

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH



ĐỒ ÁN TỐT NGHIỆP
NGÀNH CNKT Ô TÔ

**MÔ HÌNH HÓA XE TỰ HÀNH AGV VẬN CHUYỂN HÀNG
TRONG NHÀ XƯỞNG**

GVHD : PGS. TS ĐỖ VĂN DŨNG
SVTH : LÊ NGỌC HUÂN
14145093
TRẦN NHẬT MINH KHÔI
14145126



Tp. Hồ Chí Minh, tháng 01/2018

LỜI CẢM ƠN

Sau quá trình học tập tại trường và trong suốt thời gian làm đồ án tốt nghiệp. Em xin chân thành cảm ơn quý thầy cô đã dạy cho em những kiến thức, truyền đạt cho chúng em những kỹ năng để bước vào xã hội sau khi rời khỏi giảng đường đại học.

Chúng em xin chân thành cảm ơn đến thầy hướng dẫn PGS.TS Đỗ Văn Dũng, thầy Th.S Phan Nguyễn Quý Tâm đã giúp đỡ chúng em trong quá trình làm đồ án tốt nghiệp.

Cũng xin gửi lời cảm ơn đến những người bạn đã giúp đỡ trong suốt những tháng ngày học tập tại trường. Và đặc biệt cảm ơn gia đình đã tạo điều kiện để chúng em có thể hoàn thành chương trình đại học.

Sinh viên thực hiện

Lê Ngọc Huân

Trần Nhật Minh Khôi

MỤC LỤC

DANH MỤC HÌNH ẢNH	4
Danh mục các bảng	6
1.1. Lý do chọn đề tài	7
1.2. Nhiệm vụ của đề tài	7
1.3. Xe tự hành trong công nghiệp	7
1.3.1. Lịch sử phát triển của AGV(Automotive Giude Vehicle)	8
1.3.2. Phân loại AGV.....	8
1.3.2.1. Phân loại theo chức năng.....	8
1.3.2.2. Phân loại theo dạng đường đi	9
2.1. Lý thuyết về xe tự hành	11
2.2. Lý thuyết điều khiển.....	16
2.3. Bộ điều khiển PID (Proportional – Integral – Derivative).....	20
CHƯƠNG 3: CÁC LINH KIỆN SỬ DỤNG	25
3.1. Arduinio Mega 2560	25
3.2. Cầu H.....	26
3.2.1. Sơ lược về cầu H	26
3.2.2. Cầu H L298N.....	29
3.3. Động cơ DC Servo giảm tốc GA25V1	30
3.4. Cảm biến hồng ngoại.....	31
3.4.1. Tổng quát về cảm biến hồng ngoại.....	31
3.4.2. Cảm biến hồng ngoại E18-DK80N	32
3.5. Cảm biến Hall	32
3.6. LCD TFT 2.4inches.....	34
3.6.1. Thông số kỹ thuật	34
3.6.2. Sơ đồ chân LCD TFT	34
CHƯƠNG 4: THIẾT KẾ VÀ THI CÔNG MÔ HÌNH	36
4.1 Thiết kế cơ khí	36
4.1.1. Lựa chọn phương án thiết kế	36
4.1.2. Lên các bản vẽ	36
4.2. Thiết kế mạch ra chân.....	36
4.3. Xây dựng chương trình điều khiển.....	38
4.3.1. Lưu đồ giải thuật	38
4.3.2. Xây dựng bộ điều khiển PID điều khiển tốc độ cho bốn động cơ	38

4.3.3.	<i>Thiết lập lô trình</i>	41
4.4.	Xây dựng code chương trình	42
CHƯƠNG 5: MÔ HÌNH THỰC TẾ		44
5.1.	Giới thiệu mô hình	44
5.2.	Vận hành mô hình	46
5.3.	Kiểm nghiệm mô hình	48
5.3.1.	<i>Kiểm nghiệm đáp ứng của bộ PID</i>	48
5.3.2.	<i>Kiểm nghiệm hoạt động của mô hình</i>	49
5.4.	Nhược điểm	51
CHƯƠNG 6: KẾT LUẬN		52
Phụ lục A		53
Phụ lục B		54
Phụ lục C		55
DANH MỤC TÀI LIỆU THAM KHẢO		56

DANH MỤC HÌNH ẢNH

Hình 2.1: Sơ đồ về điều khiển tự động	16
Hình 2.2: Sơ đồ cấu trúc nguyên tắc điều khiển dựa vào sai lệch	17
Hình 2.3: Sơ đồ điều khiển theo nguyên tắc bù nhiễu	18
Hình 2.4: Sơ đồ điều khiển hỗn hợp	18
Hình 2.5: Sơ đồ cấu trúc phân cấp hình cây	19
Hình 2.6: Sơ đồ bộ điều khiển PID	20
Hình 2.7: Đáp ứng của khâu P	22
Hình 2.8: Đáp ứng của khâu D và bộ PD	23
Hình 2.9: Sự khác biệt giữa khâu I và PI	24
Hình 2.10: Sự đáp ứng của bộ PID	24
Hình 3.1: Arduinio Mega 2560	25
Hình 3.2: Mạch cầu H tổng quát	26
Hình 3.3: sơ đồ tổng quát của một mạch cầu H sử dụng transistor BJT	27
Hình 3.4a: Nguyên lý hoạt động của cầu H	27
Hình 3.4b: Nguyên lý hoạt động của cầu H	28
Hình 3.5: Cầu H L298N và sơ đồ chân	29
Hình 3.6: Động cơ DC Servo giảm tốc GA25V1 thực tế	30
Hình 3.7: Sơ đồ nối dây động cơ GA25V1	31
Hình 3.8: Sơ đồ mạch điện cảm biến hồng ngoại	31
Hình 3.9: Cảm biến hồng ngoại E18	32
Hình 3.10: Cảm biến Hall được gắn trên động cơ DC Servo GA25V1	33
Hình 3.11: LCD TFT 2.4 inchs thực tế	34
Hình 3.12: Sơ đồ chân LCD TFT	35
Hình 4.1: Bản vẽ mạch điện ra chân	37
Hình 4.2: Mạch ra chân thực tế	37
Hình 4.3: Lưu đồ giải thuật	38
Hình 4.4: Phần mềm Arduinio IDE	42
Hình 4.5: Chọn board và cổng com tương ứng	42
Hình 4.6: Upload chương trình	43
Hình 4.7: Cửa sổ Serial Monitor	43

Hình 5.1: Mặt trên của mô hình	44
Hình 5.2: Mặt bên của mô hình	44
Hình 5.3: Mặt trước của mô hình.....	45
Hình 5.4: Mặt dưới của mô hình.....	46
Hình 5.5: Màn hình khởi động.....	46
Hình 5.6: Bước 1	47
Hình 5.7: Bước 2	47
Hình 5.8: Bàn phím cảm ứng	47
Hình 5.9: Hiển thị thông tin sau khi nút Start được nhấn	48
Hình 5.10: Kiểm nghiệm bộ PID	48
Hình 5.11: Kiểm nghiệm chạy không vật cản	49
Hình 5.12a: Kiểm nghiệm chạy có vật cản.....	50
Hình 5.12b: Kiểm nghiệm chạy có vật cản.....	50
Hình 5.12c: Kiểm nghiệm chạy có vật cản.....	50
Hình 5.12d: Kiểm nghiệm chạy có vật cản.....	51

Danh mục các bảng

Bảng 2.1: Sơ đồ bánh xe robot tự hành.....	13
Bảng 2.2: Kí hiệu các loại bánh xe	16
Bảng 2.3: Ảnh hưởng của việc tăng từng thông số bô PID	21
Bảng 3.1: Các cổng Serial Arduunio Mega 2560	25
Bảng 3.2: Chức năng các chân LCD TFT.....	34

CHƯƠNG 1: TỔNG QUAN

1.1. Lý do chọn đề tài

Ngày nay với sự phát triển của khoa học kỹ thuật, máy móc ngày càng được sử dụng thay thế con người trong các công việc mang tính chất nặng nhọc, cần độ chính xác cao và để tăng năng suất lao động. Việc sử dụng máy móc thay thế con người mang lại hiệu suất công việc cao hơn và đây là xu hướng tất yếu trong tương lai.

AGV là một loại robot được nước ngoài sử dụng vào các ngành công nghiệp dùng để chuyên chở tự động. Tuy nhiên ở nước ta thì công nghệ này vẫn chưa được sử dụng nhiều trong thực tế. Hiện nay, hầu hết các ngành công nghiệp ở Việt Nam đều có quy mô lớn nhưng chất lượng chưa cao do những hạn chế về khoa học kỹ thuật, máy móc thiết bị còn thô sơ nên năng suất lao động chỉ ở mức trung bình lại phải thuê mướn một lượng lớn công nhân. Điều này khiến cho nền kinh tế nước ta chậm phát triển.

Nhóm em chọn đề tài “Mô hình hóa xe tự hành vận chuyển hàng hóa trong nhà xưởng” với mục tiêu nghiên cứu và tạo ra sản phẩm mang tính ứng dụng vào thực tế. Có thể thay thế con người thực hiện các công việc nặng nhọc, làm việc ở những khu vực độc hại. Nâng cao năng suất lao động, giảm thời gian vận chuyển để tăng lợi nhuận. Ngoài ra còn để tiếp cận với sự phát triển của xã hội.

1.2. Nhiệm vụ của đề tài

Mục tiêu của đề tài là tạo ra mô hình xe tự hành vận chuyển hàng trong nhà xưởng có khả năng:

- Tự vận hành theo tọa độ các điểm dừng đã cài đặt từ trước.
- Sử dụng PID để điều khiển chính xác vận tốc động cơ.
- Nhận biết được vật cản thông qua các cảm biến qua đó có thể tự tránh vật cản cũng như tìm đường đi đến đích.
- Tọa độ vị trí điểm dừng và tốc độ được cài đặt từ bàn phím.

1.3. Xe tự hành trong công nghiệp

Theo xu hướng gia tăng sự tự động hóa trong các lĩnh vực sản xuất, các hệ thống vận chuyển bên trong các nhà máy dưới dạng “hệ thống máy dẫn đường tự động” (AGV) ngày càng chiếm một vị trí quan trọng trong toàn bộ lĩnh vực xử lý vật liệu. AGV – Automated Guided Vehicle được sử dụng để thực hiện các nhiệm vụ:

- Cung cấp và chuyên chở ở các khu sản xuất và kho.

- Cung cấp và chuyên chở trong các khu vực đặc biệt như ở bệnh viện và di chuyển tài liệu trong các văn phòng.
- Vận chuyển ở các khu vực độc hại thay thế cho con người.

Trong tất cả các ứng dụng trên, người ta nhận thấy rằng AGV làm cho quá trình sản xuất linh hoạt hơn, giảm thiểu nhu cầu nhân lực, giảm thiểu thiệt hại trong kiểm kê.

1.3.1. Lịch sử phát triển của AGV(Automotive Guide Vehicle)

Hệ thống xe dẫn hướng tự động (AGVS) đã tồn tại từ năm 1953 bởi Barrett Electronics of Northbrook, bang Illinois-USA, nay là Savant Automation of Walker, bang Michigan-USA. Một nhà phát minh với giấc mơ sáng chế ra một phương pháp tự động hóa con người trên chiếc xe tải kéo mà đã được sử dụng trong các nhà máy trong nhiều năm. Lúc đầu chỉ là một chiếc xe kéo nhỏ chạy theo một đường dẫn. Hệ thống hướng dẫn đầu tiên được tạo ra khi xuất hiện các cảm biến dò theo một từ trường. AGV chỉ tồn tại ở mức này cho đến những năm 70s. Công nghệ lúc này đã điều khiển được các hệ thống để mở rộng khả năng và tính linh hoạt. Xe không chỉ còn để kéo rơ mooc trong kho, mà còn được sử dụng trong quá trình sản xuất, làm việc và các hệ thống lắp ráp ô tô.

Qua nhiều năm khi công nghệ trở nên tinh vi hơn, thì ngày nay các AGV chủ yếu được định vị bởi các hệ thống Laser LGV (Laser Guided Vehicle). Trong quá trình tự hành LGV được lập trình với các robot khác nhằm đảm bảo sản phẩm được chuyển qua các trạm, kho nơi mà chúng được giữ lại hoặc chuyển đến một vị trí khác. Ngày nay LGV đóng vai trò quan trọng trong thiết kế các nhà máy và nhà kho, đưa hàng hóa đến đúng địa điểm một cách an toàn.

1.3.2. Phân loại AGV

1.3.2.1. Phân loại theo chức năng

Xe kéo (Towing Vehicle)

Xe kéo xuất hiện đầu tiên và bây giờ vẫn còn thịnh hành. Loại này có thể kéo được nhiều toa hàng khác nhau và có thể kéo được 8000 đến 60000 pounds.

Ưu điểm của hệ thống xe kéo:

- Khả năng chuyên chở lớn.
- Có khả năng dự đoán lên kế hoạch về tính hiệu quả của việc chuyên chở cũng như đảm bảo an toàn.
- Tăng tính an toàn.

Mô hình xe kéo (Unit Load Vehicle)

Xe chở được trang bị các tầng khay chứa có thể là các nâng hạ chuyên động bằng băng tải, đai hoặc xích. Ưu điểm:

- Tải trọng được phân phôi và di chuyển theo yêu cầu.
- Thời gian đáp ứng nhanh gọn.
- Giảm hư hại.
- Đường đi linh hoạt.
- Giảm thiểu các tắc nghẽn giao thông chuyên chở.
- Lập kế hoạch hiệu quả.

Xe dây (Cart Vehicle)

Là xe có tính linh hoạt cao và rẻ tiền. Được sử dụng để chuyên chở vật liệu và các hệ thống lắp ráp.

Xe nâng (Fork Vehicle)

Có khả năng nâng các tải trọng đặt trên sàn hoặc trên các bục cao hay các khối hàng đặt trên giá.

1.3.2.2. Phân loại theo dạng đường đi

Loại không chạy theo đường dẫn (Free path navigation)

Có thể di chuyển đến vị trí bất kỳ trong không gian hoạt động. Đây là loại xe AGV có tính linh hoạt cao được định vị vị trí nhờ các cảm biến con quay hồi chuyển (Gyroscope sensor) để xác định hướng di chuyển, cảm biến laser để xác định vị trí các vật thể xung quanh trong quá trình di chuyển, hệ thống định vị cục bộ (Local Navigation Location) để xác định tọa độ tức thời.

Loại chạy theo đường dẫn (Fixed path navigation)

Đường dẫn là từ: Là loại đường dẫn có cấu tạo là dây từ chôn ngầm dưới nền sàn. Khi di chuyển, nhờ có các cảm biến cảm ứng từ mà xe có thể di chuyển theo đường dẫn. Phải tiêu tốn năng lượng cho việc tạo từ tính cho dây, đồng thời đường dẫn là cố định và không thay đổi được.

Đường ray dẫn: Chạy trên các đường ray định trước trên mặt sàn. Loại này chỉ sử dụng cho hệ thống chuyên dụng. Cho phép thiết kế đơn giản và có tốc độ cao nhưng tính linh hoạt thấp.

Đường băng kẻ trên sàn: AGV di chuyển theo các đường băng kẻ sẵn trên sàn nhờ các cảm biến nhận dạng vạch kẻ. Loại này có tính linh hoạt cao vì trong quá trình sử dụng người ta có thể thay đổi đường đi một cách dễ dàng nhờ kẻ lại các vạch dẫn. Tuy nhiên các vạch dẫn có thể bị bẩn hoặc hư hại gây khó khăn trong việc điều khiển chính xác xe.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Lý thuyết về xe tự hành

Xe tự hành hay robot di động (mobile robots, thường được gọi tắt là mobots) được định nghĩa là một loại xe robot có khả năng tự dịch chuyển, tự vận động (có thể lập trình lại được) dưới sự điều khiển tự động để thực hiện thành công công việc được giao.

Với những cảm biến, chúng có thể nhận biết môi trường xung quanh. Robot tự hành ngày càng có nhiều ý nghĩa trong các ngành công nghiệp, thương mại, y tế, các ứng dụng khoa học và phục vụ đời sống con người.

Theo lý thuyết, môi trường hoạt động của robot tự hành có thể là đất, nước, không khí, không gian vũ trụ hay sự tổ hợp giữa chúng. Địa hình bề mặt mà robot di chuyển trên đó có thể bằng phẳng hoặc thay đổi, lồi lõm.

Vấn đề của robot tự hành là làm sao để nó hoạt động, nhận biết môi trường xung quanh và thực thi các nhiệm vụ được đề ra. Vấn đề đầu tiên là di chuyển, robot tự hành nên di chuyển thế nào và cơ cấu di chuyển nào là sự lựa chọn tốt nhất. Điều hướng là vấn đề cơ bản trong nghiên cứu và chế tạo robot tự hành. Trong hiệp hội nghiên cứu về robot tự hành có hai hướng nghiên cứu khác nhau:

- Hướng thứ nhất là nghiên cứu về robot tự hành có khả năng điều hướng ở tốc độ cao nhờ thông tin thu được nhờ cảm biến, đây là loại robot có khả năng hoạt động ở môi trường trong phòng cũng như môi trường bên ngoài. Loại robot này yêu cầu khả năng tính toán đồ sộ và được trang bị cảm biến có độ nhạy cao, dải đo lớn để có thể điều khiển robot di chuyển ở tốc độ cao, trong những địa hình có môi trường phức tạp.
- Hướng thứ hai nhằm giải quyết các vấn đề các loại robot tự hành chỉ dùng để hoạt động trong môi trường trong phòng. Loại robot tự hành này có kết cấu đơn giản hơn loại trên, thực hiện những nhiệm vụ đơn giản.

Bài toán dẫn hướng cho robot tự hành được chia làm 2 loại: bài toán toàn cục (global) và bài toán cục bộ (local). Ở bài toán toàn cục, môi trường làm việc của robot hoàn toàn xác định, đường đi và vật cản là hoàn toàn biết trước. Ở bài toán cục bộ, môi trường hoạt động của robot là chưa biết trước hoặc chỉ biết một phần. Các cảm biến và thiết bị định vị cho phép robot xác định được vật cản, vị trí của nó trong môi trường giúp nó đi tới được mục tiêu. Các vấn đề gặp phải khi điều hướng cho Robot tự hành thường không giống như các loại robot khác. Để có thể điều hướng cho Robot tự hành, quyết định theo

thời gian thực phải dựa vào thông tin liên tục về môi trường thông qua các cảm biến, hoặc ở môi trường trong phòng hoặc ngoài trời, đây là điểm khác biệt lớn nhất so với kỹ thuật lập kế hoạch ngoại tuyến. Robot tự hành phải có khả năng tự quyết định về phương thức điều hướng, định hướng chuyên động để có thể tới đích thực hiện nhiệm vụ nhất định. Điều hướng cho robot tự hành là công việc đòi hỏi phải thực hiện được một số khả năng khác nhau, bao gồm: khả năng di chuyển ở mức cơ bản, ví dụ như hoạt động đi tới vị trí cho trước; khả năng phản ứng các sự kiện theo thời gian thực, ví dụ như khi có sự xuất hiện đột ngột của vật cản; khả năng xây dựng, sử dụng và duy trì bản đồ môi trường hoạt động; khả năng xác định vị trí của robot trong bản đồ đó; khả năng thiết lập kế hoạch để đi tới đích hoặc tránh các tình huống không mong muốn và khả năng thích nghi với các thay đổi của môi trường hoạt động.

Phân loại robot tự hành

Robot tự hành được chia làm 2 loại chính đó là loại robot tự hành chuyên động bằng chân và robot tự hành chuyên động bằng bánh. Ngoài ra một số loại robot hoạt động trong các môi trường đặc biệt như dưới nước hay trên không trung thì chúng được trang bị cơ cấu di chuyển đặc trưng.

- Robot tự hành di chuyển bằng chân (Legged Robot).

Ưu điểm lớn nhất của loại robot này là có thể thích nghi và di chuyển trên các địa hình gồ ghề. Hơn nữa chúng còn có thể đi qua những vật cản như hố, vết nứt sâu.

Nhược điểm chính của robot loại này chính là chế tạo quá phức tạp. Chân robot là kết cấu nhiều bậc tự do, đây là nguyên nhân làm tăng trọng lượng của robot đồng thời giảm tốc độ di chuyển. Các kỹ năng như cầm, nắm hay nâng tải cũng là nguyên nhân làm giảm độ cứng vững của robot.

Robot loại này càng linh hoạt thì chi phí chế tạo càng cao. Robot tự hành di chuyển bằng chân được mô phỏng theo các loài động vật vì thế mà chúng có loại 1 chân, loại 2,4,6 chân và có thể nhiều hơn. Dưới đây là một số loại robot điển hình chuyên động bằng chân.

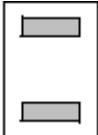
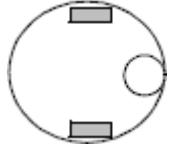
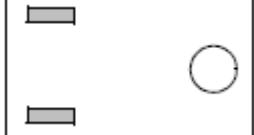
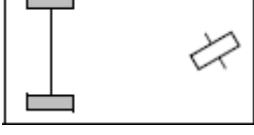
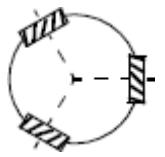
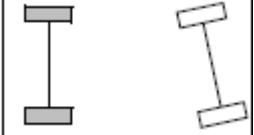
- Robot tự hành di chuyển bằng bánh

Bánh xe là cơ cấu chuyển động được sử dụng rộng rãi nhất trong công nghệ Robot tự hành. Ván đê cân bằng thường không phải là ván đê được chú ý nhiều trong robot di chuyển bằng bánh. Ba bánh là kết cấu có khả năng duy trì cân bằng nhất, tuy nhiên kết

cứu 2 bánh cũng có thể cân bằng được. Khi robot có số bánh nhiều hơn 3 thì thông thường người ta phải thiết kế hệ thống treo để duy trì sự tiếp xúc của tất cả các bánh xe với mặt đất. Vấn đề của robot loại này là về lực kéo, độ ổn định và khả năng điều khiển chuyên động,...

Sơ đồ bánh xe của robot tự hành 2 bánh, 3 bánh, 4 bánh và 6 bánh được liệt kê trong bảng dưới đây:

Bảng 2.1: Sơ đồ bánh xe robot tự hành

Số bánh	Sắp xếp	Miêu tả
		Một bánh lái phía trước, một bánh phía sau
2		Hai bánh truyền động với trọng tâm ở bên dưới trục bánh xe.
		Hai bánh truyền động ở giữa và có điểm thứ 3 tiếp xúc
		Hai bánh truyền động độc lập ở phía sau và một bánh lái ở phía trước.
		Hai bánh truyền động được nối với trục ở phía sau, một bánh lái ở phía trước.
3		Hai bánh quay tự do ở phía sau, bánh trước vừa là bánh truyền động vừa là bánh lái.
		3 bánh Swedisk được đặt ở các đỉnh của một tam giác đều, kết cấu này cho phép robot di chuyển theo đa hướng.
		2 bánh chủ động ở phía sau, hai bánh lái ở phía trước.

4		Hai bánh phía trước vừa là bánh lái vừa là bánh chủ động.
		Cả 4 bánh đều là bánh truyền động và lái.
		Hai bánh truyền động độc lập ở phía trước/sau, 2 bánh lái đa hướng ở phía sau/trước.
		Bốn bánh đa hướng.
		Hai bánh chuyển động vi sai và thêm 2 điểm tiếp xúc.
		4 bánh vừa là truyền động vừa là bánh lái.
6		Hai bánh truyền động ở giữa, thêm 4 bánh đa hướng ở xung quanh.
		Hai bánh truyền động vi sai ở giữa, bốn bánh đa hướng ở 4 góc.

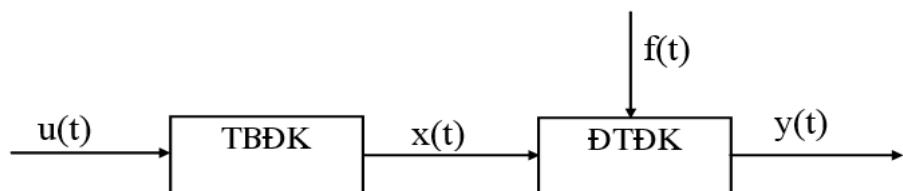
Kí hiệu của các bánh xe được thể hiện dưới bảng sau

Bảng 2.2: Kí hiệu các loại bánh xe

Kí hiệu các loại bánh xe	
	Bánh đa hướng không truyền động.
	Bánh truyền động Swedish (đa hướng).
	Bánh quay tự do tiêu chuẩn.
	Bánh truyền động tiêu chuẩn.
	Bánh vừa truyền động vừa là bánh lái.
	Bánh lái tiêu chuẩn.
	Các bánh xe được nối với nhau.

2.2. Lý thuyết điều khiển

Một hệ thống điều khiển là liên kết của nhiều thành phần tạo nên một cấu hình hệ thống nhằm tạo ra một đáp ứng mong muốn. Một cách tổng quát hệ thống điều khiển tự động được mô tả bởi sơ đồ khôi sau:



Hình 2.1: Sơ đồ về điều khiển tự động

Trong đó:

TBĐK: Thiết bị điều khiển, có nhiệm vụ tác động lên đối tượng điều khiển theo một qui luật nào đó để thỏa mãn yêu cầu công nghệ.

ĐTĐK: Đối tượng cần điều khiển (cơ cấu chấp hành), là tập hợp những phương tiện kỹ thuật như máy móc, thiết bị, khí cụ... chịu những tác động nào đó để đạt được mục đích điều khiển đề ra.

$u(t)$: Tín hiệu vào.

$y(t)$: Tín hiệu ra.

$x(t)$: Tín hiệu điều khiển tác động lên đối tượng.

$f(t)$: Tín hiệu nhiễu loạn tác động vào hệ thống.

Những nguyên tắc điều khiển cơ bản

Nguyên tắc thông tin phản hồi

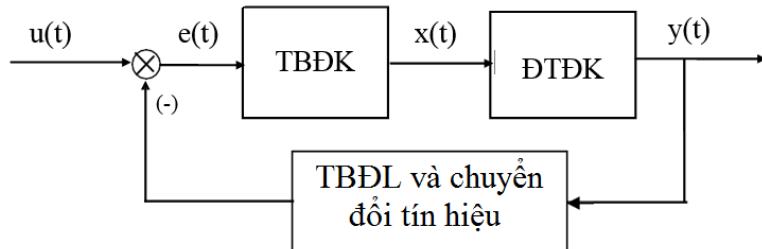
Muốn quá trình điều khiển đạt chất lượng cao, trong hệ thống phải tồn tại hai dòng thông tin: 1 từ bộ điều khiển đến đối tượng và 1 từ đối tượng ngược về bộ điều khiển (dòng thông tin ngược gọi là hồi tiếp). Điều khiển không hồi tiếp (điều khiển vòng hở) không thể đạt chất lượng cao, nhất là khi có nhiễu.

Nguyên tắc điều khiển theo sai lệch

Là nguyên tắc mà tín hiệu điều khiển $x(t)$ được thành lập dựa trên sự sai lệch của lượng ra thực tế so với yêu cầu (đặt ở đầu vào).

$$x(t) = f[y(t) - u(t)] = f[e(t)]$$

Sơ đồ cấu trúc như sau:



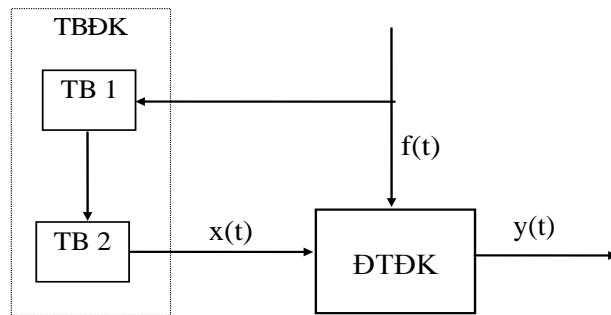
Hình 2.2: Sơ đồ cấu trúc nguyên tắc điều khiển dựa vào sai lệch

Tín hiệu đầu ra $y(t)$ được so sánh với tín hiệu đầu vào $u(t)$ nhằm tạo tín hiệu tác động lên thiết bị điều khiển (TBĐK) sau đó tạo tín hiệu tác động lên đối tượng điều khiển (ĐTĐK).

Điều khiển theo nguyên tắc bù nhiễu

Là nguyên tắc mà tín hiệu điều khiển $x(t)$ được thành lập dựa trên đo tín hiệu nhiễu và tạo hàm điều khiển để khử nhiễu ở đầu ra. $x(t) = f[f(t)]$

Những hệ thống được xây dựng theo nguyên tắc này là những hệ thống hở (không có phản hồi). Sơ đồ cấu trúc như sau:



Hình 2.3: Sơ đồ điều khiển theo nguyên tắc bù nhiễu

Trong đó:

TB 1 là thiết bị để đo nhiễu.

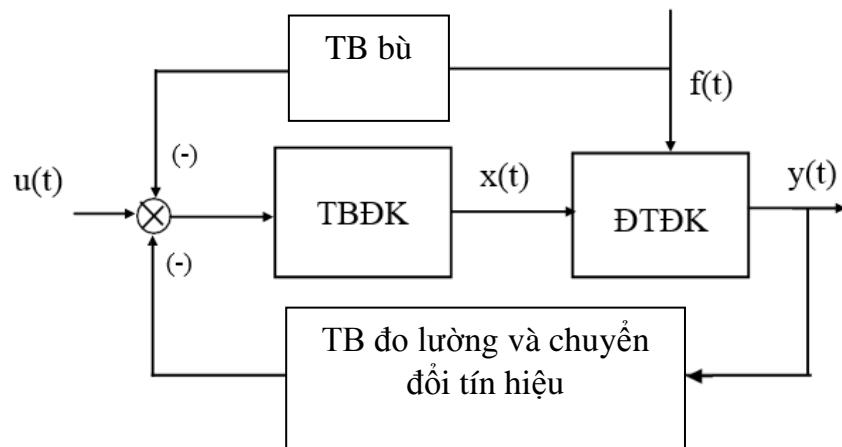
TB 2 là thiết bị để tạo ra tín hiệu điều khiển $x(t)$.

Điều khiển hỗn hợp theo sai lệch và bù nhiễu

Là nguyên tắc mà tín hiệu điều khiển $x(t)$ được thành lập dựa vào sự tổng hợp của hai phương pháp trên.

$$x(t) = f[e(t), f(t)]$$

Sơ đồ cấu trúc tổng quát như sau:



Hình 2.4: Sơ đồ điều khiển hỗn hợp

Nguyên tắc đa dạng tương xứng

Muốn quá trình điều khiển có chất lượng thì sự đa dạng của bộ điều khiển phải tương xứng với sự đa dạng của đối tượng. Tính đa dạng của bộ điều khiển thể hiện ở khả năng thu thập thông tin, lưu trữ thông tin, truyền tin, phân tích xử lý, chọn quyết định,...

Ý nghĩa: Cần thiết kế bộ điều khiển phù hợp với đối tượng.

Nguyên tắc bổ sung ngoài

Một hệ thống luôn tồn tại và hoạt động môi trường cụ thể và có tác động qua lại chặt chẽ với môi trường đó. Nguyên tắc bổ sung ngoài thừa nhận có một đối tượng chưa biết (hộp đen) tác động vào hệ thống và ta phải điều khiển cả hệ thống lẫn hộp đen.

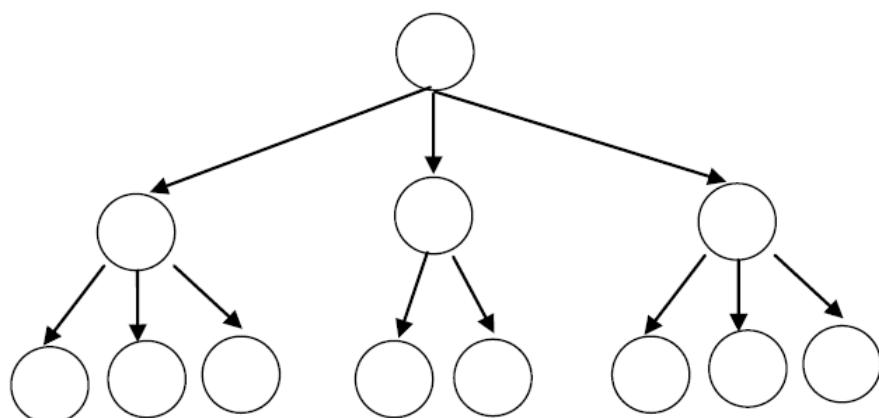
Ý nghĩa: Khi thiết kế hệ thống tự động, muốn hệ thống có chất lượng cao thì không thể bỏ qua nhiễu.

Nguyên tắc dự trữ

Vì nguyên tắc 3 luôn coi thông tin chưa đầy đủ phải đề phòng các bất trắc xảy ra và không được dùng toàn bộ lực lượng trong điều kiện bình thường. Vốn dự trữ không sử dụng, nhưng cần để đảm bảo cho hệ thống vận hành an toàn.

Nguyên tắc phân cấp

Đối với một hệ thống điều khiển phức tạp cần xây dựng nhiều lớp điều khiển bổ sung cho trung tâm. Cấu trúc phân cấp thường sử dụng là cấu trúc hình cây:



Hình 2.5: Sơ đồ cấu trúc phân cấp hình cây

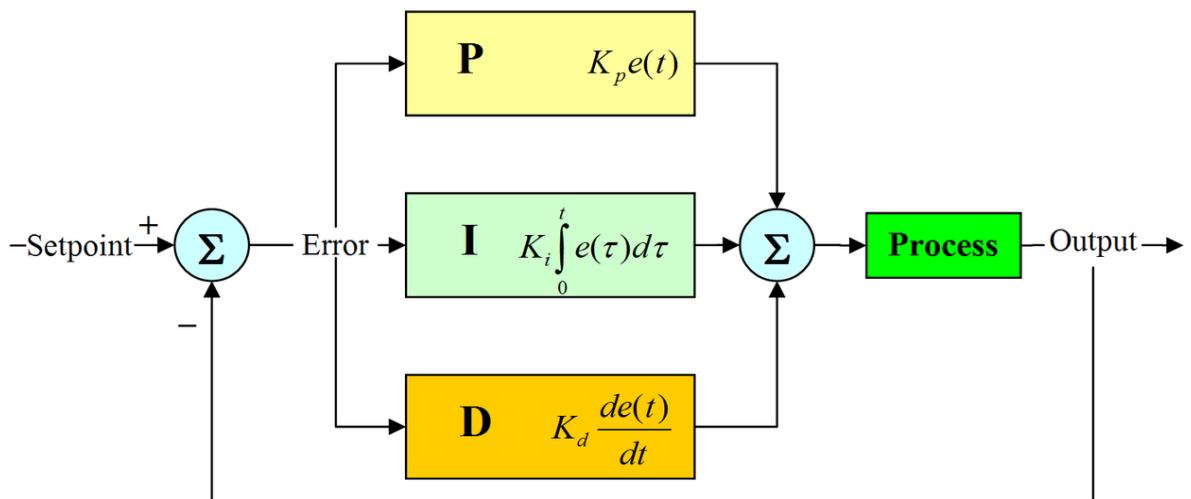
Nguyên tắc nội cân bằng

Mỗi hệ thống cần xây dựng cơ chế cân bằng nội để có khả năng tự giải quyết những biến động xảy ra.

2.3. Bộ điều khiển PID (Proportional – Integral – Derivative)

PID là thuật toán được sử dụng phổ biến trong lĩnh vực điện, điện tử. Dùng để điều khiển để động cơ tự động và giúp đạt được giá trị chuẩn mong muốn, có độ lỗi nhỏ nhất. PID là cách viết tắt của các từ Propotional (tỉ lệ), Integral (tích phân), Derivative (đạo hàm) và là giải thuật điều khiển được dùng nhiều nhất trong các ứng dụng điều khiển tự động với yêu cầu chính xác (accurate), nhanh (fast response), ổn định (small overshoot).

Sử dụng bộ điều khiển PID để điều chỉnh sai lệch giữa giá trị đo được của hệ thống (process variable) với giá trị đặt (setpoint). Bằng cách tính toán và điều chỉnh giá trị điều khiển ở ngõ ra.



Hình 2.6: Sơ đồ bộ điều khiển PID

Ảnh hưởng của việc tăng từng thông số của bộ điều khiển K_P , K_D , K_I được tóm tắt theo bảng sau:

Bảng 2.3: Ảnh hưởng của việc tăng từng thông số bộ PID

Tác động	Rise Time	Overshoot	Settling Time	S-S Error
K_P	Giảm	Tăng	Không xác định	Giảm
K_D	Giảm	Tăng	Tăng	Loại bỏ
K_I	Không xác định	Giảm	Giảm	Không xác định

Trong đó:

Rise Time: Thời gian để đầu ra vượt qua 90% mức mong muốn lần đầu tiên.

Overshoot: Độ vọt lố qua giá trị ổn định mong muốn.

Settling Time: Thời gian để hệ thống đạt được trạng thái ổn định.

S-S Error: Sự khác biệt giữa thông số đầu ra mong muốn và thông số đầu ra ổn định.

Các khâu của một hệ thống PID

- *Khâu P*

Khâu P tạo ra tín hiệu điều khiển tỉ lệ với giá trị của sai lệch. Việc này được thực hiện bằng cách nhân sai lệch e với hằng số KP – gọi là hằng số tỉ lệ.

Khâu P được tính dựa trên công thức:

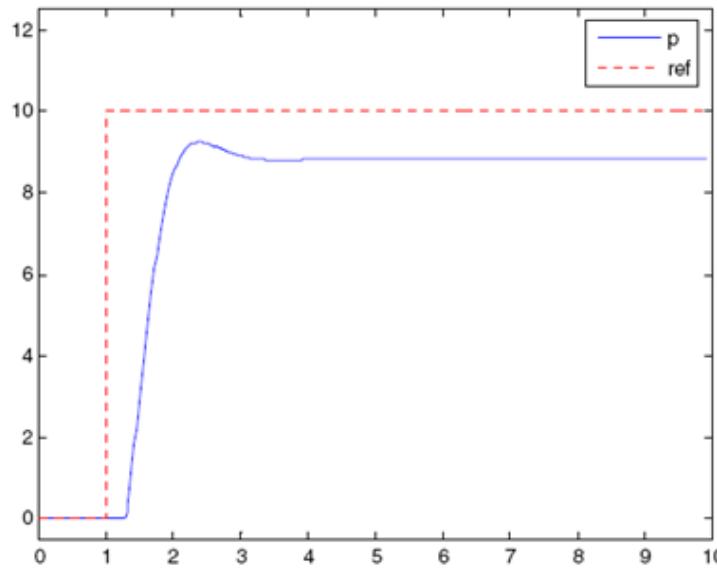
$$P_{\text{out}} = K_p e(t)$$

Với: Pout: giá trị ngõ ra

KP: hằng số tỉ lệ

e: sai lệch: $e = SP - PV$

Nếu chỉ có khâu P thì trong mọi trường hợp sai số tĩnh luôn xuất hiện, trừ khi giá trị đầu vào của hệ thống bằng 0 hoặc đã bằng với giá trị mong muốn. Trong hình sau thể hiện sai số tĩnh xuất hiện khi thay đổi giá trị đặt.



Hình 2.7: Đáp ứng của khâu P

- *Khâu D*

Khâu D cộng thêm tốc độ thay đổi sai số vào giá trị điều khiển ở ngõ ra. Nếu sai số thay đổi nhanh thì sẽ tạo ra thành phần cộng thêm vào giá trị điều khiển. Điều này cải thiện đáp ứng của hệ thống, giúp trạng thái của hệ thống thay đổi nhanh chóng và mau chóng đạt được giá trị mong muốn.

Khâu D được tính theo công thức:

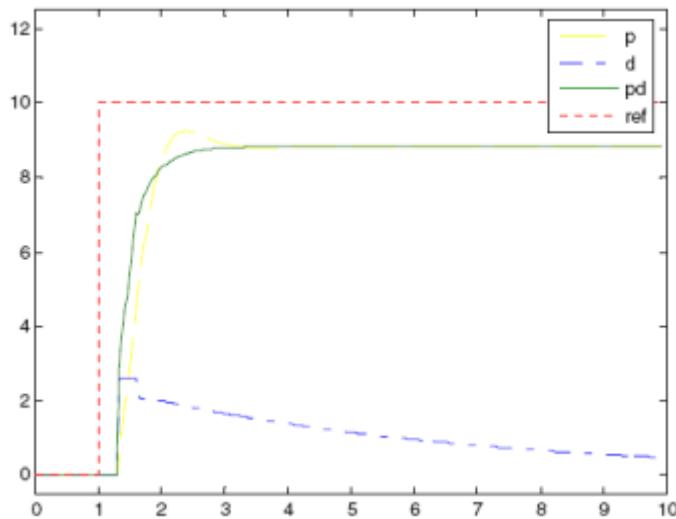
$$D_{\text{out}} = K_d \frac{de}{dt}$$

Với: D_{OUT} : ngõ ra khâu D

K_d : hệ số vi phân

e: sai số: $e = SP - PV$

Khâu D thường đi kèm với khâu P thành bộ PD, hoặc với PI để thành bộ PID.



Hình 2.8: Đáp ứng của khâu D và bộ PD

Theo hình trên, bộ PD tạo đáp ứng có thời gian tăng trưởng nhỏ hơn so với bộ P. Nếu giá trị D quá lớn sẽ làm cho hệ thống không ổn định.

- *Khâu I*

Khâu I cộng thêm tổng các sai số trước đó vào giá trị điều khiển. Việc tính tổng các sai số được thực hiện liên tục cho đến khi giá trị đạt được bằng với giá trị đặt, và kết quả là khi hệ cân bằng thì sai số bằng 0.

Khâu I được tính theo công thức:

$$I_{\text{out}} = K_i \int_0^t e(\tau) d\tau$$

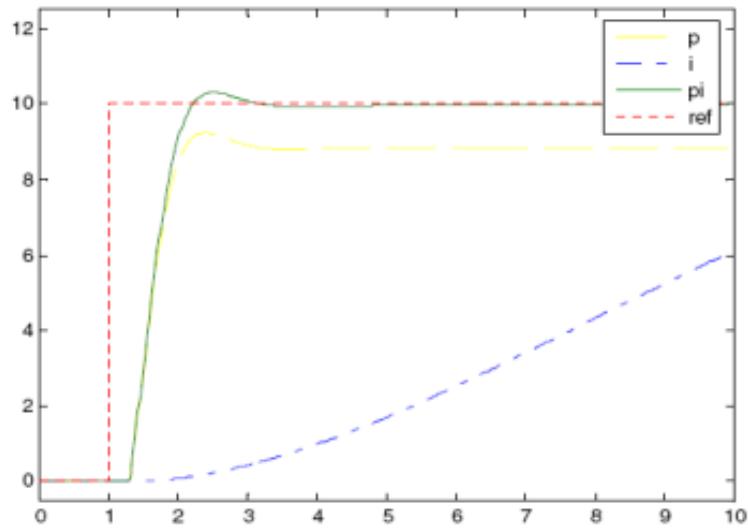
Với: I_{OUT} : giá trị ngõ ra khâu I

K_i : hệ số tích phân

e: sai số: $e = SP - PV$

Khâu I thường đi kèm với khâu P, hợp thành bộ điều khiển PI. Nếu chỉ sử dụng khâu I thì đáp ứng của hệ thống sẽ chậm và thường bị dao động.

Hình sau chỉ ra sự khác biệt giữa khâu I và PI:



Hình 2.9: Sự khác biệt giữa khâu I và PI

Ta có thể nhận thấy là khâu I làm cho đáp ứng của hệ thống bị chậm đi rất nhiều, còn khâu PI giúp triệt tiêu sai số xác lập.

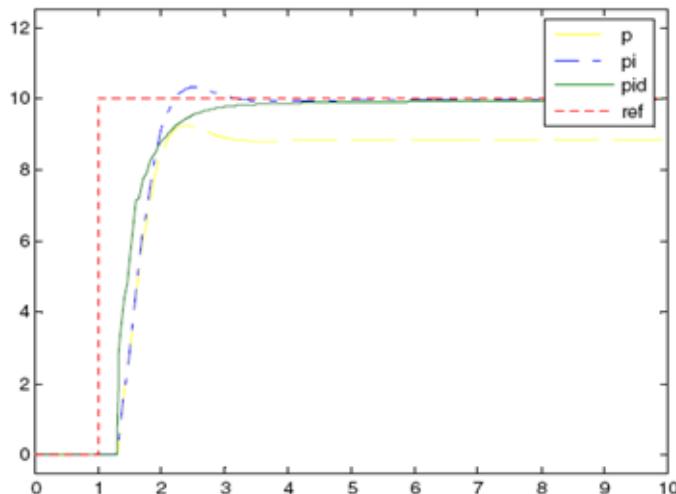
- *Bộ khâu kết hợp*

Bộ điều khiển PID là cấu trúc ghép song song giữa 3 khâu P, I và D.

Phương trình vi phân của bộ PID lý tưởng:

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}$$

Hình bên dưới thể hiện sự đáp ứng của bộ PID:



Hình 2.10: Sự đáp ứng của bộ PID

CHƯƠNG 3: CÁC LINH KIỆN SỬ DỤNG

3.1. Arduinio Mega 2560

Arduino Mega2560 là một vi điều khiển bằng cách sử dụng ATmega2560.

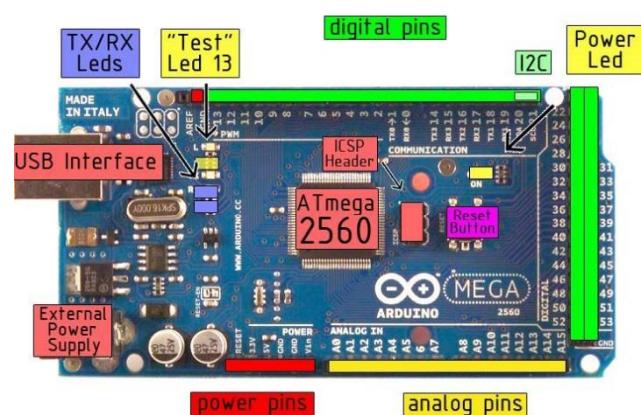
Một Arduinio Mega 2560 bao gồm:

- 54 chân digital trong đó có 15 chân có thể được sử dụng như những chân PWM là từ chân số 2 đến 13 và chân 44 45 46.
- 16 đầu vào analog (từ A0 đến A15).
- 6 ngắt ngoài: chân 2 (interrupt 0), chân 3 (interrupt 1), chân 18 (interrupt 5), chân 19 (interrupt 4), chân 20 (interrupt 3) và chân 21 (interrupt 2).
- 1 thạch anh 16 MHz.
- 1 cổng kết nối USB.
- 1 jack cắm điện.
- 1 đầu ICSP.
- 1 nút reset.
- 4 UARTs (cổng nối tiếp phần cứng).

Bảng 3.1: Các cổng Serial Arduinio Mega 2560

CỔNG SERIAL	CHÂN RX	CHÂN TX
Cổng 0	0	1
Cổng 1	19	18
Cổng 2	17	16
Cổng 3	15	14

Arduino Mega2560 khác với tất cả các vi xử lý trước giờ vì không sử dụng FTDI chip điều khiển chuyển tín hiệu từ USB để xử lý. Thay vào đó, nó sử dụng ATmega16U2 lập trình như là một công cụ chuyển đổi tín hiệu từ USB.



Hình 3.1: Arduinio Mega 2560

Thông số kỹ thuật:

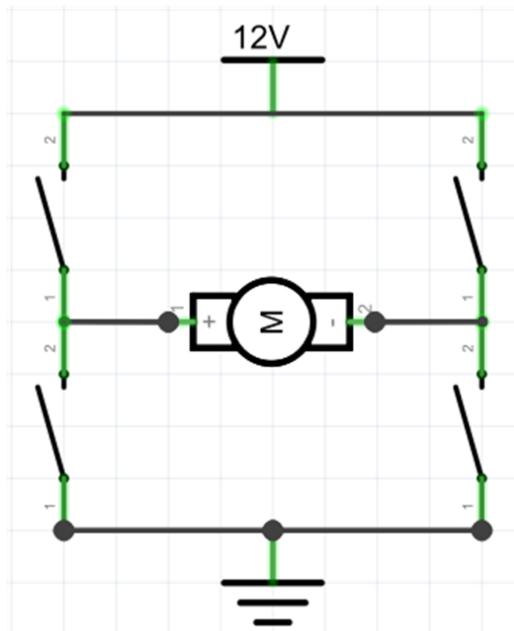
- Ví điều khiển : ATmega2560
- Điện áp hoạt động: 5V
- Điện áp đầu vào (được đề nghị): 7-12V
- Điện áp đầu vào (giới hạn): 6-20V
- Số lượng chân I / O: 54 (trong đó có 15 chân PWM)
- Số lượng chân Input Analog: 16
- Dòng điện DC mỗi I / O: 20 mA
- Dòng điện DC với chân 3.3V: 50 mA
- Bộ nhớ flash: 256 KB trong đó có 8 KB sử dụng bởi bộ nạp khởi động
- SRAM: 8 KB
- EEPROM: 4 KB
- Tốc độ đồng hồ: 16 MHz
- Chiều dài: 101,52 mm
- Bề rộng: 53,3 mm
- Cân nặng: 37 g

3.2. Cầu H

3.2.1. Sơ lược về cầu H

Xét một cách tổng quát, mạch cầu H là một mạch gồm 4 "công tắc" được mắc theo hình chữ H.

Bằng cách điều khiển 4 "công tắc" này đóng mở, ta có thể điều khiển được dòng điện qua động cơ cũng như các thiết bị điện khác tương tự.

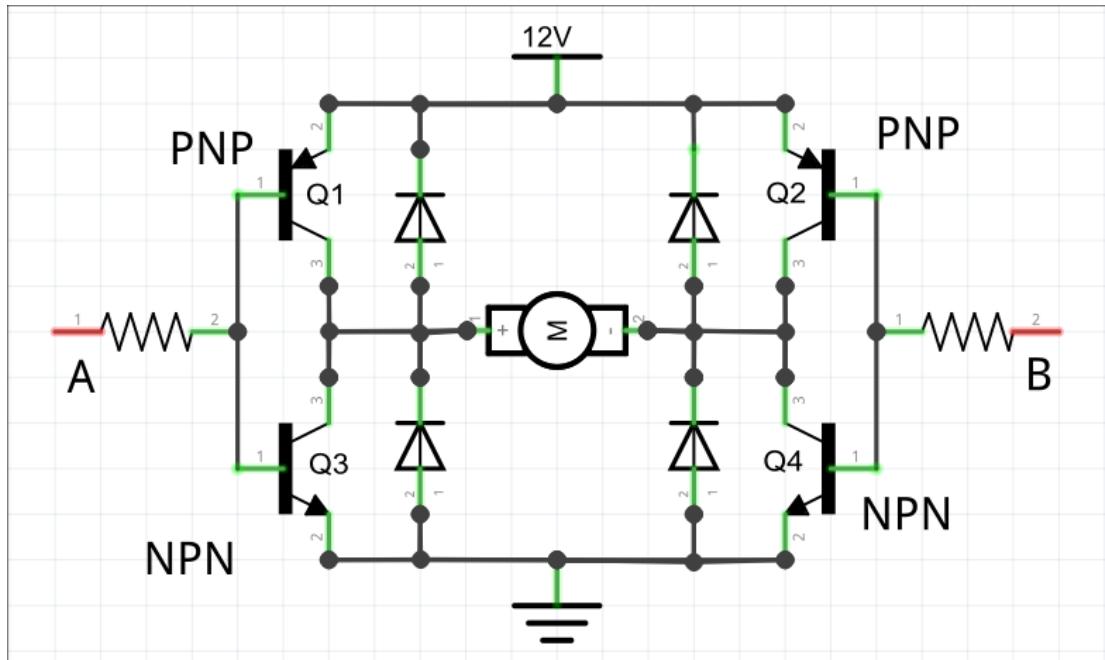


Hình 3.2: Mạch cầu H tổng quát

Nguyên lý hoạt động:

Mạch cầu H dùng transistor BJT là loại mạch được sử dụng khá thông dụng cho việc điều khiển các loại động cơ công suất thấp. Lí do đơn giản là vì transistor BJT thường có công suất thấp hơn các loại MOSFET, đồng thời cũng rẻ và dễ tìm mua, sử dụng đơn giản.

Đây là sơ đồ tổng quát của một mạch cầu H sử dụng transistor BJT.

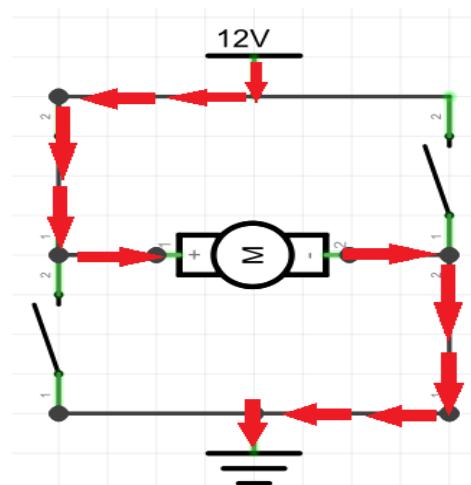


Hình 3.3: Sơ đồ tổng quát của một mạch cầu H sử dụng transistor BJT

Trong sơ đồ này, A và B là 2 cực điều khiển. 4 diode có nhiệm vụ triệt tiêu dòng điện cảm ứng sinh ra trong quá trình động cơ làm việc. Nếu không có diode bảo vệ, dòng điện cảm ứng trong mạch có thể làm hỏng các transistor.

Transistor BJT được sử dụng nên là loại có công suất lớn và hệ số khuếch đại lớn.

Theo như sơ đồ trên, ta có A và B là 2 cực điều khiển được mắc nối tiếp với 2 điện trở hạn dòng. Tùy vào loại transistor bạn đang dùng mà trị số điện trở này khác nhau. Phải đảm bảo rằng dòng điện qua cực Base của các transistor không quá lớn để làm hỏng chúng. Trung bình thì dùng điện trở 1k Ohm.



Hình 3.4a: Nguyên lý hoạt động của cầu H

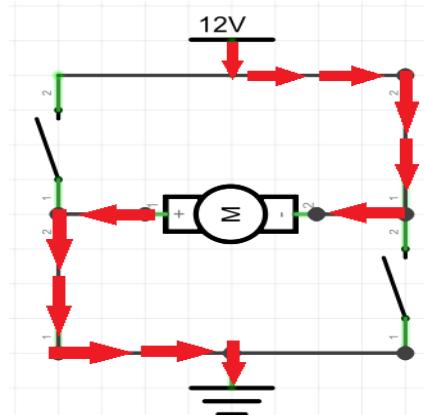
Với 2 cực điều khiển và 2 mức tín hiệu HIGH/LOW tương ứng 12V/0V cho mỗi cực, có 4 trường hợp xảy ra như sau:

A ở mức LOW và B ở mức HIGH

Ở phía A, transistor Q1 mở, Q3 đóng. Ở phía B, transistor Q2 đóng, Q4 mở. Do đó, dòng điện trong mạch có thể chạy từ nguồn 12V đến Q1, qua động cơ đến Q4 để về GND. Lúc này, động cơ quay theo chiều thuận.

A ở mức HIGH và B ở mức LOW

Ở phía A, transistor Q1 đóng, Q3 mở. Ở phía B, transistor Q2 mở, Q4 đóng. Do đó, dòng điện trong mạch có thể chạy từ nguồn 12V đến Q2, qua động cơ đến Q3 để về GND. Lúc này, động cơ quay theo chiều ngược.



Hình 3.4b: Nguyên lý hoạt động của cầu H

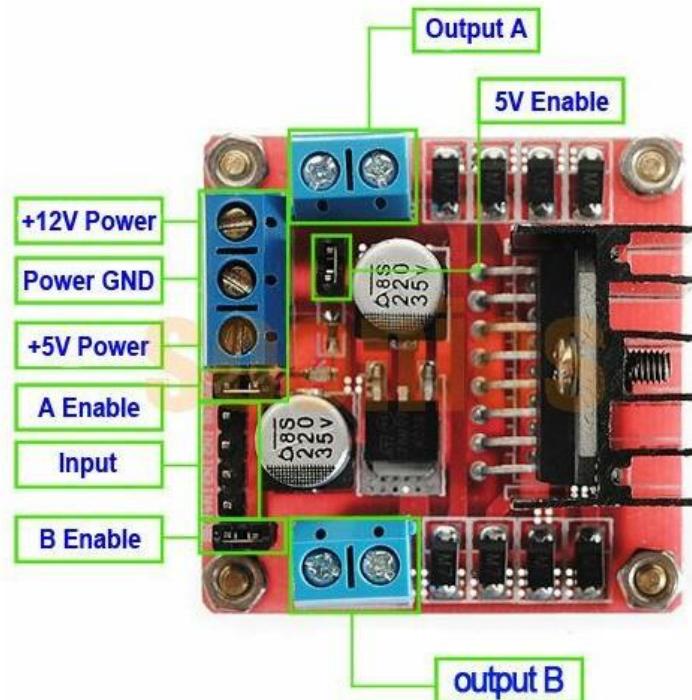
A và B cùng ở mức LOW

Khi đó, transistor Q1 và Q2 mở nhưng Q3 và Q4 đóng. Dòng điện không có đường về được GND do đó không có dòng điện qua động cơ - động cơ không hoạt động.

A và B cùng ở mức HIGH

Khi đó, transistor Q1 và Q2 đóng nhưng Q3 và Q4 mở. Dòng điện không thể chạy từ nguồn 12V ra do đó không có dòng điện qua động cơ - động cơ không hoạt động.

3.2.2. Cầu H L298N



Hình 3.5: Cầu H L298N và sơ đồ chân

Thông số kỹ thuật:

- Driver: L298N tích hợp hai mạch cầu H.
- Điện áp điều khiển: +5 V ~ +12 V.
- Dòng tối đa cho mỗi cầu H là: 2A (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 °C)
- Nhiệt độ bảo quản: -25 °C ~ +130 °C.

Các chân của cầu H L298N:

- 12V power 5V power là 2 chân cấp nguồn trực tiếp đến động cơ.
- Power GND chân này là GND của nguồn cấp cho Động cơ.
- 2 Jump A enable và B enable.
- 4 chân Input. IN1, IN2, IN3, IN4.
- Output A: nối với động cơ A.
- Output B: nối với động cơ B.

3.3. Động cơ DC Servo giảm tốc GA25V1

Động cơ DC Servo giảm tốc GA25 V1 thường được sử dụng trong các ứng dụng cần xác định tốc độ, vị trí, chiều quay của động cơ DC: Robot mè cung, robot xe hai bánh tự cân bằng,...

Động cơ DC Servo thực tế là động cơ DC thường có gắn thêm phần Encoder để có thể trả xung về vị điều khiển giúp xác định vị trí, vận tốc,...

Về cách điều khiển thì động cơ DC Servo sử dụng Driver như động cơ DC thường để điều khiển công suất động cơ, tốc độ và đảo chiều: L298, L293,... chỉ có điểm khác biệt là có thêm phần encoder để có thể hồi tiếp (feedback) xung về vị điều khiển, từ đó vị điều khiển tác động lại động cơ qua mạch công suất sử dụng các thuật toán điều khiển như PID,... để điều khiển tốc độ, vị trí,...



Hình 3.6: Động cơ DC Servo giảm tốc GA25V1 thực tế

Thông số kỹ thuật:

- Điện áp cấp cho động cơ hoạt động: 3 - 12VDC
- Điện áp cấp cho Encoder hoạt động: 3.3VDC
- Đầu Encoder 11 xung, hai kênh A-B.
- Tỷ số truyền khi qua hộp giảm tốc: 1:34
- Số xung khi qua hộp giảm tốc: 374 xung.
- Đường kính động cơ: 25mm.
- Đường kính trục: 4mm
- Tốc độ không tải: 320 rpm

- Dòng không tải: 80mA
- Tốc độ có tải: 284 rpm
- Dòng khi có tải: 600mA
- Mômen: 1.88 kgf.cm
- Công suất định mức: 13.2W
- Dòng khi động cơ bị giữ: 2.29A
- Mô men khi bị giữ: 7.96 kgf.cm

Sơ đồ nối dây:



Hình 3.7: Sơ đồ nối dây động cơ GA25V1

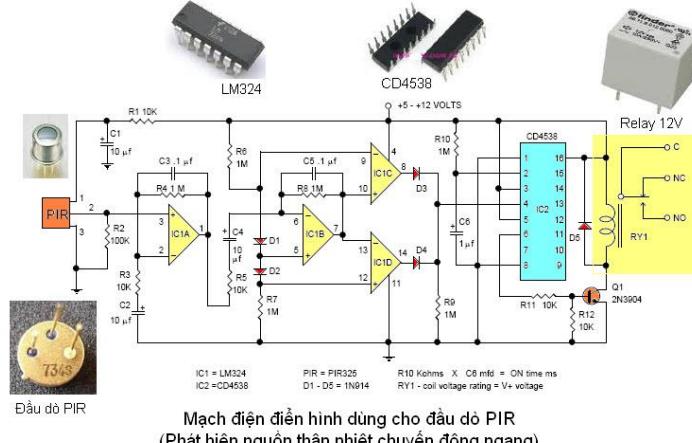
3.4. Cảm biến hồng ngoại

3.4.1. Tổng quát về cảm biến hồng ngoại

Cảm biến hồng ngoại bao gồm phần “cảm biến” và phần “hồng ngoại” (Passive Infrared), viết tắt là PIR.

Cảm biến hồng ngoại là thiết bị dùng để phát hiện tia hồng ngoại và bộ phận cảm biến sẽ phân tích và xác định nguồn phát tia hồng ngoại.

Cảm biến hồng ngoại được sử dụng với nhiều mục đích khác nhau như tìm kiếm mục tiêu tên lửa, phát hiện vật thể, quan sát trong chiến đấu, trong rừng rậm, hay trong an ninh, chống trộm.



Hình 3.8 Sơ đồ mạch điện cảm biến hồng ngoại.

Cách hoạt động của cảm biến:

Tia hồng ngoại phát ra một tần số nhất định, khi phát hiện hướng truyền có vật cản (mặt phản xạ), phản xạ vào đèn thu hồng ngoại, sau khi so sánh, đèn màu xanh sẽ sáng lên, đồng thời đầu cho tín hiệu số đầu ra (một tín hiệu bậc thấp).

3.4.2. Cảm biến hồng ngoại E18-DK80N



Hình 3.9: Cảm biến hồng ngoại E18- D80NK

Thông số kỹ thuật:

Điện áp 5V

Dòng 20mA

Khoảng cách 3-80

Kết nối: Dây nâu 5V DC.

Dây màu xanh dương GND.

Dây màu đen: tín hiệu NPN thường mở.

Nhiệt độ -25°C - 55°C.

3.5. Cảm biến Hall

Hiệu ứng Hall được khám phá bởi Edwin Herbert Hall vào năm 1879 và trở thành một trong các hiệu ứng cơ bản của vật lý học. Hiệu ứng Hall là hiệu ứng xuất hiện trên một vật dẫn có dòng điện chạy qua khi được đặt vào trong một từ trường.

Một hiệu ứng vật lý được thực hiện khi áp dụng một từ trường vuông góc lên một bản làm bằng kim loại hay chất bán dẫn hay chất dẫn điện nói chung (thanh Hall) đang có dòng điện chạy qua. Lúc đó người ta nhận được hiệu điện thế (hiệu thế Hall) sinh ra tại

hai mặt đối diện của thanh Hall. Tỷ số giữa hiệu thế Hall và dòng điện chạy qua thanh Hall gọi là điện trở Hall, đặc trưng cho vật liệu làm nên thanh Hall (một chất bán dẫn).

Hiệu ứng Hall có liên quan đến nhiều ứng dụng cảm biến. Các thiết bị dựa trên mối quan hệ tương đối đơn giản giữa hiện tại, từ trường, và điện áp có thể được sử dụng để đo vị trí, tốc độ và cường độ từ trường.

Mô hình sử dụng cảm biến Hall được gắn trên động cơ như hình sau:



Hình 3.10: Cảm biến Hall được gắn trên động cơ DC Servo GA25V1

Ứng dụng:

- Đo tốc độ động cơ.
- Phát hiện vị trí đối tượng.
- Ứng dụng trong ô tô thông minh.
- Xây dựng các khối điện tử,vv...

Thông số kỹ thuật:

- Cảm biến sử dụng: 49E Hall sensor.
- Điện áp vào: 2,3V- 5,3V.

Nguyên lý hoạt động:

Chip cảm biến 49E Hall là 1 cảm biến phô cập tuyến tính quy mô nhỏ và mức điện áp đầu ra tỉ lệ với cường độ từ trường áp dụng cho đầu nhạy cảm của nó, đối với từ trường bằng 0 thì điện áp đầu ra của nó bằng một nửa điện áp cung cấp cho nó.

Cảm biến sẽ bật khi ta đặt nó lại gần 1 nam châm và tắt nếu để nó xa nam châm, ngoài ra bạn có thể nhìn thấy sự thay đổi ở đầu ra theo từng khoảng cách với nam châm.

Chân module:

1_DOUT: Digital output.

2_AOUT: Analog output.

3_VCC: 2,3V – 5,3V.

4_GND: nối đất.

3.6. LCD TFT 2.4inches

3.6.1. Thông số kỹ thuật

- Điện áp hoạt động: 2.4-3.3V.
- Độ phân giải 240X320 với 262K màu.
- Chuẩn giao tiếp song song với 2 chế độ giao tiếp: 8bit hoặc 16 bit.
- Hỗ trợ Touchscreen và khe cắm thẻ SD.
- IC Driver LCD: ILI9325D, IC Driver Touchscreen: XPT2046.



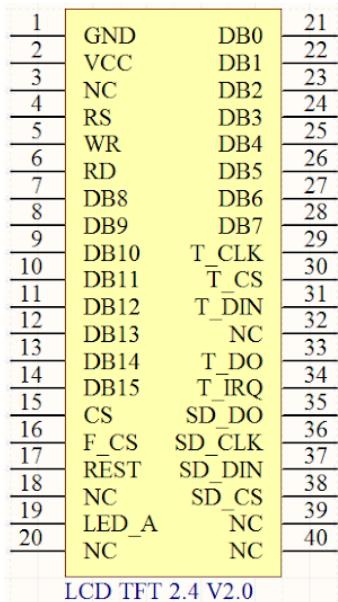
3.6.2. Sơ đồ chân LCD TFT

Hình 3.11: LCD TFT 2.4 inches thực tế

Bảng 3.2: Chức năng các chân LCD TFT

Chân	Tên	Chức năng
1	GND	Chân nối mass (0V).
2	VCC	Chân cấp nguồn (2.4V-3.3V).
4	RS	RS = 0 Ghi lệnh vào LCD. RS = 1 Ghi dữ liệu vào LCD.
5	WR	WR = 0 → 1 Ghi vào LCD.
6	RD	RD = 0 → 1 Đọc dữ liệu từ LCD.
7-14	DB8-DB15	8 bit cao của bus dữ liệu.
15	CS	CS=0: Enable LCD.

16	F_CS	F_CS = 0: Enable bộ nhớ Flash (bộ nhớ ngoài).
17	REST	Reset LCD
19	LED_A	Chân cấp nguồn cho đèn nền LCD (3.3V).
21-28	DB0-DB7	8 bit thấp của bus dữ liệu.
29-31, 33-34	T_CLK, T_CS, T_DIN, T_DO, T_IRQ	Các chân giao tiếp IC Driver Touchscreen theo chuẩn SPI.
35-38	SD_DO, SD_DCLK, SD_DIN, SD_CS	Các chân giao tiếp với thẻ nhớ SD.
3, 18, 20, 39, 40	NC	Không dùng.



Hình 3.12: Sơ đồ chân LCD TFT

CHƯƠNG 4: THIẾT KẾ VÀ THI CÔNG MÔ HÌNH

4.1 Thiết kế cơ khí

4.1.1. *Lựa chọn phương án thiết kế*

Để thực hiện mô hình có hai phương án lựa chọn dựa vào số lượng bánh xe Omni được sử dụng.

- Phương án thứ nhất: thiết kế mô hình có ba bánh xe.
- Phương án thứ hai: thiết kế mô hình có bốn bánh xe.

So sánh việc lắp ba bánh hay bốn bánh xe Omni

	3 bánh	4 bánh
Ưu điểm	3 bánh đồng phẳng mà không cần cơ cấu đặc biệt nào, dễ tương thích với các địa hình gồ ghề. 3 bánh- 3 động cơ- 3 encoder- 3 driver-> rẽ tiền hơn.	Hiệu suất cao do có 2 bánh có thể chạy ở 100% hiệu suất.
Nhược điểm	Hiệu suất không cao do không thể có nhiều hơn một bánh nằm cùng phương với chuyền động.	Cần cơ cấu tự lựa để giữ 4 bánh đồng phẳng, làm cấu trúc robot thêm phức tạp. Đắt tiền hơn 3 bánh.

Nhận thấy rằng phương án thứ hai có ưu điểm cao hơn phương án thứ nhất. Hiệu suất cao hơn so với phương án thứ nhất nên chọn phương án thứ hai để thiết kế và thi công mô hình.

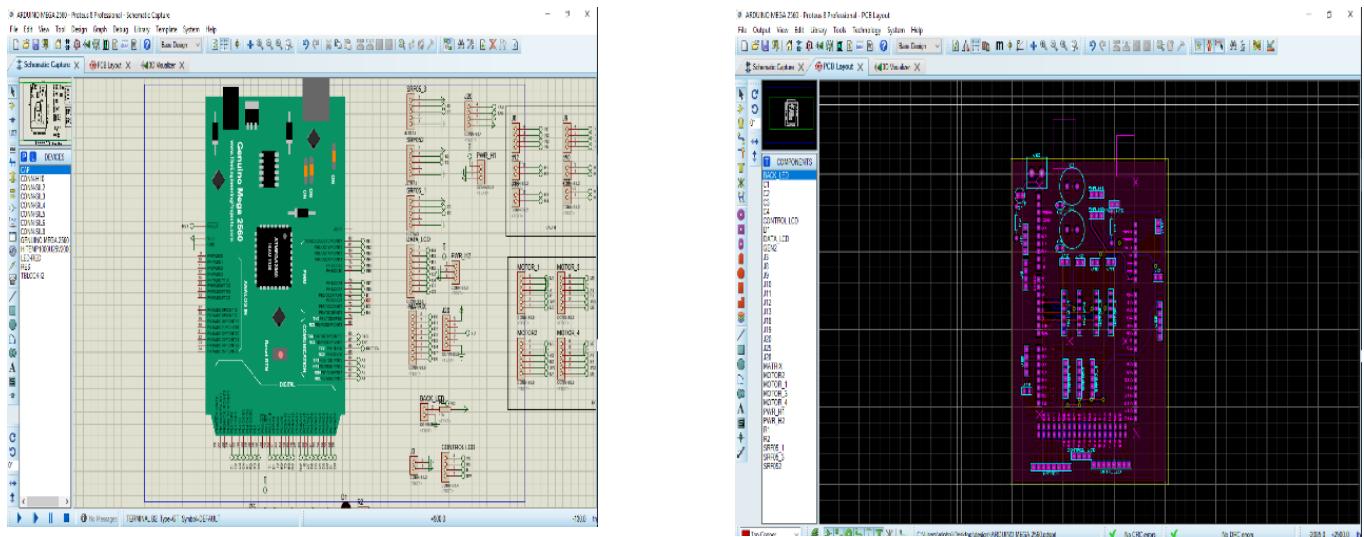
4.1.2. *Lên các bản vẽ*

Phản thiết kế cơ khí được thể hiện trong bản vẽ ở phụ lục B. Mô hình được thiết kế trên phần mềm SolidWorks 2016.

4.2. Thiết kế mạch ra chân

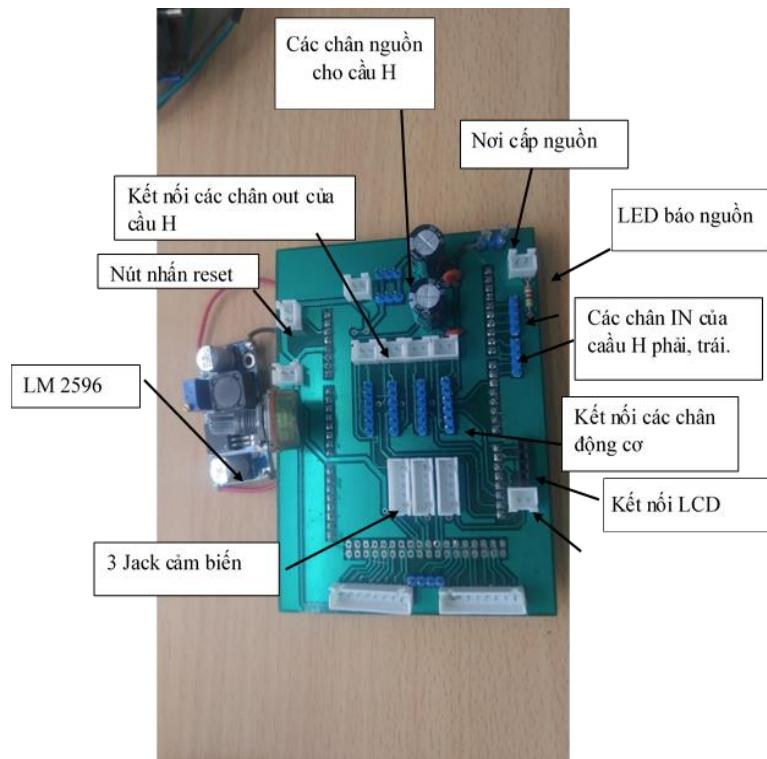
Mạch ra chân được thiết kế dưới dạng Shilde dành cho Arduinio Mega 2560.

Hình bên dưới thể hiện bản vẽ mạch điện.



Hình 4.1: Bản vẽ mạch điện ra chân

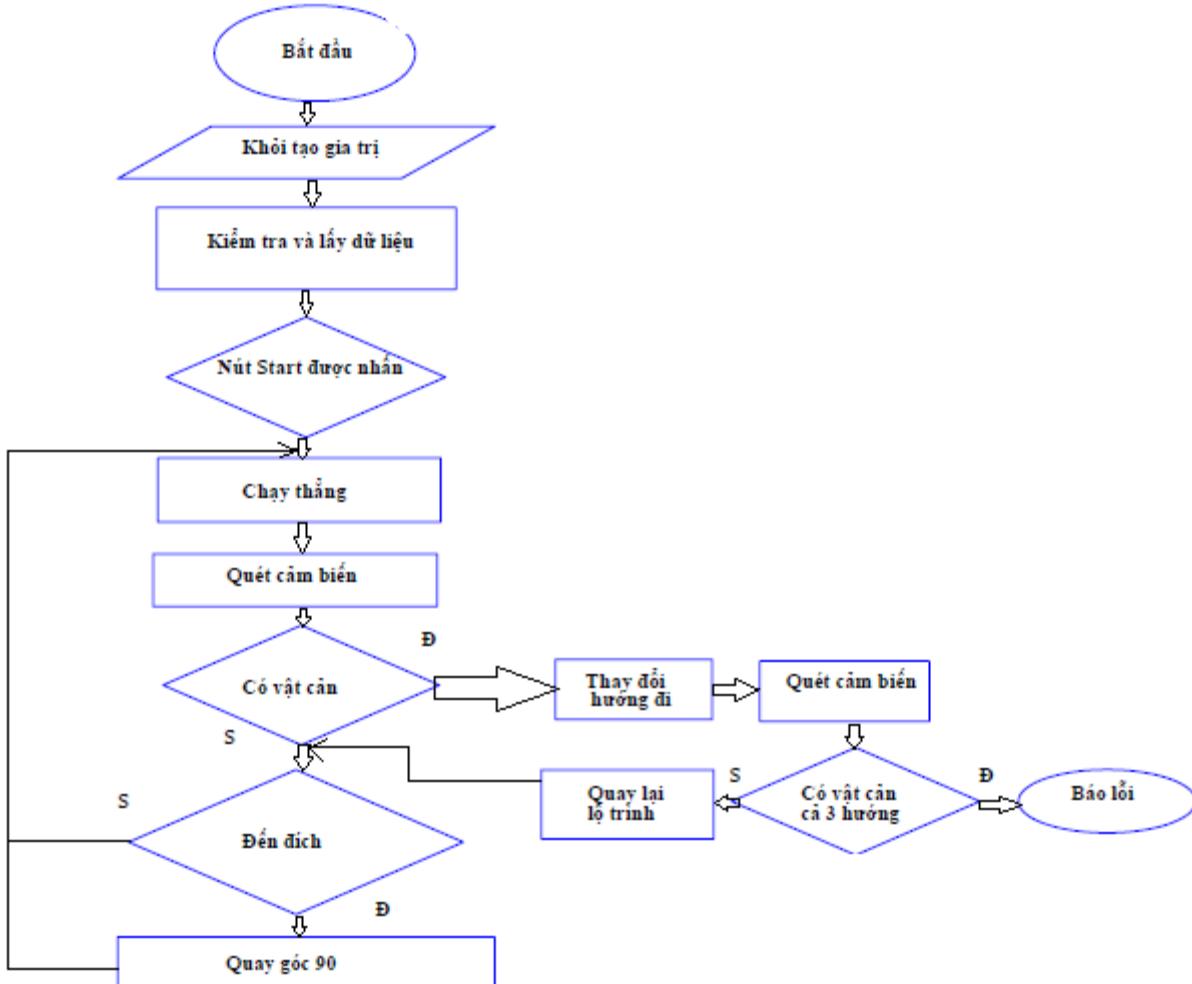
Hình ảnh thực tế bảng mạch ra chân sau khi được gia công



Hình 4.2: Mạch ra chân thực tế

4.3. Xây dựng chương trình điều khiển

4.3.1. Lưu đồ giải thuật



Hình 4.3: Lưu đồ giải thuật

4.3.2. Xây dựng bộ điều khiển PID điều khiển tốc độ cho bốn động cơ

Trong trường hợp không thể xây dựng phương pháp mô hình cho đối tượng thì phương pháp thiết kế thích hợp là phương pháp thực nghiệm. Thực nghiệm chỉ có thể tiến hành nếu hệ thống đảm bảo điều kiện: khi đưa trạng thái làm việc của hệ đến biên giới ổn định thì mọi giá trị của tín hiệu trong hệ thống điều phải nằm trong giới hạn cho phép.

Chỉnh định bằng tay: đặt $K_i = K_d = 0$. Tăng K_p nến khi hệ thống dao động tuần hoàn. Đặt thời gian tích phân bằng chu kỳ dao động. Điều chỉnh lại giá trị K_p cho phù hợp. Nếu có dao động thì điều chỉnh giá trị K_d .

Ta thiết kế bộ điều khiển PID cho một động cơ (các động cơ khác tương tự), gồm các thông số sau:

- T: thời gian lấy mẫu 10ms (chu kỳ đọc Encoder).
- xung: số xung thực tế đọc được từ encoder gắn ở động cơ.
- toodo: tốc độ thực tế của động cơ.
- tocdodat: tốc độ mong muốn.
- E, E1, E2 là các giá trị sai số tương ứng: sai số ban đầu, sai số trước đó và sai số trước đó hai lần.
- $\alpha, \beta, \gamma, K_p, K_d, K_i$: các hệ số của bộ điều khiển PID cho động cơ.
- Output, Lastoutput là các giá trị đầu ra của động cơ.

Chương trình bộ điều khiển PID được thể hiện qua giải thuật:

```
#include <TimerOne.h>
#define interA 18
int tocdodat=200;
float T;
double xung_MA=0;
double toodo_MA;
double E_MA, E1_MA, E2_MA;
double alpha_MA, beta_MA, gama_MA, Kp_MA, Kd_MA, Ki_MA;
double Output_MA, Lastoutput_MA;
void setup() {
pinMode(interA, INPUT_PULLUP);
pinMode(5,INPUT_PULLUP);
pinMode(6, OUTPUT);
pinMode(7,OUTPUT);
toodo_MA=0;
E_MA=0; E1_MA=0; E2_MA=0;
Output_MA=0; Lastoutput_MA=0;
T=0.01;
Kp_MA=5.6; Kd_MA=0.01; Ki_MA=0.04;
Serial.begin(9600);
attachInterrupt(digitalPinToInterrupt(18), docxung_MA, FALLING);
}
void docxung_MA() {
if (digitalRead(5))
xung_MA++;
else
```

```

xung_MA--;
}

double PID_MA() {
if(xung_MA<0)
xung_MA=-xung_MA;
tocdo_MA=(xung_MA/374/T)*60;
kca+=xung_MA;
xung_MA=0;
E_MA=200 - tocdo_MA;
alpha_MA=2*T*Kp_MA + Ki_MA*T*T + 2*Kd_MA;
beta_MA= T*T*Ki_MA - 4*Kd_MA - 2*T*Kp_MA;
gama_MA=2*Kd_MA;
Output_MA=(E_MA*alpha_MA + beta_MA*E1_MA + gama_MA*E2_MA +
2*T*Lastoutput_MA)/(2*T);
Lastoutput_MA=Output_MA;
E2_MA=E1_MA;
E1_MA=E_MA;
if(Output_MA>255)
Output_MA=255;
if(Output_MA<0)
Output_MA=0;
if(Output_MA>0)
{
analogWrite(6, Output_MA);
digitalWrite(7, LOW);
}
else
{
digitalWrite(6, LOW);
digitalWrite(7, LOW);
}
}
}

void loop() {
PID_MA;
Serial.print(toccdodat);Serial.print('\t');
Serial.println(tocdo_MA);
}

```

Khi đó giá trị Output sẽ là giá trị để điều khiển tốc độ động cơ để động cơ quay với tốc độ gần với giá trị đã được định trước.

4.3.3. Thiết lập lô trình

Để thực nghiệm giải thuật và điều khiển giám sát robot di động tự hành được đề cập ở các mục trên, mỗi bánh xe của robot được nối với một motor DC thông qua bộ giảm tốc và bộ encoder đặt trên trục của motor. Mỗi motor được điều khiển bởi bộ vi điều khiển trung tâm Arduino Mega thông qua mạch cầu H kép, có khả năng đọc xung từ bộ encoder tương đối. Mỗi bộ điều khiển cho phép motor chạy ON hoặc tắt OFF theo phương pháp PWM (Pulse Width Modulation) với tần số đóng ngắt 20kHz.

Để điều khiển robot di chuyển theo tọa độ (trên hệ trực Oxy) có thể sử dụng cách điều khiển robot theo đường thẳng. Trong đồ án này dùng quỹ đạo thẳng để điều khiển robot. Tức là robot chỉ chạy theo trục x hoặc trục y. Nhiệm vụ là đưa robot ở vị trí có tọa độ ban đầu (x_o, y_o) đến vị trí có tọa độ ($x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$

Sau đây là giải thuật theo quỹ đạo thẳng:

Bước 1: Đọc vị trí hiện tại và vị trí đích của robot

Vị trí hiện tại (x_o, y_o)

Vị trí đích (x_n, y_n)

Bước 2: Tính toán độ dài quãng đường đi của robot

Quãng đường $S = |x_n - x_{n-1}|$ hoặc $S = |y_n - y_{n-1}|$

Bước 3: Sau khi đến vị trí có tọa độ (x_1, y_1)

Góc quay của robot là góc 90^0

Bước 4: Cập nhật vị trí mới của robot

Tọa độ (x_1, y_1)

Bước 5: Quay lại Bước 1.

Một cách tìm đường cục bộ khá hoàn hảo là phương pháp bản đồ Nơ-ron cực (Polar Neurol Map) của Michail G. Lagoudakis (1998) dùng 16-32 cảm biến gắn trên robot, tạo trường cảm biến bao phủ vùng xung quanh robot, thuật toán sử dụng là thuật toán ánh xạ nơ-ron tương đối phức tạp. Do giới hạn của đề tài, phương pháp này không được đề cập đến.

Phương pháp được đề cập đến ở đây là sử dụng 3 cảm biến dò đường xác định khoảng cách từ đầu cảm biến đến vật cản. Tuỳ vào khoảng cách tới vật cản ứng với từng cảm biến, ta xác định được phương hướng di chuyển phù hợp cho

robot. Tuy nhiên luật dẫn hướng robot trong bài này khá đơn giản, yêu cầu vật cản phải có các cạnh để robot di chuyển tránh vật cản.

Khi robot đang hướng về đích, nếu phát hiện vật cản ở phía trước nó sẽ tiến hành chuyển hướng. Mặc định robot quay phải, tiếp theo đi vòng qua vật cản và trở về lộ trình ban đầu. Để đảm bảo điều này, trong quá trình quay hướng về đích, mỗi lần quay robot quay ta phải cập nhật tín hiệu cảm biến. Khi cả 3 cảm biến đều báo có vật cản thì báo lỗi và dừng hoạt động robot, đây cũng là một nhược điểm của sản phẩm nhóm em.

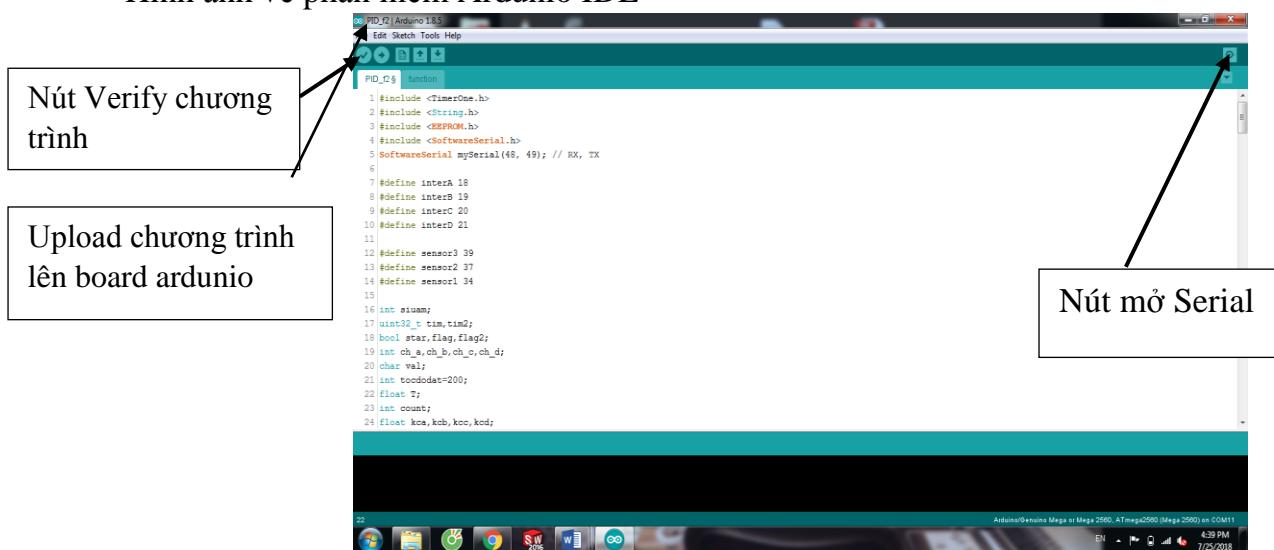
Nhược điểm chính của phương pháp trên là robot không thể xác định đầy đủ biên của vật cản do chỉ có 3 cảm biến, vì vậy robot có thể chạm vật hoặc không về tới đích.

4.4. Xây dựng code chương trình

Chương trình điều khiển được viết bằng ngôn ngữ lập trình C trên nền phần mềm Arduin IDE.

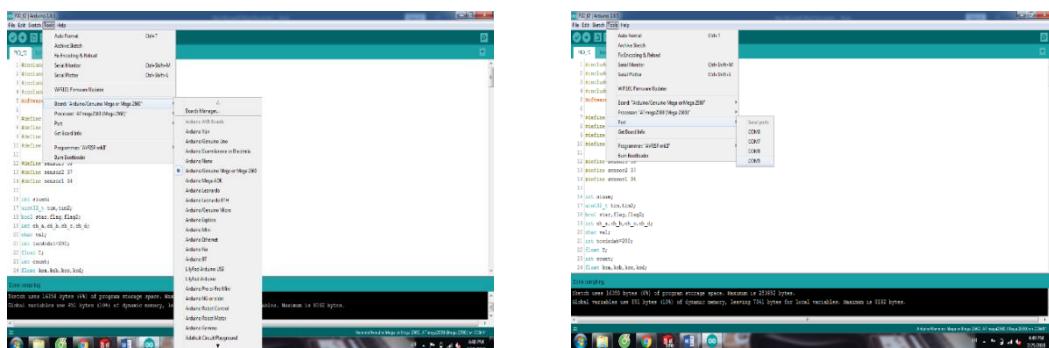
Code chương trình điều khiển được đính kèm ở phụ lục C.

Hình ảnh về phần mềm Arduin IDE



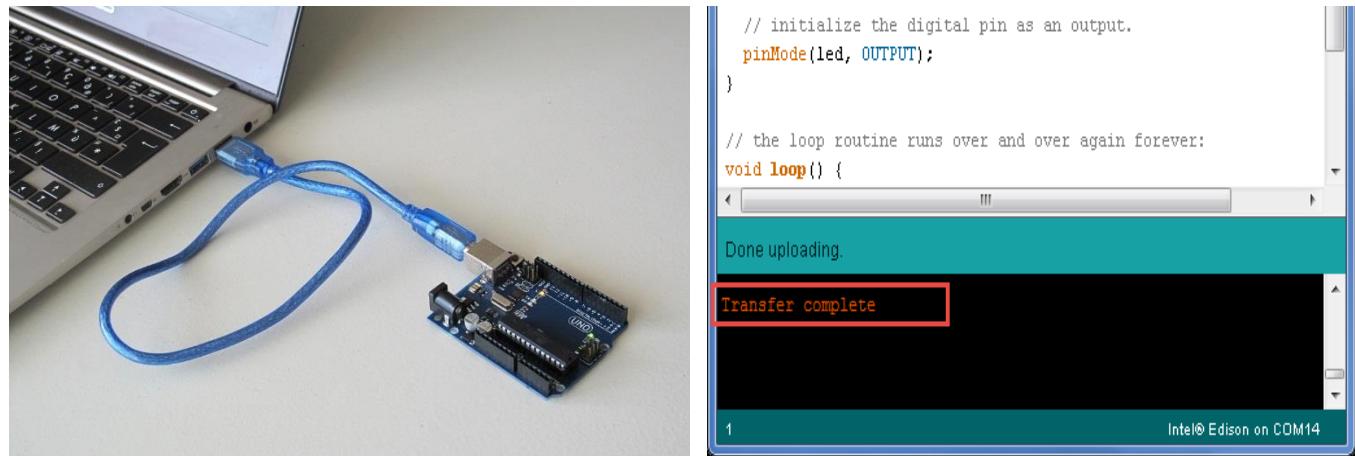
Hình 4.4: Phần mềm Arduin IDE

Để có thể Upload chương trình lên Arduin ta cần chọn loại Board Arduin và chọn Com kết nối. Hình bên dưới thể hiện cách chọn loại board và chọn port tương ứng.



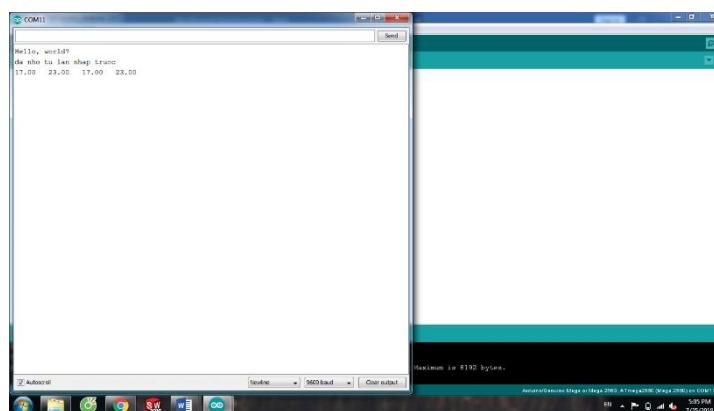
Hình 4.5: Chọn board và cổng com tương ứng

Sau đó ta kết nối máy tính với Arduinio quay cáp kết nối rồi nhấn nút Upload. Khi Upload thành công chương trình lên board arduino ta sẽ nhìn thấy dòng thông báo Done uploading.



Hình 4.6: Upload chương trình

Serial Monitor sẽ hiện thị các thông số được gọi để in ra màn hình.



Hình 4.7: Cửa sổ Serial Monitor

CHƯƠNG 5: MÔ HÌNH THỰC TẾ

5.1. Giới thiệu mô hình

Một vài hình ảnh thực tế của mô hình.

Mặt trên của mô hình



Hình 5.1: Mặt trên của mô hình

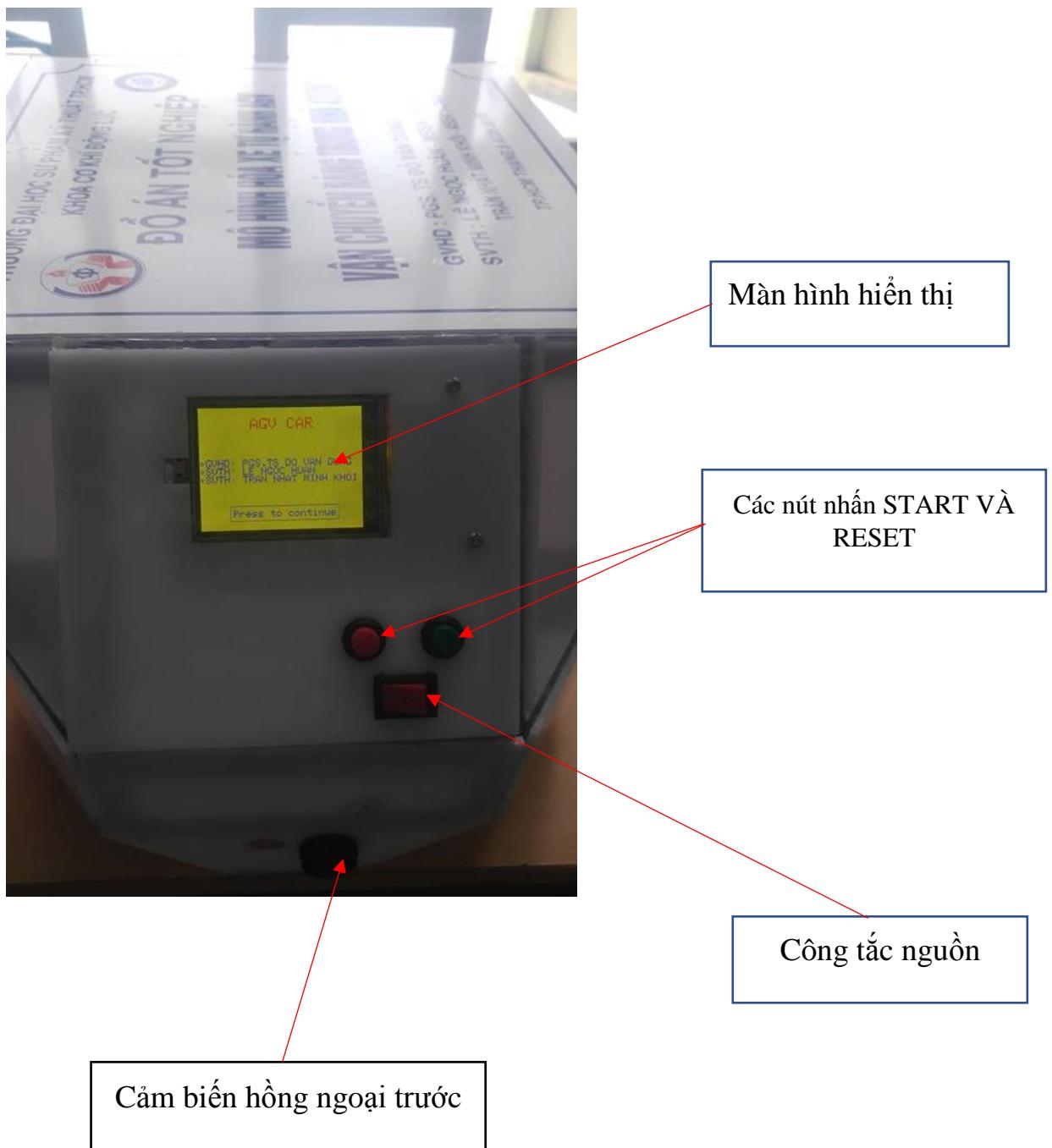
Mặt bên của mô hình



Hình 5.2: Mặt bên của mô hình

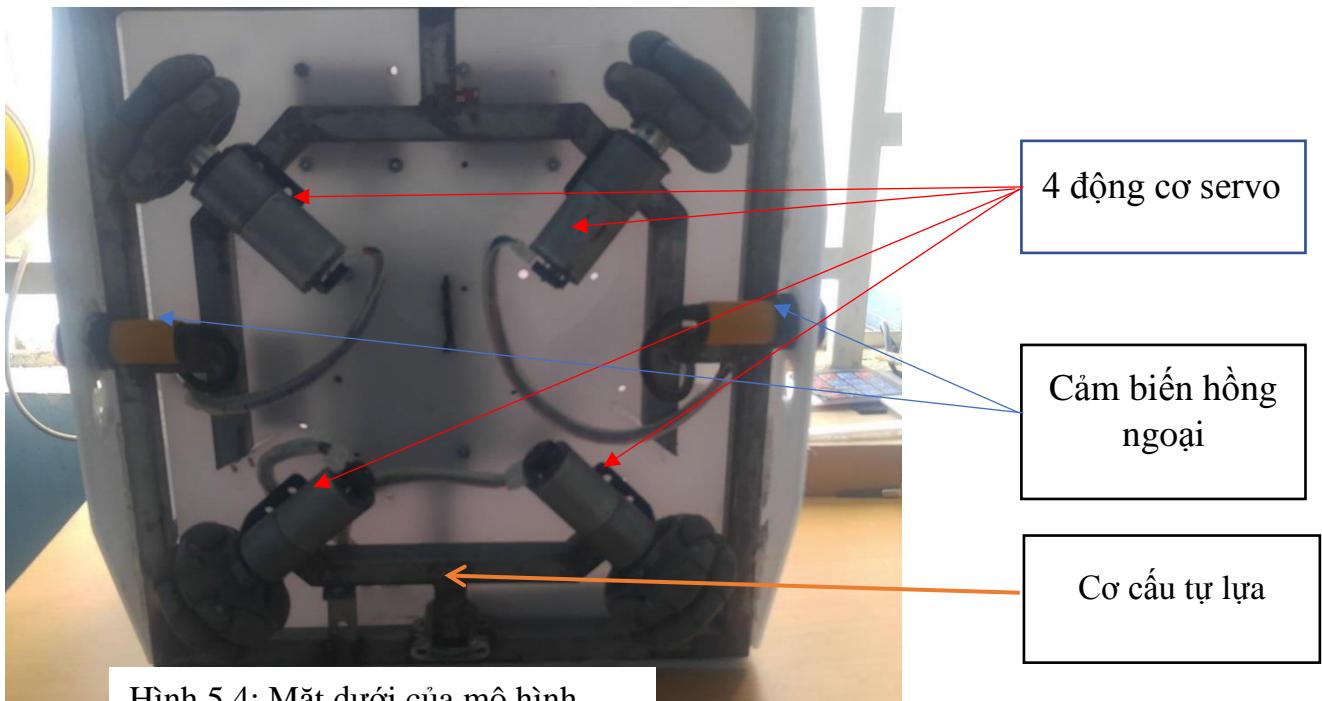
Cảm biến hồng
ngoại bên trái.

Mặt trước của mô hình



Hình 5.3: Mặt trước của mô hình

Mặt bên dưới của mô hình



Hình 5.4: Mặt dưới của mô hình

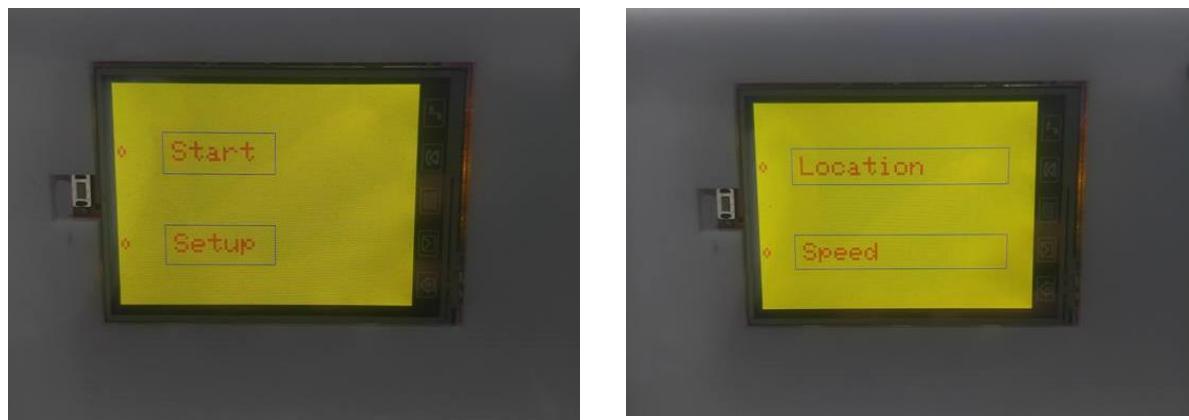
5.2. Vận hành mô hình

Sau khi công tắc nguồn được nhấn, ta tiến hành cài các thông số đầu vào.



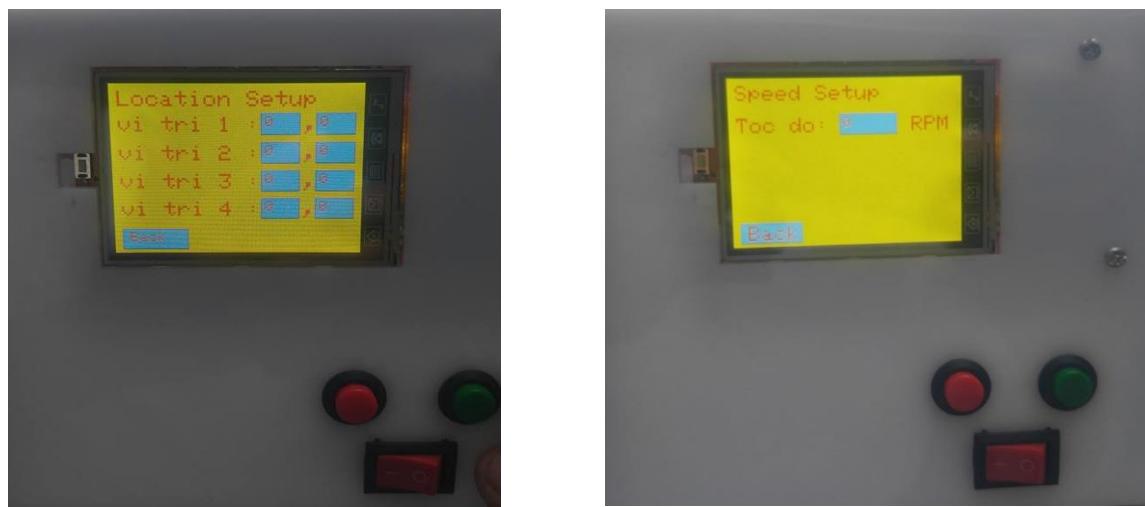
Hình 5.5: Màn hình khởi động

Chọn thẻ Setup để cài các thông số đầu vào gồm có tọa độ các điểm dừng và tốc độ mong muốn từ màn hình.



Hình 5.6: Bước 1

Ta tiến hành nhập thông số tọa độ điểm dừng và tốc độ.



Hình 5.7: Bước 2

Màn hình được lập trình sẵn bàn phím để nhập thông số



Hình 5.8: Bàn phím cảm ứng

Sau khi đã nhập xong thông số ta chọn thẻ Start, màn hình sẽ hiện thị những thông tin về vị trí của xe sau khi ta nhấn nút Start màu xanh.

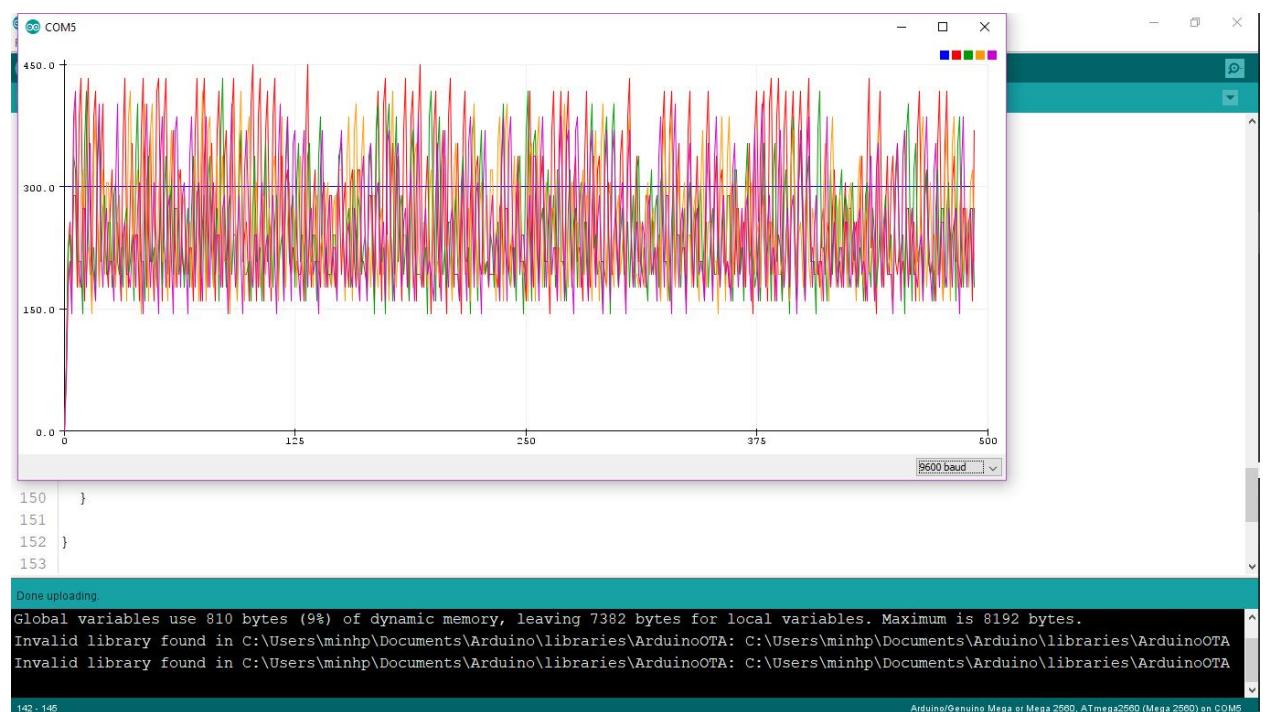


Hình 5.9: Hiển thi thông tin sau khi nút Start được nhấn

5.3. Kiểm nghiệm mô hình

5.3.1. Kiểm nghiệm đáp ứng của bộ PID

Biểu đồ sau thể hiện sự đáp ứng của bộ điều khiển PID trong việc điều khiển tốc độ cho bốn động cơ DC Servo giảm tốc.



Hình 5.10: Kiểm nghiệm bộ PID

Biểu đồ thể hiện các thông số:

- Tốc độ mong muốn động cơ hoạt động(v/ph) là đường thẳng nằm ngang.
- Các giá trị tốc độ thực tế đo được của bốn động cơ được thể hiện dưới dạng xung.

Từ biểu đồ ta nhận thấy nhờ có bộ điều khiển PID mà tốc độ thực tế (tốc độ đầu ra của bộ PID) được điều chỉnh liên tục để gần đạt được giá trị mà ta mong muốn.

5.3.2. Kiểm nghiệm hoạt động của mô hình

Mô hình được thử ở hai chế độ là chạy không vật cản và chạy có vật cản cố định

Với chế độ chạy không vật cản mô hình sẽ chạy tự động đến các vị trí được đánh dấu như hình sau



Hình 5.11: Kiểm nghiệm chạy không vật cản

Mô hình chạy đúng lộ trình từ vị trí xuất phát A đến vị trí B, C, D sau đó quay trở lại vị trí xuất phát ban đầu. Vị trí A, B, C, D là các ô màu đen như hình trên.

Với chế độ chạy có vật cản: lộ trình được giữ như cũ, vật cản được thiết kế là các hộp carton. Khi cảm biến trước phát hiện có vật cản, mô hình sẽ xử lý như sau:

- Khi gặp vật cản mô hình sẽ quay phải, sau đó sẽ đọc cảm biến hồng ngoại bên phải và tiếp tục di chuyển đồng thời sẽ lưu độ dài quãng đường di chuyển.



Hình 5.12a: Kiểm nghiệm chạy có vật cản

- Khi cảm biến bên phải không còn phát hiện vật cản, mô hình sẽ quay trái lúc này mô hình tiếp tục di chuyển thẳng cho đến khi cảm biến bên trái không còn phát



hiện vật cản.

Hình 5.12b: Kiểm nghiệm chạy có vật cản

- Khi cảm biến trái không còn phát hiện vật cản, mô hình sẽ quay trái để trở lại lô



trình ban đầu dựa vào giá trị quãng đường đi được đã được đo từ trước.

Hình 5.12c: Kiểm nghiệm chạy có vật cản

- Sau khi tránh được vật cản, mô hình quay trở lại lộ trình và đi đến đích.



Hình 5.12d: Kiểm nghiệm chạy có vật cản

- Khi tất cả các hướng đều có vật cản, hoặc vật cản quá gần mô hình sẽ không di chuyển.

5.4. Nhược điểm

Không chọn được hệ số chính xác cho bộ điều khiển PID cho bốn bánh xe nên mô hình không thể chạy thẳng 100% như mong muốn.

Góc quay của mô hình được điều chỉnh bằng hàm delay theo thời gian làm cho mô hình không quay chính xác. Với nhiều lần quay không chính xác, phương di chuyển của mô hình sẽ bị lệch.

Nguồn pin sử dụng có dung lượng thấp, thời gian sạc pin lại lâu làm ảnh hưởng đến thời gian sử dụng mô hình.

Màn hình LCD cảm ứng cho chất lượng cảm ứng không cao đôi khi bị lỗi nhập sai.

CHƯƠNG 6: KẾT LUẬN

Qua thời gian làm đồ án tốt nghiệp đã giúp chúng em cũng cố lại kiến thức đã được học, áp dụng được những kiến thức đã học vào thực tế. Trong quá trình làm đồ án chúng em đã học thêm được những bài học quý giá mà khi còn ngồi trên ghế giảng đường không hề biết đến.

Thời gian làm đồ án là thời gian để chúng em cũng cố bổ sung cho mình những phần kiến thức còn thiếu sót, từ đó hoàn thiện hơn cho mình hành trang để bước vào cuộc sống sau tốt nghiệp.

Mức độ hoàn thành của đồ án:

- Thiết kế và gia công khung xe mô hình xe tự hành AGV vận chuyển hàng trong nhà xưởng với khớp lắc giúp cho mô hình xe có thể di chuyển trên cả những địa hình không bằng phẳng.
- Xây dựng được bộ điều khiển PID điều khiển tốc độ bốn động cơ DC Servo giảm tốc.
- Mô hình được tích hợp màn hình LCD TFT cảm ứng để cài đặt vị trí điểm dừng, tốc độ mong muốn và dùng để hiện thị những thông tin.
- Nhận biết được vật cản, tránh được vật cản và đi được đến vị trí đã được định sẵn. Tuy nhiên quỹ đạo di chuyển của mô hình bị lệch một khoảng so với lý thuyết.

Phụ lục A

Sơ đồ mạch điện.

Phụ lục B

Bản vẽ thiết kế mô hình.

Bản vẽ chi tiết mô hình.

Phụ lục C

Code chương trình

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Phương Hà, Lí thuyết Điều khiển Tự động, NXB ĐH Quốc gia, 2005.
- [2]. arduino.vn
- [3]. http://en.wikipedia.org/wiki/PID_controller
- [4]. www.alldatasheet.com
- [5]. <http://bmktd.cscvn.com/forum/showthread.php?3096-Kh%C3%A2u-PID-s%E1%BB%91>.
- [6]. Jinghua Zhong, PID Controller Tuning: A Short Tutorial.
- [7]. www.clbrobot.com
- [8]. <https://vma.vn/2017/07/29/thiet-ke-che-tao-xe-tu-hanh-agv-trong-cong-nghiep/>

