỉnh sửa hàng năm với những phiên bản có những tính năng mới hữu ích hơn cho người thiết kế. Hiện nay phiên bản mới nhất của phần mềm là Altium 19.



Hình 2.35: Giao diện làm việc của Altium 17.

Altium Designer có một số đặc trưng sau:

- Giao diện thiết kế, quản lý và chỉnh sửa thân thiện, dễ dàng biên dịch, quản lý file, quản lý phiên bản cho các tài liệu thiết kế.
- Hỗ trợ mạnh mẽ cho việc thiết kế tự động, đi dây tự động theo thuật toán tối ưu, phân tích lắp ráp linh kiện. Hỗ trợ việc tìm các giải pháp thiết kế hoặc chỉnh sửa mạch, linh kiện, netlist có sẵn từ trước theo các tham số mới.
- Mở, xem và in các file thiết kế mạch dễ dàng với đầy đủ các thông tin linh kiện, netlist, dữ liệu bản vẽ, kích thước, số lượng...
- Hệ thống các thư viện linh kiện phong phú, chi tiết và hoàn chỉnh bao gồm tất cả các linh kiện nhúng, số, tương tự...
- Đặt và sửa đổi tượng trên các lớp cơ khí, định nghĩa các luật thiết kế, tùy chỉnh các lớp mạch in, chuyển từ schematic sang PCB, đặt vị trí linh kiện trên PCB.

- Mô phỏng mạch PCB 3D, đem lại hình ảnh mạch điện trung thực trong không gian 3 chiều, hỗ trợ MCAD-ECAD, liên kết trực tiếp với mô hình STEP, kiểm tra khoảng cách cách điện, cấu hình cho cả 2D và 3D.

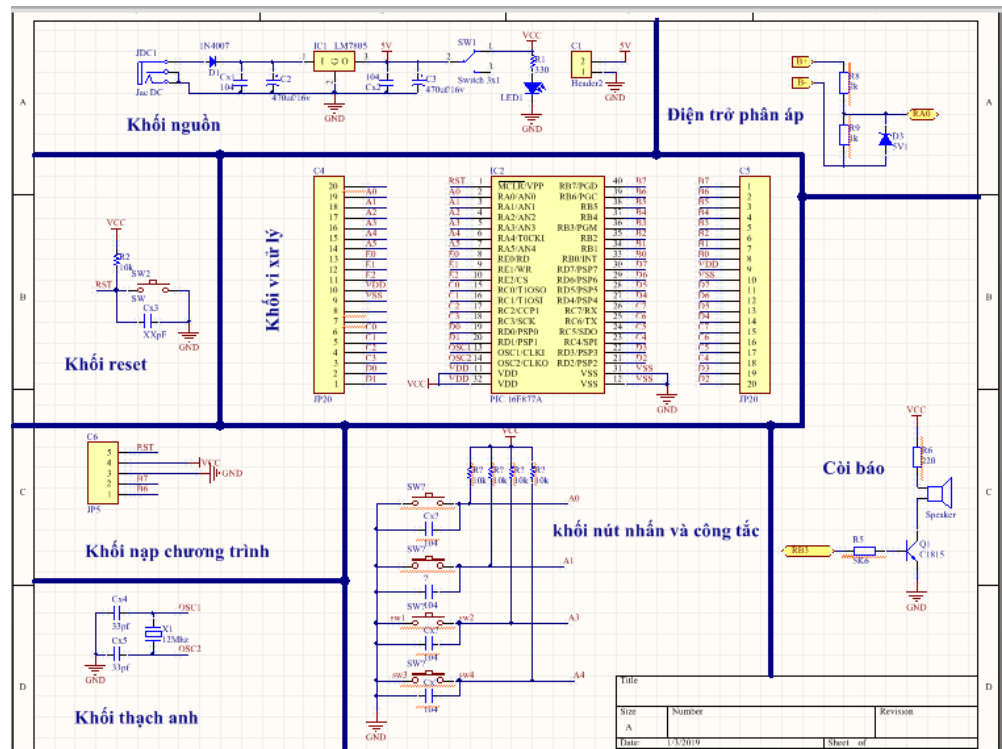
- Hỗ trợ thiết kế PCB sang FPGA và ngược lại.

Từ đó, chúng ta thấy Altium designer có nhiều điểm mạnh so với các phần mềm khác như đặt luật thiết kế, quản lý đề tài mô phỏng dễ dàng, giao diện thân thiện,...

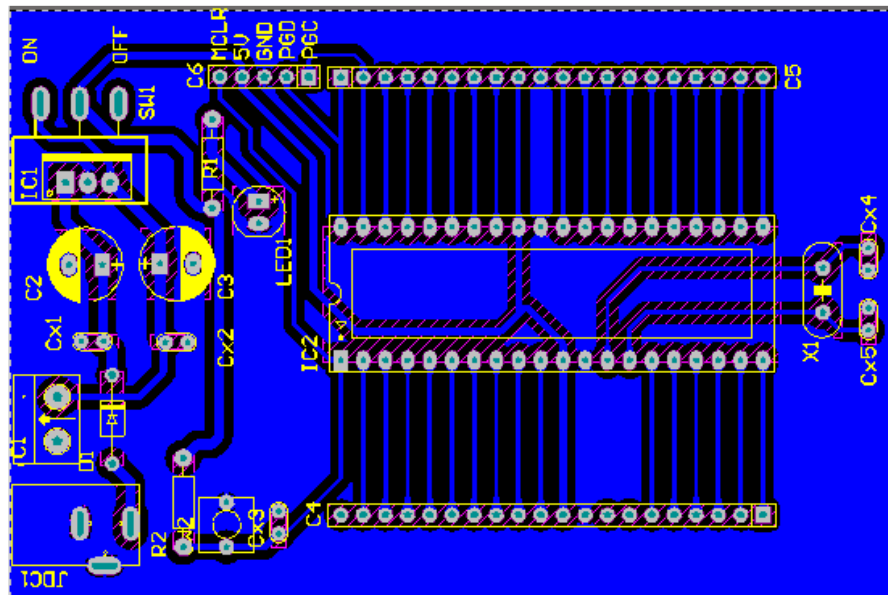
Việc thiết kế mạch điện tử trên phần mềm altium designer có thể được tóm tắt gồm các bước như sau:

- Đặt ra các yêu cầu bài toán.
- Lựa chọn linh kiện.
- Thiết kế mạch nguyên lý.
- Lựa chọn các chân linh kiện để chuyển sang mạch in Update mạch nguyên lý sang mạch in.
- Lựa chọn kích thước mạch in sắp xếp các vị trí các loại linh kiện như điện trở, tụ điện, IC...
- Đặt kích thước các loại dây nối.
- Đi dây trên mạch.
- Kiểm tra toàn mạch.

Bằng việc sử dụng phần mềm thiết kế mạch Alitum designer đồ án đã thiết kế được sơ đồ nguyên lý và mạch in PCB khối điều khiển cho mô hình robot lau nhà thông minh



Hình 2.36: Sơ đồ nguyên lý khối điều khiển



Hình 2.37: Sơ đồ mạch in khối điều khiển

CHƯƠNG 3: XÂY DỰNG PHẦN MỀM ĐIỀU KHIỂN

3.1 Lập trình cho hệ thống thời gian thực

Khái niệm về điều khiển thời gian thực

RTOS (real time operation system) là một hệ điều hành đa nhiệm dành cho các ứng dụng thời gian thực, như trong các hệ thống nhúng, robotics, điều khiển tự động.

Một hệ thống real-time được hiểu là một hệ thống làm việc với các sự kiện tức thời (real-time). Tuy nhiên, không phải mọi hệ thống đều có thể thực hiện được những quyết định tức thời hay đáp trả lại các sự kiện một cách tức thời như mong muốn. Khi xây dựng các ứng dụng phần mềm chúng ta luôn mong muốn thời gian trễ để đưa ra một lệnh hay một quyết định là nhỏ nhất. Các hệ thống đáp ứng bao giờ cũng có một thời gian trễ nhất định. Khái niệm “hệ thống thời gian thực” ở đây được hiểu ngầm như một hệ thống đáp ứng sự kiện với một thời gian trễ chấp nhận được.

Như vậy, sự tin cậy của một hệ thống thời gian thực không chỉ phụ thuộc vào sự chính xác của kết quả, mà còn phụ thuộc vào thời điểm đưa ra kết quả, hệ thống có lỗi khi yêu cầu về thời gian không được thỏa mãn.

Một hệ thống thời gian thực có các đặc điểm tiêu biểu sau :

- Tính bị động: Hệ thống phải phản ứng với các sự kiện xuất hiện vào các thời điểm thường không biết trước.
- Tính nhanh nhạy: Hệ thống phải xử lý thông tin một cách nhanh chóng để có thể đưa ra kết quả phản ứng một cách kịp thời.
- Tính đồng thời: Hệ thống phải có khả năng phản ứng và xử lý đồng thời nhiều sự kiện diễn ra.
- Tính tiên định: Dự đoán trước được thời gian phản ứng tiêu biểu, thời gian phản ứng chậm nhất cũng như trình tự đưa ra các phản ứng. Nếu một bộ điều khiển phải xử lý đồng thời nhiều nhiệm vụ, ta phải tham gia quyết định được về trình tự thực hiện các công việc và đánh giá được thời gian

xử lý mỗi công việc. Như vậy, người sử dụng mới có cơ sở để đánh giá về khả năng đáp ứng tính thời gian thực của hệ thống.

Xử lý thời gian thực:

Trong các hệ thống điều khiển với nhiều đầu vào và đáp ứng ở đầu ra thì số lượng công việc phải xử lý là khá lớn. Vì vậy ta phải chia khối lượng công việc đó thành nhiều tác vụ nhỏ. Mỗi tác vụ xử lý một công việc nhất định, có thể là quét bàn phím, đọc cảm biến, điều khiển động cơ,... Tại một thời điểm vi điều khiển chỉ thực hiện một tác vụ, nhưng với thời gian chuyển đổi rất nhanh giữa các tác vụ ta có thể coi như các tác vụ được thực hiện đồng thời.

Có hai kỹ thuật phổ biến cho lập trình hệ thống đa nhiệm là: trạng thái máy (State Machine) và lập lịch (Time Slicing).

State Machine chia các tác vụ thành các trạng thái (state). Tại mỗi thời điểm sẽ xác định một trạng thái và chỉ chuyển trạng thái khác khi có một điều kiện nào đó xảy ra.

Time Slicing gán cho mỗi tác vụ một chu kỳ thời gian nào đó. Khi thực hiện xong tác vụ này, vi xử lý sẽ chờ để nhảy sang thực hiện tác vụ khác. Các tác vụ cứ thế được thực hiện tuần tự.

3.2 Sử dụng trình biên dịch CCS

Sự ra đời của một loại vi điều khiển đi kèm với việc phát triển phần mềm ứng dụng cho việc lập trình cho con vi điều khiển đó. Vi điều khiển chỉ hiểu và làm việc với hai con số 0 và 1. Ban đầu để việc lập trình cho VĐK là làm việc với dãy các con số 0 và 1. Sau này khi kiến trúc của Vi điều khiển ngày càng phức tạp, số lượng thanh ghi lệnh nhiều lên, việc lập trình với dãy các số 0 và 1 không còn phù hợp nữa, đòi hỏi ra đời một ngôn ngữ mới thay thế. Và ngôn ngữ lập trình Assembly. Ở đây ta không nói nhiều đến Assembly. Sau này khi lập trình cho Vi điều khiển một cách ngắn gọn và dễ hiểu hơn đã dẫn đến sự ra đời của ngôn ngữ C ra đời, nhu cầu dùng ngôn ngữ C để thay cho ASM trong việc mô tả các lệnh nhiều chương trình soạn thảo và biên dịch C cho Vi điều khiển : Keil C, HT-PIC, MikroC, CCS...

Trong đề tài này đã chọn CCS là trình biên dịch để viết chương trình điều khiển cho robot vì CCS là một công cụ lập trình C mạnh cho Vi điều khiển PIC.

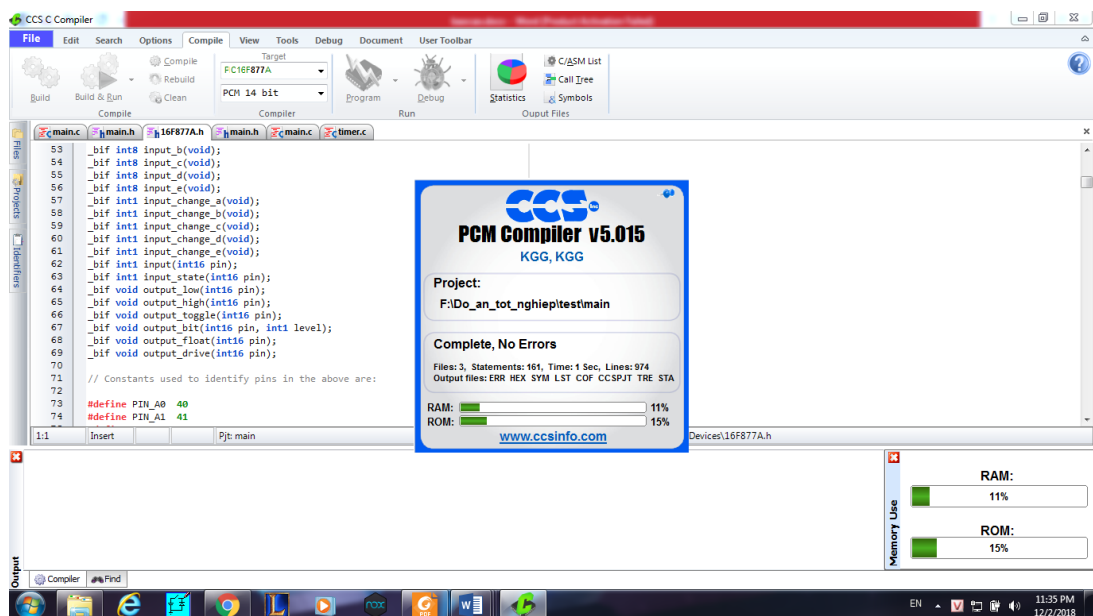
Giới thiệu về CCS

CCS là trình biên dịch lập trình ngôn ngữ C cho Vi điều khiển PIC của hãng Microchip.

Chương trình là sự tích hợp của 3 trình biên dịch riêng biệt cho 3 dòng PIC khác nhau đó là:

- PCB cho dòng PIC 12-bit opcodes
- PCM cho dòng PIC 14-bit opcodes
- PCH cho dòng PIC 16 và 18-bit

Tất cả 3 trình biên dịch này được tích hợp lại vào trong một chương trình bao gồm cả trình soạn thảo và biên dịch là CCS, phiên bản mới nhất là PCWHD Compiler Ver 5.081. Trong đề tài này đã sử dụng phiên bản PCWHD Compiler Ver 5.015.



Hình 3.1: Giao diện làm việc của CCS

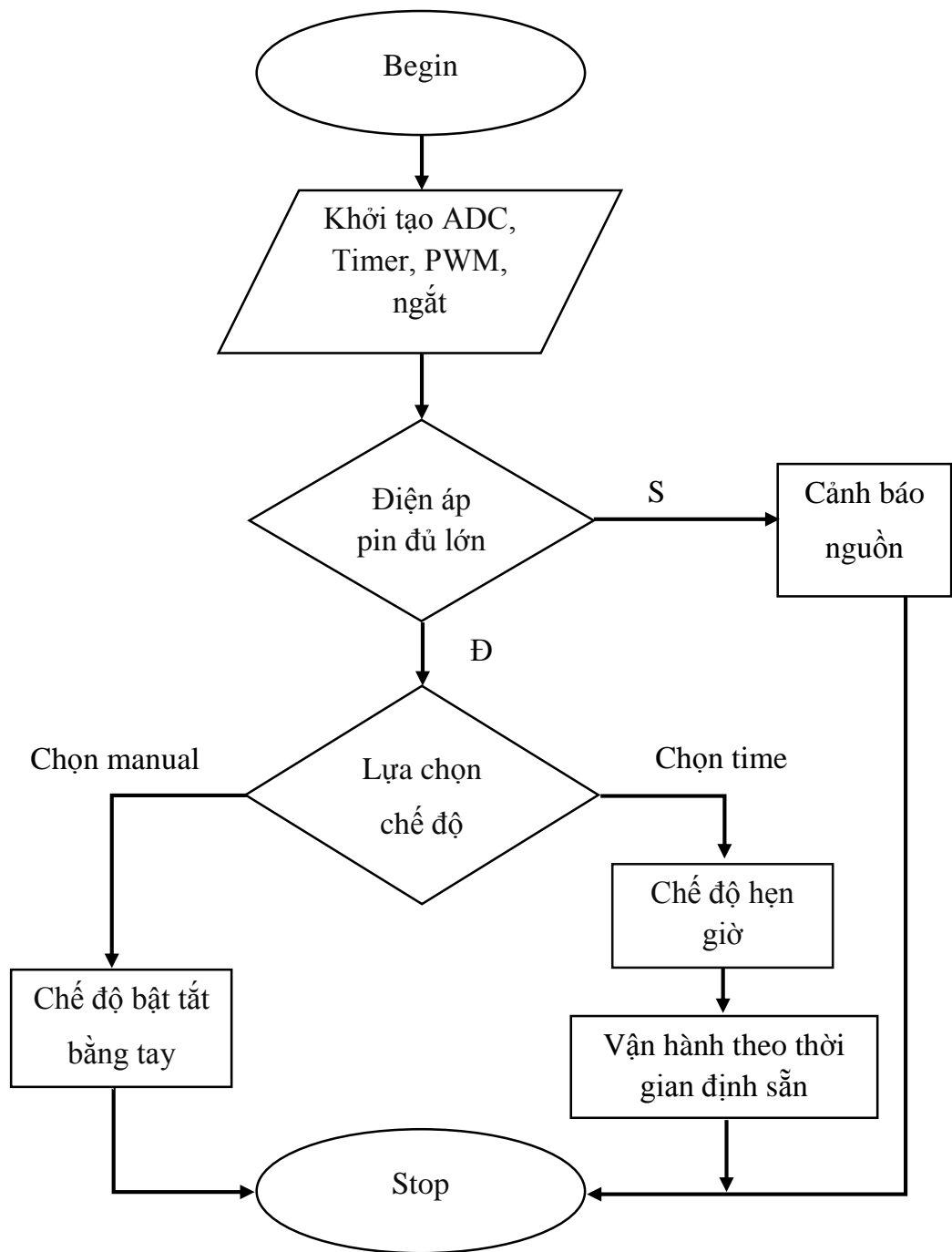
Giống như nhiều trình biên dịch C khác cho PIC, CCS giúp cho người sử dụng nắm bắt nhanh được vi điều khiển PIC và sử dụng PIC trong các dự án. Các

chương trình điều khiển sẽ được thực hiện nhanh chóng và đạt hiệu quả cao thông qua việc sử dụng ngôn ngữ lập trình cấp cao – Ngôn ngữ C.

3.3 Lưu đồ thuật toán điều khiển robot

3.3.1 Lưu đồ thuật toán chung

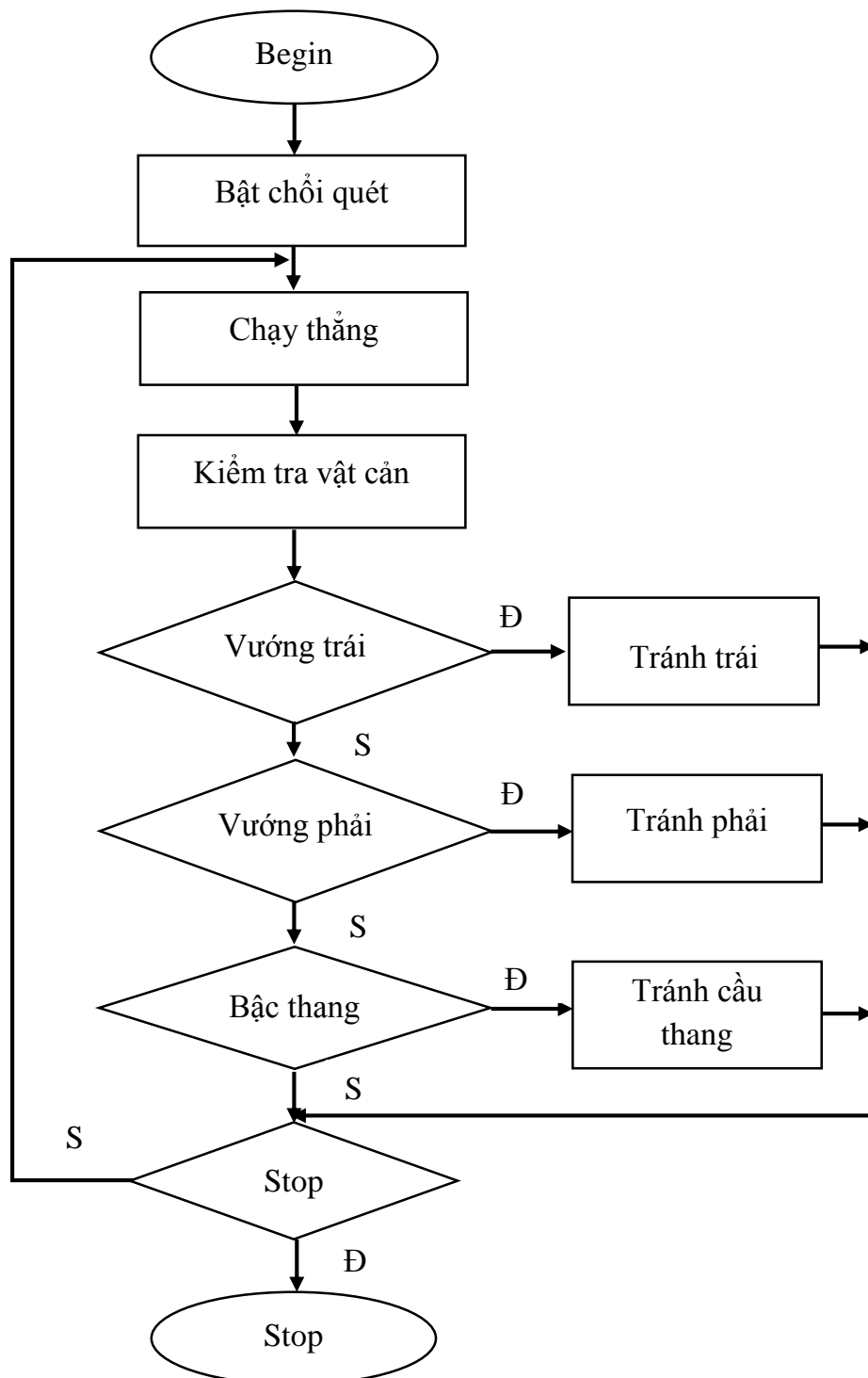
Chương trình điều khiển robot có thể diễn tả bằng lưu đồ thuật toán sau:



Hình 3.2: Lưu đồ thuật toán của robot.

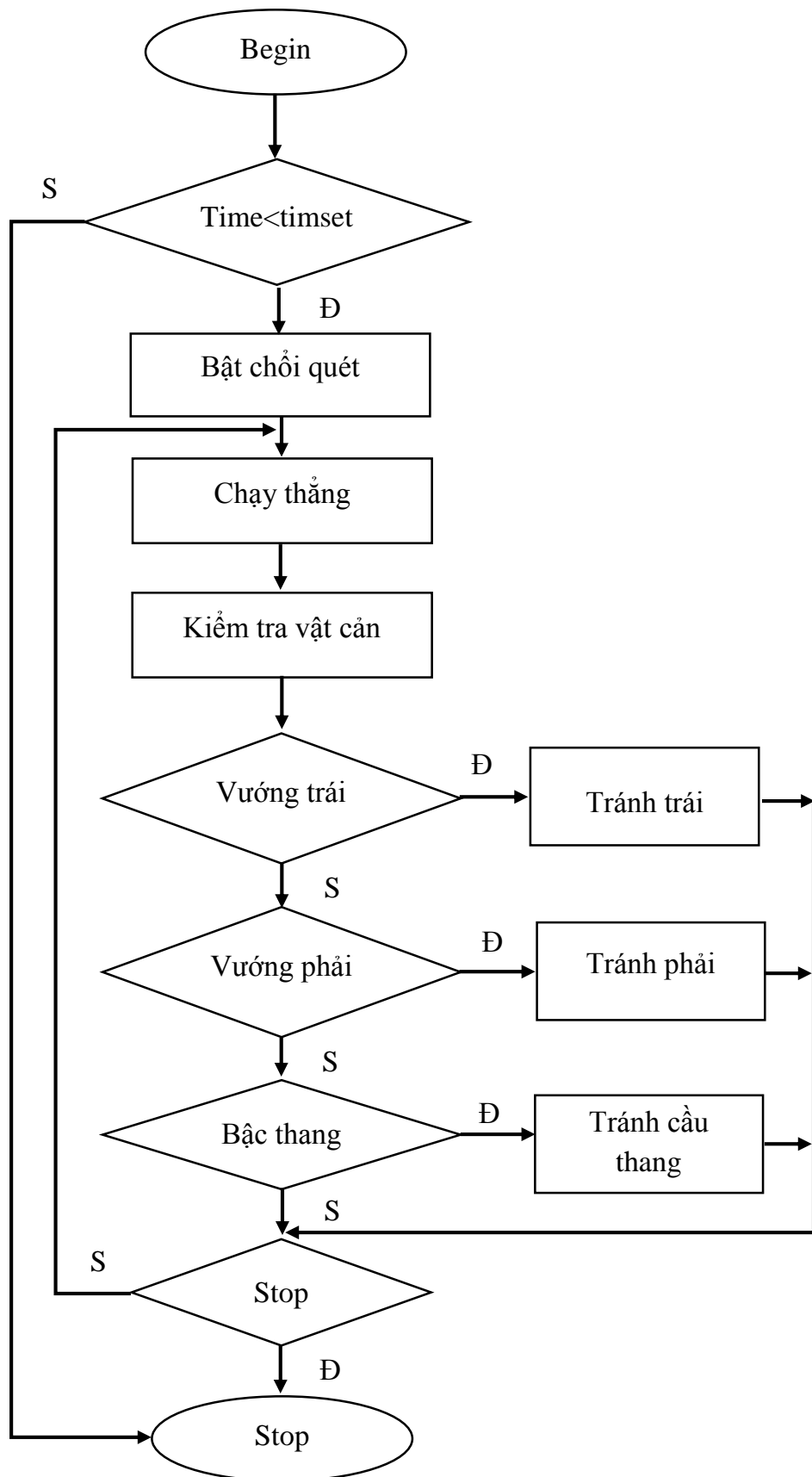
3.3.2 Lưu đồ thuật toán chế độ bật tắt bằng tay

Sau khi khởi tạo ADC, Timer, PWM, ngắt và kiểm tra nguồn từ pin, robot đi vào trạng thái chờ. Nếu nút Manual được nhấn robot sẽ làm việc ở chế độ bật tắt bằng tay, quá trình hoạt động của chế độ Manual có thể được biểu diễn như lưu đồ sau:



Hình 3.3: Lưu đồ thuật toán chế độ Manual

3.3.3 Lưu đồ thuật toán chế độ hẹn giờ



Hình 3.4: Lưu đồ thuật toán chế độ Time

3.4 Sử dụng ADC để đo điện áp pin

Để đo được điện áp DC của pin thông qua cầu phân áp ta sử dụng bộ ADC 10bit của PIC16F877A.

Do sử dụng điện áp tham chiếu từ nguồn 5V cấp cho vi xử lý nên độ phân giải (step size) là: $SS = \frac{5000mV}{1023} = 4,887mV$ (3.1)

Chọn chu kỳ lấy mẫu bằng chu kỳ lệnh của vi xử lý.

Chân RA0 được chọn là chân đo điện áp sau cầu phân áp, chọn kênh chuyển đổi ADC là kênh 0.

Đoạn mã chương trình khởi tạo ADC:

```
setup_adc(ADC_CLOCK_INTERNAL);
setup_adc_ports(AN0);
set_adc_channel(0);
```

Do điện áp được đưa vào bộ ADC và điện áp thực trên Pin có tỉ lệ là 1:3 nên điện áp trên Pin sẽ bằng 3 lần kết quả từ bộ ADC.

Đoạn mã chương trình đọc điện áp Pin:

```
Void DocDienAp(void)
{
    adc = read_adc();
    DienApDC= 3*5000*(float)adc / 1023;
}
```

3.5 Cài đặt bộ hẹn giờ bằng Timer

Chọn Timer1 làm bộ định thời để cài đặt bộ hẹn giờ cho robot.

Mạch sử dụng thạch anh 4MHz nên tần số hệ thống sẽ là :

$$f = \frac{1}{4MHz} = 1MHz \quad (3.2)$$

Cài đặt Timer1 ở chế độ bộ chia 8 nên tần số của bộ Timer1 còn lại là

$1\text{MHz}/8 = 0.125\text{MHz} = 125\text{KHz}$ và chu kì sẽ là $8\mu\text{s}$.

Tạo hàm timer đếm với đơn vị 1 phút:

Chọn giá trị bắt đầu đếm cho Timer1 là 3036, Timer1 là timer 16 bit nên sẽ đếm thêm $65536 - 3036 = 62500$ xung nữa thì tràn, mỗi xung có thời gian là $8\mu\text{s}$ nên thời gian để Timer1 tràn sẽ là $62500 \times 8\mu\text{s} = 500\text{ms} = 0,5\text{s}$

Để tạo thời gian đơn vị là 1 phút ta cần cho Timer1 tràn 120 lần.

Ta có đoạn mã chương trình:

```
setup_timer_1(T1_internal/T1_div_by_8);
set_timer1(3036);
void timer(void)
{ if(trm1if==1)
  { trm1if=0;
    set_timer1(3036);
    time++;
    if(time==(120*time_set))//1ph
    { ato=0;
      time=0;}
  }
}
```

3.6 Bài toán di chuyển của robot

Để điều khiển robot đi theo một quỹ đạo nào đó, ví dụ như di chuyển theo một đường cong nào đó thực chất là ta thay đổi tỉ lệ tốc độ 2 bánh xe của robot.

Gọi tỉ lệ tốc độ quay giữa 2 bánh xe là R:

- $R = \frac{V_t}{V_p}$ (3.3) với V_t là tốc độ bánh xe bên trái, V_p là tốc độ bánh xe bên phải. Khoảng cách giữa 2 bánh xe là L.
- Với $R = 1$, robot tiến thẳng.
- Với $0 < R < 1$, robot đi theo hình tròn ngược chiều kim đồng hồ với đường kính D, sao cho:

$$R = \left(\frac{D-L}{D+L} \right) \quad (3.4)$$

- Với $R > 1$, robot đi theo hình tròn thuận chiều kim đồng hồ với đường kính D , sao cho:

$$R = \left(\frac{D+L}{D-L} \right) \quad (3.5)$$

- Với $R = 0$, robot quay tròn tại chỗ với tâm là bánh trái.
- Với $R = \infty$, robot quay tròn tại chỗ với tâm là bánh phải.
- Với $R = -1$, hai bánh xe quay ngược chiều và đồng tốc, robot quay tròn tại chỗ, tâm là điểm giữa hai bánh xe.

Tổ hợp các quỹ đạo cơ bản ở trên lại với nhau ta có thể cho robot di chuyển theo một hình dạng bất kỳ nào đó.

Điều khiển động cơ bằng điều chế độ rộng xung:

Để điều khiển tốc độ động cơ theo ý muốn mà giữ nguyên nguồn điện cấp vào động cơ trong đề tài em sử dụng phương pháp điều chế độ rộng xung (Pulse Width Modulation – PWM).

Chu kỳ PWM của Pic16f877a được tính theo công thức:

$$\text{PWM period} = [(PR2 + 1)] \times 4 \times TOSC \times (\text{TMR2 prescale value}) \quad (3.6)$$

Trong đó: TOSC là chu kỳ của thạch anh tạo dao động.

TMR2 prescale value giá trị chia trước của Timer2.

Mạch vi xử lý sử dụng thạch anh F_{OSC} 4MHz nên chu kỳ là:

$$TOSC = \frac{1}{F_{OSC}} = \frac{1}{4MHz} = 250ns \quad (3.7)$$

Chọn TMR2 prescale value = 16. Chọn PR2 = 255.

Chu kỳ PWM: PWM period = $[(255 + 1)] \times 4 \times 250ns \times (16) = 4096us$

Tần số PWM = $\frac{1}{4096us} = 244Hz$.

Để điều khiển cả 2 động cơ bằng PWM, em sử dụng cả 2 khối CCP1 và CCP2.

Đoạn mã chương trình khởi tạo PWM, Timer2:

```
setup_timer_2(T2_DIV_BY_16,255,1);//tan so la 244hz
setup_ccp1(CCP_PWM);//che do pwm
set_pwm1_duty(0);//du ty la 0%
setup_ccp2(CCP_PWM);//che do pwm
set_pwm2_duty(0);
```

Để điều khiển chiều động cơ ta chỉ cần thay đổi mức tín hiệu xuất ra ở 2 chân RC0 (đối với động cơ phải) và RC3 (đối với động cơ trái) đồng thời thay đổi giá trị trong hàm *set_pwm1_duty()* để điều chỉnh tốc độ động cơ.

Ví dụ đoạn mã chương trình điều khiển động cơ bên phải tiến lên:

```
void _phaitien(void)
{
    output_bit(ENABLEA,1);
    output_high(pin_c0);
    set_pwm1_duty(duty);
}
```

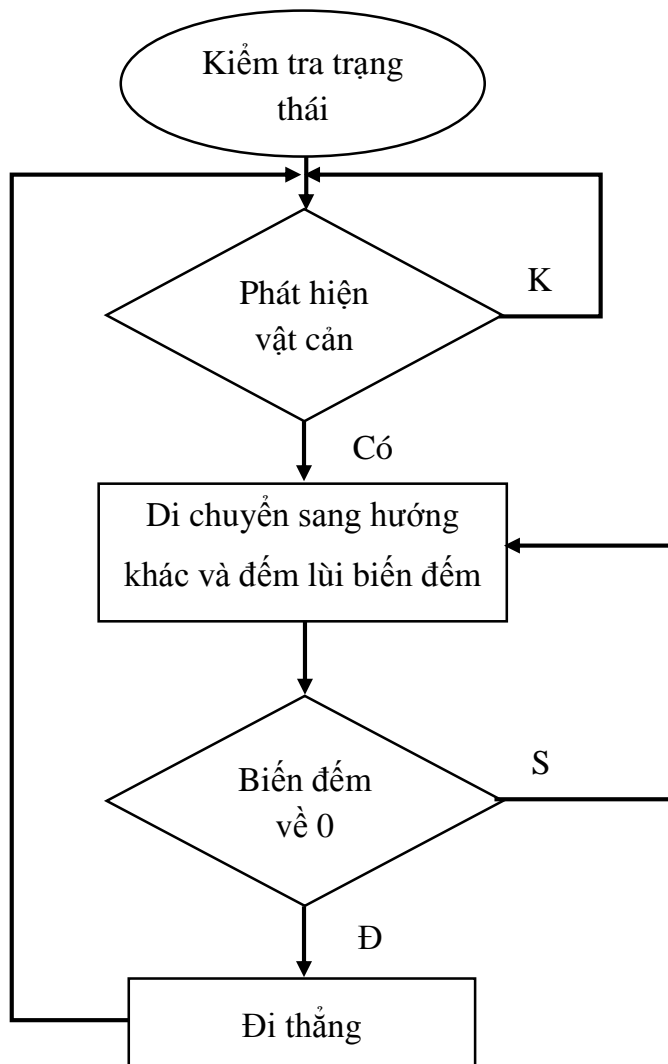
3.7 Tránh vật cản và cầu thang

Vành dò vật cản phía trước và cảm biến dưới gầm cho ta biết robot có gặp vật cản hay độ cao hay không. Sau đây là thuật toán tránh vật cản và độ cao của robot:

Bước 1: Kiểm tra trạng thái công tắc hành trình và cảm biến. Nếu không có vật cản hay cầu thang thì tiếp tục bước 1. Nếu có vật cản hoặc cầu thang, thực hiện bước 2.

Bước 2: Robot di chuyển sang hướng khác đồng thời biến đếm thời gian bắt đầu đếm ngược từ giá trị đặt trước. Thực hiện bước 3.

Bước 3: Nếu biến đếm chưa về 0 mà tiếp tục gặp vật cản hoặc độ cao quay lại bước 2. Nếu biến đếm về 0 robot đi thẳng. Sau đó quay lại bước 1.



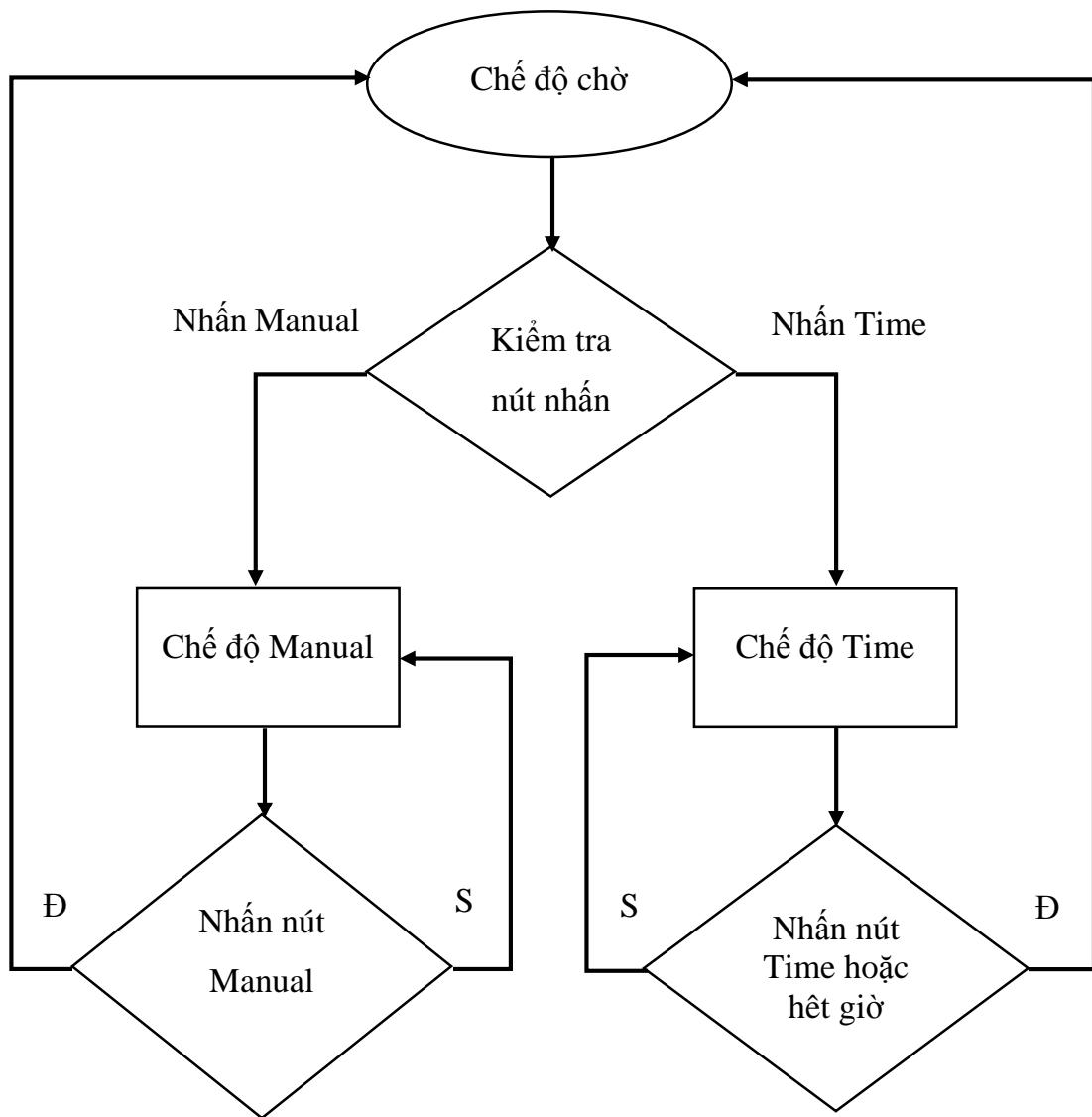
Hình 3.5: Lưu đồ thuật toán tránh vật cản và cầu thang

3.8 Xử lý bàn phím

Sau khi bật công tắc nguồn và điện áp Pin lớn hơn mức cho phép, robot sẽ ở chế độ chờ.

Nếu nhấn nút Manual, robot sẽ hoạt động đến khi ta nhấn nút Manual lần nữa để dừng. Và robot sẽ quay về chế độ chờ.

Nếu nhấn nút Time, robot sẽ hoạt động tới khi hết thời gian đặt hoặc ta nhấn phím Time lần nữa để robot quay về chế độ chờ.



Hình 3.6: Lưu đồ chương trình kiểm tra nút

3.9 Kết luận chương

Trên cơ sở thiết kế phần cứng đã được thiết kế ở chương 2, trong chương này đồ án đã xây dựng được phần mềm điều khiển cho hệ thống robot lau nhà thông minh. Phần mềm có các chức năng như: kiểm tra điện áp pin có cao hơn ngưỡng hoạt động đặt trước không mới cho robot hoạt động, nếu điện áp pin thấp hơn ngưỡng hoạt động đặt trước sẽ đưa ra cảnh báo và không cho phép robot hoạt động. Có thể điều khiển tốc độ di chuyển của robot bằng phần mềm. Thực hiện chọn chế độ làm việc cho robot lau nhà thông minh bằng nút nhấn. Khi hoạt động robot có thể vừa di chuyển tránh

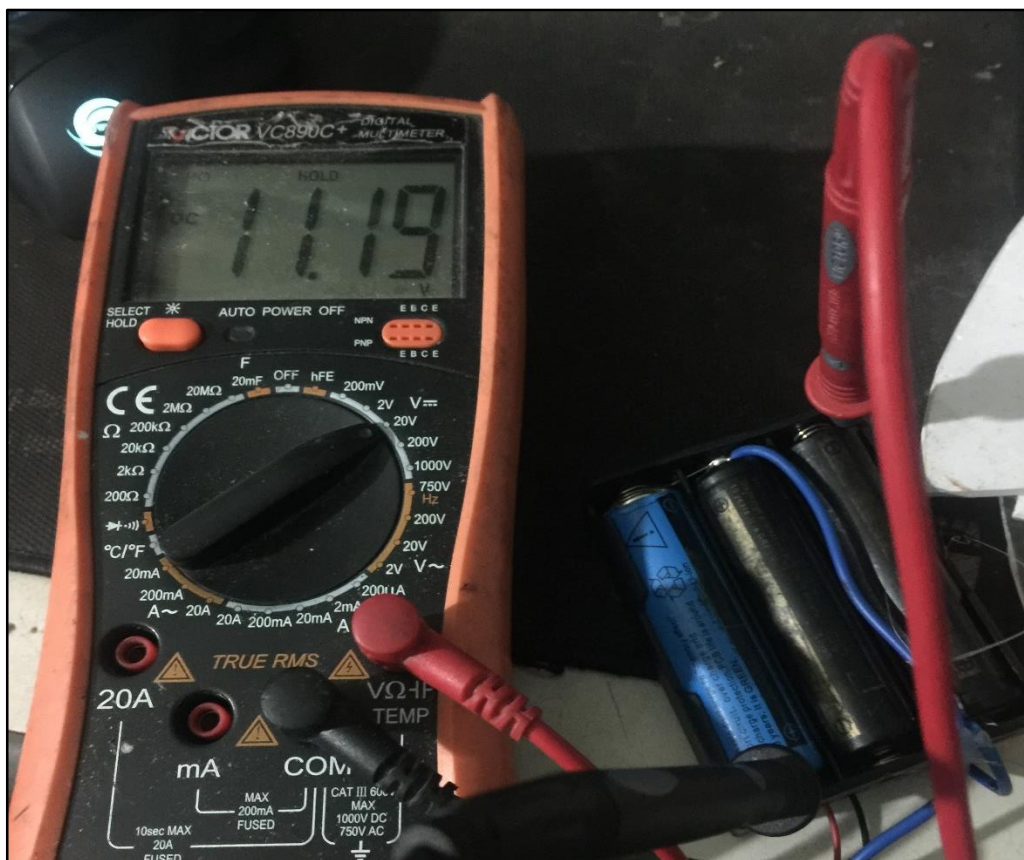
vật cản và cầu thang, vừa làm sạch sàn nhà. Robot có thể hoạt động trong một khoảng thời gian định trước ở chế độ Time.

KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

A. Kiểm tra trước khi hoạt động

Tiến hành đo đặc nguồn cấp, mạch công suất, mạch cấp nguồn, kiểm tra phím bấm giao tiếp, các công tắc hành trình và cảm biến trước khi thử nghiệm các hoạt động của robot.

- Kiểm tra tình trạng Pin: Dùng đồng hồ đo ở chế độ đo điện áp, kiểm tra cực dương và cực âm của bộ 3 cell Pin còn sử dụng tốt nếu điện áp đạt được từ 11.1 - 12,6(V).



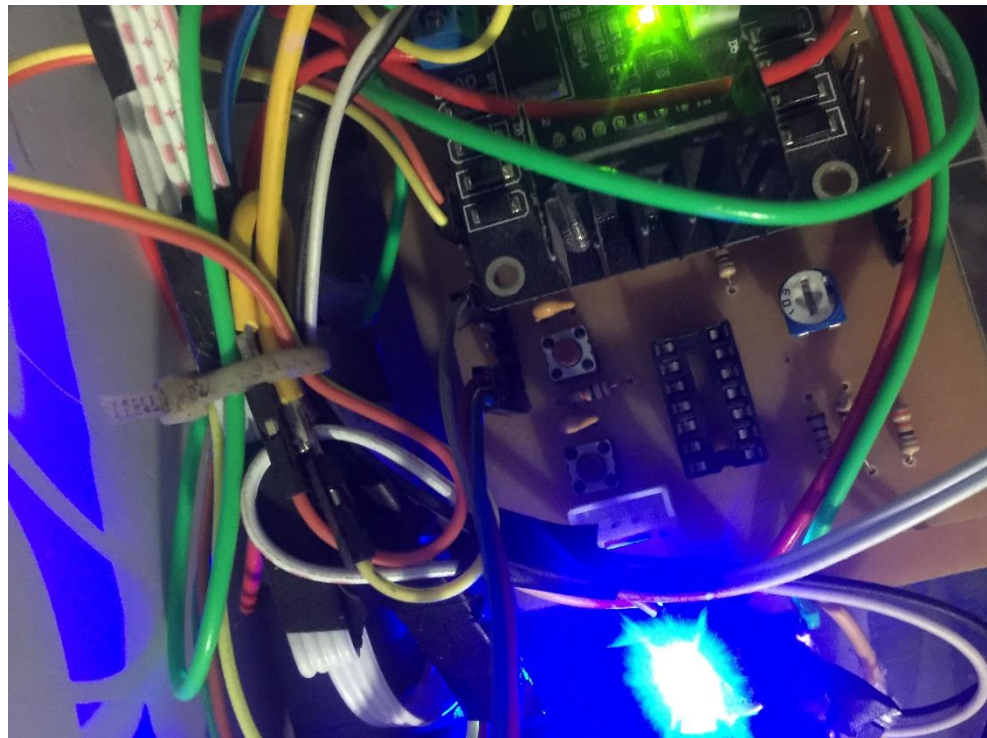
Hình 4.1: Kiểm tra nguồn Pin

- Kiểm tra nguồn cấp cho vi xử lý, khối nguồn công suất cấp cho động cơ: cắm jack và bật công tắc nguồn cho các khối, nếu đèn báo của tất cả các khối đều sáng chứng tỏ phần mạch điều khiển và mạch công suất cho robot đã được cấp nguồn và sẵn sàng hoạt động.



Hình 4.2: Kiểm tra khối công suất

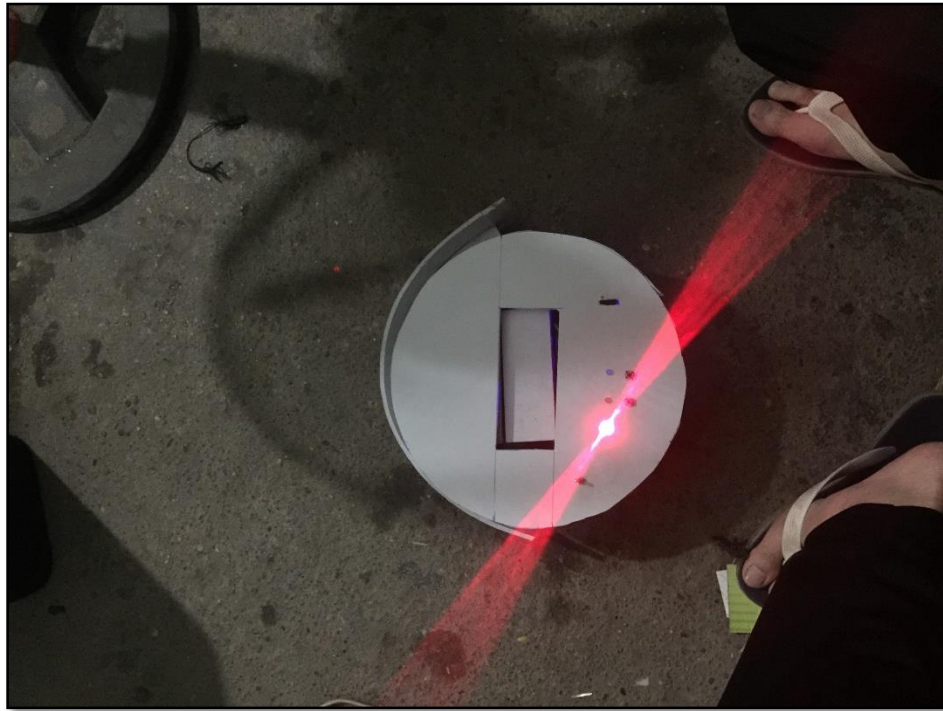
- Kiểm tra phím nhấn: kiểm tra hoạt động của phím nhấn đã đáp ứng đúng với yêu cầu chức năng đề ra hay chưa.



Hình 4.3: Kiểm tra phím nhấn

B. Chạy thử mô hình

Robot di chuyển tránh vật cản



Hình 4.4: Vị trí bắt đầu chạy của robot

Sau khi chạy được 1 đoạn đường robot gặp vật cản



Hình 4.5: Robot gặp vật cản và dừng lại

Sau đó robot di chuyển sang phải



Hình 4.6: Robot di chuyển sang hướng khác

Robot tiếp tục di chuyển thẳng



Hình 4.7: Robot đi thẳng

Robot tránh rơi

Hình 4.8: Robot gặp cạnh bàn



Hình 4.9: Robot dừng và lùi



Hình 4.10: Rác robot thu được

C. Kết luận

Đề tài đã nghiên cứu phương pháp điều khiển cũng như thiết kế, chế tạo robot lau nhà dựa trên nhu cầu thực tiễn. Hệ thống mới chỉ ở mức độ đơn giản, còn nhiều vấn đề phức tạp để có thể đưa vào thực tiễn. Tuy nhiên, robot lau nhà đã đạt yêu cầu đề ra.

Mô hình được hoàn thiện bao gồm cả phần thiết kế cơ khí, mạch điện tử và chương trình điều khiển. Nghiên cứu và ứng dụng vi điều khiển thành công trong việc điều khiển robot lau nhà tự động thành công

Kết quả đạt được:

- Robot làm việc hiệu quả, sàn nhà sạch bụi, mảnh vụn.
- Robot di chuyển và tránh vật cản linh hoạt.
- Khối mạch vi điều khiển và công suất hoạt động an toàn.
- Các phím nhấn chọn chế độ hoạt động tốt.

- Robot hoạt động tốt khi hẹn giờ.
- Bộ nạp pin hoạt động tốt.
- Có thể nạp, thay đổi code chương trình dễ dàng.
- Bộ đọc điện áp và cảnh báo hoạt động ổn định.

Các điểm hạn chế

- Robot chỉ hoạt động tốt ở các sàn gỗ và gạch men, hạn chế hoạt động ở những nơi có độ nhấp nhô cao, không gian chật hẹp.
- Tiếng ồn hơi lớn khi robot hoạt động.
- Chưa làm sạch tốt các góc.
- Còn lặp lại những diện tích đã làm việc.
- Dung lượng pin còn khiêm tốn.
- Robot chưa tự động nạp điện được mà vẫn cần sự trợ giúp của con người

D. Hướng phát triển cho đề tài

Với điều kiện hiện thực tế em xin đề xuất một số hướng phát triển của đề tài như sau:

- Robot cần được nâng cấp khả năng tự sạc nguồn khi cạn.
- Robot cần phát triển khả năng nhận dạng không gian.
- Robot cần lập bản đồ và vạch ra kế hoạch di chuyển tối ưu để đi hết diện tích sàn với thời gian ngắn nhất.
- Robot có khả năng hoạt động theo thời gian biểu được định sẵn.
- ❖ Ví dụ về phát triển khả năng tìm nguồn sạc cho robot

Mục tiêu

Khi sắp hết Pin, hoặc thực hiện xong nhiệm vụ, robot tự tìm đến một vị trí nhất định trong phòng dừng lại, tự động nạp đầy Pin. Sau khi nạp đầy Pin robot lại tiếp tục công việc của mình.

Phương pháp đề cử: Tìm đường bằng phương pháp xử lý ảnh.

Trên robot ta đặt một hệ thống camera gồm: một camera thông thường và một camera hồng ngoại tích hợp đèn chiếu tia hồng ngoại. Hạn chế khi sử dụng một camera đến từ sự ảnh hưởng của màu sắc hay họa tiết môi trường phức tạp. Đó cũng là hạn chế chung của xử lý ảnh trong không gian 2D cho robot tự hành tránh vật cản. Bằng việc sử dụng thêm camera cảm nhận độ sâu ta thu được đầy đủ hơn thông tin từ môi trường, nhờ vậy mà những hạn chế trên được khắc phục. Thông tin thêm ở đây là giá trị khoảng cách tới vật, lúc này công việc xử lý ảnh phức tạp hơn do phải làm việc trong không gian 3D. Ảnh từ thế giới thực được thu về thông qua hai camera, sau đó qua khâu hiệu chỉnh, khắc phục méo trước khi cho ra bản đồ độ sâu. Công việc này được thực hiện bằng các giải thuật đặc biệt với sự hỗ trợ của thư viện OpenCV. Ta thu được đầy đủ thông tin của vật cản trong môi trường như chiều cao, bề rộng hay khoảng cách từ camera tới vật sau khâu khôi phục 3D (3D reconstruction). Từ đó robot có thể lập ra bản đồ độ sâu chứa thông tin vị trí vật trong không gian phía trước robot. Và từ bản đồ

độ sâu robot có thể lập ra kế hoạch di chuyển, nhận diện vật cản và không gian trống để đưa ra hướng đi tối ưu cho robot đồng thời có thể tìm đường tới điểm đích mong muốn.

Phương pháp này tỏ ra ưu điểm hơn nhiều so với một số phương pháp tìm đường khác như:

- Có thể phát hiện vật cản chính xác, hiệu quả, không phụ thuộc hình dạng, kích thước hay màu sắc vật, có thể phát hiện các vật thể trong không trung.
- Robot có thêm “cái nhìn” về môi trường, từ đó hỗ trợ việc dẫn đường cho robot tốt hơn.
- Ít bị ảnh hưởng bởi nhiễu.
- Việc có thêm camera xử lý ảnh giúp phát triển nhiều tính năng thông minh hơn nữa cho robot ví dụ như ra lệnh cho robot bằng cách ra hiệu bằng tay cho robot v.v...

PHỤ LỤC

➤ Chương trình điều khiển động cơ

```

void start(void)

{
    _phaitien();

    _traitien();

}

void stop(void)

{

    output_bit(ENABLEA,0);

    output_high(pin_c0);

    set_pwm1_duty(0);

    output_bit(ENABLEB,0);

    output_high(pin_c3);

    set_pwm2_duty(0);

}

void _phaitien(void)

{

    output_bit(ENABLEA,1);

    output_high(pin_c0);

    set_pwm1_duty(duty);

}

void _traitien(void)

```

```

{
    output_bit(ENABLEB,1);

    set_pwm2_duty(-duty);

    output_low(pin_c3);
}

void _phailui(void)
{
    output_bit(ENABLEA,1);

    output_low(pin_c0);

    set_pwm1_duty(duty);
}

void _trailui(void)
{
    output_bit(ENABLEB,1);

    set_pwm2_duty(-duty);

    output_high(pin_c3);
}

void back(void)
{
    _trailui();

    _phailui();
}

```

➤ **Chương trình điều khiển tránh vật cản và độ cao**

```

void RePhai(void)

{

if(countRight<=0) Return;

else{

countRight--;

if(countRight<=0) _phaitien();

}

}

void ReTrai(void)

{

if(countLeft<=0) Return;

else{

countLeft--;

if(countLeft<=0) _traitien();

}

}

void tranhroi(void)

{

if(countBack<=0)

Return;

else{

countBack--;

```



```

if(countBack<=0)

{

    _traitien();

    _phailui();

    delay_ms(1000);

    _phaitien();

}

}

}

void _xacminhR(void)

{

    if(HT1==1 ) Return;

    else

    {

        if(countRight<=0) _phailui();

        countRight=timeRight;

    }

}

void _xacminhL(void)

{

    if(HT2==1) Return;

    else

    {

```

```
if(countLeft<=0) _trailui();
```

```
countLeft=timeLeft;
```

```
}
```

```
}
```

```
void _xacminhCB(void)
```

```
{
```

```
if(CB==0) Return;
```

```
else
```

```
{
```

```
if(countBack<=0)back();
```

```
countBack=timeBack;
```

```
}
```

```
}
```

➤ Chương trình đọc nút nhấn

```
void _manual(void)
```

```
{
```

```
if(sw==0) {
```

```
while(sw==0);
```

```
manual++;
```

```
if>manual>1) manual=0;
```

```
}
```

```
}
```

```

void _auto(void)

{

if(sw_ato==0){

while(sw_ato==0);

ato++;

if(ato>1) ato=0;

}

}

```

➤ **Chương trình cài đặt thời gian**

```

void timer(void)

{

if(trm1if==1)

{

trm1if=0;

set_timer1(3036);

time++;

if(time==(120*time_set))/1ph

{

ato=0;

time=0;

}

}

}

```

➤ **Chương trình chính:**

```
void main()

{

    set_tris_a(0xff);

    set_tris_D(0x00);

    ato=0;

    manual=0;

    on=0;

    on_auto=0;

    time=0;

    pin=0;

    output_bit(PIN_D0,1); //den manual off

    output_bit(PIN_D1,1); //den auto off

    output_bit(PIN_B2,0); //motor quet off

    output_bit(PIN_B3,0); //buzzer off

    setup_timer_2(T2_DIV_BY_16,255,1); //tan so la 244hz

    setup_ccp1(CCP_PWM); //che do pwm

    set_pwm1_duty(0); //du ty la 0%

    setup_ccp2(CCP_PWM); //che do pwm

    set_pwm2_duty(0);

    setup_timer_1(T1_internal/T1_div_by_8);

    set_timer1(3036);
```

```

setup_adc(ADC_CLOCK_INTERNAL);

setup_adc_ports(AN0);

set_adc_channel(0);

stop();

    while(true)

{

DocDienAp();

_manual();

_auto();

    if(DienApDC<pinset)

{

output_bit(PIN_B3,1); //buzzer on

pin=0;

manual=0;

ato=0;

time=0;

}

    else {

output_bit(PIN_B3,0); //buzzer off

pin=1;

}

    if(on==0&&manual==1&&pin==1) {start(); on=1;}

    if(on_auto==0&&ato==1&&pin==1) {start(); on_auto=1;}

```

```

if(manual==1 &&on_auto==0 &&pin==1) {

    output_bit(PIN_D0,0); //den manual on

    output_bit(PIN_D1,1); //den auto off

    output_bit(PIN_B2,1); //motor quet on

    RePhai();

    ReTrai();

    trahroi();

    _xacminhR();

    _xacminhL();

    _xacminhCB();

    delay_ms(10);

}

else if(ato==1 &&on==0 &&pin==1) {

    timer();

    output_bit(PIN_D0,1); //den auto off

    output_bit(PIN_D1,0); //den auto on

    output_bit(PIN_B2,1); //motor quet on

    RePhai();

    ReTrai();

    trahroi();

    _xacminhR();

    _xacminhL();

```

```

    _xacminhCB();

    delay_ms(10);

}

else if(manual==0||ato==0||pin==0) {

    output_bit(PIN_D0,1); //den manual off

    output_bit(PIN_D1,1); //den auto on

    output_bit(PIN_B2,0); //motor quiet off

    stop();

    on=0;

    on_auto=0;

    time=0;

}

}

}

```

TÀI LIỆU THAM KHẢO

Các tài liệu

1. Lê Minh Triền. Đồ án tốt nghiệp, Ứng dụng vi điều khiển thiết kế máy hút bụi thông minh. ĐHGTVT – 2013.
2. Nguyễn Thiện Phúc. Robot giống người. Nxb KHKT.
3. Sascha Kolski. Mobile Robots Perception & Navigation. Publisher: InTech 2007.
4. Zoran Gacovski. Mobile Robots Current Trends. Published: October 26th 2011.

Các website:

1. <http://www.google.com.vn/>
2. <http://www.irobot.com/forum/>
3. <http://www.dientuvietnam.net/forums/>
4. <http://www.alldatasheet.com/>
5. [http:// www.microchip.com](http://www.microchip.com).
6. <http://www.dieukhien.net/vn/>
7. <http://en.wikipedia.org/>
8. <http://codientu.org/>
9. <http://picvietnam.com/>
10. <https://www.youtube.com/>