

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



ĐỒ ÁN TỐT NGHIỆP

**NGHIÊN CỨU THUẬT TOÁN DI CHUYỂN
MOBILE ROBOT**

GVHD: Ths. Trần Thanh Lam

SVTH

MSSV

Nguyễn Hoàng Danh

18146090

Nguyễn Tấn Duy

18146092

Đỗ Đình Long

18146157

Tp. Hồ Chí Minh, tháng 7 năm 2022

DANH SÁCH THÀNH VIÊN THAM GIA
HỌC KỲ I NĂM HỌC 2022-2023

STT	Họ và tên	MSSV	Tỉ lệ % hoàn thành
1	Nguyễn Hoàng Danh	18146090	100%
2	Nguyễn Tấn Duy	18146092	100%
3	Đỗ Đình Long	18146157	100%

Ghi chú:

- Tỷ lệ % = 100%: Mức độ phần trăm của từng sinh viên tham gia.
- Trưởng nhóm: Nguyễn Hoàng Danh

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ký tên

Ths. Trần Thanh Lam

LỜI CẢM ƠN

Để có được kết quả đồ án như ngày hôm nay là một chặng đường đầy khó khăn thử thách đối với nhóm thực hiện. Nhóm chúng em xin cảm ơn đến thầy Võ Lâm Chương và thầy Trần Thanh Lam đã cho tụi em những hướng đi đúng đắn, tận tình hướng dẫn và chỉ bảo cho chúng em, truyền cảm hứng và động lực để nhóm hoàn thành tốt đồ án môn học này.

Bên cạnh đó chúng em rất biết ơn đến những thầy cô tại trường đại học Sư Phạm Kỹ Thuật, đặc biệt là bộ môn cơ điện tử đã giảng dạy chúng em bằng cả tâm huyết, nhiệt tình hỗ trợ chúng em trong khoảng thời gian lên lớp và giải đáp thắc mắc đồ án của chúng em.

Cuối cùng, dù đã rất cố gắng để có thể hoàn thiện thật tốt đồ án môn học, nhưng với điều kiện thời gian cũng như kinh nghiệm vẫn còn hạn chế, nhóm chúng em vẫn không tránh khỏi những thiếu sót và sai sót. Chúng em rất mong nhận được sự góp ý và những lời nhận xét quý báu từ các thầy cô để nhóm có thể hoàn thiện và nâng cao kiến thức. Nhóm chúng em xin tiếp nhận bằng cả lòng biết ơn.

Xin chân thành cảm ơn.

Sinh viên thực hiện

Nguyễn Hoàng Danh

Nguyễn Tấn Duy

Đỗ Đình Long

LỜI NÓI ĐẦU

Thế giới đã trải qua bốn cuộc cách mạng công nghiệp vĩ đại của lịch sử con người. Và bây giờ là năm 2022 của thế kỉ XXI, con người đang sống và đóng góp từng ngày cho sự phát triển của nhân loại trong cuộc cách mạng công nghiệp lần thứ tư – “Công nghiệp 4.0”. Năm 2013, một từ khóa mới là “Công nghiệp 4.0” bắt đầu nổi lên xuất phát từ một báo cáo của chính phủ Đức đề cập đến cụm từ này nhằm nói tới chiến lược công nghệ cao, điện toán hóa ngành sản xuất mà không cần sự tham gia của con người. Đây được gọi là cuộc cách mạng số, thông qua các công nghệ như Internet vạn vật (IoT), trí tuệ nhân tạo (AI), thực tế ảo (VR), tương tác thực tại ảo (AR), mạng xã hội, điện toán đám mây, di động, phân tích dữ liệu lớn (SMAC)... để chuyển hóa toàn bộ thế giới thực thành thế giới số.

Để phục vụ cho nền công nghiệp đang phát triển vô cùng năng động và đang là xu thế trên toàn thế giới, không thể thiếu đến Robot. Robot là một loại thiết bị 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 có một trong các đặc điểm sau đây: do con người sáng tạo ra, có khả năng nhận biết môi trường xung quanh và tương tác với những vật thể trong môi trường, có khả năng đưa ra các lựa chọn dựa trên môi trường và được điều khiển một cách tự động theo những trình tự đã được lập trình trước, có thể điều khiển được bằng các lệnh để có thể thay đổi tùy theo yêu cầu của người sử dụng, có thể di chuyển quay hoặc tịnh tiến theo một hay nhiều chiều và khéo léo trong vận động.

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

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

Nội dung cơ bản gồm 8 chương. Có các nội dung cơ bản như sau:

Chương 1: Tổng quan về đề tài

Chương 2: Tổng quan về Mobile Robot

Chương 3: Cơ sở lý thuyết

Chương 4: Thiết kế cơ khí cho Mobile Robot

Chương 5: Xây dựng hệ thống Điện – Điều khiển

Chương 6: Xây dựng App tương tác với người dùng

Chương 7: Xây dựng giải thuật điều khiển

Chương 8: Kết quả và thực nghiệm

Chương 9: Kết luận và hướng phát triển

Xin chân thành gửi lời cảm ơn tới thầy Võ Lâm Chương và thầy Trần Thanh Lam đã giúp chúng em hoàn thành đồ án này. Do kiến thức còn hạn chế nên đồ án còn nhiều sai sót. Chúng em xin nhận sự giúp đỡ của các thầy và các bạn để đồ án được hoàn thiện hơn. Chúng em xin chân thành cảm ơn.

MỤC LỤC

LỜI NÓI ĐẦU	i
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	1
1.1. Giới thiệu	1
1.2. Lý do chọn đề tài	4
1.3. Tính cấp thiết của đề tài.....	4
1.4. Ý nghĩa của đề tài	5
1.5. Mục tiêu của đề tài	5
1.6. Phương pháp nghiên cứu	5
CHƯƠNG 2: TỔNG QUAN VỀ MOBILE ROBOT	6
2.1. Lịch sử phát triển	6
2.2. Phân loại	8
2.3. Đặt vấn đề.....	10
2.4. Giới hạn đề tài	12
CHƯƠNG 3: CƠ SỞ LÝ THUYẾT	14
3.1. Lựa chọn cơ cấu chuyển động cho Mobile Robot.....	14
3.2. Microcontroller Unit STM32F103C8T6	15
3.2.1 Giới thiệu về STM32F103C8T6	15
3.2.2. Cấu hình chi tiết của STM32F103C8T6	16
3.3. Cảm biến gia tốc, con quay đàn hồi chuyển và từ kế	17
3.3.1. Giới thiệu về IMU	17
3.3.2. Ứng dụng của IMU.....	18
3.3.3. Nguyên lý hoạt động của từng loại	19
3.3.4. Cảm biến là loại phổ biến nhất cho các ứng dụng hiện nay	20
3.4. Cảm biến siêu âm đo khoảng cách	20
3.4.1 Giới thiệu về cảm biến siêu âm	20
3.4.2 Nguyên lý hoạt động của cảm biến siêu âm.....	20
3.5. Bộ lọc Kalman	21

3.5.1	Bộ lọc là gì?.....	21
3.5.2.	Giới thiệu bộ lọc Kalman	21
3.5.3.	Bản chất bộ lọc Kalman	22
CHƯƠNG 4: THIẾT KẾ CƠ KHÍ CHO MOBILE ROBOT		23
4.1.	Một số nghiên cứu về Omni Robot trên thế giới và Việt Nam	23
4.2.	Thiết kế kỹ thuật cơ khí.....	24
4.2.1.	Tính toán động học cho robot	24
4.2.2.	Tính toán động lực học cho robot	25
4.2.3.	Tính toán cho động cơ.....	27
4.2.4.	Tính toán động học cho một số chuyển động cơ bản.....	28
4.2.5.	Tính toán lựa chọn động cơ.....	30
4.2.6.	Tính toán chọn số cảm biến.....	31
CHƯƠNG 5: XÂY DỰNG HỆ THỐNG ĐIỆN – ĐIỀU KHIỂN.....		35
5.1.	Hệ thống điều khiển Mobile Robot	35
5.1.1.	Điều khiển động cơ	35
5.1.2.	Xác định tọa độ và góc quay của Mobile Robot	36
5.1.3	Phát hiện vật cản.....	37
5.2.	Hệ thống điện Mobile Robot	38
5.2.1	Tổng quan hệ thống điện.....	38
5.2.2.	An toàn điện	38
5.2.3.	Các thiết bị được sử dụng trên Mobile Robot	39
CHƯƠNG 6: XÂY DỰNG GIAO DIỆN APP TƯƠNG TÁC VỚI NGƯỜI DÙNG		60
6.1	Giới thiệu về tổng quan giao diện.....	60
6.2	Phần mềm thiết kế App – Thunkable	60
6.3	Cơ sở dữ liệu – Firebase RealTime Database.....	61
6.4	Thiết kế App điều khiển	62
CHƯƠNG 7: XÂY DỰNG GIẢI THUẬT ĐIỀU KHIỂN		66

7.1.	Tổng quan sơ đồ điều khiển	66
7.2.	Lý thuyết về bộ điều khiển PID.....	68
7.2.1.	Định nghĩa	68
7.2.2.	Giải thuật PID số	69
7.2.3	Giải thích thư viện pid_controller.c và pid_controller.h	70
7.3.	Giải thuật di chuyển đa hướng Mobile Robot	74
7.4.	Giải thuật di chuyển đến vị trí cố định của Mobile Robot	75
CHƯƠNG 8: KẾT QUẢ THỰC NGHIỆM		77
8.1.	Thi công mô hình.....	77
8.2.	Kết quả thu được cảm biến gia tốc MPU9250	82
8.3.	Kết quả thu được từ cảm biến khoảng cách US-015	83
8.4.	Kiểm tra tính ổn định của bộ PID.....	84
8.5.	Phần mềm App Mobile Robot	85
8.6.	Di chuyển đa hướng Mobile Robot	85
8.7.	Di chuyển tự động	86
8.8.	Di chuyển theo vị trí cố định	86
8.9.	Thực nghiệm:.....	87
CHƯƠNG 9: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		88
9.1.	Về lý thuyết	88
9.2.	Về thực nghiệm	88
9.3.	Ưu điểm, nhược điểm	88
9.4.	Hướng phát triển.....	88
TÀI LIỆU THAM KHẢO.....		89

DANH MỤC CÁC TỪ VIẾT TẮT

AGV	Automatic Guided Vehicle
ANSI	American National Standards Institute
ARM	Advanced RISC Machine
AVR	Automatic Voltage Regulator
CSS	Cascading Style Sheets
DMP	Data Management Platform
DOM	Document Object Model
DSP	Digital Motion Processor
GPS	Global Positioning System
HDMI	High-Definition Multimedia Interface
HTML	Hyper Text Markup Language
I2C	Inter-Integrated Circuit
IACS	International Association of Classification Societies
IMU	Inertial Measurement Unit
PI	Proportional Integral
PIC	Programmable Intelligent Computer
PID	Proportional Integral Derivative
PoE	Power over Ethernet

MỤC LỤC HÌNH ẢNH

Hình 2. 1 Military Robot.....	21
Hình 2. 2 iRobot 510 PackBot.....	21
Hình 2. 3 Robot dò line.....	25
Hình 2. 4 Robot hút bụi Xiaomi Mijia Gen	26
Hình 2. 5 Robomow RS622.....	28
Hình 2. 6 Động cơ DC Servo JGB37-520 DC Geared Motor	31
Hình 3. 1 Mô hình động học và động lực học của Robot.....	14
Hình 3. 2 Mô hình động học và động lực học của Robot khi quay một góc θ	14
Hình 3. 3 STM32F103C8T6.....	16
Hình 3. 4 Các giá trị của IMU	17
Hình 3. 5 Cảm biến từ trường.....	19
Hình 3. 6 Hiệu ứng Hall.....	19
Hình 3. 7 Nguyên lý hoạt động của cảm biến siêu âm	21
Hình 3. 8 Bản chất bộ lọc Kalman.....	22
Hình 4. 1 WowWee Rovio Robot.....	23
Hình 4. 2 Robot Omni do đại học bách khoa thành phố Hồ Chí Minh thiết kế ...	23
Hình 4. 3 Robot SPARK IV	24
Hình 4. 4 Hệ số ma sát khô và hệ số ma sát ướt.....	26
Hình 4. 5 Chuyển động tịnh tiến.....	29
Hình 4. 6 Chuyển động quay	29
Hình 4. 7 Sử dụng cảm biến	32
Hình 4. 8 Sử dụng xử lý ảnh.....	32
Hình 4. 9 Mô phỏng cách một cảm biến quét trong không gian	33
Hình 4. 10 Cách phân bố cảm biến trên Mobile Robot	34
Hình 5. 1 Sơ đồ khối hệ thống điều khiển Mobile Robot	35
Hình 5. 2 Sơ đồ khối điều khiển động cơ	36
Hình 5. 3 Giải thuật điều khiển động cơ PID	36
Hình 5. 4 Sơ đồ khối xác định vị trí của Mobile Robot trong không gian	37

Hình 5. 5 Các góc quét của cảm biến siêu âm	37
Hình 5. 6 Sơ đồ cấp nguồn 24V	38
Hình 5. 7 Sơ đồ cấp nguồn 5V	38
Hình 5. 8 Đặc tính của aSMD300.....	39
Hình 5. 9 Động cơ DC Servo JGB37-520 DC Geared Motor	39
Hình 5. 10 MPU6050.....	40
Hình 5. 11 Mô tả thanh ghi 26: FSYNC và DLPF	550
Hình 5. 12 Bảng giá trị FSYNC trên thanh ghi 26: FSYNC và DLPF	41
Hình 5. 13 Bảng giá trị DLPF trên thanh ghi 26: FSYNC và DLPF.....	42
Hình 5. 14 Mô tả thanh ghi 27: Gyroscope Configuration.....	42
Hình 5. 15 Giá trị FS_SEL trên thanh ghi 27: Gyroscope Configuration	43
Hình 5. 16 Thanh ghi 27: Accelerometer Configuration.....	44
Hình 5. 17 Giá trị FS_SEL trên thanh ghi 28: Accelerometer Configuration	44
Hình 7. 1 Sơ đồ khối giải thuật di chuyển	68
Hình 7. 2 Ví dụ minh họa cho thuật toán.....	75
Hình 7. 3 Mô phỏng vị trí trên khu vực cố định	75
Hình 5. 24 Bảng mô tả trên thanh ghi 65 - 66: Temperature Measurements	48
Hình 5. 26 Bảng mô tả trên thanh ghi 03-08: Magnematic Measurement	48
Hình 5. 27 Góc roll, pitch và yaw trên MPU9250.....	48
Hình 5. 28 US-015	49
Hình 5. 29 Thông số kỹ thuật	49
Hình 5. 30 Cách xác định vật.....	50
Hình 5. 31 Nguyên lý hoạt động.....	50
Hình 5. 32 Chế độ InputCapture	51
Hình 5. 33 Timer trong chế độ InputCapture	52
Hình 6. 1 Sơ đồ khối tổng quát giao diện điều khiển	53
Hình 6. 2 Giao diện của thunkable	54
Hình 6. 1 Nguyên lý hoạt động của Firebase.....	55
Hình 6. 4 Thiết kế App điều khiển	55
Hình 6. 5 Quá trình Save giá trị điều khiển lên Firebase.....	56
Hình 6. 6 Cấu trúc của chương trình trên Firebase.....	56

Hình 7. 1 Sơ đồ khối giải thuật di chuyển	58
Hình 7. 2 Ví dụ minh họa cho thuật toán.....	59
Hình 7. 3 Mô phỏng vị trí trên khu vực cố định	60
Hình 7. 1 Sơ đồ khối giải thuật di chuyển	68
Hình 7. 2 Ví dụ minh họa cho thuật toán.....	75
Hình 7. 3 Mô phỏng vị trí trên khu vực cố định	75
Hình 8. 7 Giao diện App điều khiển	85
Hình 8. 8 Chế độ di chuyển đa hướng	85
Hình 8. 9 Chế độ di chuyển tự động	86
Hình 8. 10 Chế độ di chuyển theo vị trí cố định.....	86

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu

Khi nhắc đến Robot thì không thể nào bỏ qua một loại robot đang rất thịnh hành . Đó chính là Robot tự hành (Mobile Robot) hay còn gọi là Autonomous Mobile Robots (AMRs). Đây là loại robot hội tụ đầy đủ những yếu tố đặc trưng của kỷ nguyên máy tính – tự động hóa với hệ thống liên kết thể giới thực và ảo. Hiện nay, AMRs không chỉ phục vụ cho công nghiệp mà chúng còn đang được con người sử dụng một cách đầy thông minh và tối ưu trong đời sống xã hội: y tế, giáo dục, nhà hàng khách sạn,... thậm chí là phục vụ nhu cầu sinh hoạt tại nhà.

Một Mobile Robot , là một con robot có khả năng di chuyển trong xung quanh (vận động). Chúng có khả năng điều hướng trong môi trường không được kiểm soát mà không cần đến các thiết bị hướng dẫn vật lý hoặc cơ điện. Ngoài ra, rô bốt di động có thể dựa vào các thiết bị hướng dẫn cho phép chúng đi theo một lộ trình điều hướng được xác định trước trong không gian được kiểm soát tương đối (AGV - xe dẫn đường tự hành).

Mobile Robot là một ngành định hướng giải pháp đang phát triển nhanh chóng, kết hợp khoa học kỹ thuật và công nghệ thông tin với các ngành như khoa học nhận thức, Trí tuệ nhân tạo và nhiều ngành khác. Chính sự tương tác liên ngành này đã giúp chúng ta có thể làm chủ được sự phức tạp vốn có của Mobile Robot. Mobile Robot có khả năng di chuyển độc lập và thực hiện các hành động nhất định. Về cơ bản, ngoài khả năng di chuyển, chúng có thể hoạt động tự chủ mà không cần sự can thiệp của con người. Có một sự khác biệt chung giữa các thiết bị bán tự trị và hoàn toàn tự động, dựa trên nguồn điện của robot; nguồn điện bên ngoài, chẳng hạn như trạm sạc, có tác dụng hạn chế quyền tự chủ.

Mặc dù lĩnh vực Mobile Robot hiện đang ở giai đoạn phát triển tương đối sớm, các nguyên mẫu và thậm chí một số sản phẩm nổi tiếp đang bắt đầu được tìm thấy trong một loạt các lĩnh vực. Mobile Robot cũng đã tìm được đường vào các ứng dụng hàng ngày trong nước với việc phát minh ra máy hút bụi di động, có thể hoạt động với khả năng tự chủ gần như hoàn toàn. Mobile Robot đã trở nên phổ biến hơn trong các môi trường thương mại và công nghiệp. Các bệnh viện đã sử dụng Mobile Robot tự động để di chuyển vật liệu trong nhiều năm. Các nhà kho đã lắp đặt hệ thống rô bốt di động để di chuyển hiệu quả nguyên liệu từ các kệ chứa hàng đến khu vực thực hiện đơn

hàng. Mobile Robot cũng là một trọng tâm chính của nghiên cứu hiện nay và hầu hết mọi trường đại học lớn đều có một hoặc nhiều phòng thí nghiệm tập trung vào nghiên cứu Mobile Robot. Mobile Robot cũng được tìm thấy trong các cơ sở công nghiệp, quân sự và an ninh. Những đổi mới được thực hiện trong lĩnh vực Trí tuệ nhân tạo và các ứng dụng của chúng được thiết lập để nâng cao đáng kể mức độ liên quan và sử dụng các Mobile Robot, độc lập trong tương lai.

Ưu nhược điểm của Mobile Robot:

- ***Ưu điểm :***

Mobile Robot là loại robot có khả năng thao tác ở nhiều địa điểm khác biệt nhau, không ở cố định một vị trí nào. Mobile robot có đặc tính khá linh động, khả năng đó đến từ việc có các bộ phận chuyển động như bánh xe, chân, tay, cánh quạt... Mobile Robot “phải biết” định vị và “thu nhận” được thông tin đầy đủ về môi trường xung quanh, sau đó nó sẽ quyết định thực hiện hành động nào cho phù hợp.

Vì vậy , Mobile Robot sẽ thường được kết hợp thêm các cảm biến nhằm giúp cho robot có thể cảm nhận được môi trường xung quanh trong lúc di chuyển. Ngoài ra, Mobile robot còn có thể kết nối với một hệ thống điều khiển trên máy tính cũng như hệ thống cung cấp điện năng cho các tác vụ của cảm biến và hệ thống chuyển động. Dựa trên các đặc tính của từng loại robot, một số loại Mobile robot bắt buộc phải mang theo nguồn điện, camera, micro, bộ cảm biến và các bộ xử lý. Tuy nhiên, do các Mobile Robot đều có một tải trọng nhất định, nên cần phải tính toán trọng lượng các vật một cách tỉ mỉ chi tiết trong quá trình thiết kế Robot. Một đặc điểm quan trọng khác nữa là các Mobile Robot cần phải có tính tự động một cách tương đối, nghĩa là phải có khả năng tự làm một hành động nào đó mà không cần có sự can thiệp của con người.

- ***Nhược điểm:***

Khác với robot cố định, để vận hành được **Mobile Robot** thì cần có thêm những yêu cầu cao hơn, đòi hỏi quá trình thiết kế chế tạo robot đầu tư nhiều hơn. Trong khi **việc vận hành đối với robot cố định** không quá phức tạp, chỉ cần không gian được giới hạn sẵn để thực các công việc lặp đi lặp lại, còn hệ thống Mobile Robot hoạt động trong không gian mở, thay đổi liên tục và luôn có những biến số phức tạp.

Ứng dụng của Mobile Robot:

Mobile Robot áp dụng cho rất nhiều loại công việc khác nhau từ công nông nghiệp cho đến xây dựng, từ dò bom mìn ở các nơi nguy hiểm đến thăm dò các mỏ dầu khí trên biển, xử lý vấn đề ô nhiễm bụi bẩn của môi trường, y học sinh học hóa học , vận chuyển hàng hóa,...

- **Ngành lắp ráp**

- Lắp ráp sửa chữa xe: di chuyển lớp xe kho chính sang kho lưu trữ trung gian để sửa chữa các trạm báo chí.
- Thiết bị điện trên ô tô: PCBs di chuyển và các thiết bị nhỏ lẻ dùng và dòng phụ tùng thiết phụ bổ sung trên xe ô tô.
- Phụ kiện: vận chuyển totes đi đến từ các trạm ép phun.

- **Kỹ thuật số**

- Semiconductor WF: vận tải WIP Intrabay giữa stockers để xử lý các thiết bị công cụ, lưới caro vận tải cho quá trình in ảnh.
- Đóng gói bán dẫn sản xuất và kiểm tra: di chuyển khay chip IC qua xe đẩy hàng.
- Mobile Device Sản xuất: di chuyển tote của PCBs trong lắp ráp và sửa chữa các thiết bị cầm tay.
- Trung tâm dữ liệu: giám sát tài nguyên môi trường và xử lý sự cố trong máy chủ showroom lớn.

- **Thực phẩm**

- Nhà máy thực phẩm: di chuyển các hộp các tông để đóng gói các nguyên liệu.
- Cơ sở phục vụ: di chuyển totes.

- **Logistics**

- Kho hàng: thương mại điện tử thực hiện đơn hàng, phân phối và vận chuyển các đơn hàng.
- Sản xuất Sản phẩm và tiêu dùng.

- **Sản xuất đồ trang sức**

- Di chuyển khuôn mẫu trang sức hoàn thành trạm đúc.
- Sản xuất kính mát: băng tải ảo để di chuyển hộp kính mát từ ASRS đài sắp xếp lại và sau đó mang lại cho phần hoàn thiện đến tải.

- Vận chuyển trang thiết bị sản xuất.
- Đảm bảo quá trình vận hành suôn sẻ.
- **Khách san:** Dịch vụ giao hàng và dẫn khách hàng đến từng phòng đã được đặt sẵn.
- **Bệnh viện:** Dụng cụ thiết bị phẫu thuật tại bệnh viện.

1.2. Lý do chọn đề tài

Nhằm phục vụ giúp cho nền công nghiệp đang phát triển vô cùng năng động và đang là một trong những xu thế trên toàn thế giới, không thể thiếu đến Mobile Robot. Mobile Robot là một loại thiết bị điện tử có thể hiện thực hóa công việc một cách tự động bằng cách điều khiển, lập trình trên máy tính hoặc các bo mạch trên những máy tính nhúng vi điều khiển. Mobile Robot phải có một trong các đặc điểm sau đây: được sáng tạo ra bởi con người, có khả năng nhận biết môi trường xung quanh và tương tác với những vật thể được đặt ngẫu nhiên trong môi trường, có khả năng đưa ra các lựa chọn nhanh và chính xác, được điều khiển tự động theo trình tự lập trình sẵn, có thể điều khiển bằng các lệnh có thể thay đổi theo yêu cầu của người dùng, có thể tiến trên các bề mặt mà không cần cài đặt.

Để góp phần phát triển khoa học công nghệ nước nhà, nhóm chúng em đã chọn “**Mobile Robot, chuyển động đa hướng, bánh xe Omni**”. Từ những suy nghĩ đó chúng em sử dụng những kiến thức mình có để nghiên cứu chế tạo Mobile Robot trong phạm vi đồ án tốt nghiệp với ước mơ đóng góp vào sự phát triển của nước nhà chế tạo Robot của trong thời gian sắp tới giúp đất nước phát triển.

1.3. Tính cấp thiết của đề tài

Hiện nay Mobile Robot được ứng dụng rất nhiều trong các lĩnh vực trong công nghiệp, nông nghiệp lẫn quá trình sinh hoạt hằng ngày. Và nhờ phát triển mạnh mẽ của công nghệ, các khu công nghiệp hay các mái ấm gia đình đều có xuất hiện Mobile Robot, việc này giúp đẩy nhanh năng suất làm việc và cuộc sống của các hộ gia đình trở nên dễ dàng và tiết kiệm thời gian hơn, nhanh chóng, hiệu quả và giúp làm giảm sức lao động của con người giúp công việc dễ dàng hơn. Tuy nhiên, do điều kiện các doanh nghiệp hay của các hộ gia đình nhà máy ở Việt Nam còn khá hạn chế trong lĩnh vực của công nghệ này nên cần phải có giải pháp nhanh chóng để tiết kiệm thời gian hơn nữa.

1.4. Ý nghĩa của đề tài

Với đề tài này, nhóm mong muốn tạo ra một loại Robot có khả năng tự di chuyển trong một khu vực nhất định được xác định từ trước, có thể thay thế con người làm những việc nguy hiểm để đảm bảo an toàn, tránh những nguy hiểm có thể xảy ra. Ngoài ra có thể tiết kiệm được thời gian cũng như chi phí. Với đề tài này, nhóm nghiên cứu mong muốn góp phần nâng cao năng suất lao động, mang lại lợi ích cho doanh nghiệp và sự phát triển của đất nước.

1.5. Mục tiêu của đề tài

- Nghiên cứu, thiết kế Mobile Robot có ngoại hình gần gũi phù hợp với các ứng dụng của doanh nghiệp, dễ dàng tương tác.
- Xác định được vị trí của Robot trong khu vực hoạt động.
- Phát hiện vật cản có thể dừng lại hoặc né tránh

1.6. Phương pháp nghiên cứu

Để thực hiện đề tài này, nhóm thực hiện sử dụng phương pháp chủ yếu là kết hợp giữa nghiên cứu lý thuyết và thực nghiệm kết quả. Về nghiên cứu lý thuyết, nhóm đã tham khảo các tài liệu về cơ khí, lập trình, điện - điện tử, các tài liệu nghiên cứu trong và ngoài nước, các đề tài chế tạo Mobile Robot đã được làm ra. Về thực nghiệm, nhóm đã tiến hành thiết kế, lắp đặt thi công mô hình Mobile Robot, Ngoài ra, nhóm cũng kết hợp các kiến thức chuyên ngành cơ điện tử, kiến thức đa ngành, sử dụng các phần mềm máy tính để tính toán, thiết kế, mô phỏng, xây dựng mô hình thực nghiệm, qua đó có thể đáp ứng các đặc điểm của Mobile Robot cho phù hợp với yêu cầu.

CHƯƠNG 2: TỔNG QUAN VỀ MOBILE ROBOT

2.1. Lịch sử phát triển

Mobile Robot được giới thiệu lần đầu tiên vào năm 1939-1945, trong Thế chiến thứ hai. Nó là kết quả của các thành tựu khoa học và công nghệ trong các lĩnh vực nghiên cứu và phát triển như khoa học máy tính và điều khiển học, hầu hết là bom bay có thể phát nổ ở khoảng cách mục tiêu nhất định, ví dụ, để sử dụng trong hệ thống định vị dẫn đường và điều khiển của radar. Tên lửa V1 và V2 có hệ thống "lái tự động" và hệ thống kích nổ, là tiền thân của đầu đạn hạt nhân tự dẫn hiện đại.

Năm 1961 đến năm 1963: Đại học Johns Hopkins phát triển "Beast". Beast sử dụng hệ thống định vị để di chuyển. Khi pin yếu, nó sẽ tự động tìm và cắm vào ổ cắm.

Năm 1969: Mowbot là Robot đầu tiên tự động cất cỏ. The Stanford Cart Line Follower là một Robot di động có thể điều hướng bằng cách phát hiện các đường màu trắng, sử dụng máy ảnh để quan sát. Nó bao gồm một "kênh vô tuyến" được kết nối với một máy tính trung tâm để thực hiện các phép tính.

Năm 1980: Thị hiếu tiêu dùng đối với Robot tăng lên, Robot dùng trong gia đình được mua và bán. Ví dụ, RB5X vẫn còn tồn tại cho đến ngày nay và một số mẫu robot HERO. Robot Stanford Cart đủ tiên tiến để điều khiển con tàu vượt qua các chương ngại vật và tạo bản đồ nơi nó sẽ đến.

Năm 1989: Mark Tinden phát minh ra robot BEAM.

Năm 1990: Cha đẻ của Robot công nghiệp, Joseph Engelberger, đã làm việc với các đồng nghiệp và phát minh ra robot y tế tự động do Helpmate tiếp thị.

Năm 1996-1997: NASA phóng Mars Pathfinder với 2 Robot Rovers và Sojourner lên Sao Hỏa. Mars Surface Rover được điều khiển từ mặt đất. Sojourner được trang bị hệ thống phòng ngừa rủi ro cao. Hệ thống này cho phép Sojourner điều hướng địa hình sao Hỏa một cách độc lập.

Năm 1999: Sony giới thiệu Aibo, một Robot có thể đi bộ, quan sát và tương tác với môi trường xung quanh. Robot quân sự điều khiển từ xa PackBot cũng được giới thiệu. 2001: Dự án Swaim-Bots được khởi động, Swaim-Bots như một bầy côn trùng. Chúng bao gồm một số lượng lớn các robot riêng lẻ có thể tương tác với nhau và cùng nhau hoàn thành các nhiệm vụ phức tạp.

Năm 2002: Roomaba, một robot nhà di động, đảm nhận công việc dọn dẹp. Khi tiếp tục phát triển, có rất nhiều loại robot phục vụ con người và dần tỏ ra thân thiện hơn.

Năm 2004: Robosapien, một robot đồ chơi do Mark Tilden thiết kế, được phát hành. Trong Dự án Centibots, 100 robot làm việc cùng nhau để tạo ra bản đồ của một khu vực không xác định và tìm các đối tượng trong môi trường đó. Trong Thử thách lớn đầu tiên của DARPA, các robot tự động đã cùng nhau cạnh tranh trên sa mạc.

Năm 2006: Sony ngừng sản xuất Aibo và Helpmate. PatrolBot trở nên phổ biến khi các robot di động tiếp tục cạnh tranh với nhau để trở thành sản phẩm độc quyền. Bộ An ninh Nội địa Hoa Kỳ đã từ bỏ dự án MDARS-I nhưng gây quỹ cho dự án MDARS-E, một loại robot an ninh tự động khác. TALON-Sword ra đời, một robot tự động có sẵn trên thị trường với súng phóng lựu và các tùy chọn kết hợp vũ khí khác. Asimo của Honda biết đi như người và leo cầu thang chỉ bằng hai chân.

Năm 2008: Boston Dynamics đã phát hành một video toàn cảnh về một thể hệ BigDog mới có thể đi trên mặt đất bằng giá và lấy lại thăng bằng khi bị va chạm từ bên cạnh.

Năm 2010: Thử thách chế tạo rô bốt tự động đa mục đích quốc tế có một đội xe tự hành lập bản đồ môi trường đô thị năng động rộng lớn, xác định và theo dõi con người cũng như các đối tượng thù địch.

Năm 2016: hướng dẫn cho rô bốt di động có thẻ RFID là một cách mới để sử dụng thẻ RFID. Người ta chứng minh rằng robot luôn đến mục tiêu với sai số đo khoảng cách, ngay cả trong các phép đo khoảng cách, ngay cả khi các phép đo khoảng cách và góc không chính xác. Anh ấy cũng có khả năng chọn con đường đúng theo nhiều cách khác nhau.

Năm 2017: ARGOS Challenge đã phát triển robot để sử dụng trong điều kiện khắc nghiệt tại các trạm dầu khí ngoài khơi..

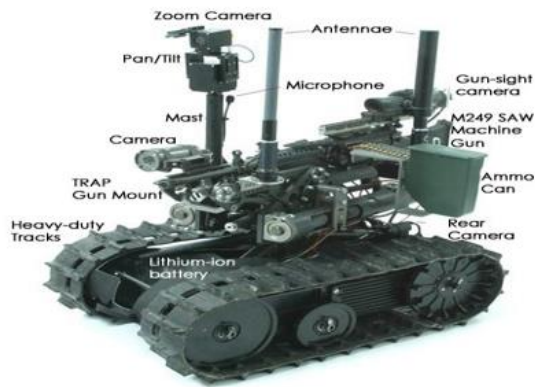
2.2. Phân loại

Mobile Robot có thể được phân loại thành:

- Môi trường mà chúng phục vụ:
 - *Robot trên đất liền hoặc robot sinh hoạt thường* được gọi là Phương tiện trên mặt đất không người lái (Unmanned Ground Vehicles -UGVs). Chúng thường có bánh xe hoặc bánh xích, nhưng cũng bao gồm Robot hai chân hoặc hơn nhiều chân (hình người, động vật hoặc côn trùng).
 - *Robot vận chuyển và giao hàng* có thể di chuyển vật liệu và vật tư tiêu hao trong môi trường làm việc nặng nhọc.
 - *Robot bay* thường được gọi là Máy bay không người lái (Unmanned Aerial Vehicles - UAVs)..
 - *Robot dưới nước* thường được gọi là phương tiện tự động dưới nước Water Robot (Autonomous Underwater Vehicles - AUVs).
 - *Các robot vùng cực*, được thiết kế để điều hướng trong các môi trường đầy băng giá được sử dụng nhiều tại các trạm nghiên cứu ở Bắc Cực và Nam Cực.
- Robot được sử dụng để di chuyển nhiều nhất chủ yếu là:
 - *Robot có chân(LEG)* : chân giống người (android) hoặc chân giống động vật.
 - *Robot có bánh xe vượt địa hình.*
 - *Robot có bánh ray bám đường.*
- Các loại điều khiển

Điều khiển từ xa bằng tay trên hệ thống máy tính hoặc giao diện điện thoại.

Robot điều khiển từ xa bằng tay được điều khiển thông qua sóng vô tuyến, Wifi, hồng ngoại hoặc bằng Bluetooth... Robot điều khiển từ xa giúp con người tránh ra khỏi những nguy hiểm. Ví dụ Robot điều khiển từ xa cầm tay bao gồm có: Robot quân sự Military-Robot và iRobot's PackBot...

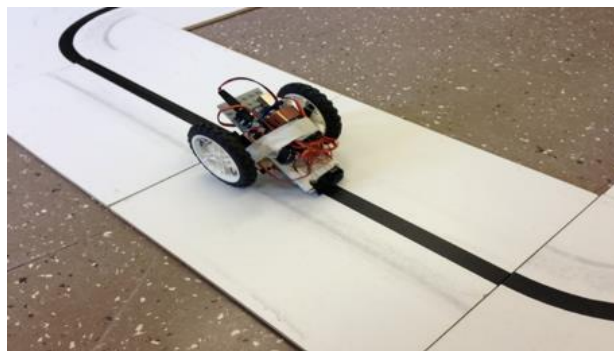


Hình 2. 1 Military Robot



Hình 2.2 iRobot 510 PackBot

Một số Robot tự động đầu tiên là những con Robot di chuyển trên theo lộ trình. Chúng có thể đi theo các đường được sơn khắc lên trên sàn trên trần nhà hay cáp điện trên sàn nhà. Hầu hết những Robots này hoạt động với một thuật toán đơn giản là theo dõi lộ trình trong bộ cảm biến trung tâm, chúng không thể tránh các chướng ngại vật, chúng chỉ dừng lại khi có thứ gì đó cản đường chúng. Khá nhiều mẫu của loại Robot này vẫn được bán bởi FMC, Egemin, HK-Sytem và một vài hãng nhỏ lẻ khác như Egemin và một vài công ty khác.



Hình 2.3 Robot dò line

Chuyển động ngẫu nhiên:

Robot tự hoạt động độc lập với những chuyển động ngẫu nhiên, về cơ bản đó là những chuyển động nảy bật trên tường, các bức tường được phát hiện do sự cản trở về mặt vật lý như là máy hút bụi Roomba hoặc với bộ nhận biết điện tử của máy cắt cỏ Friendly Robotics.

*Hình 2.2 Robot hút bụi Xiaomi Mijia Gen**Hình 2.3 Robomow RS622*

2.3. Đặt vấn đề

Trong thế kỷ XXI, cuộc sống con người ngày càng trở nên bận rộn để làm những công việc hỗ trợ trong cuộc sống của con người ngày càng thuận tiện và dễ hơn. Robot dịch vụ có thể mang lại sự tiện nghi cho người dùng bằng cách thực hiện các công việc nhàm chán, lặp đi lặp lại, nguy hiểm, không có tính nổi trội...

Tuy nhiên, dựa trên kết quả nghiên cứu và thực nghiệm về Mobile Robot đặc biệt các robot phục vụ dịch vụ chuyên nghiệp như: quét dọn nhà cửa, phục vụ công việc trong gia đình, nội trợ, quản lý không gian, giúp đỡ cho con người,... còn chưa tối ưu và hiệu quả. Điều này là do quá trình nghiên cứu còn hạn chế, chúng ta vừa mới bắt đầu phát triển Robot dịch vụ trong thời gian ngắn, lực lượng nghiên cứu về lĩnh vực này hiện tại còn rất mỏng tại Việt Nam.

Vì vậy, ngoài những nghiên cứu đầu tiên này, vẫn chưa có một nghiên cứu có tính hệ thống hoặc là một sản phẩm Mobile Robot hoàn chỉnh đáp ứng được yêu cầu trong khi nhu cầu thật sự về Robot dịch vụ là có sự an toàn và cấp thiết.

Home Robot là loại robot an toàn với người dùng, tiết kiệm và dễ sử dụng giúp con người dùng thực hiện một số công việc gia đình. Mặc dù mục tiêu cuối cùng của Home Robot là một Robot thông minh giúp việc dọn dẹp trong gia đình và thay thế công việc đa năng của một “ô sin” thì thực tế phát triển của Home Robot hiện nay chỉ là những Robot nhỏ lẻ và chỉ làm một việc đơn nhiệm như hút bụi, lau dọn nhà, lau cửa kính hoặc cắt cỏ. Đây là một thị trường đã từng bị quên lãng ở Mỹ nhưng lại được Nhật quan tâm rất nhiều. Trong khi chính phủ Mỹ phân bổ rất nhiều tiền cho việc phát triển các robot quân sự và không quan tâm đến phát triển Home Robot thì ngược lại, Nhật Bản lại tập trung nguồn lực cho R&D của Service Robot dân sự hơn là quân sự. Nếu tình trạng này tiếp tục kéo dài thì Nhật Bản sẽ là quốc gia thống trị thị trường Home

Robot trên thế giới và bỏ xa Mỹ ở lại phía sau. Ở Mỹ, công nghệ robot được coi là tài sản tuyệt đối của quốc gia với nhiều bí mật quân sự và chỉ sau này mới được chuyển giao cho các lợi ích dân sự. Trong khi đó, Nhật Bản coi công nghệ Robot là nền tảng cho sự cạnh tranh quốc tế của Nhật Bản trong tương lai. Chính phủ Nhật Bản ước tính ngành công nghiệp robot sẽ phát triển mạnh mẽ và nước Nhật sẽ đi lên với sự thành công của Home Robot. Chỉ có một loại Home Robot của Mỹ thành công trên thị trường đó là robot hút bụi iRobot Roomba. Giá thành của Roomba Robot có giá chỉ vài trăm USD nhưng hoạt động khá tốt và dễ sử dụng. Các loại robot gia dụng trước Roomba hoặc là quá mức phức tạp như robot công nghiệp hoặc là quá đơn giản như các loại robot đồ chơi. Trên thực tế khá nhiều robot công nghiệp được nghĩ là phức tạp lại thì lại kém hơn nhiều so với Roomba Robot.

Nhật Bản có đủ động lực để phát triển ngành công nghiệp robot. Với hơn 1/5 dân số Nhật Bản có tuổi thọ trên 65 tuổi, nước Nhật đang dựa vào robot để thay thế nguồn nhân lực trẻ và chăm sóc người già. Vì vậy, đối với nước Nhật, một cuộc cách mạng robot là tiên quyết và bắt buộc. Hàn Quốc cũng là quốc gia đặt mục tiêu phát triển rất nhiều cho Home Robot với giá thành rẻ để mỗi gia đình Hàn Quốc sẽ có một Home Robot trong thập kỷ này.

Mặc dù thị trường của Home Robot hiện nay còn nhỏ nhưng một khi ngành công nghiệp này khởi sắc nó sẽ có sức tăng trưởng mạnh và trở thành một lĩnh vực rất hữu ích và rất sinh lợi.

Khả năng định hướng trong không gian:

Các vấn đề về điều hướng hiện nay được phân chia thành 2 loại: định hướng trong nhà (Indoor) và định hướng ngoài trời, địa hình (Odoor). Quá trình định hướng cho Mobile Robot gồm 4 bước: nhận diện cảnh quan môi trường, xác định vị trí đứng, thiết kế quỹ đạo và tạo chuyển động cho Robot. Với môi trường có cấu trúc, quy trình nhận biết cho phép tạo ra bản đồ hoặc mô hình không gian phục vụ cho bài toán định vị Robot và thiết kế quỹ đạo robot. Đối với môi trường phi cấu trúc không ổn định, robot phải có khả năng học cách xác định được hướng đi của chính mình bằng cách xác định môi trường. Do đó, lĩnh vực xác định hướng đi cho Mobile Robot là một lĩnh vực có thể áp dụng các phương pháp trí tuệ nhân tạo như quá trình phát hiện môi trường, suy diễn và

tìm hướng đi tối ưu có thể được áp dụng. Vấn đề xác định vị và lập bản đồ đã từng là những chủ đề nghiên cứu trọng tâm ở Mobile Robot trong quá khứ.

Quá trình định vị là quá trình Robot xác định được vị trí hiện tại của nó đang ở đâu trong không gian làm việc. Để đạt được mục tiêu này, một số cảm biến phải được triển khai để thu thập dữ liệu về trạng thái của robot và môi trường xung quanh nó. Dữ liệu này thường bị nhiễu và có sai số tích lũy, đòi hỏi kỹ thuật lọc động và sử dụng kỹ thuật phối hợp cảm biến để có được các phép đo chính xác hơn. Phương pháp định vị có thể là cục bộ hoặc toàn cục. Phương pháp đơn giản nhất là định vị cục bộ vì Robot cập nhật định kỳ vị trí của nó so với điểm xuất phát. Ngược lại, các phương pháp định vị toàn cục không yêu cầu kiến thức về vị trí của điểm xuất phát. Để khắc phục sự không chắc chắn của thông tin được đo bởi các cảm biến, chúng ta cần sử dụng phương pháp xác suất. Các phương pháp định vị được sử dụng thường dựa trên nguyên lý bộ lọc Bayes kết hợp với thuật toán đệ quy để ước tính vị trí và hướng từ phương trình mô tả chuyển động của robot. Thời gian tính toán của phép lọc Bayes là lâu, vì vậy nhiều nghiên cứu gần đây đã tập trung vào việc tìm ra các phương pháp đơn giản hóa để giảm khối lượng tính toán. Hơn trong số những đơn giản hóa này dẫn đến nhiều thuật toán định vị khác nhau, được chia thành hai loại tùy thuộc vào cách mô tả độ tin cậy của dữ liệu. Nếu dữ liệu được mô tả bằng các hàm phân phối Gaussian, chúng ta có thể sử dụng phương pháp lọc Kalman. Nếu dữ liệu được mô tả bởi nhiều hàm phân phối xác suất khác nhau, chúng ta có thể sử dụng các thuật toán vị trí dựa trên quy trình Markov. Phương pháp định vị dựa trên phân phối Gaussian và bộ lọc Kalman chỉ có hiệu quả đối với bài toán định vị cục bộ. Phương pháp vị trí Markov có thể là phương pháp tô pô, phương pháp mạng tinh thể và phương pháp sử dụng các mẫu rời rạc của các giá trị trạng thái. Nếu dữ liệu robot được mô tả bằng các mẫu ngẫu nhiên rời rạc, chúng ta có thể sử dụng kỹ thuật lọc hạt để giúp xác định tốt hơn vị trí của Robot.

2.4. Giới hạn đề tài

Mục tiêu của nhóm là làm một chiếc **Mobile Robot** trong nhà (indoor) có khả năng phục vụ con người về các công việc đơn giản trong nhà: lau nhà, hút bụi, quản lý không gian, an ninh trong nhà,...

Điều khiển được động cơ và thu được giá trị từ các cảm biến hỗ trợ cho Mobile Robot.

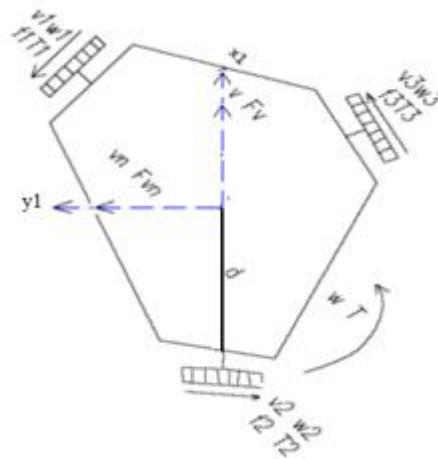
Vì lý do dịch bệnh mọi người không thể làm việc trực tiếp với nhau vì vậy việc làm đồ án môn học đối với chúng em rất khó khăn mong thầy cô có thể thông cảm cho tụi em.

CHƯƠNG 3: CƠ SỞ LÝ THUYẾT

3.1. Lựa chọn cơ cấu chuyển động cho Mobile Robot

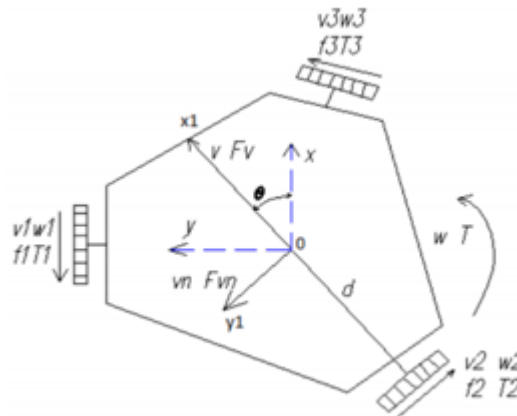
Chuyển động của robot là do chuyển động được tạo ra bởi các bánh xe. Các bánh xe chuyển động do động cơ lắp ở mỗi trục bánh quay và sinh ra lực kéo. Vậy ta xét đến cấu tạo của robot, trong đó các lực kéo đặt ở các bánh xe. Vận tốc của robot được tính dựa vào vận tốc của các bánh xe. Từ đó ta có mô hình robot như sau đây:

Khi robot chưa quay:



Hình 3.1 Mô hình động học và động lực học của Robot

Khi robot quay một góc θ :



Hình 3.2 Mô hình động học và động lực học của Robot khi quay một góc θ

Robot gồm 3 bánh đặt lệch nhau góc 120° , tiếp tuyến với một đường tròn có đường kính 350mm. Gọi các bánh xe lần lượt là 1, 2 và 3. Chọn 2 hệ trục tọa độ như sau: hệ trục tọa độ Oxy gắn cố định với mặt phẳng sàn, hệ trục tọa độ $O_{x_1y_1}$ gắn với robot như hình vẽ. Tại thời điểm ban đầu, đặt một giả thiết rằng hai hệ trục tọa độ này trùng với

nhau. Tại thời điểm t , hai hệ trục tọa độ lệch nhau góc θ . Đây chính là góc quay của robot so với vị trí ban đầu.

Sau đây là các kí hiệu dùng trong hình vẽ:

- Vị trí robot trong hệ trục tọa độ Oxy theo phương Ox: x .
- Vị trí robot trong hệ trục tọa độ Oxy theo phương Oy: y .
- Khoảng cách giữa mỗi bánh và tâm robot: d [m] (Hình 2.4).
- Vận tốc dài của các bánh xe: v_1, v_2, v_3 [m/s].
- Vận tốc góc của bánh: $\omega_1, \omega_2, \omega_3$ [rad/s].
- Lực do động cơ sinh ra trên các bánh: f_1, f_2, f_3 [N].
- Mô men kéo do động cơ sinh ra đặt lên các bánh : T_1, T_2, T_3 [Nm].
- Vận tốc theo 2 phương trên hệ $O_{x_1y_1z_1}$ của robot: v, v_n, ω_1 [m/s] (z_1 trùng z).
- Vận tốc góc của robot: ω [rad/s].

3.2. Microcontroller Unit STM32F103C8T6

3.2.1 Giới thiệu về STM32F103C8T6

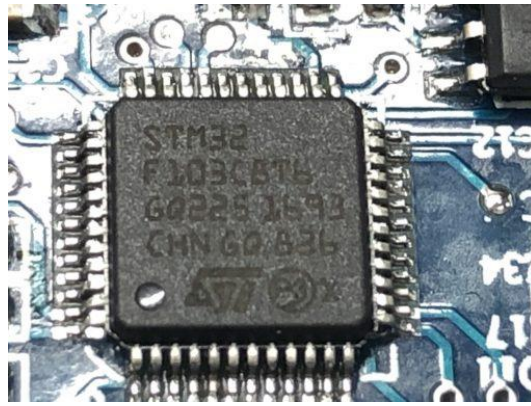
STM32 là một trong những dòng chip vi xử lý phổ biến của ST với nhiều họ thông dụng như F0, F1, F2, F3, F4..... STM32F103 thuộc họ F1 với lõi là ARM COTEX M3. STM32F103 là vi điều khiển 32bit, tốc độ tối đa là 72Mhz. Giá thành khá rẻ nếu so với các dòng vi điều khiển có chức năng tương tự. Mạch nạp cũng như công cụ lập trình đa dạng và dễ sử dụng.

STM32 được ứng dụng chính: dùng cho driver nhằm điều khiển ứng dụng, thiết bị cầm tay, máy tính và phụ kiện, GPS, các ứng dụng trong công nghiệp, PLC, biến tần, máy quét, hệ thống cảnh báo, thiết bị liên lạc nội bộ (bộ đàm),...

Phần mềm lập trình: IAR Embedded Workbench, Keil C,...

Thư viện lập trình: nguồn thư viện lập trình phong phú cho STM32 như: STM32Cube HAL, STM32snippets, STM32Cube LL, Mbed core. Standard Peripheral Libraries. Mỗi thư viện đều có ưu điểm và khuyết điểm riêng, ở đây nhóm xin phép sử dụng Standard Peripheral Libraries vì bộ thư viện này đã ra đời khá lâu và thông dụng, hỗ trợ nhiều ngoại vi và dễ hiểu bản chất của lập trình.

Mạch nạp: có khá nhiều loại mạch nạp như : ULINK, J-LINK , CMSIS-DAP, STLINK... ở đây nhóm sử dụng Stlink vì giá thành khá rẻ và debug lỗi cũng tốt.



Hình 3.3 STM32F103C8T6

3.2.2. Cấu hình chi tiết của STM32F103C8T6

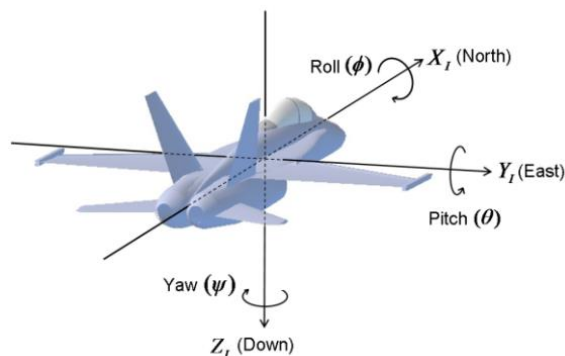
- ARM 32-bit Cortex M3 với tần số clock tối đa là 72Mhz.
- Bộ nhớ:
 - 64 kbytes bộ nhớ Flash (bộ nhớ lập trình).
 - 20kbytes SRAM.
- Tần số clock, reset và quản lý nguồn.
 - Điện áp hoạt động 2.0V -> 3.6V.
 - Programmable voltage detector (PVD), Power on reset (POR), Power down reset (PDR).
 - Sử dụng thạch anh ngoài từ 4Mhz -> 20Mhz.
 - Thạch anh nội dùng dao động RC ở mode 8Mhz hoặc 40khz.
 - Nguồn thạch anh ngoài 32.768kHz được sử dụng cho RTC.
- Trường hợp điện áp thấp:
 - Mode: ngủ, ngừng hoạt động hoặc hoạt động ở chế độ chờ.
 - Cấp nguồn ở chân Vbat bằng pin để hoạt động bộ RTC và sử dụng lưu trữ data khi mất nguồn cấp chính.
- 2 bộ ADC 12bit trong đó mỗi bộ có 9 kênh.
 - Khoảng giá trị chuyển đổi từ 0 – 3.6V.
 - Lấy mẫu nhiều kênh hoặc 1 kênh.
 - Có cảm biến nhiệt độ nội.
- DMA: bộ chuyển đổi giúp tăng tốc độ xử lý chương trình do không có sự can thiệp quá sâu từ CPU.
 - 7 kênh DMA.

- Hỗ trợ DMA cho ADC, I2C, SPI, UART.
- 7 bộ Timer.
 - 3 timer 16 bit hỗ trợ các mode IC/OC/PWM.
 - 1 timer 16 bit hỗ trợ để điều khiển động cơ với các mode bảo vệ như ngắt input, dead-time..
 - 2 watchdog timer dùng để bảo vệ và kiểm tra lỗi.
 - 1 sysTick timer 24 bit đếm xuống dùng cho các ứng dụng như hàm Delay....
- Hỗ trợ 9 kênh giao tiếp bao gồm:
 - 2 bộ I2C(SMBus/PMBus).
 - 3 bộ USART(ISO 7816 interface, LIN, IrDA capability, modem control).
 - 2 SPIs (18 Mbit/s).
 - 1 bộ CAN interface (2.0B Active)
 - USB 2.0 full-speed interface
- Kiểm tra lỗi CRC và 96-bit ID.

3.3. Cảm biến gia tốc, con quay đàn hồi chuyển và từ kế

3.3.1. Giới thiệu về IMU

Inertial Measurement Unit - Đơn vị đo lường quán tính cung cấp thông tin về vận tốc góc và góc nghiêng của hệ thống. Dựa trên nguyên lý hoạt động của 2 cảm biến gia tốc kế (gia tốc kế) và con quay hồi chuyển (gyroscope). Tuy nhiên, khái niệm này được sử dụng khá lỏng lẻo, vì vậy IMU có thể được coi như một đơn vị đo lường quán tính của hệ thống, khi được xử lý, cung cấp giá trị cảm biến để giúp hệ thống hoạt động tốt nhất.



Hình 3.4 Các giá trị của IMU

3.3.2. Ứng dụng của IMU

Cảm biến IMU được sử dụng trong các ứng dụng như rô bốt tự cân bằng, máy bay quadcopters, điện thoại thông minh, v.v. Cảm biến IMU giúp chúng ta xác định vị trí của đối tượng được gắn vào cảm biến trong không gian ba chiều. Chúng được sử dụng để phát hiện hướng của điện thoại thông minh hoặc trong các thiết bị như Fitbit sử dụng cảm biến IMU để theo dõi chuyển động. Module cảm biến IMU có thể được tích hợp gồm hai hoặc nhiều cảm biến như:

- Cảm biến gia tốc (acceleometer)
- Cảm biến góc quay (gyroscope)
- Cảm biến từ trường (magnetometer)
- Áp suất kế để đo độ cao (altimeter)

Từ các giá trị đọc được từ cảm biến, thông thường sẽ cần phải cần các thuật toán hiệu chuẩn, lọc và sao lưu để tạo ra các giá trị cảm biến có thể sử dụng cho nhu cầu của ứng dụng cảm biến trong từng trường hợp cụ thể.

3.3.3. Nguyên lý hoạt động của từng loại

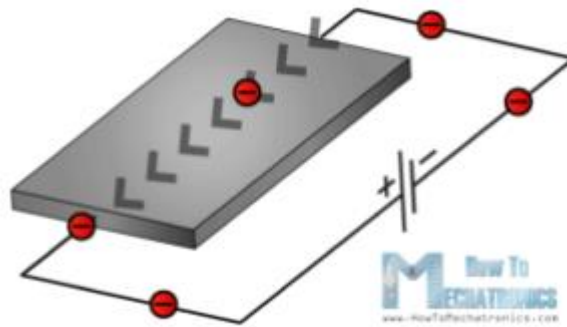
- Cảm biến gyroscope:

Nếu bạn đặt chip IMU ở vị trí cố định, giá trị trả về là $Gyro = [0.0, 0.0, 0.0]$ vì không có vòng quay nào cả. Con quay hồi chuyển chỉ đo tốc độ quay, không đo trực tiếp góc quay. Vì vậy, nếu bạn quay mô-đun một góc nhất định và sau đó dừng lại, giá trị của con quay hồi chuyển sẽ tăng và sau đó giảm xuống 0.

- Cảm biến gia tốc kế:

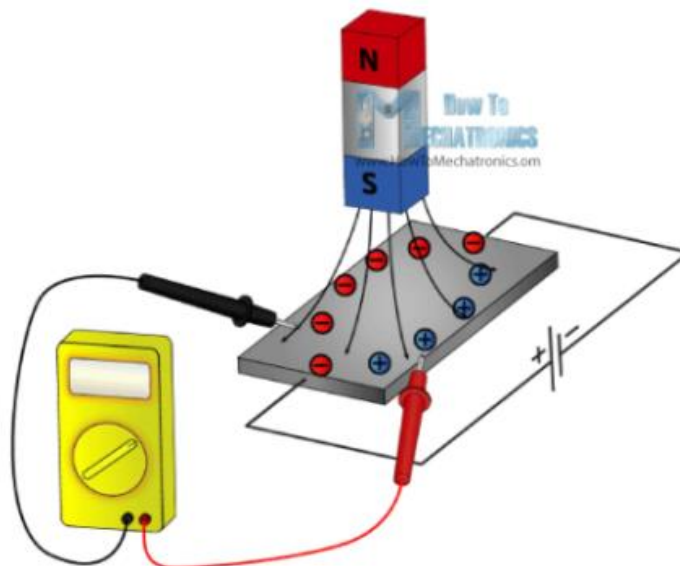
Nguyên tắc để đo góc dùng gia tốc kế là phân tích sự tác dụng của trọng lực lên các trục ta có thể dùng nó để tính các góc lệch roll và pitch.

- Cảm biến từ trường:



Hình 3.5 Cảm biến từ trường

Cảm biến từ trường hầu như hoạt động dựa trên hiệu ứng Hall. Ta cấp nguồn cho dòng electron chạy qua mạch điện, có một tấm dẫn điện như hình.



Hình 3.6 Hiệu ứng Hall

Khi có từ trường bao quanh, lực Loren làm cho dòng electron trong mạch chuyển động theo hai hướng. Nếu chúng ta đo hiệu điện thế, chúng ta biết kích thước của nguồn từ trường này. Nếu trục z được chọn làm trục vuông góc với mặt đất, góc yaw là góc khi trục z được quay. Nó thường được sử dụng để đo góc hàm với một công thức đơn giản. Trên thực tế, chúng ta có thể tính toán góc hàm bằng con quay hồi chuyển, nhưng nếu chúng ta kết hợp một cảm biến từ tính, nó sẽ cho kết quả chính xác hơn và loại bỏ nhược điểm của việc sử dụng con quay hồi chuyển.

- Cảm biến áp suất kế (Barometer):

Barometer tính toán độ cao bằng cách sử dụng tính chất độ cao thay đổi theo áp suất, càng lên cao áp suất càng giảm. Ta chỉ cần áp dụng công thức để chuyển đổi.

3.3.4. Cảm biến là loại phổ biến nhất cho các ứng dụng hiện nay

- MPU6050 accelerometer and gyroscope.
- Magnetometer HMC5883L.
- Barometer MS5611.

3.4. Cảm biến siêu âm đo khoảng cách

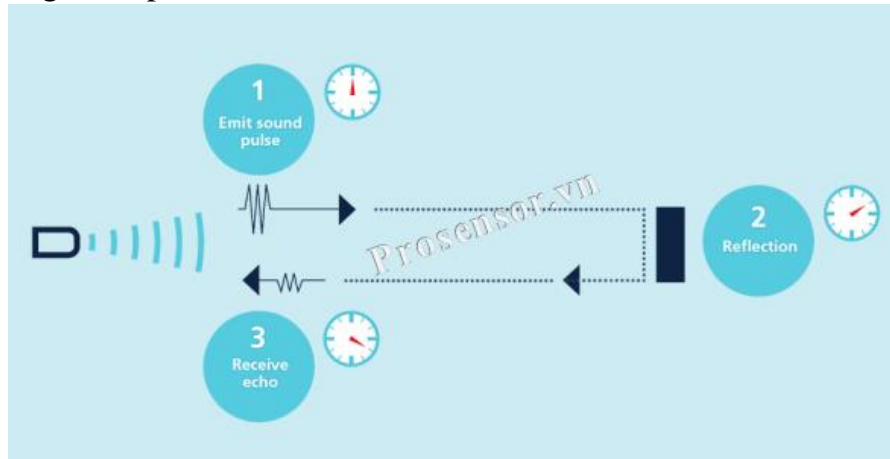
3.4.1 Giới thiệu về cảm biến siêu âm

Cảm biến siêu âm là một thiết bị đo nhỏ gọn bao gồm một máy phát siêu âm và một mô-đun điện tử để đo liên tục hoặc phát hiện mức giới hạn của chất lỏng, khoảng cách đến chướng ngại vật, nước thải, bùn, chất rắn, v.v. Chất kết dính, nhựa trong các mạch đóng và mở khác nhau, vân vân.

3.4.2 Nguyên lý hoạt động của cảm biến siêu âm

Cảm biến siêu âm phát ra các xung âm thanh tần số cao, ngắn với tốc độ không đổi. Chúng bay trong không khí với tốc độ âm thanh. Nếu chúng va vào một vật thể, chúng sẽ bị phản xạ trở lại cảm biến dưới dạng tiếng vang. Sau đó, cảm biến sẽ tính toán phạm vi tới mục tiêu dựa trên thời gian trôi qua từ khi gửi tín hiệu đến khi nhận được tiếng vọng. Vì khoảng cách đến một vật thể được xác định bằng cách đo thời gian truyền và nhận của một tiếng vang chứ không phải bởi cường độ của âm thanh, nên cảm biến siêu âm rất tốt trong việc triệt tiêu nhiễu. Hầu như tất cả các vật liệu phản xạ âm thanh có thể được phát hiện bất kể màu sắc của chúng. Ngay cả vật liệu trong suốt hoặc lá mỏng cũng không có vấn đề gì đối với cảm biến siêu âm. Cảm biến siêu âm có thể nhìn xuyên qua không khí đầy bụi và sương mù. Có thể đo mức trong giếng và ống nghiệm,

cũng như tạo điều kiện phát hiện các chai nhỏ trong ngành đóng gói. Ngay cả những sợi cáp mỏng cũng được phát hiện chính xác.



Hình 3.7 Nguyên lý hoạt động của cảm biến siêu âm

3.5. Bộ lọc Kalman

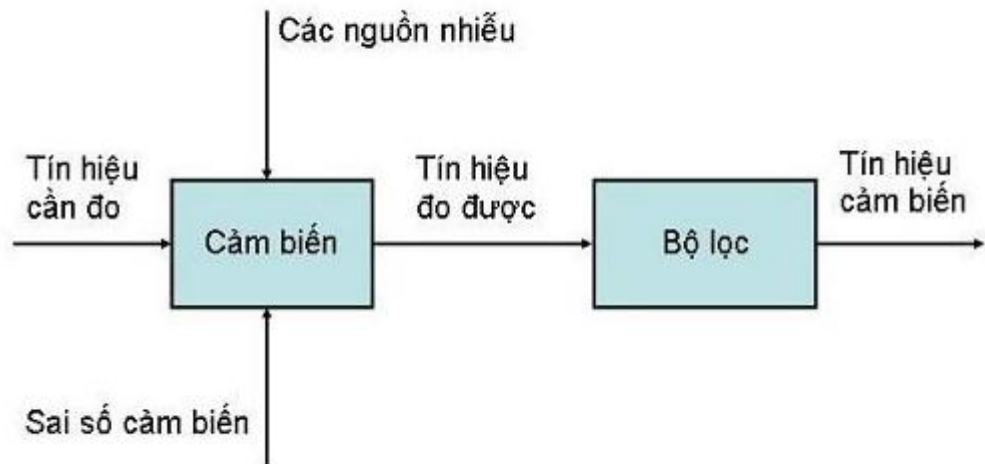
3.5.1 Bộ lọc là gì?

Đầu tiên, lọc là một quá trình loại bỏ những gì không có giá trị hoặc lợi ích và giữ lại những gì có giá trị. Tiếng anh gọi là filter (giống như bộ lọc để pha cà phê). Người ta có thể tưởng tượng rằng bộ lọc cà phê cũng đảm nhận công việc của bộ lọc, lọc bỏ caffeine trộn với nước và giải phóng cà phê khỏi cơ thể. Về xử lý tín hiệu, Beetle được thiết kế để lọc tín hiệu sạch. (phải được tìm thấy) Từ trong tín hiệu trộn lẫn giữa tín hiệu sạch và nhiều tín hiệu bẩn (không cần thiết).

3.5.2. Giới thiệu bộ lọc Kalman

Trước tiên tên Kalman là tên của người nghĩ ra cái bộ lọc đấy. Vào năm 1960, R.E Kalman đã công bố bài báo nổi tiếng về một giải pháp truy hồi để giải quyết bài toán lọc thông tin rời rạc tuyến tính (discrete data linear filtering). Tên đầy đủ của bài báo là "A New Approach to Linear Filtering and Prediction Problems". Từ đó đến nay cùng với sự phát triển của tính toán kỹ thuật số, bộ lọc Kalman đã trở thành chủ đề nghiên cứu sôi nổi và được ứng dụng trong nhiều ngành kỹ thuật công nghệ khác nhau : trong tự động hóa, trong định vị cũng như trong viễn thông (và nhiều lĩnh vực khác nữa).

3.5.3. Bản chất bộ lọc Kalman



Hình 3.8 Bản chất bộ lọc Kalman

Thực ra nó cũng chỉ đơn giản là từ một tín hiệu cần đo, khi chúng ta đo, sẽ có những sai số từ cảm biến, ảnh hưởng đến tín hiệu cần đo, môi trường đo có nhiễu... Tất cả những thứ này, tổng hợp lại, sẽ cho ta một kết quả đo. Học mạch lọc Kalman như thế nào? Đây là một câu hỏi lớn không lớn, nhỏ không nhỏ, nhưng tính quan trọng của nó dành cho những người mới học là không thể chối cãi. Tuy có thể chưa thấu đáo, tường tận, song hy vọng nó có thể giúp các bạn có những ý tưởng cơ bản về mạch lọc Kalman, và bắt đầu nghiên cứu nó một cách phổ biến. Một trong những kiến thức nền tảng để có thể hiểu về nguyên lý của bộ lọc Kalman chính là xác suất.

**** theo <https://tailieumienphi.vn/doc/bo-loc-kalman-3d9ptq.html> - Bộ lọc Kalman,

CHƯƠNG 4: THIẾT KẾ CƠ KHÍ CHO MOBILE ROBOT

4.1. Một số nghiên cứu về Omni Robot trên thế giới và Việt Nam

Trên thế giới, Omni robot đã được phát triển từ rất lâu và ngày càng xuất hiện nhiều ứng dụng đa dạng do được tích hợp các công nghệ mới nhất.

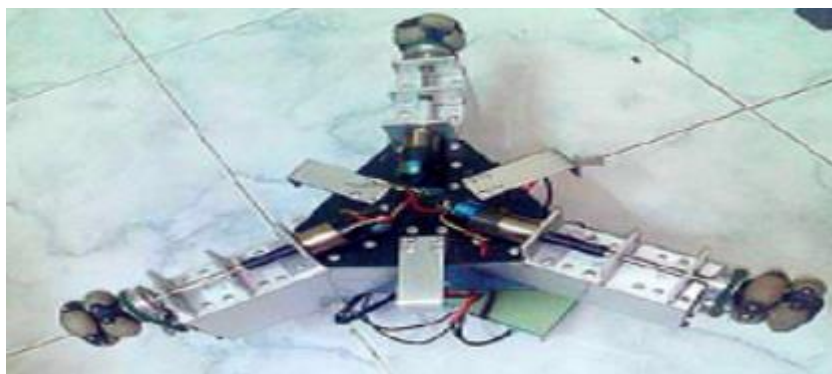


Hình 4.1 WowWee Rovio Robot

Tính năng:

- Có thể điều khiển qua mạng wifi.
- Có thể điều khiển qua hệ thống internet.
- Có thể tự di chuyển xung quanh nhà do có nguồn dự trữ.

Ở các trường đại học, Những nghiên cứu về Omni Robot ở Việt Nam đã được thực hiện trong thời gian gần đây bởi các sinh viên và giáo viên. Những nghiên cứu này đã đạt được những kết quả ban đầu, tạo ra được sản phẩm thực nghiệm tốt.



Hình 4.2 Robot Omni do đại học bách khoa thành phố Hồ Chí Minh thiết kế



Hình 4.3 Robot SPARK IV

Omni Robot tránh vật cản sử dụng 3 cảm biến hồng ngoại do nhóm sinh viên đại học bách khoa Hà Nội chế tạo.

Mục đích của hệ thống:

Dựa theo sơ đồ cấu trúc Mobile Robot của các nhóm nghiên cứu trước đã thực hiện được bám đối tượng và tránh chướng ngại vật trong quá trình hoạt động, đồ án tính toán động học và quỹ đạo chuyển động cho robot để xác định được thuật toán điều khiển cho từng quỹ đạo.

4.2. Thiết kế kỹ thuật cơ khí

4.2.1. Tính toán động học cho robot

Gọi $v_x(t)$, $v_y(t)$, $\omega(t)$ là vận tốc tức thời của robot trên hệ trục tọa độ Oxy.

$$\begin{aligned} v_x(t) &= \frac{dx(t)}{dt} \\ v_y(t) &= \frac{dy(t)}{dt} \\ \omega(t) &= \frac{d\theta(t)}{dt} \end{aligned} \quad (4.1)$$

Chiếu các vận tốc trên lên hệ trục tọa độ $O_{x_1y_1}$ gắn với robot, ta có:

$$\begin{aligned} v(t) &= v_x(t) \cos(\theta(t)) + v_y(t) \sin(\theta(t)) \\ v_n(t) &= -v_x(t) \sin(\theta(t)) + v_y(t) \cos(\theta(t)) \\ \omega_1(t) &= \omega(t) \end{aligned} \quad (4.2)$$

Viết lại các công thức trên dưới dạng ma trận, ta được công thức, ta nhận được công thức liên hệ giữa vận tốc theo 2 hệ trục tọa độ khác nhau:

$$\begin{bmatrix} v(t) \\ vn(t) \\ \omega_1(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix} \quad (4.3)$$

Xét trong hệ trục gắn với robot, $O_{x_1y_1}$, giả sử robot chuyển động với các vận tốc là $v(t)$, $vn(t)$ và $\omega(t)$, ta có công thức tính vận tốc của các bánh là:

$$\begin{aligned} v_1(t) &= -v(t)\sin(\pi/3) + vn(t)\cos(\pi/3) + \omega_1(t)d \\ v_2 &= -vn(t) + \omega(t)d \end{aligned} \quad (4.4)$$

$$v_3(t) = v(t)\sin(\pi/3) + vn(t)\cos(\pi/3) + \omega_1(t)d$$

Viết lại hệ phương trình (4.4) ta thu được dạng ma trận:

$$\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} -\sin(\pi/3) & \cos(\pi/3) & d \\ 0 & -1 & d \\ \sin(\pi/3) & \cos(\pi/3) & d \end{bmatrix} \begin{bmatrix} v(t) \\ vn(t) \\ \omega_1(t) \end{bmatrix} \quad (4.5)$$

Giải hệ phương trình trên ta có:

$$\begin{aligned} v(t) &= \frac{\sqrt{3}}{3}(v_3(t) - v_1(t)) \\ vn(t) &= \frac{1}{3}(v_3(t) + v_1(t)) - \frac{2}{3}v_2(t) \\ \omega_1(t) &= \frac{1}{4d}(v_1(t) + v_2(t) + v_3(t)) \end{aligned} \quad (4.6)$$

Các công thức trên cho phép ta tính vận tốc của các robot(vận tốc theo các phương của trục tọa độ và vận tốc robot) khi đã biết vận tốc các bánh xe.

4.2.2. Tính toán động lực học cho robot

Theo phương trình các phương trình tính gia tốc và gia tốc góc cho một vật

$$M\dot{v} = F$$

$$J\dot{\omega} = T$$

Ta có:

$$\begin{aligned} m \frac{dv(t)}{dt} &= F_v(t) - F_{Bv}(t) - F_{Cv}(t) \\ m \frac{dvn(t)}{dt} &= F_{vn}(t) - F_{Bvn}(t) - F_{Cvn}(t) \end{aligned} \quad (4.7)$$

$$J \cdot \frac{d\omega(t)}{dt} = T(t) - T_{B\omega}(t) - T_{C\omega}(t)$$

Trong đó:

$m[\text{kg}]$: Khối lượng robot

$J[\text{kgm}^2]$: Mô men quán tính của robot

$F_{Bv}, F_{Bvn}[\text{N}]$: Lực ma sát nhớt dọc theo phương Ox_1 và Oy_1 tác dụng lên robot

$T_{B\omega}[\text{Nm}]$: Mô men ma sát nhớt tác dụng lên robot

$F_{Cv}, F_{Cvn}[\text{N}]$: Lực ma sát khô dọc theo phương Ox_1 và Oy_1 tác dụng lên robot

$T_{C\omega}[\text{Nm}]$: Mô men ma sát khô tác dụng lên robot

Lực ma sát nhớt tỉ lệ với vận tốc của robot, xem hình 4.4.:

$$F_{Bv}(t) = B_v v(t)$$

$$F_{Bvn}(t) = B_{vn} vn(t)$$

$$F_{Bvn}(t) = B_{\omega} \omega(t)$$

$B_v, B_{vn}[\text{N}/(\text{m/s})]$: Hệ số ma sát nhớt

$B_{\omega}[\text{N}/(\text{rad/s})]$: Hệ số ma sát nhớt do vận tốc quay

Độ lớn của lực ma sát khô là hằng số, xem hình 4.4:

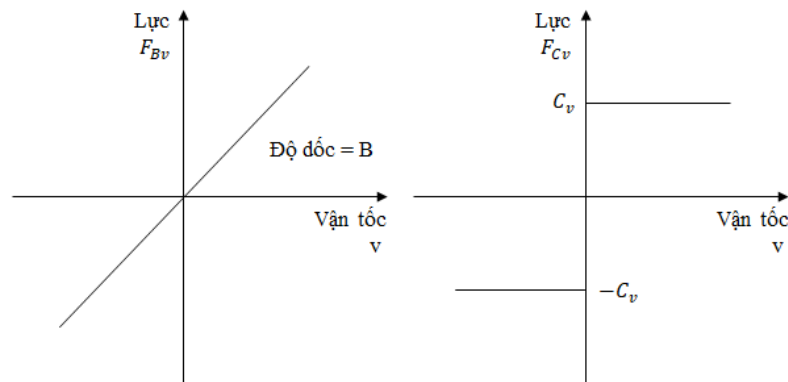
$$F_{Cv}(t) = C_v \text{sign}(v(t))$$

$$F_{Cvn}(t) = C_{vn} \text{sign}(vn(t)) \quad (4.8)$$

$$T_{C\omega}(t) = C_{\omega} \text{sign}(\omega(t))$$

$C_v, C_{vn}[\text{N}]$: Hệ số ma sát khô dọc theo phương Ox_1 và Oy_1

$C_{\omega}[\text{Nm}]$: Hệ số ma sát khô theo ω



Hình 4.4 Hệ số ma sát khô và hệ số ma sát ướt

Mối quan hệ giữ lực kéo của robot, mô men quay và lực quay trên mỗi bánh được đưa vào công thức sau:

$$F_v(t) = (f_3(t) - f_1(t)) \sin \frac{\pi}{3}$$

$$F_{vn}(t) = -f_2(t) + (f_3(t) + f_1(t)) \cos \frac{\pi}{3}$$

$$T(t) = (f_1(t) + f_2(t) + f_3(t))d$$

Lực tác dụng lên robot dọc theo phương O_{x_1} và O_{y_1} : F_v, F_{vn} [N]

Mô men robot (theo ω): T [Nm]

Lực kéo trên mỗi bánh được ước lượng theo mô men kéo có thể xác định được bằng cách sử dụng dòng điện của động cơ, được miêu tả bằng công thức dưới đây :

$$f(t) = \frac{T(t)}{r} \quad (4.11)$$

$$T(t) = K \cdot i(t)$$

r: bán kính bánh xe

l : Hệ số của hộp tốc độ

r[m]: Bán kính bánh xe

$K_t [\frac{Nm}{A}]$: Hằng số mô men động cơ

i_j [A] : Dòng động cơ

4.2.3. Tính toán cho động cơ

$$u_j(t) = L \frac{di_j(t)}{dt} + Ri_j(t) + K_v \omega_{mj}(t) \quad (4.12)$$

$$T_j(t) = K_t i_j$$

L[H]: Độ tự cảm của động cơ

R[Ω]: Điện trở của động cơ

K_v [V/(rad/s)]: Hằng số suất điện động ngược

u_j [V]: Điện áp động cơ

ω_{mj} [rad/s]: Vận tốc quay của động cơ

T_{mj} : Mô men động cơ thứ j

Trong trạng thái ổn định, độ tự cảm L bằng 0, nên ta có thể viết lại công thức:

$$\begin{aligned} u_j(t) &= R i_j(t) + K_v \omega_{mj}(t) \\ \frac{u_j(t)}{i_j(t)} &= R + \frac{K_v \omega_{mj}(t)}{i_j(t)} \end{aligned} \quad (4.13)$$

Kết hợp công thức ở trên ta có công thức:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) + K \text{sign}(x) \quad (4.14)$$

$$x(t) = [v(t) \quad v_n(t) \quad \omega(t)] \quad (4.15)$$

Công thức này trình bày cách tính gia tốc của robot theo phương v , v_n và gia tốc góc cho robot.

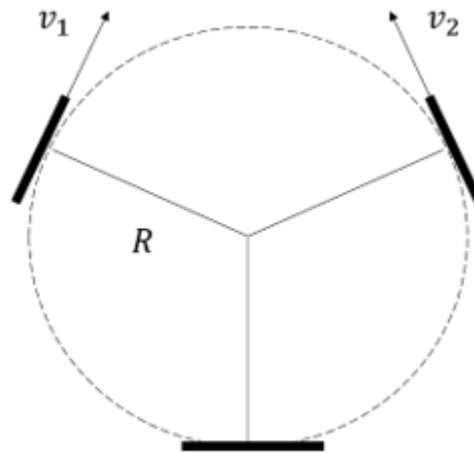
Trong đó ta có:

$$\begin{aligned} A &= \text{diag}(A_{11}, A_{22}, A_{33}) \\ A_{11} &= -\frac{3K_t^2 l^2}{2r^2 RM} - \frac{B_v}{M} \\ A_{22} &= -\frac{3K_t^2 l^2}{2r^2 RM} - \frac{B_{v_n}}{M} \\ A_{33} &= -\frac{3dK_t^2 l^2}{r^2 RJ} - \frac{B_\omega}{J} \\ B &= \frac{lK_t}{rR} \begin{bmatrix} -\frac{\sqrt{3}}{2M} & 0 & \frac{\sqrt{3}}{2M} \\ \frac{1}{2M} & \frac{1}{M} & \frac{1}{2M} \\ \frac{d}{J} & \frac{d}{J} & \frac{d}{J} \end{bmatrix} \\ K &= \text{diag}\left(-\frac{C_v}{M}, -\frac{C_{v_n}}{M}, -\frac{C_\omega}{M}\right) \end{aligned}$$

4.2.4. Tính toán động học cho một số chuyển động cơ bản

Các chuyển động dùng để lập trình cho robot gồm có:

Chuyển động tịnh tiến: Khi cho 2 bánh bất kỳ trong 3 bánh quay cùng lúc theo 1 phương nào đó (2 bánh đó quay ngược chiều nhau) thì sẽ làm cho robot chuyển động tịnh tiến.



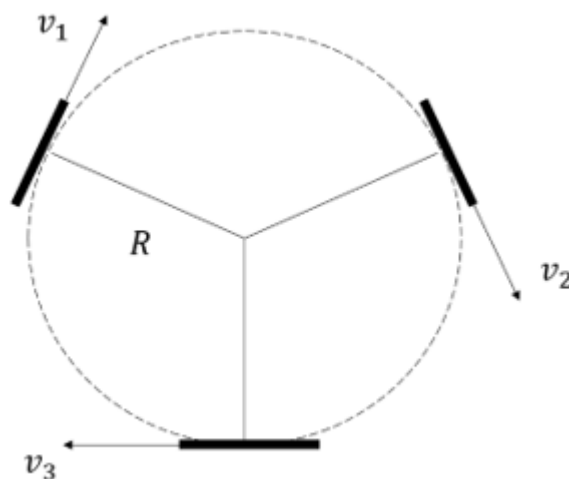
Hình 4.5 Chuyển động tịnh tiến

Ví dụ: Khi cho 2 bánh xe V1 và V2 theo phương như hình vẽ 2.4 với r là bán kính của bánh xe Omni. Giả sử 2 bánh xe cùng quay với vận tốc ω (quay ngược chiều nhau) có vector vận tốc dài là $V1$ và $V2$.

Tổng hợp 2 chuyển động trên ta có vector chuyển động tịnh tiến của robot:

$$v_{tt} = 2v_1 \cos \varphi = 2v_2 \cos \varphi = 2\omega r \cos 30^\circ = \sqrt{3}\omega r$$

Chuyển động quay: Khi cho 3 bánh xe quay cùng chiều thì sẽ làm cho robot quay quanh tâm của nó (hình 2.5). Tùy vào góc quay mà độ lớn vận tốc và hướng quay của các bánh sẽ khác nhau. Căn cứ vào vận tốc các bánh mà ta có thể điều khiển góc quay và vận tốc của robot.



Hình 4.6 Chuyển động quay

4.2.5. Tính toán lựa chọn động cơ

Khối lượng ước tính $m = 10\text{kg}$, vận tốc $v = 0.25\text{m/s}$, bán kính bánh xe chính $r = 50\text{mm}$, hệ số ma sát của bánh xe $\mu = 0.3$, $g = 9.8\text{m/s}^2$.

Theo định luật 2 Newton:

$$\Sigma F = F_w - F_f = ma$$

$$F_w = ma + F_f$$

$$F_w = ma + \mu mg$$

$$F_w = 31.9\text{ N}$$

Momen của động cơ

$$T = F_w r = 1.595\text{ N.m}$$

Số vòng quay của trục động cơ (vòng/phút)

$$N = \frac{v \cdot 1000}{\pi \cdot D} \text{ (vòng/phút)}$$

$$N = \frac{15 \cdot 1000}{\pi \cdot 100} = 48 \text{ (vòng/phút)}$$

$$\omega = \frac{N \cdot 2\pi}{60} = 5.03 \text{ (rad/s)}$$

Công suất cần thiết:

$$P = T\omega = 10.2\text{ W}$$

Hiệu suất của hộp số: $\eta_1 = 0.97$

Hiệu suất của động cơ: $\eta_2 = 0.85$

Hiệu suất của hệ thống: $\eta = \eta_1 \eta_2 = 0.8$

$$P_t = \frac{P}{\eta} = 12.75\text{ W}$$

Với $T = 1.585\text{ N.m}$, $P_t = 12.75\text{ W}$, $N = 48\text{ v/p}$.

Ta chọn động cơ DC Servo JGB37-520 DC Geared Motor.

Thông tin sản phẩm: Động cơ DC Servo JGB37-520 DC Geared Motor



Hình 4.7 Động cơ DC Servo JGB37-520 DC Geared Motor

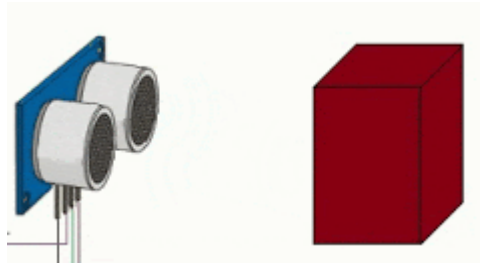
Thông số kỹ thuật:

- Điện áp sử dụng: 12VDC
- Đường kính: 37mm
- Encoder: Cảm biến từ trường Hall, có 2 kênh AB lệch nhau giúp xác định chiều quay và vận tốc của động cơ, đĩa Encoder trả ra 11 xung/1 kênh/ 1 vòng (nếu đo tín hiệu đồng thời của cả hai kênh sẽ thu được tổng 22 xung / 1 vòng quay của Encoder).
 - Cách tính số xung của mỗi kênh trên 1 vòng quay của trục chính động cơ = Tỷ số truyền x số xung của Encoder, ví dụ tỷ số 150:1 thì số xung Encoder trả ra cho 1 vòng quay của trục chính động cơ sẽ là $11 \times 150 = 1650$ xung / 1 kênh.
 - Điện áp cấp cho Encoder hoạt động: 3.3~5VDC, lưu ý cấp quá áp hoặc ngược chiều sẽ làm cháy Encoder ngay lập tức
 - Công suất : từ 7W – 15W.

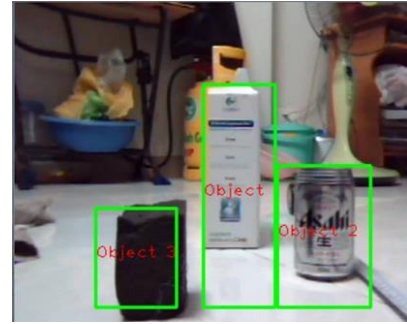
4.2.6. Tính toán chọn số cảm biến

Cách thức di chuyển và ta cũng đã chọn được động cơ cho Mobile Robot vậy giờ làm sao để robot có thể di chuyển một cách tự do trong phạm vi khu vực bạn cho robot di chuyển?

Có rất nhiều cách để cho Mobile robot nhận diện được vật cản trước mắt để cử như sử dụng cảm biến được gắn xung quanh để robot nhận diện được khoảng cách tới vật cản mà tránh né, nếu như cần chính xác hơn khu vực di chuyển phức tạp hơn thì có thể sử dụng xử lý ảnh để có thể nhận diện đối tượng trước mặt mà di chuyển thông minh hơn trong hệ thống công nghiệp.



Hình 4.8 Sử dụng cảm biến



Hình 4.9 Sử dụng xử lý ảnh

Vì giới hạn của đề tài này sử dụng cảm biến gắn xung quanh để robot nhận biết là cách hiệu quả và ít tốn kém nhất. Chọn cảm biến US015 để sử dụng cho đề tài này.

Cảm biến siêu âm US015 là một dạng cảm biến module. Cảm biến này thường chỉ là một bản mạch, hoạt động theo nguyên lý thu phát sóng siêu âm bởi 2 chiếc loa cao tần.

Nguyên lý hoạt động:

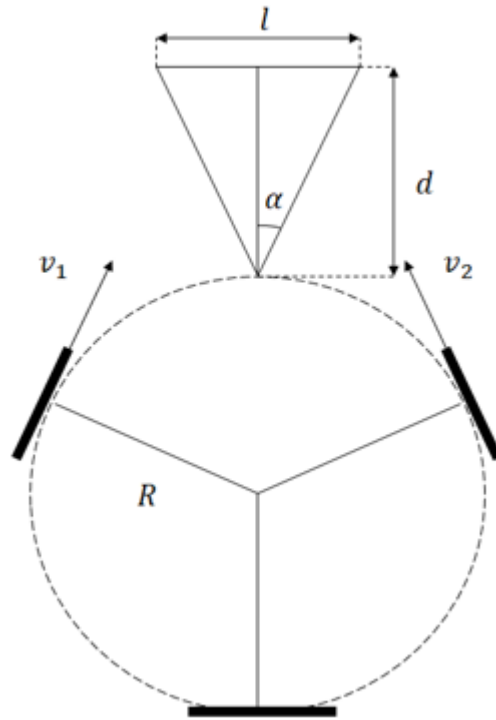
Để đo khoảng cách bằng cảm biến siêu âm US-015, ta sẽ phát 1 xung rất ngắn (5 microSeconds) từ chân Trig. Tiếp theo, 1 xung HIGH ở chân Echo sẽ được cảm biến tạo ra và phát đi cho đến khi nhận lại được sóng phản xạ ở chân này. Lúc này, độ rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và phản xạ lại.

Trong không khí, tốc độ âm thanh đạt mức 340 m/s (hằng số), tương đương với 29,412 microSeconds/cm ($106 / (340 \cdot 100)$).

Khi đã tính được thời gian, ta sẽ chia cho 29,412 để ra giá trị khoảng cách.

Tính toán chọn số lượng cảm biến:

Di chuyển cơ bản của robot là di chuyển tịnh tiến vì vậy ta gắn cảm biến vào hướng di chuyển thẳng của cảm biến như hình vẽ.



Hình 4.10 Mô phỏng cách một cảm biến quét trong không gian

d (mm) khoảng cách phát hiện vật của US-015.

α ($^{\circ}$) góc cảm biến.

l (mm) khoảng quét vật của cảm biến.

R (mm) bán kính của robot.

v_1, v_2 (m/s) vận tốc của hai bánh trước.

Theo thông số của cảm biến US-015 ta có:

$$d_{max} = 4m = 4000mm$$

$$\alpha = 15^{\circ}$$

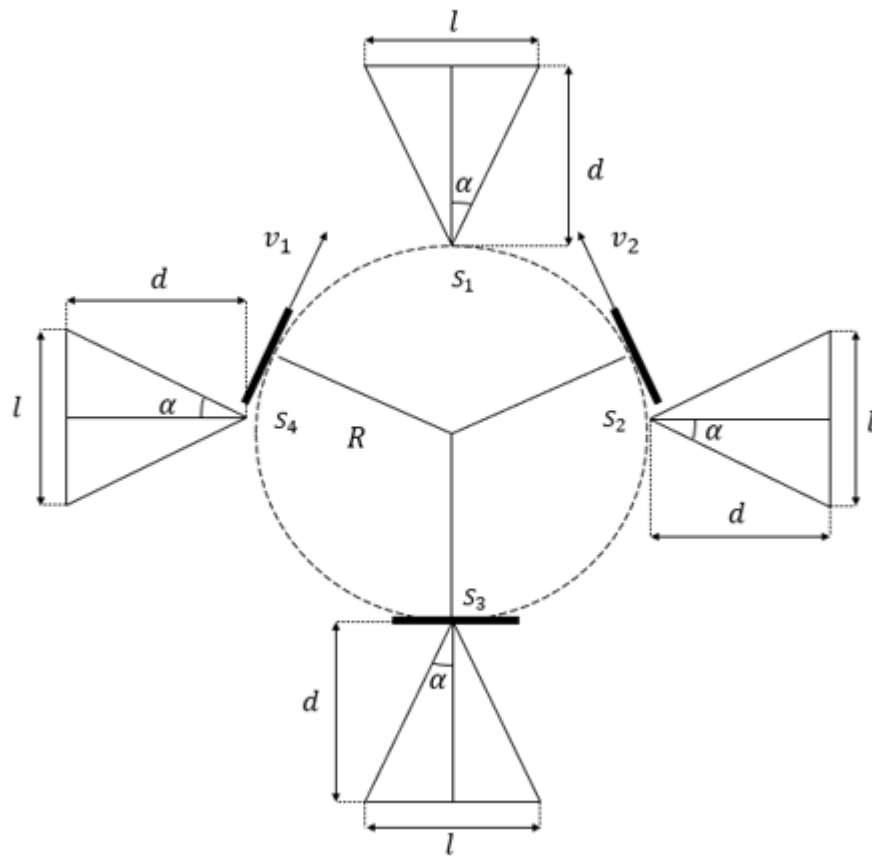
$$\frac{l_{max}}{2} = \sin(\alpha) \times d_{max} = \sin(15^{\circ}) \times 4000 = 1035.3 \text{ (mm)}$$

$$\Rightarrow l_{max} = 2070.55 \text{ (mm)}$$

Ví dụ: Chọn $d = 100mm$ ta tìm được khoảng quét :

$$\frac{l}{2} = \sin(\alpha) \times d = \sin(15^{\circ}) \times 100 = 26 \text{ mm} \Rightarrow l = 52 \text{ mm}$$

Tránh trường hợp 2 cảm biến quét cùng một khoảng quét ta chọn 4 cảm biến phân bố như hình:

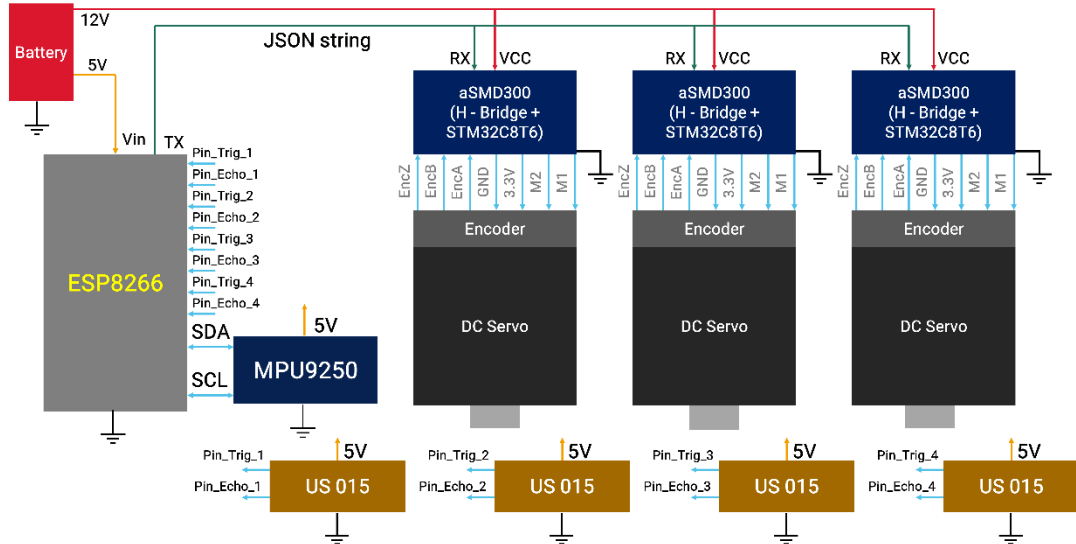


Hình 4.11 Cách phân bố cảm biến trên Mobile Robot

CHƯƠNG 5: XÂY DỰNG HỆ THỐNG ĐIỆN – ĐIỀU KHIỂN

5.1. Hệ thống điều khiển Mobile Robot

Mạch điều khiển robot có nhiệm vụ điều khiển động cơ, đọc tín hiệu từ encoder và MPU6050 để xác định vị trí Robot, đọc tín hiệu cảm biến siêu âm để phát hiện vật cản.



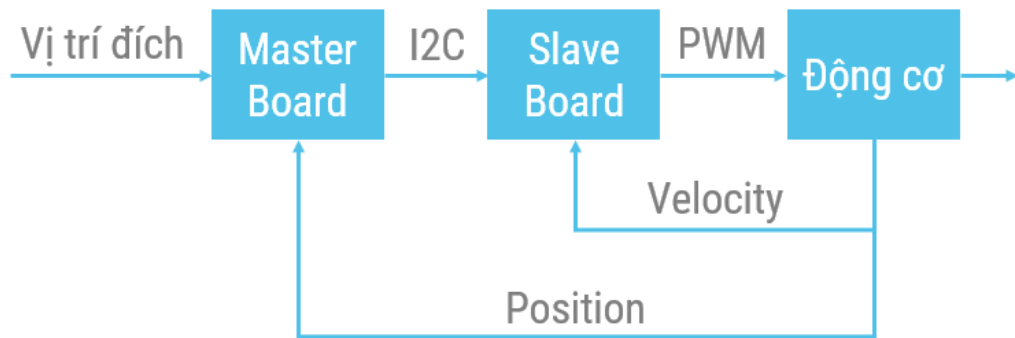
Hình 5.1 Sơ đồ khối hệ thống điều khiển Mobile Robot

Trong đó:

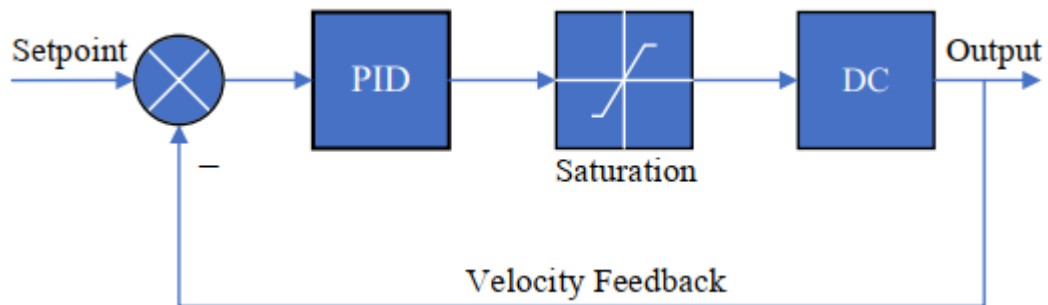
- aSMD350 Driver là khối xử lý trung tâm H-Bridge và STM32F103C8T6, giúp điều khiển động cơ cũng như xử lý các tín hiệu cảm biến.
- MPU6050(GY-521) xác định góc xoay.
- Encoder xác định vận tốc xe.
- Cảm biến siêu âm US-015.

5.1.1. Điều khiển động cơ

Để điều khiển động cơ ổn định ở tốc độ mong muốn, nhóm sử dụng một aSMD350 cho mỗi động cơ. Các Driver giao tiếp với nhau qua giao thức I2C. Bộ điều khiển sẽ tính toán ra các giá trị vận tốc. Mạch điều khiển gửi giá trị của vận tốc về bộ Master. Driver nhận được dữ liệu vận tốc tiến hành điều khiển tốc độ động cơ và đọc giá trị encoder để phản hồi vận tốc. Driver sử dụng bộ điều khiển PID để điều khiển động cơ. Giá trị vận tốc tính được từ encoder sẽ được đưa về mạch Master để tính toán vị trí của Robot và Driver đọc để điều khiển PID cho động cơ.



Hình 5.2 Sơ đồ khối điều khiển động cơ



Hình 5.3 Giải thuật điều khiển động cơ PID

5.1.2. Xác định tọa độ và góc quay của Mobile Robot

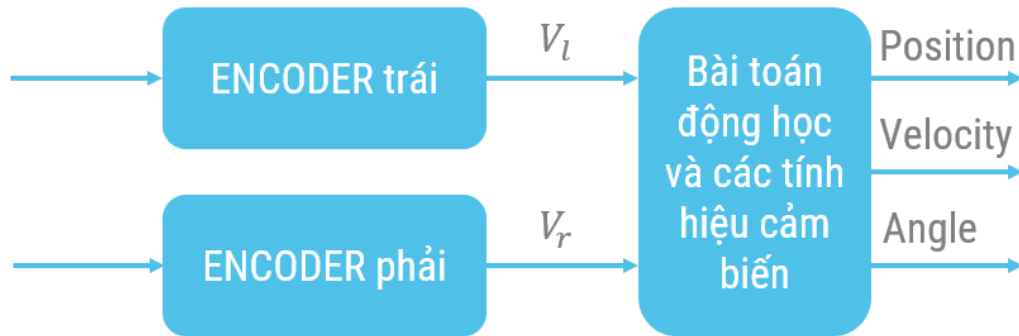
Định vị robot là vấn đề quan trọng trong các nghiên cứu về mobile robot. Ngày nay có nhiều phần cứng hiện đại và việc phát triển giải thuật tính toán dẫn đến tăng độ chính xác và tin cậy trong việc định vị robot. Nhìn chung định vị robot được chia làm ba phương pháp.

- Phương pháp thứ nhất là sử dụng các cảm biến gắn trên robot như: GPS, cảm biến gia tốc... Đặc điểm của phương pháp này có sai số lớn thích hợp với các ứng dụng ngoài trời.

- Phương pháp thứ hai là sử dụng các tia Laser (RP lidar), camera... Đặc điểm của phương pháp này là có độ chính xác cao phù hợp với các ứng dụng trong nhà nhưng chi phí cao và quá trình tính toán phức tạp.

- Phương pháp thứ ba là phương pháp được dùng trong hầu hết trên robot di động vì đơn giản và dễ thực hiện. Đó là sử dụng bộ mã hóa số vòng quay bánh xe (encoder).

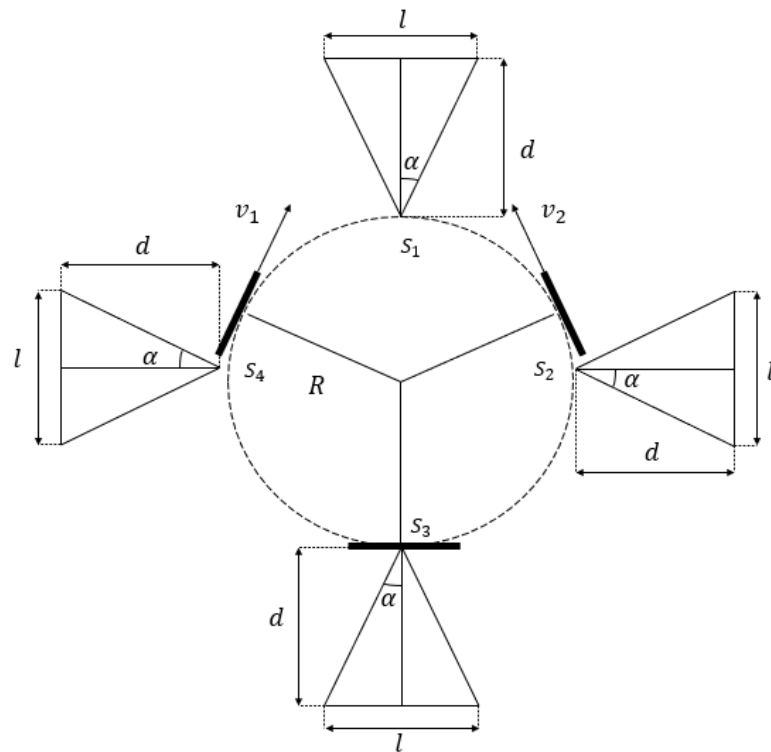
Trong phạm vi đề tài và thời gian nghiên cứu có hạn nên nhóm em sẽ kết phương pháp thứ nhất và thứ ba để định vị Mobile Robot.



Hình 5.4 Sơ đồ khối xác định vị trí của Mobile Robot trong không gian

5.1.3 Phát hiện vật cản

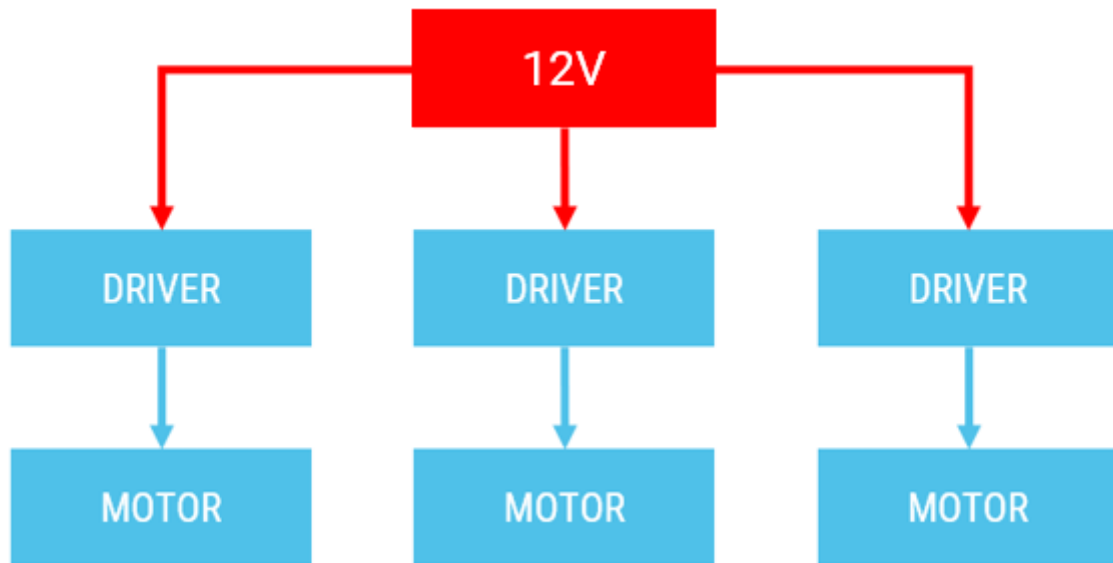
Robot cần phát hiện vật cản trong quá trình di chuyển để tránh va chạm gây hư hỏng robot hoặc va chạm với con người làm tổn thương con người. Ngoài các vật cản đã biết trước khi tạo bản đồ, robot cũng có thể gặp phải các vật cản xuất hiện bất ngờ. Để phát hiện các vật cản này có nhiều phương pháp có thể sử dụng như lidar, camera, cảm biến radar, cảm biến siêu âm... Trong đề tài này nhóm sử dụng cảm biến siêu âm để phát hiện vật cản. Cảm biến siêu âm có đặc điểm ít bị nhiễu bởi môi trường, độ chính xác ở mức độ vừa phải, dễ lập trình điều khiển.



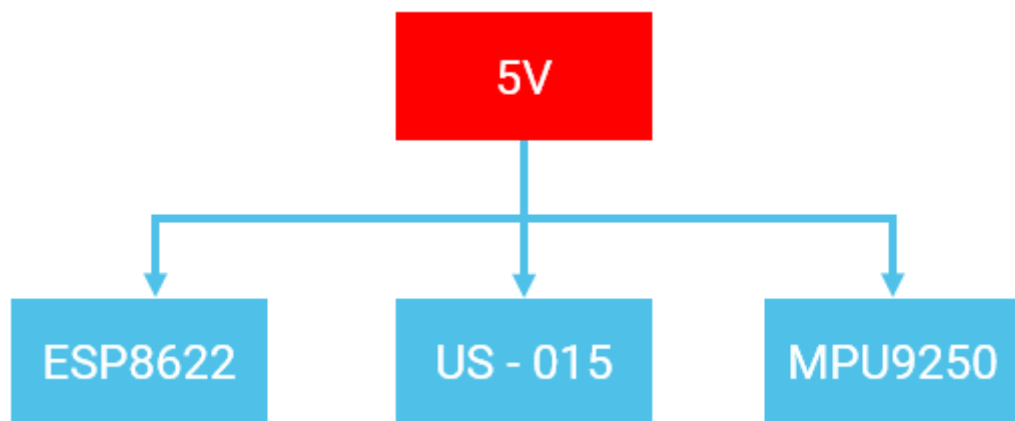
Hình 5.5 Các góc quét của cảm biến siêu âm

5.2. Hệ thống điện Mobile Robot

5.2.1 Tổng quan hệ thống điện



Hình 5.6 Sơ đồ cấp nguồn 12V



Hình 5.7 Sơ đồ cấp nguồn 5V

5.2.2. An toàn điện

Để đảm bảo an toàn điện cho hệ thống và bảo vệ cho các thiết bị điện tránh hư hỏng, nhóm bố trí các mạch ổn áp ở đầu vào của thiết bị điện. Khi gặp sự cố, mạch sẽ tự ngắt. Nhờ vậy đảm bảo an toàn cho hệ thống robot.

5.2.3. Các thiết bị được sử dụng trên Mobile Robot

5.2.3.1. Automotive 300W Single Motor Driver



aSMD300

automotive 300W Single Motor Driver

Specifications

Dimension	64x53x12 (mm)
Voltage range	12-30VDC
Rated Current	12A*
Peak Current	20A*
MCU	STM32 open source PID control
Protection	Fast fuse/over current/ over

Hình 5.8 Đặc tính của aSMD300

aSMD300 là một bộ Driver được tích hợp MCU STM32. Dễ dàng sử dụng lắp ráp để điều khiển nhiều động cơ một lúc. Gồm nhiều cách giao tiếp như: I2C, Uart, CAN.

5.2.3.2. Động cơ

Động cơ DC Servo JGB37-520 DC Geared Motor



Hình 5.9 Động cơ DC Servo JGB37-520 DC Geared Motor

Thông số kỹ thuật:

- Điện áp sử dụng: 12VDC
- Đường kính: 37mm

5.2.3.3. Cảm biến gia tốc MPU9250



Hình 5. 10 MPU9250

Thông tin kỹ thuật:

- Điện áp sử dụng: 3~5VDC
- Điện áp giao tiếp: 3~5VDC
- Chuẩn giao tiếp: I2C
- Giá trị Gyroscopes trong khoảng: +/- 250 500 1000 2000 degree/sec
- Giá trị Acceleration trong khoảng: +/- 2g, +/- 4g, +/- 8g, +/- 16g
- Giá trị Magnetometer trong khoảng: +/- 4800 μ T
- Board mạch mạ vàng, linh kiện hàn tự động bằng máy chất lượng tốt nhất.

Cách sử dụng MPU9250 bằng STM32:

Khởi tạo:

- Thanh ghi 117: Xác định tình trạng truy cập MPU9250 (WHO_AM_I)

BIT	NAME	FUNCTION
[7:0]	WHOAMI	Register to indicate to user which device is being accessed.

Thanh ghi này giúp người dùng xác định xem cảm biến MPU9250 có đang được truy cập. Từ đó xác định khả năng hoạt động được hay không của cảm biến. Dữ liệu của thanh ghi là giá trị ID 8-bit 0x73.

- Thanh ghi 25: Xác định tần số lấy mẫu (Sample Rate Divider):

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
19	25	SMPLRT_DIV[7:0]							

Hình 5.10. Mô tả thanh ghi 25: Sample Rate Divider

Ts lấy mẫu = Gyroscope Output Rate / (1 + SMPLRT_DIV)

Với:

- Ts lấy mẫu là do chúng ta chọn:
- Gyroscope Output Rate:
 - = 1KHz nếu Digital Low Pass Filter (DLPF) is enable (được chọn ở thanh ghi 26: Configuration: DLPF_CFG = 1 -> 6)
 - = 8KHz nếu Digital Low Pass Filter (DLPF) is disable (được chọn ở thanh ghi 26: Configuration: DLPF_CFG = 0 or 7)
- SMPLRT_DIV: bộ chia Sample Rate => được tính khi đã có hết các thông số trên

Ví dụ: 1s đo 1000 lần => 1kHz

- 1 kHz = 1 / (1 + SMPLRT_DIV) => SMPLRT_DIV = 0 (decimal)
=> giá trị gán cho thanh ghi 25 là: 0x00 (Hex)
- Thanh ghi 26: cấu hình nguồn xung ngoại và bộ lọc số cho Gyrometer

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1A	26	-	-	EXT_SYNC_SET[2:0]			DLPF_CFG[2:0]		

Hình 5.11. Mô tả thanh ghi 26: FSYNC và DLPF

Thanh ghi này cấu hình cho nguồn xung ngoại (The External Frame Synchronization -FSYNC) và bộ lọc số (The Digital Low Pass Filter – DLPF, có nhắc đến trong thanh ghi 25) để cài đặt cho cả con quay hồi chuyển (gyroscope) và gia tốc kế (accelerometers).

Giá trị mẫu được ghi lại ở bit đáng kể tối thiểu (LSB – least significant bit) trong thanh ghi dữ liệu của cảm biến, nó được xác định bởi giá trị của EXT_SYNC_SET theo bảng sau.

EXT_SYNC_SET	FSYNC Bit Location
0	Input disabled
1	TEMP_OUT_L[0]
2	GYRO_XOUT_L[0]
3	GYRO_YOUT_L[0]
4	GYRO_ZOUT_L[0]
5	ACCEL_XOUT_L[0]
6	ACCEL_YOUT_L[0]
7	ACCEL_ZOUT_L[0]

Hình 5.12. Bảng giá trị FSYNC trên thanh ghi 26: FSYNC và DLPF

DLPF được cấu hình bởi DLPF_CFG. Gia tốc kế và con quay hồi chuyển được sàng lọc theo các cách khác nhau mà đại diện mỗi cách được đại diện bởi giá trị của DLPF_CFG được nêu trong bảng sau.

FCHOICE		DLPF_CFG	Gyroscope			Temperature Sensor	
<1>	<0>		Bandwidth (Hz)	Delay (ms)	Fs (kHz)	Bandwidth (Hz)	Delay (ms)
x	0	x	8800	0.064	32	4000	0.04
0	1	x	3600	0.11	32	4000	0.04
1	1	0	250	0.97	8	4000	0.04
1	1	1	184	2.9	1	188	1.9
1	1	2	92	3.9	1	98	2.8
1	1	3	41	5.9	1	42	4.8
1	1	4	20	9.9	1	20	8.3
1	1	5	10	17.85	1	10	13.4
1	1	6	5	33.48	1	5	18.6
1	1	7	3600	0.17	8	4000	0.04

Hình 5.13. Bảng giá trị DLPF trên thanh ghi 26: FSYNC và DLPF

Tốc độ lấy mẫu của con quay hồi chuyển (Gyro Sample Rate) được chọn: 1 kHz (thanh ghi 25) và DLPF có băng thông 92 Hz => giá trị gán cho thanh ghi 26 là 0x02.

- Thanh ghi 27: Cấu hình cho bộ đo Góc (Gyroscope Configuration)

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

Hình 5.14. mô tả thanh ghi 27: Gyroscope Configuration

XG_ST, YG_ST, ZG_ST được chọn thì nó sẽ tự kiểm tra (self test).

Chọn FS_SEL cho khoảng đo (full scale range) cho các đầu ra bộ đo Góc (the Gyroscope outputs):

FS_SEL	Full Scale Range
0	$\pm 250^\circ/\text{s}$
1	$\pm 500^\circ/\text{s}$
2	$\pm 1000^\circ/\text{s}$
3	$\pm 2000^\circ/\text{s}$

Hình 5.15. Giá trị FS_SEL trên thanh ghi 27: Gyroscope Configuration

Chọn độ phân giải cho Gyroscope là $\pm 250^\circ/\text{s}$ \Rightarrow Giá trị gán cho thanh ghi 27 là: 0x00

- Thanh ghi 28: Cấu hình cho bộ đo Gia tốc (Accelerometer Configuration)

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		-		

Hình 5.16. Thanh ghi 27: Accelerometer Configuration

XA_ST, YA_ST, ZA_ST được set = 1 thì nó sẽ tự kiểm tra giống thanh ghi 27 (self test)

Chọn AFS_SEL cho full scale range cho các đầu ra bộ đo Gia tốc (the Accelerometer outputs).

AFS_SEL	Full Scale Range
0	$\pm 2g$
1	$\pm 4g$
2	$\pm 8g$
3	$\pm 16g$

Hình 5.17. Giá trị FS_SEL trên thanh ghi 28: Accelerometer Configuration

Chọn độ phân giải cho Accelerometer là $\pm 2g$ \Rightarrow giá trị gán cho thanh ghi 28 là: 0x00

- Thanh ghi 29: cấu hình bộ lọc Accelerometer (Accelerometer Configuration)

BIT	NAME	FUNCTION
[7:6]	Reserved	
[5:4]	Reserved	
[3]	accel_fchoice_b	Used to bypass DLPF as shown in table 2 below. NOTE: This register contains accel_fchoice_b (the inverted version of accel_fchoice as described in the table below).
[2:0]	A_DLPFCFG	Accelerometer low pass filter setting as shown in table 2 below.

Accelerometer Data Rates and Bandwidths (Normal Mode)

ACCEL_FCHOICE	A_DLPF_CFG	Output			
		Bandwidth (Hz)	Delay (ms)	Noise Density (ug/rHz)	Rate (kHz)
0	X	1.13 K	0.75	250	4
1	0	460	1.94	250	1
1	1	184	5.80	250	1
1	2	92	7.80	250	1
1	3	41	11.80	250	1
1	4	20	19.80	250	1
1	5	10	35.70	250	1
1	6	5	66.96	250	1
1	7	460	1.94	250	1

Hình 5.18 Mô tả và bảng giá trị trên thanh ghi 29: Accelerometer Configuration

Thông số bộ lọc số Accelerometer được chọn theo bảng dưới với băng thông 92 Hz để phù hợp với tần số lấy mẫu 1 kHz (sample rate tính ở thanh ghi 25)

⇒ Chọn DLPF_CFG = 2

⇒ Gán giá trị cho thanh ghi 29 là 0x02

- Thanh ghi 56: Cấu hình cho Interrupt (Interrupt Enable)

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
38	56		-		FIFO_OFLOW_EN	I2C_MST_INT_EN	-	-	DATA_RDY_EN

Hình 5.19. Mô tả và bảng giá trị trên thanh ghi 56: Interrupt Enable

FIFO_OFLOW_EN: khi bit này được set lên 1, bit này cho phép tràn bộ đệm FIFO để tạo ra 1 sự kiện ngắt.

I2C_MST_INT_EN: khi bit này được set lên 1, bit này kích hoạt bất kỳ nguồn ngắt từ Master I2C nào để tạo ra 1 sự kiện ngắt.

DATA_RDY_EN: khi bit này được set lên 1, bit này kích hoạt sự kiện ngắt Data Ready, thứ mà xảy ra mỗi khi hoàn thành thao tác ghi vào các thanh ghi cảm biến

- Thanh ghi 107: Quản lý năng lượng (Power Manegement)

Thanh ghi này cho phép người dùng cấu hình cho mode năng lượng và nguồn clock. Nó cũng cung cấp 1 bit để reset cho toàn bộ thiết bị và 1 bit để vô hiệu hóa cảm biến nhiệt độ.

Bit	Tên	GT	Chức năng
[7]	H_RESET	1	Reset toàn bộ thanh ghi bên trong về giá trị khởi tạo của nó. Sau khi việc reset hoàn tất, H_RESET tự động chuyển về 0.
		0	Tắt chế độ Reset
[6]	SLEEP	1	MPU-9250 chuyển về ngủ
		0	MPU-9250 hoạt động bình thường
[5]	CYCLE	1	Khi được đặt và ngủ và chờ không được đặt, chip sẽ đập giữa giấc ngủ và lấy một mẫu với tốc độ được xác định bởi LP_ACCECL_ODR ĐĂNG KÝ LƯU Ý: Khi tất cả các trục gia tốc được vô hiệu hóa qua PWR_MGMT_2 Lên ở tốc độ được xác định bởi các thanh ghi tương ứng ở trên, nhưng sẽ không lấy bất kỳ mẫu nào.
		0	Hoạt động bình thường
[4]	GYRO_STANDBY	1	Khi được đặt, drive con quay và mạch PLL được bật, nhưng các sense path bị vô hiệu hóa. Đây là chế độ công suất thấp cho phép kích hoạt nhanh các con quay.
		0	Drive con quay và mạch PLL được tắt
[3]	PD_PTAT	Tắt điện áp nội PTAT và PTAT ADC	
[2:0]	CLKSEL[2:0]	0	Dao động 20MHz bên trong

		1:5 Tự động chọn nguồn đồng hồ tốt nhất hiện có - PLL Nếu sẵn sàng, khác sử dụng bộ tạo dao động bên trong 6 Dao động 20MHz bên trong 7 Dừng đồng hồ và tiếp tục máy phát thời gian trong RESET (sau khi tải OTP, nghịch đảo của bit PU_SLEEP_MODE sẽ được ghi vào CLKSEL[0])
--	--	--

Khi bật nguồn, nguồn clock của MPU-9250 mặc định lấy từ bộ dao động nội. Tuy nhiên, nó rất được khuyến khích rằng thiết bị được cấu hình bằng cách sử dụng một trong các con quay hồi chuyển (hoặc một nguồn xung ngoại) như là bộ tham chiếu xung để cải thiện khả năng hoạt động.

- Thanh ghi 106: Các điều khiển bởi người dùng (User Control)

Bit	Tên	Chức năng
[7]	Reserve	
[6]	FIFO_EN	1 - Kích hoạt chế độ hoạt động FIFO. 0 - Tắt truy cập FIFO từ giao diện nối tiếp. Để vô hiệu hóa FIFO ghi bởi DMA, hãy sử dụng thanh ghi FIFO_EN. Để vô hiệu hóa FIFO có thể ghi từ DMP, tắt DMP.
[5]	I2C_MST_EN	1 - Kích hoạt môđun I/F Master I2C; Ghim ES_DA và ES_SCL được phân lập từ chân SDA/ SDI và SCL/ SCLK. 0 - Tắt mô -đun I/F chính I2C; Ghim ES_DA và ES_SCL được điều khiển một cách hợp lý bởi các chân SDA/ SDI và SCL/ SCLK.

		Lưu ý: DMP sẽ chạy khi được bật, ngay cả khi tắt cả các cảm biến bên trong bị tắt, trừ khi tốc độ mẫu được đặt thành 8kHz.
[4]	I2C_IF_DIS	1 - Đặt lại môđun slave I2C và chỉ đặt giao diện nối tiếp ở chế độ SPI. Bit Auto này xóa sau một chu kỳ đồng hồ.
[3]	Reserved	
[2]	FIFO_RST	1 - Đặt lại môđun FIFO. Đặt lại là không đồng bộ. Bit Auto này xóa sau một chu kỳ đồng hồ.
[1]	I2C_MST_RST	1 - Đặt lại môđun chính I2C. Đặt lại là không đồng bộ. Bit Auto này xóa sau một chu kỳ đồng hồ. Lưu ý: bit này chỉ nên được đặt khi Master I2C đã treo. Nếu bit này được đặt trong một giao dịch chính I2C hoạt động, nô lệ I2C sẽ bị treo, sẽ yêu cầu máy chủ lưu trữ lại slave.
[0]	SIG_COND_RST	1 - Đặt lại tất cả đường dẫn tín hiệu số con quay, đường dẫn tín hiệu số và đường dẫn tín hiệu số TEMP. Bit này cũng xóa tất cả các thanh ghi cảm biến. SIG_COND_RST là một xung rộng một clk8m.

- Cùng với các thanh ghi configuration khác để cấu hình cho cảm biến MPU-9250: FIFO_EN, I2C_MST_CTRL, FIFO_COUNTH, FIFO_W_R,

Thanh ghi 59 – 64: Bộ đo Accelerometer

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT[7:0]							

Hình 5.20. Bảng mô tả trên thanh ghi 59 – 64: Accelerometer Measurements

Các thanh ghi này ghi lại các giá trị của phép đo gia tốc trước đó.

Mỗi phép đo gia tốc kể 16 bit có một thang đo đầy đủ được xác định trong ACCEL_FS (Thanh ghi 28). Đối với mỗi cài đặt thang đo, độ nhạy của máy đo gia tốc trên mỗi LSB trong ACCEL_xOUT được hiển thị trong bảng bên dưới.

AFS_SEL	Full Scale Range	LSB Sensitivity
0	$\pm 2g$	16384 LSB/g
1	$\pm 4g$	8192 LSB/g
2	$\pm 8g$	4096 LSB/g
3	$\pm 16g$	2048 LSB/g

Hình 5.21 Giá trị phép đo trên thanh ghi 59 – 64: Accelerometer Measurements

- Thanh ghi 67 - 72: Bộ đo vận tốc Gyro

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
43	67	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT[7:0]							

Hình 5.22 Thanh ghi 67 - 72: Bộ đo vận tốc Gyro (Gyroscope Measurements)

Các thanh ghi này ghi lại các giá trị của phép đo vận tốc góc trước đó.

Mỗi phép đo vận tốc góc kể 16 bit có một thang đo đầy đủ được xác định trong FS_SEL (Thanh ghi 27). Đối với mỗi cài đặt thang đo, độ nhạy của máy đo vận tốc góc trên mỗi LSB trong GYRO_xOUT được hiển thị trong bảng bên dưới.

FS_SEL	Full Scale Range	LSB Sensitivity
0	± 250 °/s	131 LSB/°/s
1	± 500 °/s	65.5 LSB/°/s
2	± 1000 °/s	32.8 LSB/°/s
3	± 2000 °/s	16.4 LSB/°/s

Hình 5.23 Bảng giá trị phép đo trên thanh ghi 67 - 72: Bộ đo vận tốc Gyro (Gyroscope Measurements)

- Thanh ghi 65 – 66: Đo nhiệt độ

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
41	65	TEMP_OUT[15:8]							
42	66	TEMP_OUT[7:0]							

Hình 5.24 Bảng mô tả trên thanh ghi 65 - 66: Temperature Measurements

Thanh ghi này ghi lại các giá trị của phép đo nhiệt độ trước đo.

Giá trị nhiệt độ (oC) được lấy ra từ thanh ghi có thể tính như sau:

$$\text{Giá trị nhiệt độ (oC)} = \frac{\text{Giá trị thô lấy ra từ thanh ghi}}{340} + 36.53$$

- Thanh ghi CNTL 1: Control 1 (AK8963 – Cảm biến từ kế)

Addr	Register name	D7	D6	D5	D4	D3	D2	D1	D0
Read-only register									
0AH	CNTL1	0	0	0	BIT	MODE3	MODE2	MODE1	MODE0
	Reset	0	0	0	0	0	0	0	0

Hình 5.25 Bảng mô tả trên thanh ghi CNTL 1: Control 1

MODE[3:0]: các mode vận hành:

- “0000”: mode tắt nguồn
- “0001”: mode đo đơn
- “0010”: mode đo liên tục 1
- “0110”: mode đo liên tục 2
- “0100”: mode đo kích hoạt ngoại
- “1000”: mode tự đo
- “1111”: mode truy cập FUSE ROM

- Thanh ghi bộ đo từ kế

Addr	Register name	D7	D6	D5	D4	D3	D2	D1	D0
Read-only register									
03H	HXL	HX7	HX6	HX5	HX4	HX3	HX2	HX1	HX0
04H	HXH	HX15	HX14	HX13	HX12	HX11	HX10	HX9	HX8
05H	HYL	HY7	HY6	HY5	HY4	HY3	HY2	HY1	HY0
06H	HYH	HY15	HY14	HY13	HY12	HY11	HY10	HY9	HY8
07H	HZL	HZ7	HZ6	HZ5	HZ4	HZ3	HZ2	HZ1	HZ0
08H	HZH	HZ15	HZ14	HZ13	HZ12	HZ11	HZ10	HZ9	HZ8
Reset		0	0	0	0	0	0	0	0

Hình 5.26 Bảng mô tả trên thanh ghi 03-08 AK8963: Magnematic Measurement

Dữ liệu từ trường của cảm biến từ kế:

- HXL[0:7]: dữ liệu trục x 8 bit thấp
- HXH[8:15]: dữ liệu trục y 8 bit cao
- HYL[0:7]: dữ liệu trục y 8 bit thấp
- HYH[8:15]: dữ liệu trục y 8 bit cao
- HZL[0:7]: dữ liệu trục z 8 bit thấp
- HZH[8:15]: dữ liệu trục z 8 bit cao

Các bước thực hiện

- Cấu hình:

Chọn tần số lấy mẫu và bộ chia của nó dựa trên công thức:

$$Ts \text{ lấy mẫu} = \text{Gyroscope Output Rate} / (1 + \text{SMPLRT_DIV})$$

Với:

- Ts lấy mẫu là do chúng ta chọn:
- Gyroscope Output Rate:
 - = 1 kHz nếu Digital Low Pass Filter (DLPF) is enable (được chọn ở thanh ghi 26: Configuration: DLPF_CFG = 1 -> 6)
 - = 8 kHz nếu Digital Low Pass Filter (DLPF) is disable (được chọn ở thanh ghi 26: Configuration: DLPF_CFG = 0 or 7)
- SMPLRT_DIV: bộ chia Sample Rate => được tính khi đã có hết các thông số trên

Ví dụ: 1s đo 1000 lần => 1 kHz và Gyroscope Output Rate = 1 kHz

$$1\text{kHz} = 1 / (1 + \text{SMPLRT_DIV}) \Rightarrow \text{SMPLRT_DIV} = 0 \text{ (decimal)} \Rightarrow \text{pData} = 0x00(\text{Hex}).$$

- Tắt bộ lọc số và cấu hình cho nguồn xung ngoại cung cấp cho MPU9250:

Dựa vào phần 2.1.2. Thanh ghi 26, tắt bộ lọc số DLPF_CFG = 0 và chọn xung nội làm nguồn cung cấp xung cho MPU6050 EXT_SYNC_SET = 0 => pData = 0x00.

- Cấu hình cho con quay hồi chuyển (Gyroscope):

Chọn thang đo ± 250 o/s => FS_SEL = 0 được set ở bit 3 - 4 => 0000 0000

=> pData = 0x00 (Hex).

- Cấu hình cho gia tốc kế (Accelerometer):

Chọn thang đo $\pm 2g$ => AFS_SEL = 0 được set ở bit 3 - 4 => 0000 0000

=> pData = 0x00 (Hex).

- Cấu hình quản lý năng lượng:

Chọn con quay hồi chuyển trực X làm bộ nguồn xung clock cho MPU9250 (được khuyến cáo) => 0000 0001 => pData = 0x01 (hex).

- Đọc giá trị đầu ra:

- Đọc giá trị từ thanh ghi:

- ❖ Cảm biến gia tốc (Accelerometer)

Với rawAccelData là mảng 6 giá trị rawAccelData[6]. Khi đọc từ thanh ghi ACCEL_XOUT_H, ACCEL_XOUT_L, ACCEL_YOUT_H, ACCEL_YOUT_L, ACCEL_ZOUT_H, ACCEL_ZOUT_L, lệnh HAL_I2C_Mem_Read() sẽ tự động đọc các thanh ghi liên kề với từng giá trị 8 bit đo được ở mỗi trục được gán vào 1 biến trong mảng rawAccelData[6].

- ❖ Cảm biến con quay hồi chuyển (Gyroscope)

Cũng tương tự như cảm biến gia tốc (Accelerometer), có đến 6 thanh ghi lưu trữ giá trị 8 bit đo được từ 3 trục x, y, z trong hệ tọa độ của cảm biến MPU9250. Lệnh HAL_I2C_Mem_Read() đọc lần lượt từng giá trị ở mỗi thanh ghi và gán vào trong mảng rawGyroData[6].

- ❖ Cảm biến từ kế (Magnetometer)

Giống như hai bộ cảm biến gia tốc và con quay hồi chuyển trên, cảm biến từ kế (Magnetometer) cũng thực hiện phép đo trên hệ trục tọa độ của cảm biến MPU9250. Tại mỗi trục sẽ có kết quả trả về là 1 giá trị 16 bit được chia nhỏ thành 2 giá trị 8 bit được lưu vào 2 thanh ghi. Các thanh ghi

được sử dụng là từ AK8963_XOUT_L (địa chỉ thanh ghi: 0x03) đến AK8963_XOUT_H (địa chỉ thanh ghi: 0x08).

- Xử lý dữ liệu thô

Vì mỗi gia tốc của mỗi trục đo được 1 giá trị 16 bit mà lệnh HAL_I2C_Mem_Read chỉ đọc được giá trị 8 bit từ các thanh ghi, cho nên cần ghép 8 bit đầu và 8 bit cuối từ mỗi thiết bị đo của một trục nào đó thành 1 giá trị 16 bit hoàn chỉnh

❖ Cảm biến gia tốc kế (Accelerometer)

```
accelCountX = (int16_t)(rawAccelData[0]<<8 | rawAccelData [1]);
```

```
accelCountY = (int16_t)(rawAccelData[2]<<8 | rawAccelData [3]);
```

```
accelCountZ = (int16_t)(rawAccelData[4]<<8 | rawAccelData [5]);
```

❖ Cảm biến con quay hồi chuyển (Gyroscope)

```
gyroCountX = (int16_t)(rawGyroData[0]<<8 | rawGyroData [1]);
```

```
gyroCountY = (int16_t)( rawGyroData [2]<<8 | rawGyroData [3]);
```

```
gyroCountZ = (int16_t)( rawGyroData [4]<<8 | rawGyroData [5]);
```

❖ Cảm biến từ kế (Magnetometer)

```
magCountX = (int16_t)( rawMagData [0]<<8 | rawMagData [1]);
```

```
magCountY = (int16_t)( rawMagData [2]<<8 | rawMagData [3]);
```

```
magCountZ = (int16_t)( rawMagData [4]<<8 | rawMagData [5]);
```


- Sau đó cần chia với thang đo tương ứng để được giá trị thực:

- ❖ Cảm biến gia tốc kế (Accelerometer)

$$Ax = (\text{float})\text{accelCountX} * 2/32768.0;$$

$$Ay = (\text{float}) \text{ accelCountY} * 2/32768.0;$$

$$Az = (\text{float}) \text{ accelCountZ} * 2/32768.0;$$

- ❖ Cảm biến con quay hồi chuyển (Gyroscope)

$$Gx = (\text{float})\text{gyroCountX} * 250/32768.0;$$

$$Gy = (\text{float})\text{gyroCountY} * 250/32768.0;$$

$$Gz = (\text{float})\text{gyroCountZ} * 250/32768.0;$$

- Cảm biến từ kế (Magnetometer)

$$Mx = (\text{float})\text{magCountX} * 4912.0/8190.0;$$

$$My = (\text{float})\text{magCountY} * 4912.0/8190.0;$$

$$Mz = (\text{float})\text{magCountZ} * 4912.0/8190.0;$$

- Tính các góc quay quanh các trục hệ tọa độ MPU9250

Để tính các góc: roll – góc quay quanh trục x, pitch – góc quay quanh trục y, yaw – góc quay quanh trục z, cần chuyển đổi các phép Quaternion ước lượng thành các góc Euler

Một Quaternion có thể được mô tả như sau

$$\mathbf{q} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T = [q_w \quad q_x \quad q_y \quad q_z]^T$$

$$|\mathbf{q}|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = q_w^2 + q_x^2 + q_y^2 + q_z^2 = 1$$

Có thể kết hợp với các góc quay quanh các trục nêu trên với phương trình sau:

$$\mathbf{q}_0 = \mathbf{q}_w = \cos(\alpha/2)$$

$$\mathbf{q}_1 = \mathbf{q}_x = \sin(\alpha/2) \cos(\beta_x)$$

$$\mathbf{q}_2 = \mathbf{q}_y = \sin(\alpha/2) \cos(\beta_y)$$

$$\mathbf{q}_3 = \mathbf{q}_z = \sin(\alpha/2) \cos(\beta_z)$$

Tại đây: trong đó α là một góc quay đơn (giá trị trong radian của góc quay) và $\cos(\beta_x)$, $\cos(\beta_y)$ và $\cos(\beta_z)$ là “cosin hướng” của các góc giữa ba trục tọa độ và trục của vòng quay.

Để hiểu rõ hơn các “cosin hướng” với các quaternion sau:

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{q}_w = \cos(\text{rotation angle}/2) \\ \mathbf{q}_1 &= \mathbf{q}_x = \sin(\text{rotation angle}/2) \cos(\text{angle between axis of rotation and x axis}) \\ \mathbf{q}_2 &= \mathbf{q}_y = \sin(\text{rotation angle}/2) \cos(\text{angle between axis of rotation and y axis}) \\ \mathbf{q}_3 &= \mathbf{q}_z = \sin(\text{rotation angle}/2) \cos(\text{angle between axis of rotation and z axis}) \end{aligned}$$

Nếu trục quay trùng với trục x:

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{q}_w = \cos(\alpha/2) \\ \mathbf{q}_1 &= \mathbf{q}_x = \sin(\alpha/2) \cdot 1 \\ \mathbf{q}_2 &= \mathbf{q}_y = \sin(\alpha/2) \cdot 0 \\ \mathbf{q}_3 &= \mathbf{q}_z = \sin(\alpha/2) \cdot 0 \end{aligned}$$

Nếu trục quay trùng với trục y:

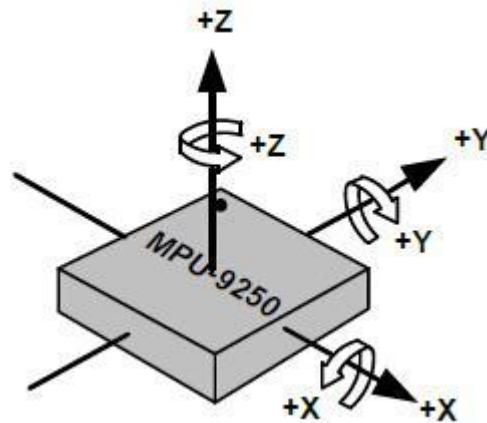
$$\begin{aligned} \mathbf{q}_0 &= \mathbf{q}_w = \cos(\alpha/2) \\ \mathbf{q}_1 &= \mathbf{q}_x = \sin(\alpha/2) \cdot 0 \\ \mathbf{q}_2 &= \mathbf{q}_y = \sin(\alpha/2) \cdot 1 \\ \mathbf{q}_3 &= \mathbf{q}_z = \sin(\alpha/2) \cdot 0 \end{aligned}$$

Nếu trục quay hợp một góc 45° giữa các trục x và y:

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{q}_w = \cos(\alpha/2) \\ \mathbf{q}_1 &= \mathbf{q}_x = \sin(\alpha/2) \cdot 0.707 \dots \\ \mathbf{q}_2 &= \mathbf{q}_y = \sin(\alpha/2) \cdot 0.707 \dots \\ \mathbf{q}_3 &= \mathbf{q}_z = \sin(\alpha/2) \cdot 0 \end{aligned}$$

Do đó, các trục X và Y ảnh hưởng qua lại đối với trục quay mới. Các góc Euler có thể được lấy từ Độ tứ thông qua các mối quan hệ:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan \frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)} \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \arctan \frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)} \end{bmatrix}$$



Hình 5.27 Góc roll, pitch và yaw trên MPU9250

Tuy nhiên, các hàm Arctan và Arcsin được triển khai trong các ngôn ngữ máy tính chỉ tạo ra kết quả giữa $-\pi/2$ và $\pi/2$ và trong ba lần quay giữa $-\pi/2$ và $\pi/2$, người ta không có được tất cả các hướng có thể. Để tạo tất cả các hướng, người ta cần thay thế các hàm Arctan trong mã máy tính bằng atan2:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \text{asin}(2(q_0q_2 - q_3q_1)) \\ \text{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix}$$

5.2.3.4. Cảm biến siêu âm đo khoảng cách vật cản

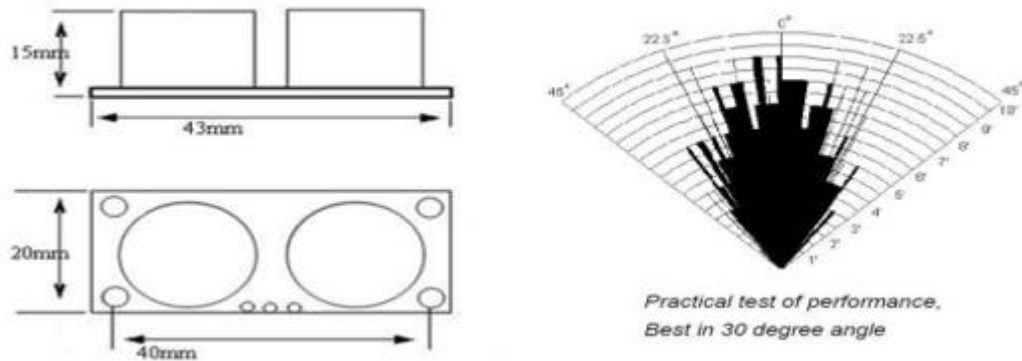


Hình 5.28 US-015

Thông số kỹ thuật:

- Điện thế hoạt động: DC 5V
- Dòng tiêu thụ: 2.2mA
- Khoảng nhiệt độ hoạt động: 0 - 70°C
- Chuẩn tín hiệu: TTL
- Góc quét: < 15 độ
- Khoảng cách nhận: 2 ~ 400cm

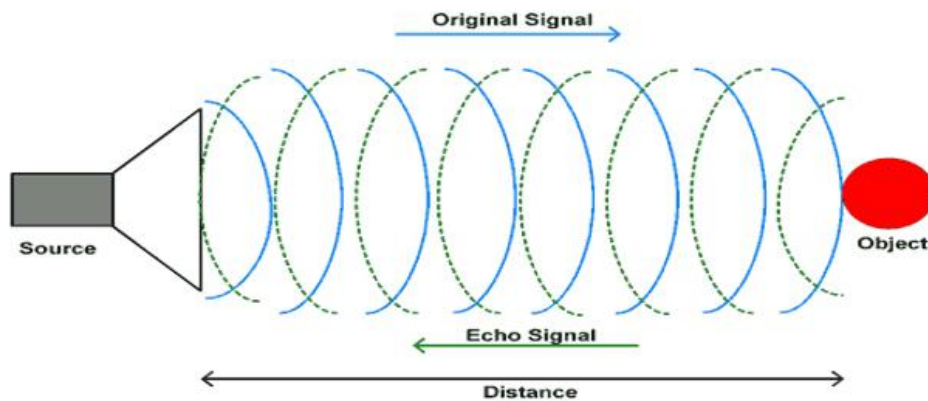
- Độ chính xác: $0.3\text{cm} + 1\%$.



Hình 5.29 Thông số kỹ thuật US-015

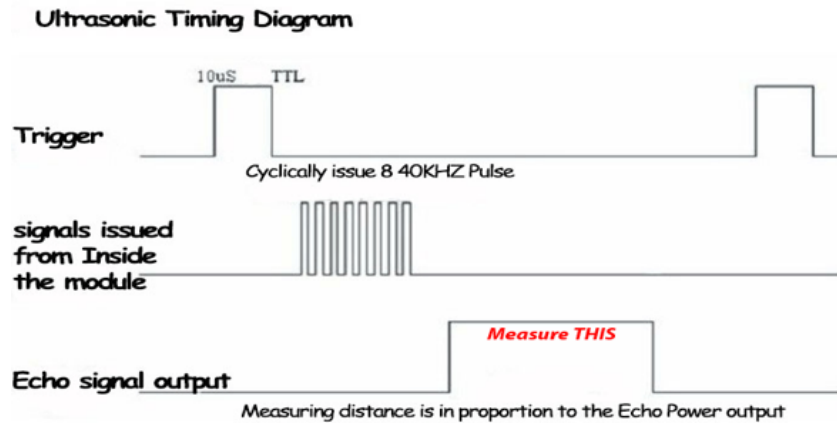
Nguyên lý hoạt động của cảm biến:

Hoạt động theo nguyên tắc sonar hoặc radar. Cảm biến siêu âm phát ra siêu âm, siêu âm sau khi chạm vào vật cản sẽ phản xạ lại và được thu hồi bởi cảm biến. Ta có thể đo khoảng cách bằng cách xác định thời gian từ lúc phát đến lúc nhận được phản xạ.



Hình 5.30 Cách xác định vật

- Trên chân Trig ta sẽ phát 1 xung rất ngắn độ rộng khoảng $10\mu\text{s}$.
- Tiếp theo đó từ bộ phát của cảm biến sẽ phát ra chuỗi xung 8 chu kỳ có tần số 40kHz đồng thời 1 xung ở chân Echo sẽ nhảy từ mức thấp lên cao và sẽ tiếp tục ở mức cao cho đến khi chạm vật cản và trở về thì nó sẽ nhảy xuống mức thấp.
- Lúc này động rộng của xung sẽ bằng với thời gian sóng siêu âm được phát từ cảm biến và phản xạ lại.



Hình 5.31 Nguyên lý hoạt động

Trong không khí, tốc độ âm thanh đạt mức 340 m/s (hằng số), tương đương với $29,412 \text{ cm}/\mu\text{s}$ ($106 / (340 \cdot 100)$).

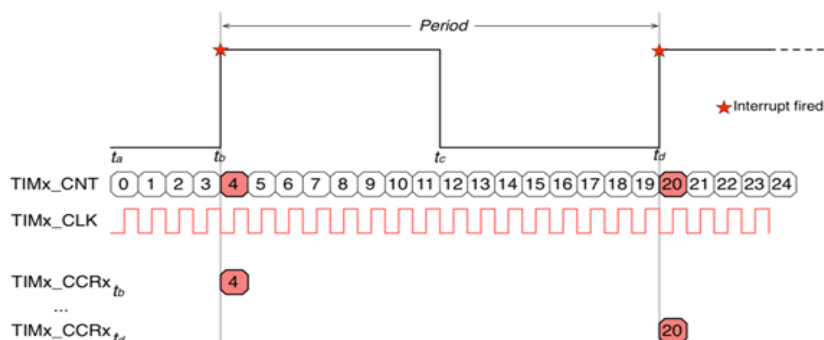
Khi đã tính được thời gian, ta sẽ chia cho 29,412 để ra giá trị khoảng cách.

Đo khoảng cách cảm biến bằng chế độ Input Capture

Như đã đề cập ở trên module sẽ gửi tín hiệu ở mức cao tỉ lệ với khoảng cách đo được. Tín hiệu này sẽ duy trì ở mức cao trong vòng vài micro giây, vì vậy ta sẽ sử dụng chế độ InputCapture với Interrupts trong Timer2 để đo khoảng cách chính xác hơn. Phương pháp này sẽ tránh các lỗi thường gặp đối với các phương pháp cũ.

Input Capture là gì?

Input Capture là chế độ bắt sườn đầu vào ở chế độ này, timer sẽ bắt sự kiện xung từ bên ngoài vào, cụ thể là bắt sườn lên hoặc sườn xuống của xung, thường ứng dụng trong các hoạt động đo tần số hoặc độ rộng xung.



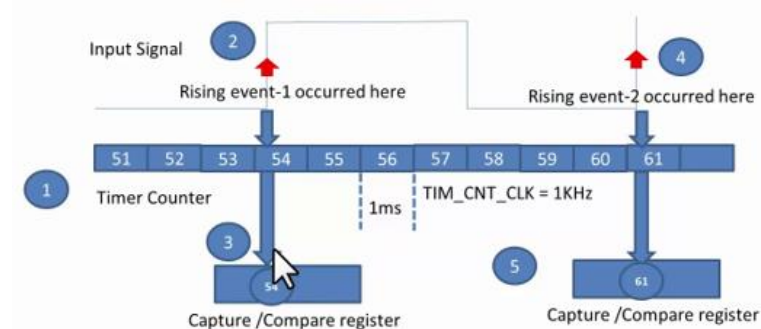
Hình 5.32 Chế độ InputCapture

STM32 Timer chế độ Input Capture

Input Capture có 2 chế độ là bắt xung sườn lên (Rising) hoặc sườn xuống (Falling)

Nguyên lý hoạt động như sau:

- Sau khi khởi động Timer, mỗi khi có xung sườn lên (hoặc xuống) tại đầu vào Channelx của Timer. Thanh ghi TIMx_CCRx (x là Timer số 1,2,3,4...) sẽ được nạp giá trị của thanh ghi Counter TIMx_CNT.
- Bit CC1IF được đặt lên 1 (Cờ ngắt), Bit CC1OF sẽ được set lên 1 nếu 2 lần cờ liên tiếp cờ CC1IF được đặt lên 1 mà không xóa. Nghĩa là mỗi khi có sự kiện xung lên sẽ có ngắt xảy ra, lập trình viên phải đọc giá trị lưu vào CCR1 sau đó xóa cờ CC1IF, nếu không cờ CC1OF sẽ được bật lên báo tràn và dữ liệu sẽ không được ghi vào nữa.
- Một sự kiện ngắt được sinh ra nếu Bit CC1IE được đặt lên 1.
- Một sự kiện DMA sinh ra nếu Bit CC1DE được đặt lên 1.



Hình 5.33 Timer trong chế độ InputCapture

Thuật toán

Sử dụng chế độ Input Capture để chụp giá trị cạnh lên ở mức cao (High) và giá trị cạnh xuống ở mức thấp (Low)

- Khoảng thời gian đầu tiên khi phát hiện vật sẽ được ghi lại khi chân Echo bật lên mức cao giá trị Timer sẽ được ghi lại vào thanh ghi CCR1 và chúng ta sẽ có khoảng thời gian T1
- Sau đó sẽ kích hoạt một tín hiệu ngắt để chụp ở trên cạnh rơi
- Khi chân Echo ở mức thấp giá trị Timer cũng sẽ được ghi lại ở trên mép cạnh rơi ở mức thấp và chúng ta sẽ có khoảng thời gian T2
- Độ rộng xung được tính bằng khoảng thời gian chênh lệch $T = T2 - T1$
- Khi đã tính được thời gian, ta sẽ nhân cho $29,412 \text{ cm}/\mu\text{s}$ để ra giá trị khoảng cách.

5.2.3.5 *Acquy*

- Bình Acquy GS GTZ5S được sản xuất tại tập đoàn sản xuất ắc quy GS Nhật Bản nổi tiếng thế giới.

- Loại bình: VRLA, siêu kín siêu bền, miễn bảo dưỡng.

- Ưu điểm Acquy GS: tuổi thọ lâu dài, chất lượng ổn định, thiết kế phù hợp với điều kiện khí hậu tại Việt Nam.

- Dung lượng: 12V – 3.5Ah (10 hours). - Kích thước (mm): 112 x 70 x 85 (dài x rộng x cao).

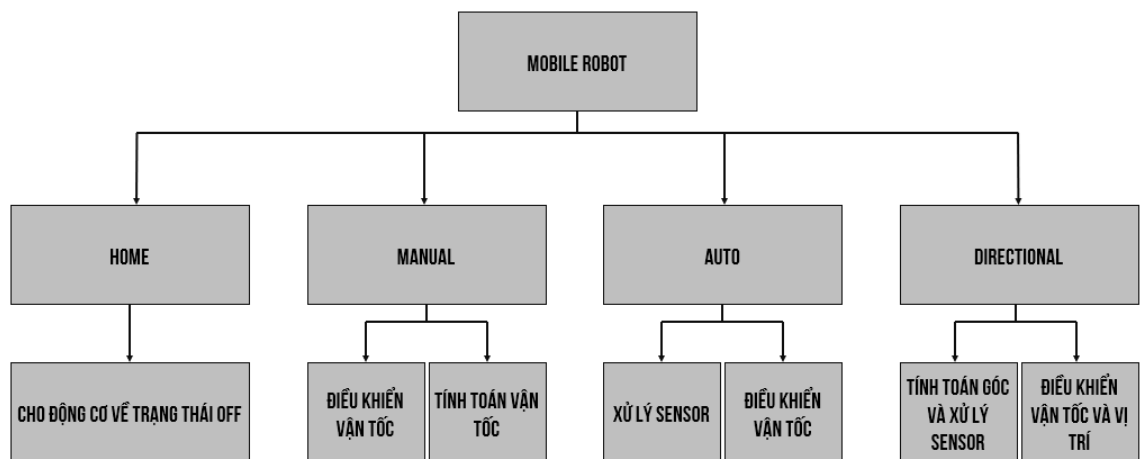
CHƯƠNG 6: XÂY DỰNG GIAO DIỆN APP TƯƠNG TÁC VỚI NGƯỜI DÙNG

Để đáp ứng nhu cầu tự động và giám sát từ xa Mobile Robot cần phải có một giao diện tương tác với người sử dụng nhằm phối hợp các cơ cấu cơ khí, các ngoại vi hoạt động đồng bộ và đặc biệt là hợp nhất với nhau.

6.1 Giới thiệu về tổng quan giao diện

App sẽ điều khiển được các chức năng như sau:

- **Manual:** Chức năng giúp cho người dùng có thể di chuyển Mobile Robot theo hướng mà mình muốn di chuyển. Thường được ứng dụng trong các nhà máy vận chuyển đồ có khối lượng tương đối giúp người dùng có thể linh hoạt hơn trong việc di chuyển robot đến các nơi khó để Mobile Robot di chuyển tự động đến được.
- **Auto:** Chức năng này làm cho Mobile Robot có thể di chuyển tự do trong khu vực kiểm soát. Nhờ có các sensor giúp cho Mobile Robot có thể di chuyển mà không va đập vào các vật cản. Được ứng dụng trong các Mobile Robot trong gia đình như lau nhà hoặc hút bụi.
- **Directional:** Chức năng này cho chúng ta một bản đồ cố định theo hình bàn cờ khi cần Mobile Robot di chuyển đến đúng toàn bộ mình mong muốn. Được sử dụng cho các nhà máy có các khu đặt hàng cố định Mobile Robot để di chuyển.



Hình 6. 2 Sơ đồ khối tổng quát giao diện điều khiển

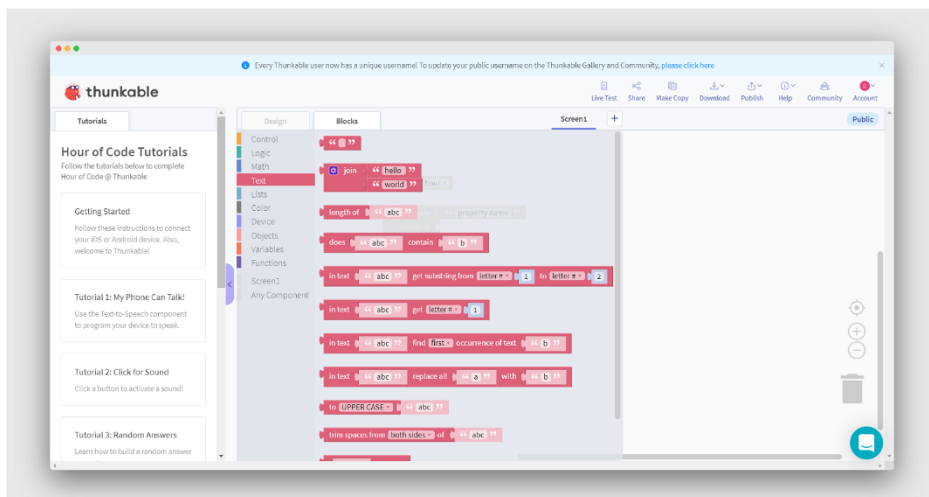
6.2 Phần mềm thiết kế App – Thunkable

Thunkable là một website được xây dựng lên từ công nghệ của MIT App Inventor bởi hai trong số những kỹ sư MIT đầu tiên của MIT Inventor vào đầu năm 2016. Nền tảng này dành cho những nhà lập trình phát triển ứng dụng chuyên nghiệp hơn với chất

lượng cao hơn, hoặc xây dựng các ứng dụng mạnh mẽ hơn cho doanh nghiệp, cho cộng đồng hoặc chỉ cho bản thân.

Do được xây dựng dựa trên MIT App Inventor, Thunkable có tất cả các đặc điểm nêu trên của MIT App Inventor. Ngoài ra, Thunkable có những cải tiến sau đây:

- Material Design: Nhược điểm lớn nhất của MIT App Inventor là về trình thiết kế ở Thunkable được hỗ trợ các Material Design đa dạng dễ dàng sử dụng và cực kỳ hợp với xu hướng hiện tại
- Có rất nhiều tiện ích hỗ trợ như: Google map, đọc các dạng tài liệu như PDF hay word,...
- Kích thước của ứng dụng có thể rất cao.
- Nhóm phát triển Thunkable đã bắt đầu đưa vào thử nghiệm phiên bản iOS của ngôn ngữ Drag and Drop.



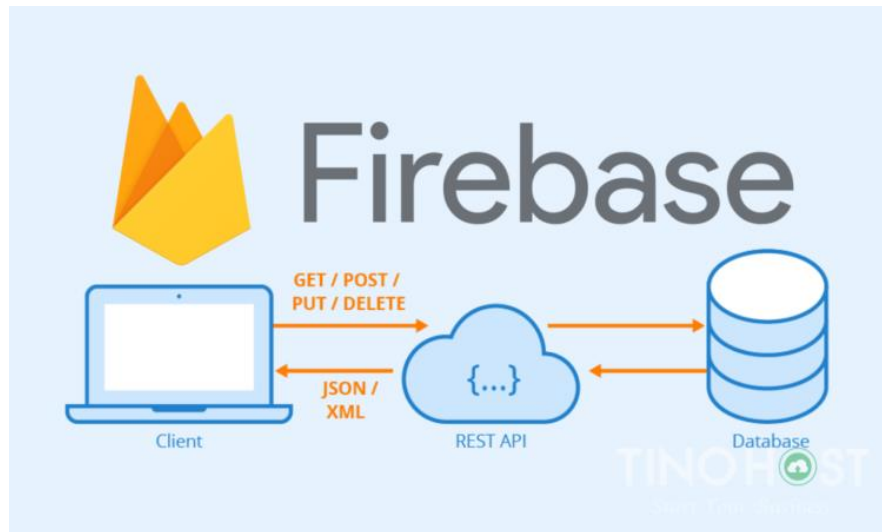
Hình 6. 3 Giao diện của thunkable

6.3 Cơ sở dữ liệu – Firebase RealTime Database

Firebase chính là một dịch vụ cơ sở dữ liệu được hoạt động ở trên nền tảng đám mây (Cloud). Đi kèm với đó là một hệ thống máy chủ mạnh mẽ của Google. Hệ thống có chức năng chính là giúp cho người dùng có thể lập trình ứng dụng thông qua cách đơn giản hóa những thao tác với các cơ sở dữ liệu.

Firebase Realtime Database có dạng một JSON đã được đồng bộ thời gian đến với tất cả các kết nối client. Để có được hoạt động này thì các lập trình viên cần phải đăng ký tài khoản ở trên Firebase. Dữ liệu ở trong database sẽ tự động cập nhật một cách liên

tục khi phát triển ứng dụng. Sau khi đã được cập nhật thì những dữ liệu này sẽ được truyền tải thông qua các kết nối SSL có 2048 bit.



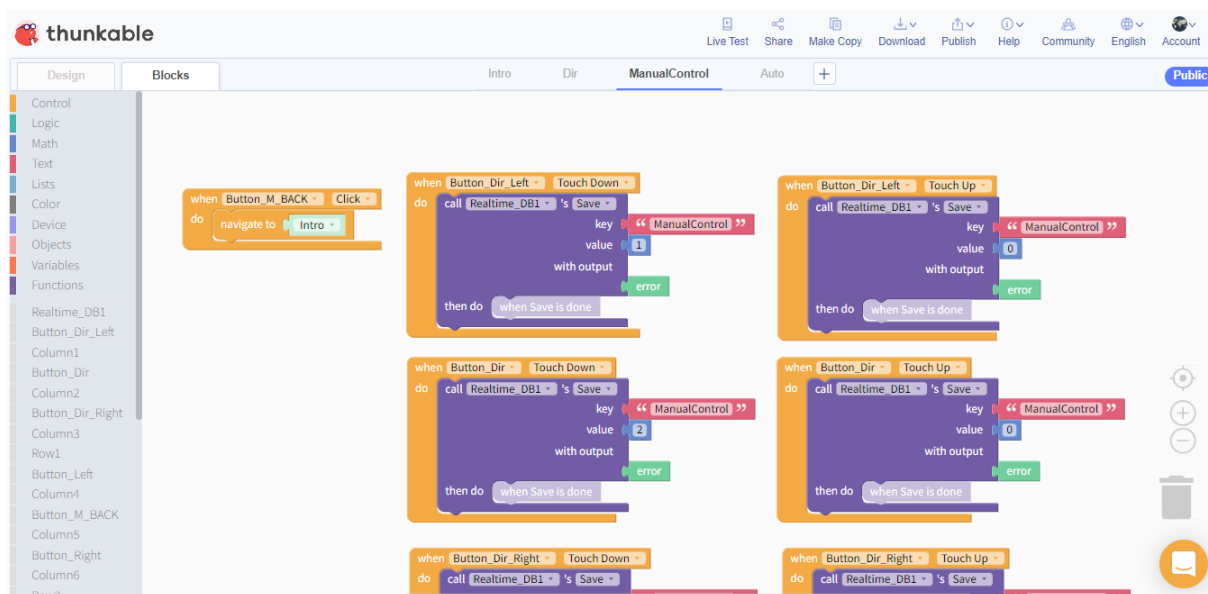
Hình 6. 4 Nguyên lý hoạt động của Firebase

6.4 Thiết kế App điều khiển



Hình 6. 5 Thiết kế App điều khiển

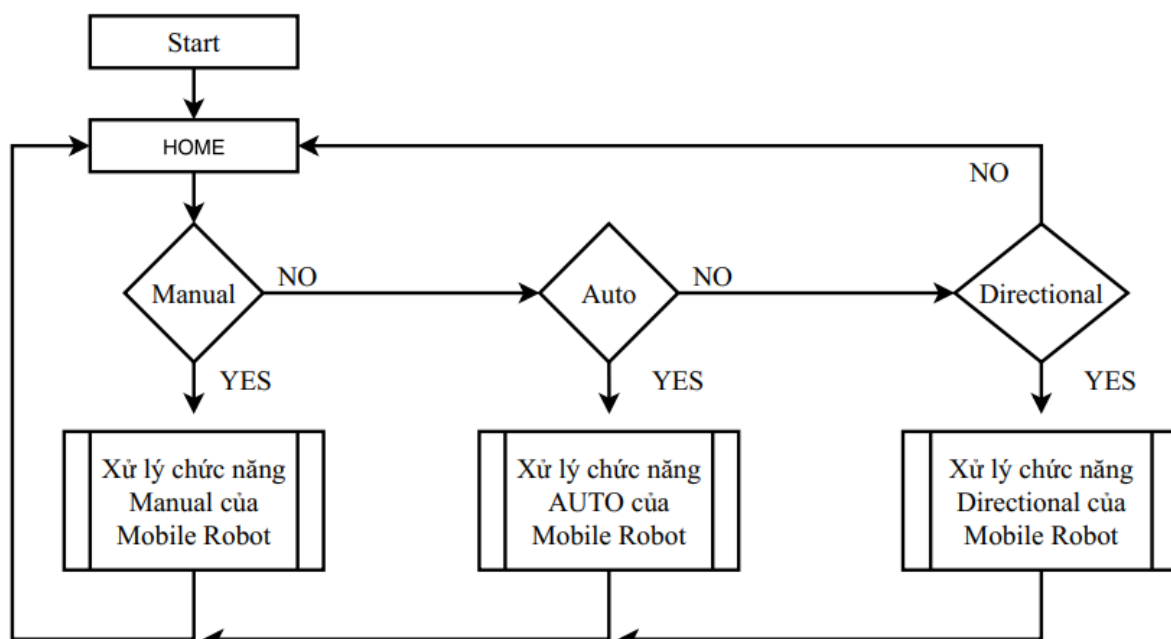
- Sử dụng thunkable để xây dựng một App điều khiển có thể đưa tín hiệu lên Firebase.
- Sau khi dữ liệu đã được đưa lên trên Firebase thì đồng thời ESP8266 sẽ GET data từ “key” của Firebase mà lấy được data. Giải thích về “key” bởi vì cấu trúc dữ liệu của Firebase là JSON vì vậy sẽ có “key” và “value” ví dụ như {“ManualControl”: “0”} thì Manual là key và 0 là value. Sau đó ESP8266 sẽ dùng UART truyền một chuỗi JSON về ba bộ cầu H và mỗi bộ cầu H sẽ có một STM32F103 để xử lý PID và điều khiển động cơ.

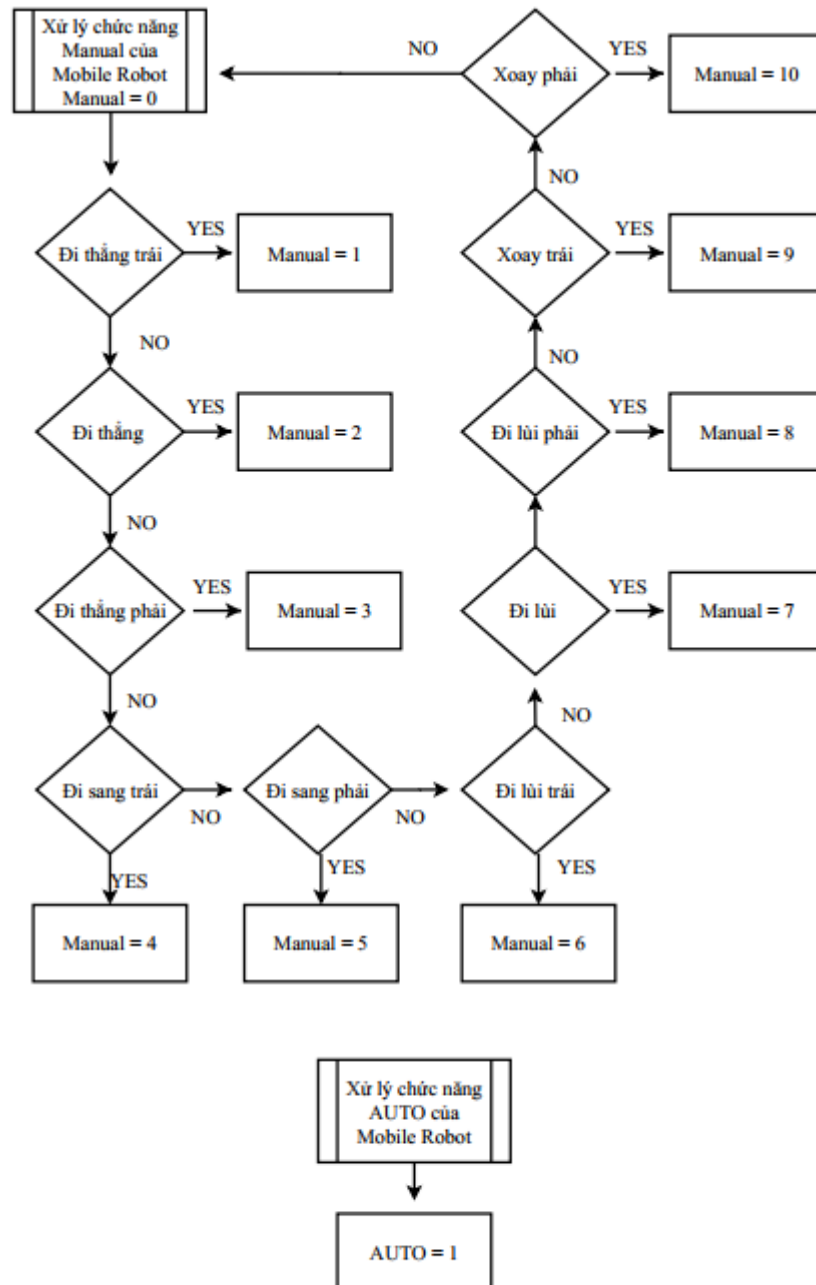


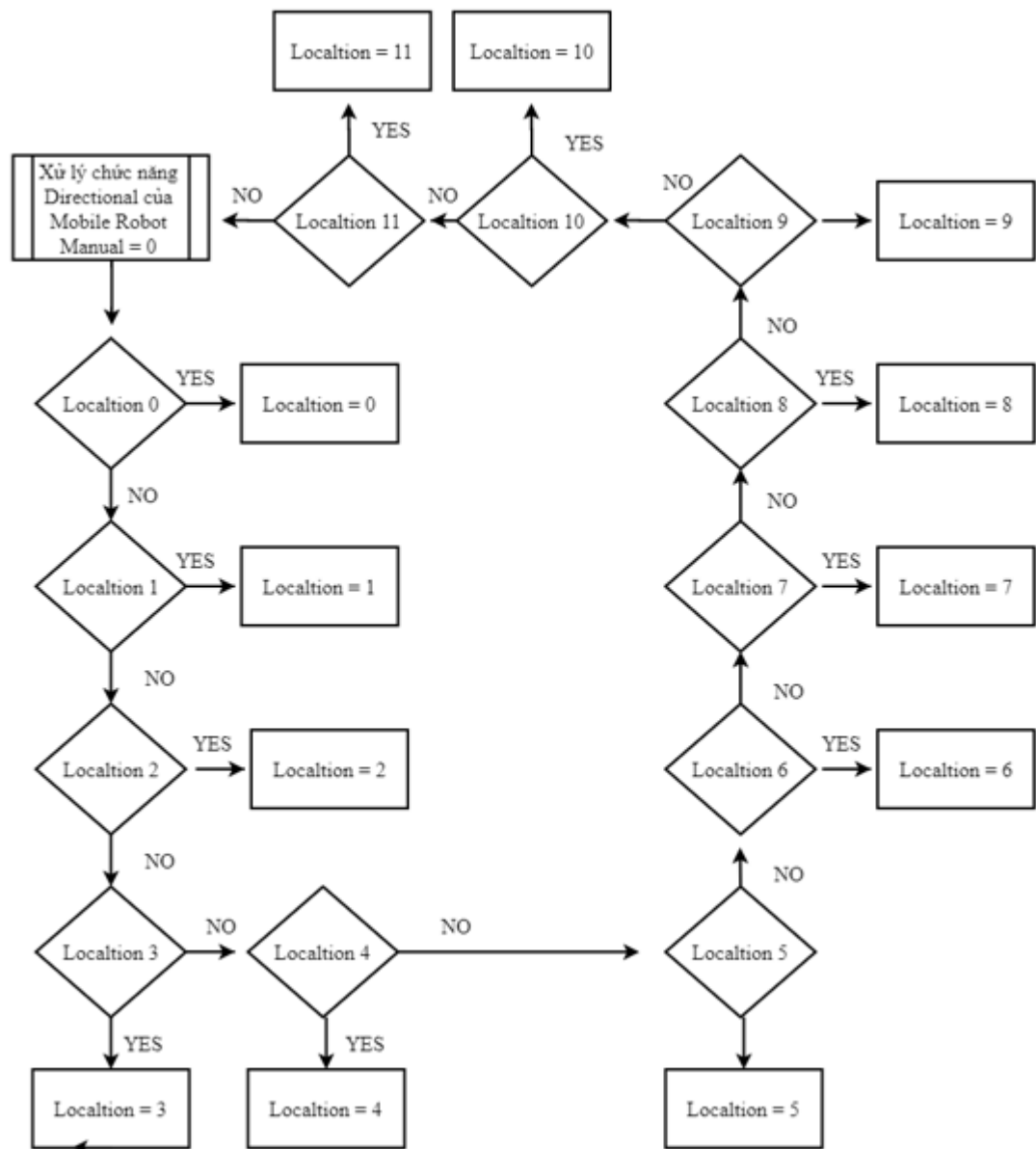
Hình 6. 6 Quá trình Save giá trị điều khiển lên Firebase



Hình 6. 7 Cấu trúc của chương trình trên Firebase

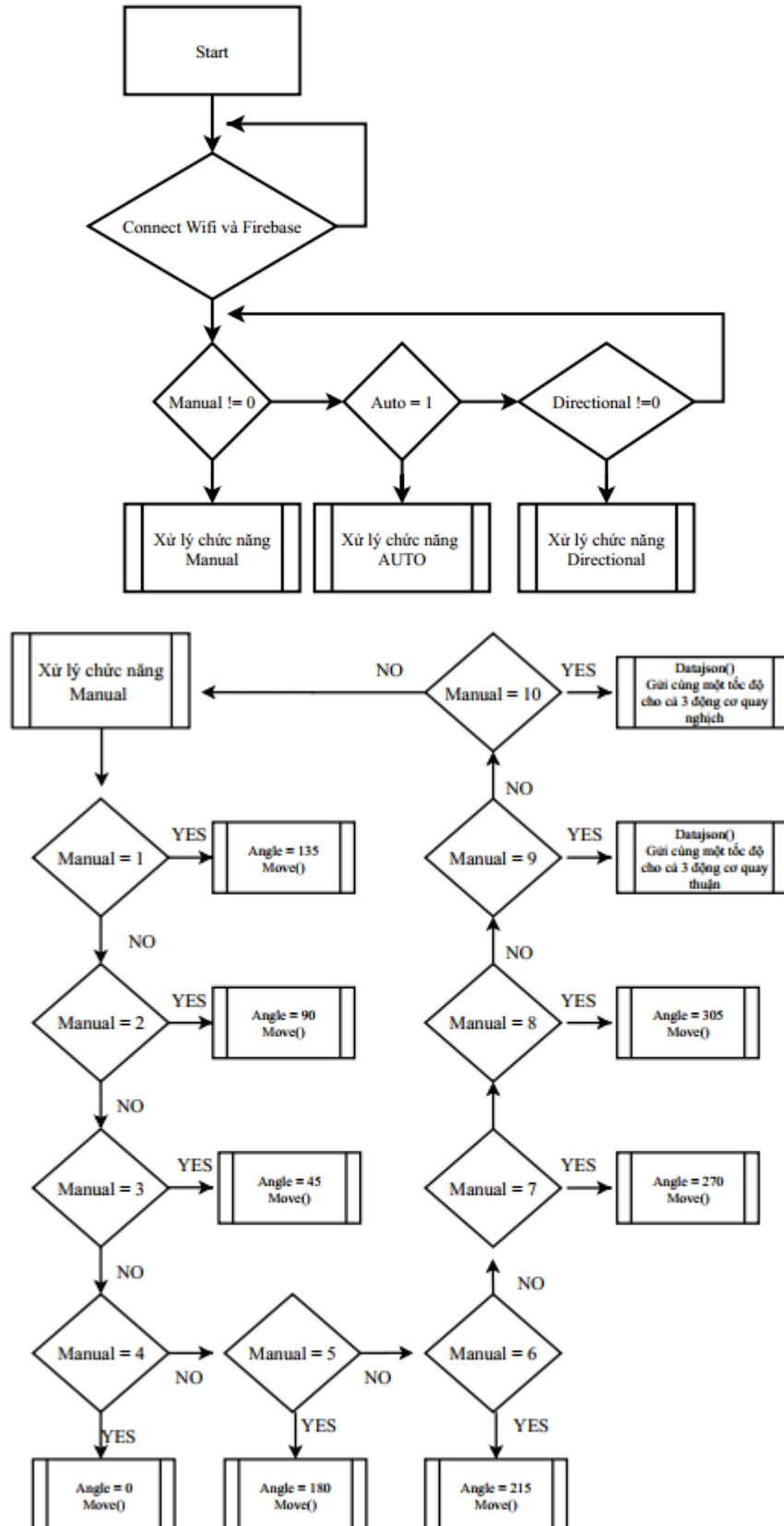


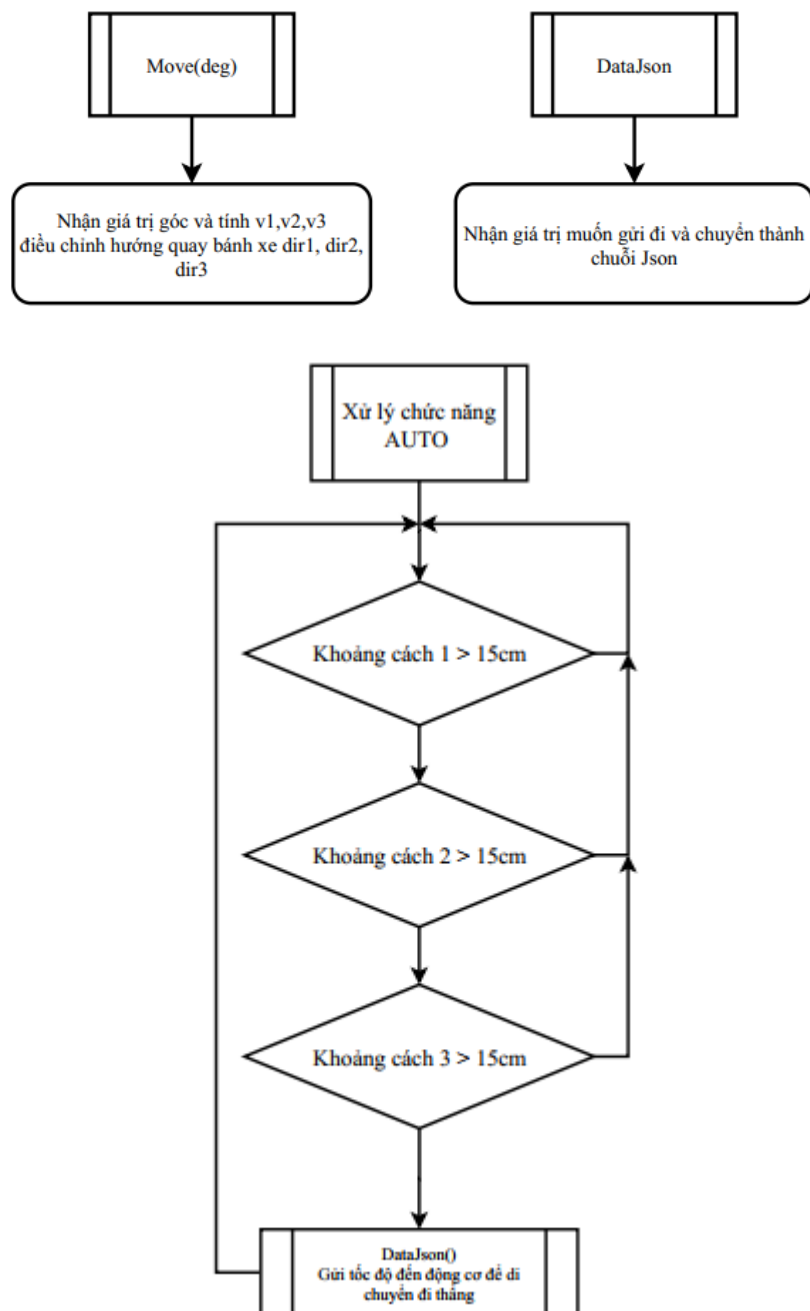


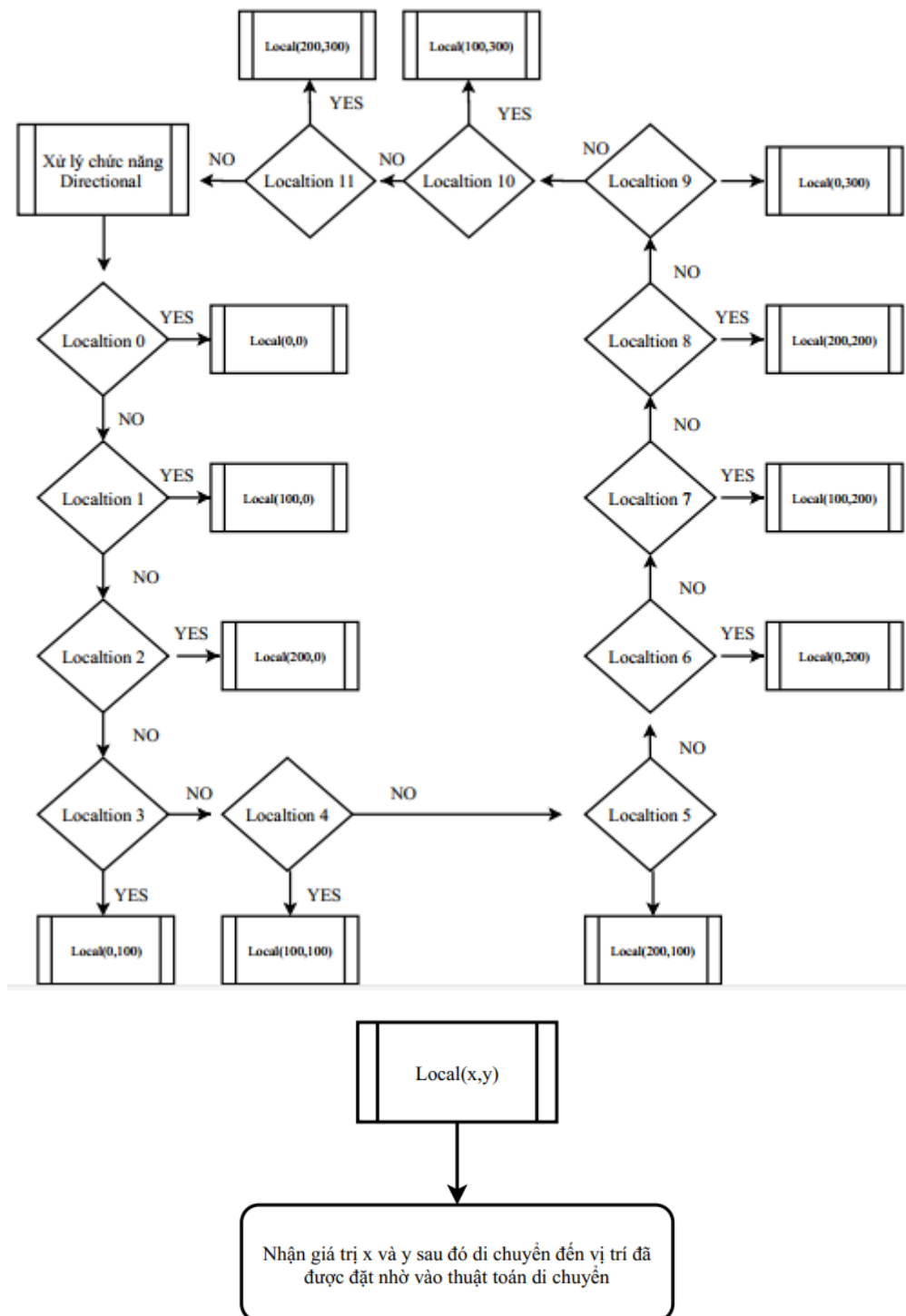


CHƯƠNG 7: XÂY DỰNG GIẢI THUẬT ĐIỀU KHIỂN

7.1. Tổng quan sơ đồ điều khiển







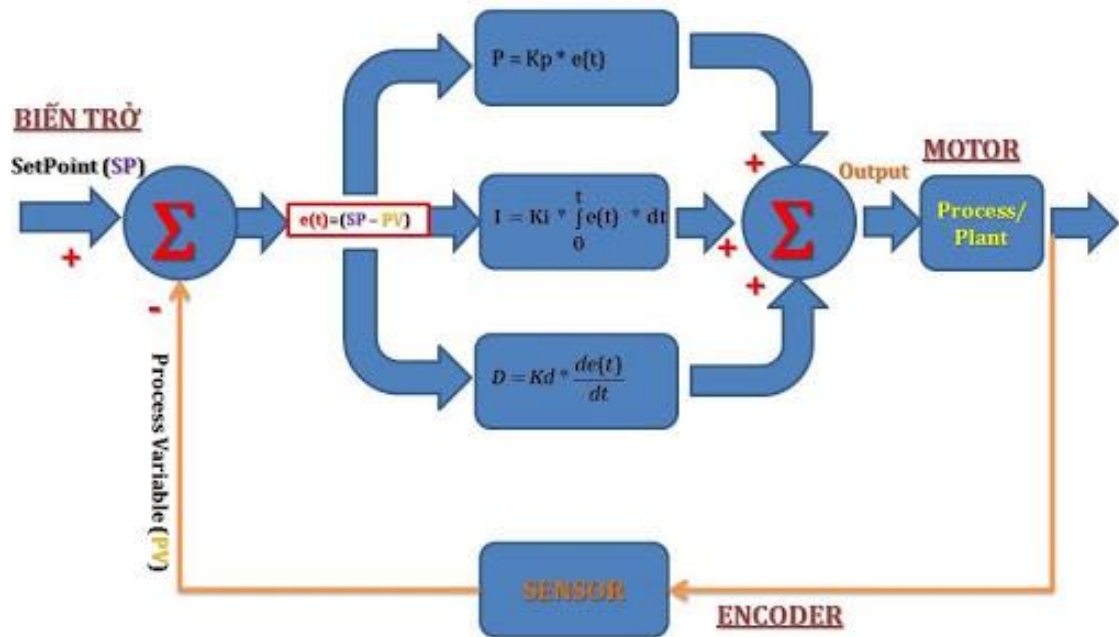
Hình 7. 1 Sơ đồ khối giải thuật di chuyển

7.2. Lý thuyết về bộ điều khiển PID

7.2.1. Định nghĩa

PID là cách viết tắt của các từ Propotional (tỉ lệ), Integral (tích phân) và Derivative (đạo hàm). Tuy xuất hiện rất lâu nhưng đến nay PID vẫn là giải thuật điều khiển được dùng nhiều nhất trong các ứng dụng điều khiển tự động. Bộ điều khiển PID là một bộ

điều khiển vòng kín được sử dụng rộng rãi trong hệ thống điện, hệ thống tự động, điện tử. Mục tiêu của bộ điều khiển PID là điều chỉnh giá trị điều khiển ở ngõ ra Output sao cho sai lệch Error $e(t) = (SP - PV)$ giữa giá trị đo được của hệ thống PV (Process Variable) với giá trị cài đặt SP (SetPoint) nhỏ nhất có thể (~ 0), đạt được sự ổn định và có đáp ứng nhanh.



7.2.2. Giải thuật PID số

$$u_k = u_k^P + u_k^D + u_k^I$$

Khâu P : $u_k^P = K_p e_k$

Khâu D : $u_k^D = K_D \left(\frac{e_k - e_{k-1}}{T} \right)$ với T : Thời gian lấy mẫu.

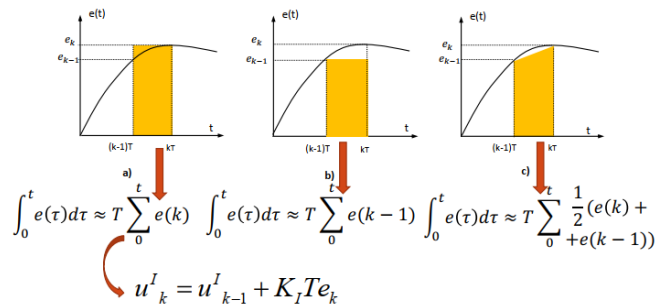
Khâu I:

Có 3 cách:

Tích phân lùi: Dùng e_k kéo về tính hình chữ nhật.

Tích phân tiến: Dùng e_{k-1} kéo về tính hình chữ nhật.

Tích phân hình thang: Nối e_k và e_{k-1} lại với nhau và tính hình thang.



7.2.3 Giải thích thư viện pid_controller.c và pid_controller.h

Vì là một thư viện đã được các chuyên gia tạo ra nên thuật toán ở đây chúng em chỉ giải thích trên phương diện những gì mình biết không được đầy đủ chỉ giải thích những hàm mà chúng em sử dụng.

Trong tệp pid.controller.h các biến được định nghĩa và tạo các kiểu dữ liệu để tệp pid_controller.c xử lý. Một số dạng chính được định nghĩa:

```
typedef enum
{
    MANUAL,          // PID tắt đi cho người dùng tùy chỉnh output
    AUTOMATIC        // PID mở lên tính và ra output
}
PIDMode;

typedef enum
{
    DIRECT,          // Quay thuận (CW)
    REVERSE          // Quay nghịch (CCW)
}
PIDDirection;

typedef struct
{
    //
    // Input to the PID Controller
    //
    float input;     // Input vào PID

    //
    // Previous input to the PID Controller
    //
    float lastInput; // Input trước đó của PID

    //
    // Output of the PID Controller
    //
    float output;    // Output của PID
```

```
// Gain constant values that were passed by the user
// These are for display purposes
//
float dispKp;
float dispKi;
float dispKd;

//
// Gain constant values that the controller alters for
// its own use
//
float alteredKp;
float alteredKi;
float alteredKd;

//
// The Integral Term
//
float iTerm;

//
// The interval (in seconds) on which the PID controller
// will be called
//
float sampleTime;

//
// The values that the output will be constrained to
//
float outMin;
float outMax;

//
// The user chosen operating point
//
float setpoint;

//
// The sense of direction of the controller
// DIRECT: A positive setpoint gives a positive output
// REVERSE: A positive setpoint gives a negative output
//
PIDDirection controllerDirection;

//
// Tells how the controller should respond if the user has
// taken over manual control or not
// MANUAL: PID controller is off.
// AUTOMATIC: PID controller is on.
//
PIDMode mode;
```

Một số dạng biến trên được định nghĩa bằng C++ và tệp pid_controller.c sẽ xử lý một số vấn đề có liên quan đến PID như sau.

Hàm này dùng để giới hạn giá trị lại.

```
//
#define CONSTRAIN(x,lower,upper) ((x)<(lower)?(lower):((x)>(upper)?(upper):(x)))
```

PIDInit sẽ chọn các giá trị cho PID

PIDControl *pid : Chọn điều khiển vị trí hay là điều khiển tốc độ.

float kp, **float** ki, **float** kd: Chọn thông số cho bộ điều khiển.

float sampleTimeSeconds: Chọn thời gian lấy mẫu.

float minOutput, **float** maxOutput: Giới hạn đầu ra và đầu vào.

PIDMode mode: Chọn Manual hay là Automatic.

PIDDirection controllerDirection chọn chiều cho động cơ.

```
void PIDInit(PIDControl *pid, float kp, float ki, float kd,
             float sampleTimeSeconds, float minOutput, float maxOutput,
             PIDMode mode, PIDDirection controllerDirection)
{
    pid->controllerDirection = controllerDirection;
    pid->mode = mode;
    pid->iTerm = 0.0f;
    pid->input = 0.0f;
    pid->lastInput = 0.0f;
    pid->output = 0.0f;
    pid->setpoint = 0.0f;

    if(sampleTimeSeconds > 0.0f)
    {
        pid->sampleTime = sampleTimeSeconds;
    }
    else
    {
        // If the passed parameter was incorrect, set to 1 second
        pid->sampleTime = 1.0f;
    }

    PIDOutputLimitsSet(pid, minOutput, maxOutput);
    PIDTuningsSet(pid, kp, ki, kd);
}
```

Các giá trị sẽ được trở vào các giá trị của chúng. Trong đó có hàm PIDtuningsSet nhiệm vụ của hàm này là sẽ lấy các giá trị đã được set:

```
pid->dispKp = kp;
pid->dispKi = ki;
pid->dispKd = kd;
```

Sau xử lý theo PID số:

```
pid->alteredKp = kp;
pid->alteredKi = ki * pid->sampleTime;
pid->alteredKd = kd / pid->sampleTime;
```

Và xử lý hướng cho động cơ.

```
void
PIDTuningsSet(PIDControl *pid, float kp, float ki, float kd)
{
    // Check if the parameters are valid
    if(kp < 0.0f || ki < 0.0f || kd < 0.0f)
    {
        return;
    }

    // Save the parameters for displaying purposes
    pid->dispKp = kp;
    pid->dispKi = ki;
    pid->dispKd = kd;

    // Alter the parameters for PID
    pid->alteredKp = kp;
    pid->alteredKi = ki * pid->sampleTime;
    pid->alteredKd = kd / pid->sampleTime;

    // Apply reverse direction to the altered values if necessary
    if(pid->controllerDirection == REVERSE)
    {
        pid->alteredKp = -(pid->alteredKp);
        pid->alteredKi = -(pid->alteredKi);
        pid->alteredKd = -(pid->alteredKd);
    }
}
```

Hàm PIDCompute là hàm quan trọng nhất trong thư viện hàm này sẽ tính toán và cho ra output.

`error = (pid->setpoint) - (pid->input);` : Tính sai số giữa giá trị đo của hệ thống và giá trị đặt.

`pid->iTerm += (pid->alteredKi) * error;` : xử lý khâu I bằng cách cộng dồn giá trị như lý thuyết của PID số ở trên.

`pid->iTerm = CONSTRAIN((pid->iTerm), (pid->outMin), (pid->outMax));` : Giải quyết vấn đề giới hạn giá trị của khâu I.

`dInput = (pid->input) - (pid->lastInput);` : Ở bộ điều khiển của hệ thống này thư viện sử dụng đạo hàm của input thay vì sử dụng đạo hàm của sai số (error) như PID số.

`pid->output = (pid->alteredKp) * error + (pid->iTerm) - (pid->alteredKd) * dInput;` : Tính ra OUTPUT bằng các giá trị trên công cả ba khâu P,D,I dạng rõ ràng hơn của phương trình trên như sau:

Khâu P: `(pid->alteredKp) * error = kp * error`

Khâu I: `(pid->iTerm) = (pid->iTerm) + ki * pid->sampleTime * error`

Khâu D: `(pid->alteredKd) * dInput = kd / pid->sampleTime * dInput`

Chỉ khác phương trình trên lý thuyết là khâu D thay vì dùng đạo hàm của sai số ta sử dụng đạo hàm của Input.

```

bool
PIDCompute(PIDControl *pid)
{
    float error, dInput;

    if(pid->mode == MANUAL)
    {
        return false;
    }

    // The classic PID error term
    error = (pid->setpoint) - (pid->input);

    // Compute the integral term separately ahead of time
    pid->iTerm += (pid->alteredKi) * error;

    // Constrain the integrator to make sure it does not exceed output bounds
    pid->iTerm = CONSTRAIN( (pid->iTerm), (pid->outMin), (pid->outMax) );

    // Take the "derivative on measurement" instead of "derivative on error"
    dInput = (pid->input) - (pid->lastInput);

    // Run all the terms together to get the overall output
    pid->output = (pid->alteredKp) * error + (pid->iTerm) - (pid->alteredKd) * dInput;

    // Bound the output
    pid->output = CONSTRAIN( (pid->output), (pid->outMin), (pid->outMax) );

    // Make the current input the former input
    pid->lastInput = pid->input;

    return true;
}

```

7.3. Giải thuật di chuyển đa hướng Mobile Robot

$$\begin{aligned}
 v_1(t) &= -v(t)\sin(\pi/3) + vn(t)\cos(\pi/3) + \omega_1(t)d \\
 v_2(t) &= -vn(t) + \omega_1(t)d
 \end{aligned}
 \tag{4.4}$$

$$v_3(t) = v(t)\sin(\pi/3) + vn(t)\cos(\pi/3) + \omega_1(t)d$$

Theo như công thức (4.4) ta đã thu được 3 hệ số $v_1(t)$, $v_2(t)$, $v_3(t)$ với $v(t) = \cos(\alpha)$ và $vn(t) = \sin(\alpha)$, α là góc mà Mobile Robot di chuyển và $\omega_1(t) = 0$ ta suy ra được công thức:

$$v_1(t) = -\cos(\alpha)\sin(\pi/3) + \sin(\alpha)\cos(\pi/3)$$

$$v_2(t) = -\sin(\alpha)$$

$$v_3(t) = \cos(\alpha)\sin(\pi/3) + \sin(\alpha)\cos(\pi/3)$$

Sau khi tính được hệ số vận tốc ta cần thêm một vận tốc tiếp tuyến và truyền vào thuật toán một góc α .

Về hướng quay của bánh xe thì ta chỉ cần xét dấu của hệ số nếu dương thì quay thuận và ngược lại.

```

void Move(int deg)
{
    vx = cos(deg * PI / 180);
    vy = sin(deg * PI / 180);

    v1 = vx;
    v2 = sqrt(3) / 2 * vy - 0.5 * vx;
    v3 = -(sqrt(3) / 2 * vy + 0.5 * vx);

    DIR1 = v1 < 0 ? 1 : 0;
    DIR2 = v2 < 0 ? 1 : 0;
    DIR3 = v3 < 0 ? 1 : 0;

    v1 = Vtt * abs(v1);
    v2 = Vtt * abs(v2);
    v3 = Vtt * abs(v3);

    DC1 = v1;
    DC2 = v2;
    DC3 = v3;
}

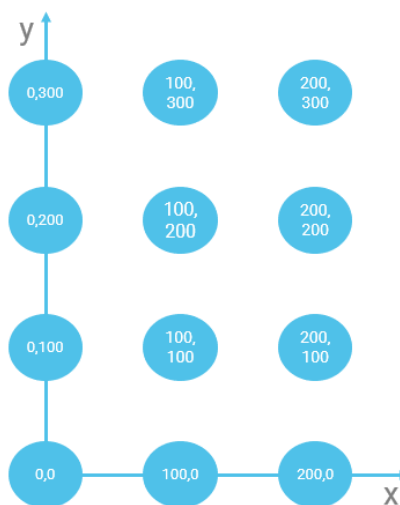
```

Hình 7. 2 Ví dụ minh họa cho thuật toán

7.4. Giải thuật di chuyển đến vị trí cố định của Mobile Robot



Xây dựng thuật toán:



Hình 7. 3 Mô phỏng vị trí trên khu vực cố định

Gọi x_n , y_n , x_o và y_o là tọa độ điểm x di chuyển tới, y di chuyển tới, x hiện tại và y hiện tại.

Trình tự di chuyển như sau: Di chuyển theo trục y -> Quay ngược hoặc thuận 90 độ -> Di chuyển theo trục x.

Với quãng đường trục y được tính như sau: $y = y_n - y_o$ nếu $y > 0$ thì đi thẳng và ngược lại.

Góc quay được tính như sau nếu: $x_n - x_o > 0$ thì quay thuận 90 độ và ngược lại.

Quãng đường di chuyển trục x được tính như sau: $x = x_n - x_o$

Ví dụ: Mobile Robot đang ở vị trí Home (0,0) và muốn di chuyển đến vị trí (200,100):

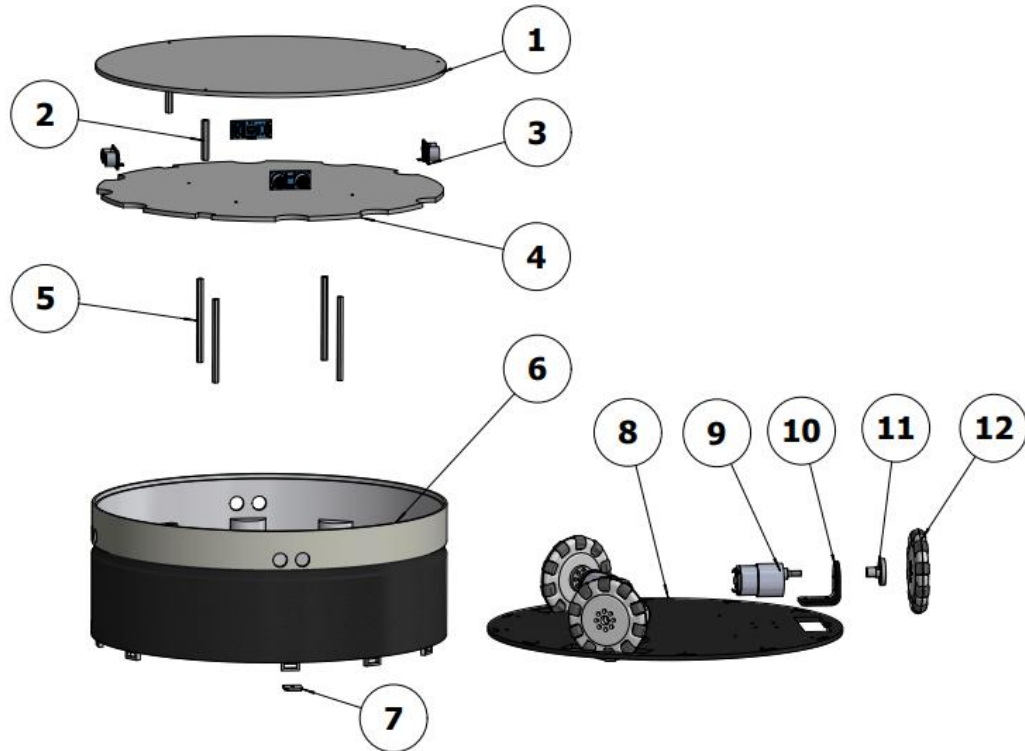
Ta tính được

- Quãng đường trục y $= y_n - y_o = 100 - 0 = 100$ (cm)
- Quãng đường trục x $= x_n - x_o = 200 - 0 = 200$ (cm)
- Vì $x > 0$ nên Robot sẽ quay thuận 90 độ.

Robot di chuyển như sau : Quay thuận theo trục y 100 cm sau đó quay thuận 90 độ và cuối cùng là quay thuận 200 cm để đến vị trí (200,100).

CHƯƠNG 8: KẾT QUẢ THỰC NGHIỆM

8.1. Thi công mô hình

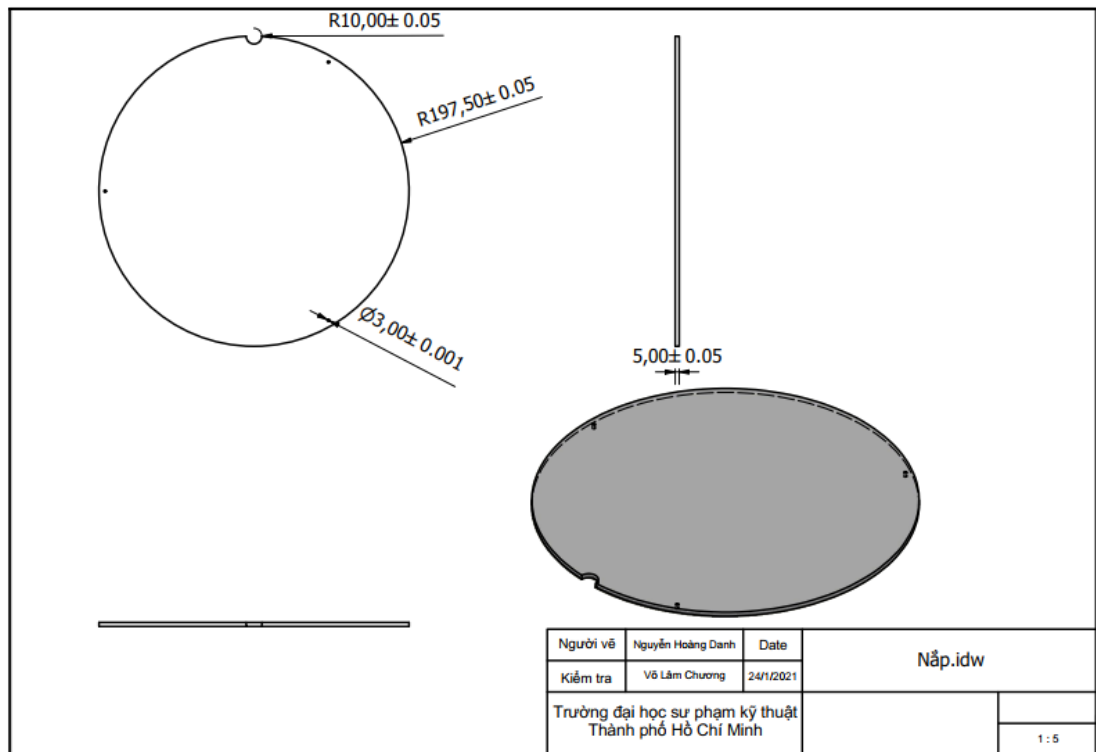


Hình 8.1 Bản vẽ phân rã của Robot

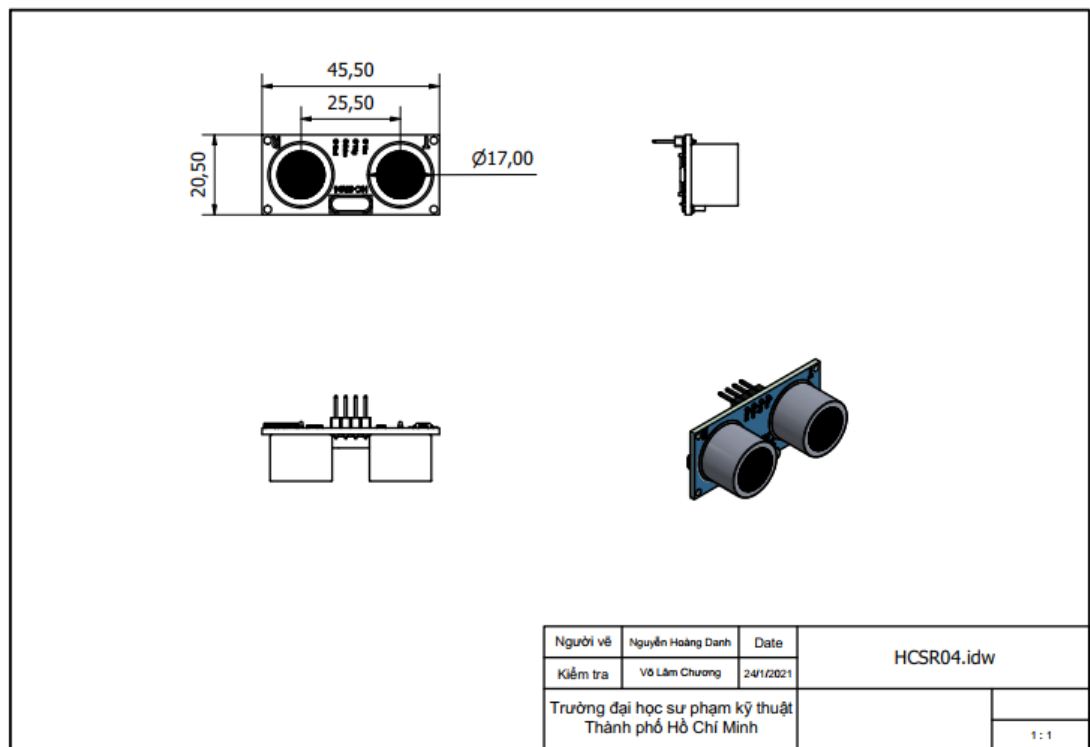
Chi tiết	Tên	Chức năng
1	Nắp	Che phần bên trong robot đồng thời bảo vệ robot
2	Trụ đỡ 45mm	Đỡ nắp trên
3	Cảm biến	Cảm biến siêu âm SR-HC04 giúp robot tránh vật cản
4	Nắp ngăn	Tăng độ thẩm mỹ cho robot
5	Trụ đỡ 100mm	Đỡ nắp che
6	Thân	Chứa các bộ phận hoạt động của robot
7	Then	Liên kết cho đế và thân robot
8	Đế	Giữ bánh xe và đỡ các phần điện của robot
9	Động cơ	Giúp robot chuyển động
10	Gá đỡ động cơ	Đỡ động cơ dễ gá trên đế
11	Chốt	Liên kết động cơ và bánh xe
12	Bánh xe	Giúp robot chuyển động

Hình 8. 2 Bảng các chi tiết và chức năng

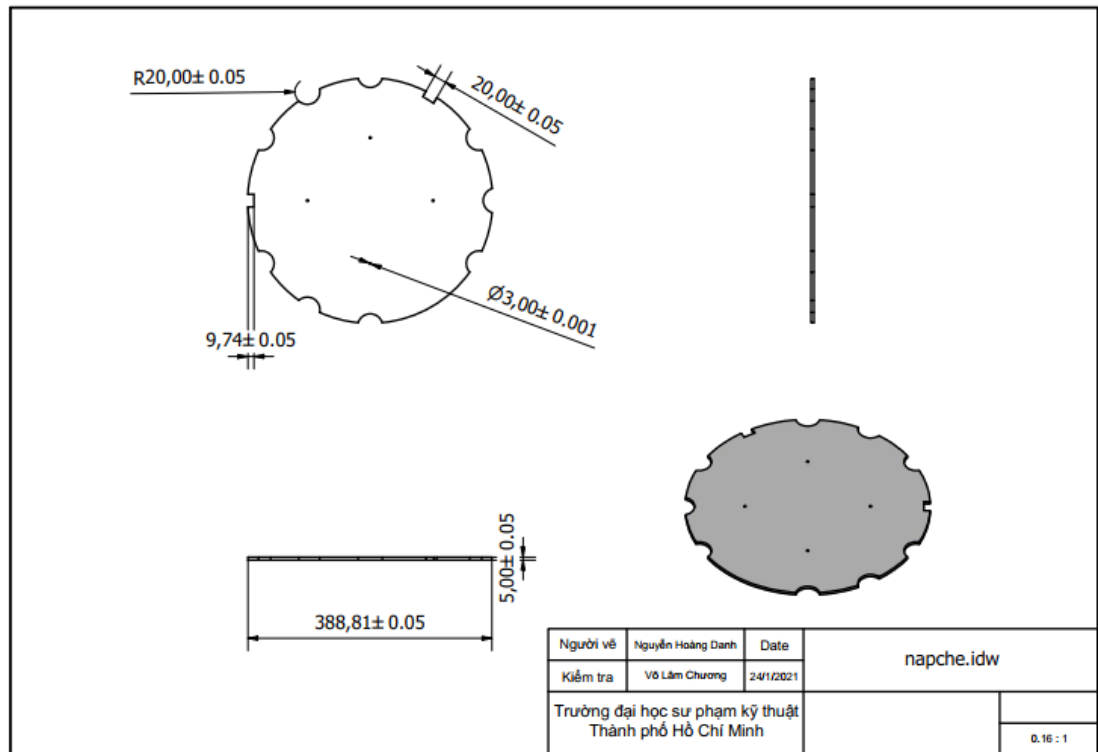
Chi tiết 1:



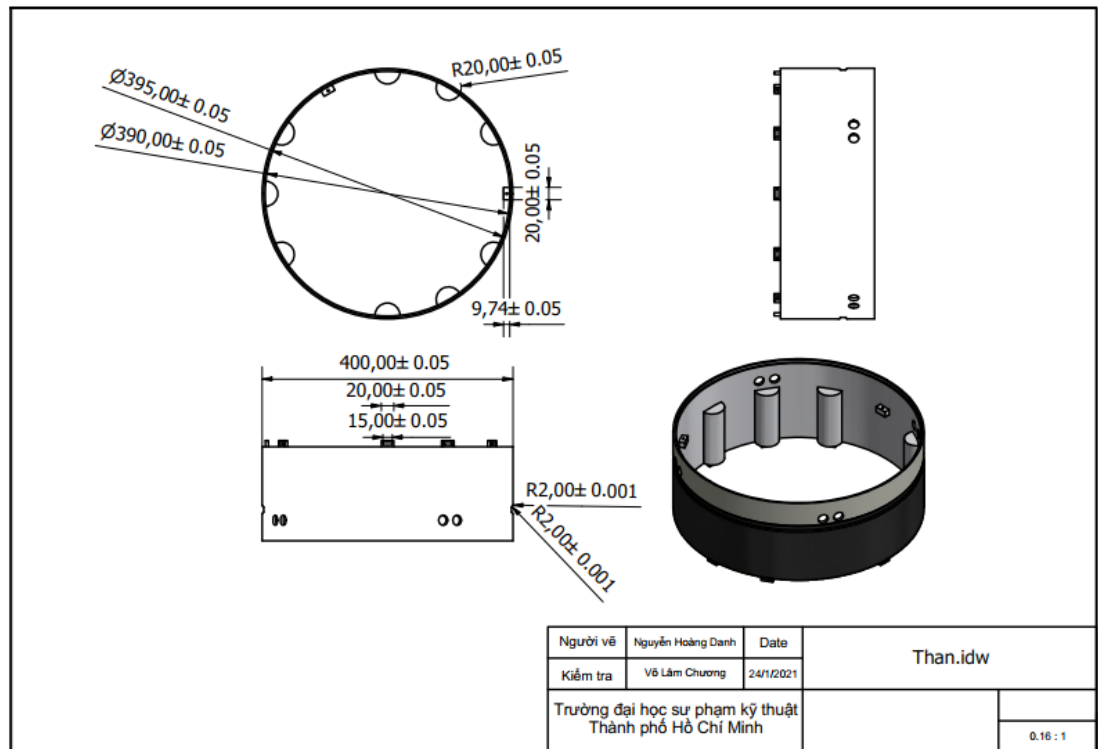
Chi tiết 3



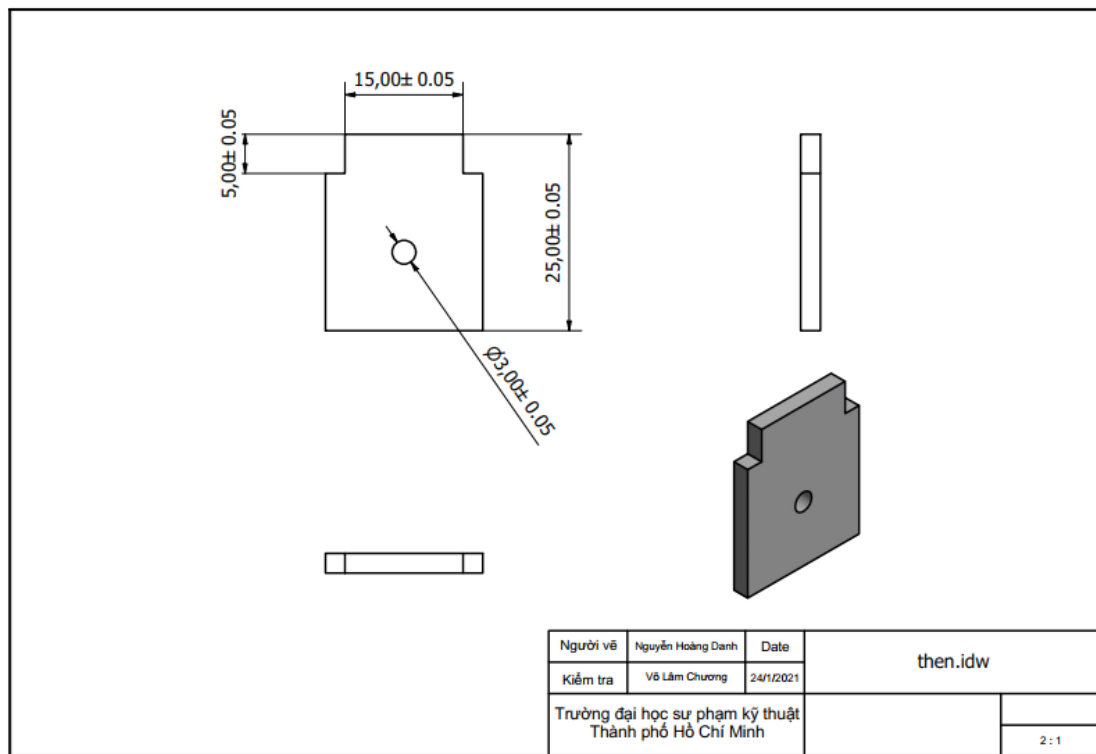
Chi tiết 4:



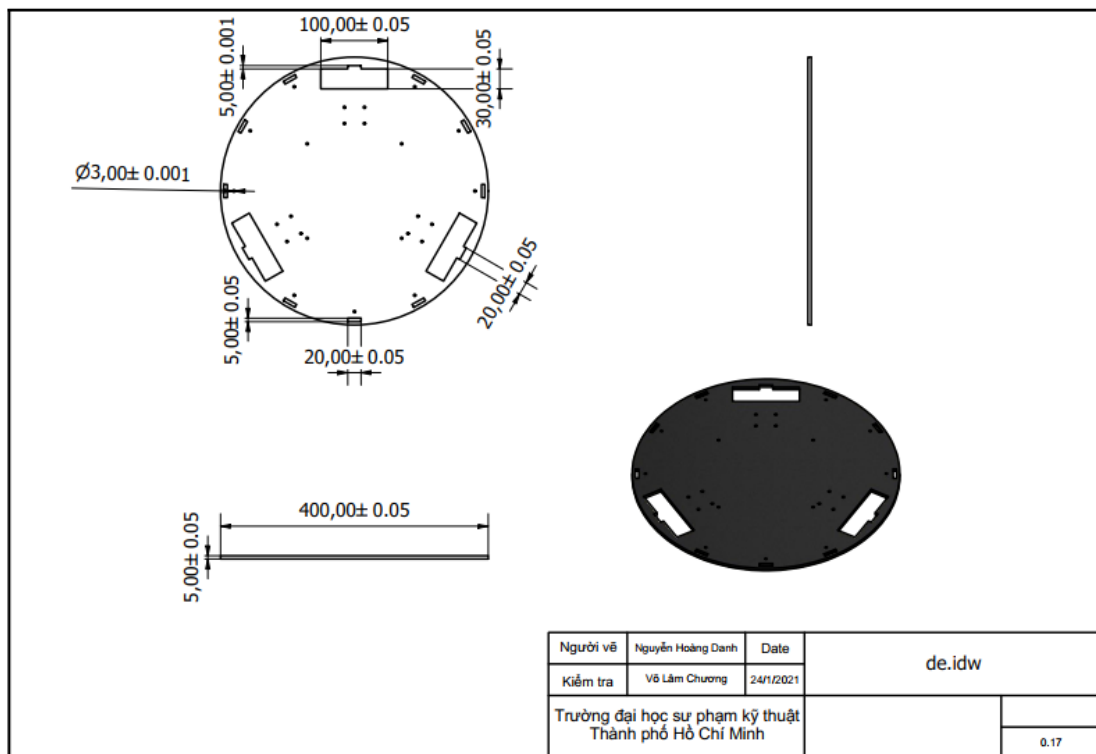
Chi tiết 6:



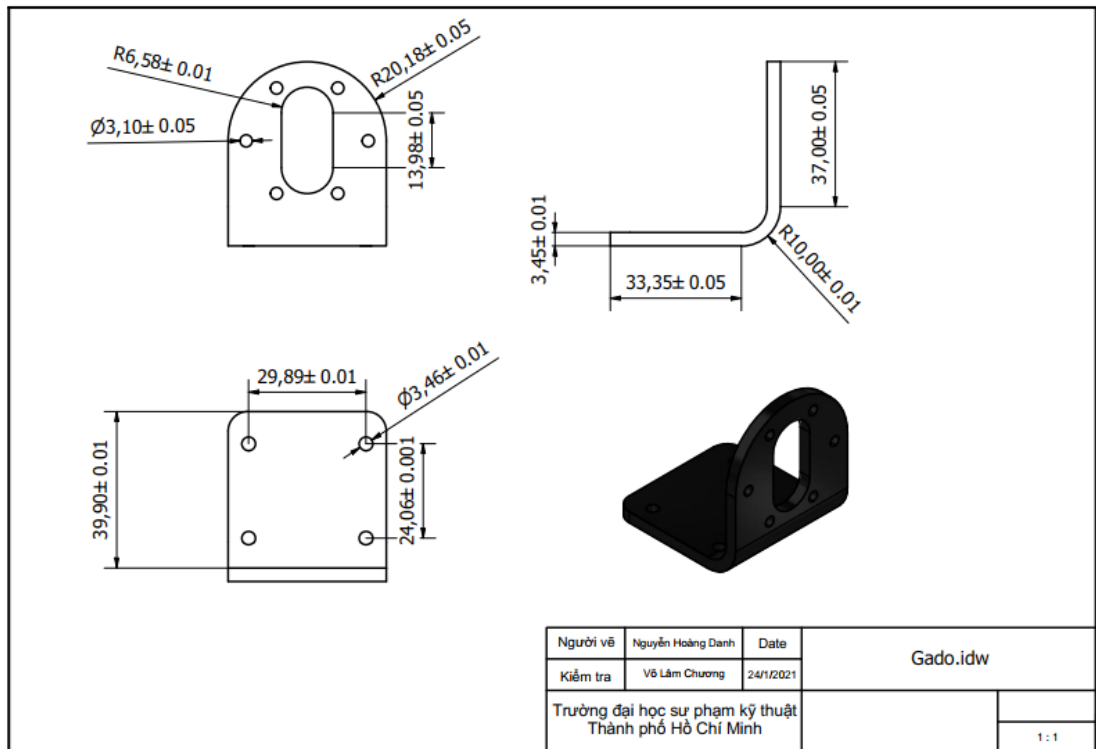
Chi tiết 7:



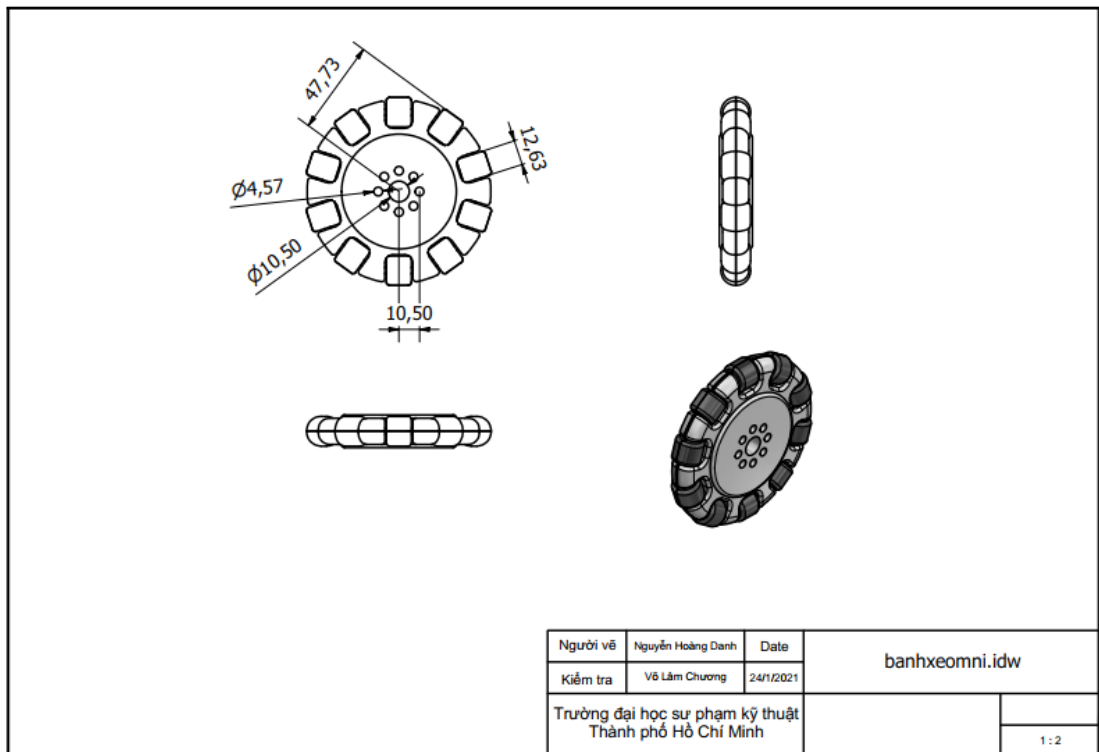
Chi tiết 8:



Chi tiết 10:



Chi tiết 12:



Mô hình sản phẩm hoàn thành:



Hình 8. 3 Mobile Robot

8.2. Kết quả thu được cảm biến gia tốc MPU9250

Kết quả góc X:

Watch 1			
Name	Value	Type	
Angle_X	45	uint	
Angle_Y	0	uint	
<Enter expression>			

Kết quả góc Y:

Watch 1			
Name	Value	Type	
Angle_X	0	uint	
Angle_Y	35	uint	
<Enter expression>			

Kết quả này thu được khi nghiêng góc $X = 45^\circ - Y = 0$ và $X = 0 - Y = 35^\circ$.

8.3. Kết quả thu được từ cảm biến khoảng cách US-015



Hình 8. 4 Thực nghiệm US-015

- Kết quả cảm biến 1

Name	Value	Type
Distance	8.4829998	float
Distance1	0	float
Distance2	0	float

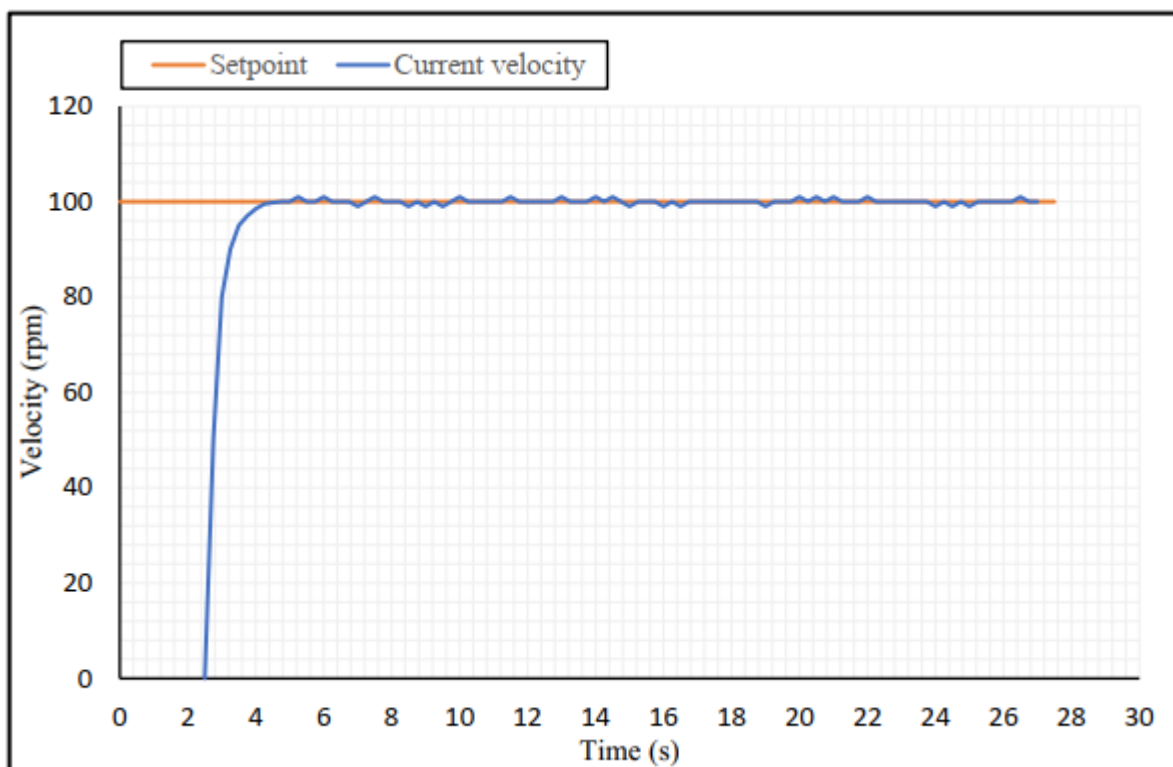
- Kết quả cảm biến 2

Distance	0	float
Distance1	8.15999985	float
Distance2	0	float

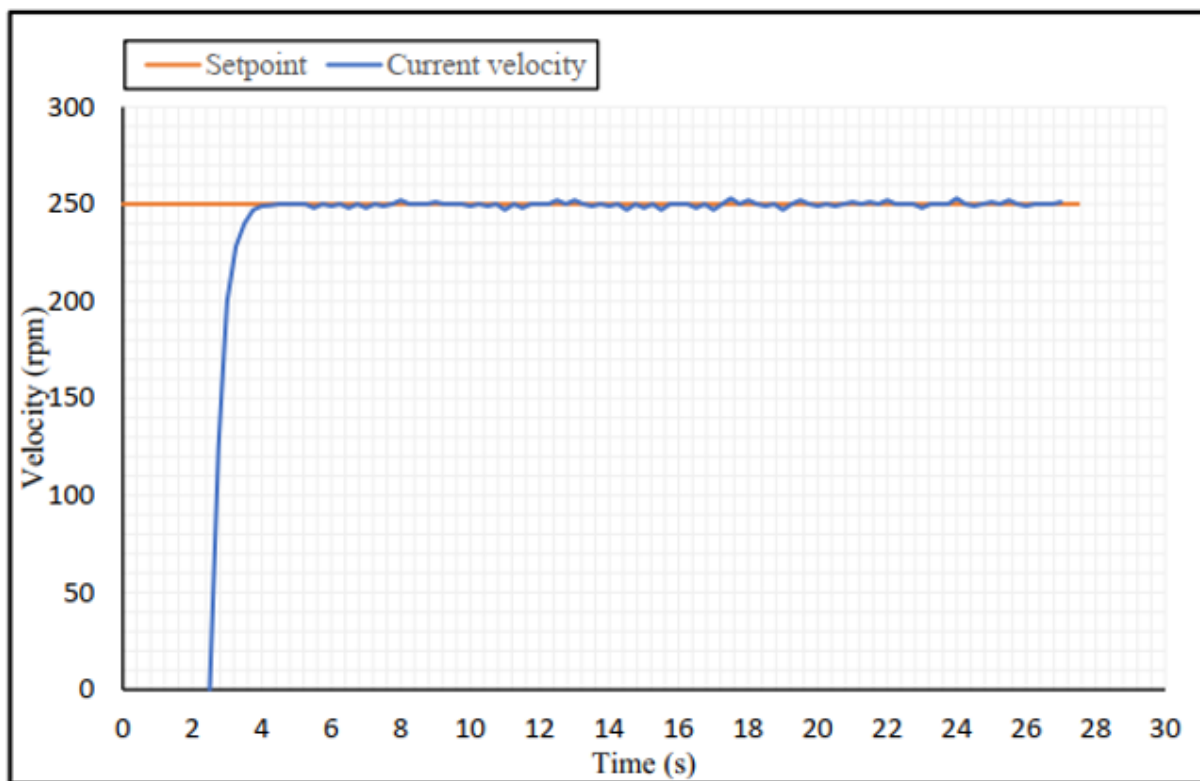
- Kết quả cảm biến 3

Watch 1		
Name	Value	Type
Distance	0	float
Distance1	0	float
Distance2	8.14299965	float

8.4. Kiểm tra tính ổn định của bộ PID



Hình 8. 5 Đồ thị hiển thị vận tốc theo thời gian khi Setpoint là 100rpm



Hình 8. 6 Đồ thị hiển thị vận tốc theo thời gian khi Setpoint là 250rpm

8.5. Phần mềm App Mobile Robot

App được xây dựng và thiết kế dựa trên phần mềm Thunkable bằng cách đưa tín hiệu lên Firebase. Sau khi dữ liệu đã được đưa lên trên Firebase thì đồng thời ESP8266 sẽ GET data từ “key” của Firebase mà lấy được data

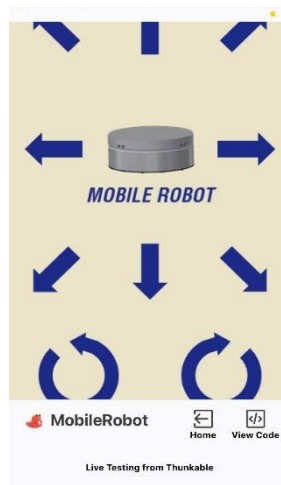


Hình 8. 7 Giao diện điều khiển app

8.6. Di chuyển đa hướng Mobile Robot

Một trong những chế độ có thể kể đến của Mobile Robot đó là có thể di chuyển đa hướng dựa trên việc sử dụng bánh xe Omni. Một loại bánh xe được dùng khá phổ biến trong các dòng Mobile Robot.

Ở chế độ này có thể điều khiển Robot di chuyển theo hướng mà mình người dùng muốn di chuyển giúp cho người dùng có thể điều khiển Robot đến vị trí mà mình mong muốn cũng như là những nơi khó khăn trong việc định vị.



Hình 8. 8 Chế độ di chuyển đa hướng

8.7. Di chuyển tự động

Một trong những chế độ làm nên sự thuận lợi của Mobile Robot đó là di chuyển tự động trong khu vực được kiểm soát, có thể nhận biết được vật và tránh được vật cản.

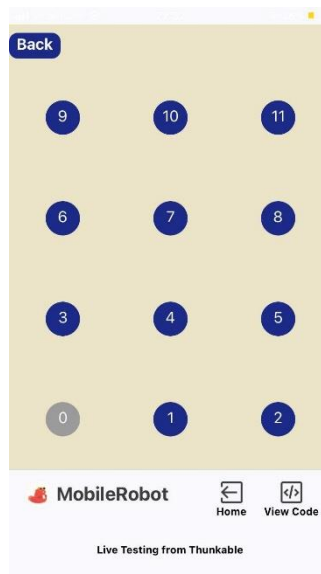
Có thể ứng dụng chế độ này trong những loại Mobile Robot giúp việc nhà, vận chuyển hàng hóa trong nhà máy và những khu công nghiệp.



Hình 8. 9 Chế độ di chuyển tự động

8.8. Di chuyển theo vị trí cố định

Trên màn hình của App sẽ thể hiện 12 điểm cố định với khoảng cách bằng nhau được đánh dấu sẵn. Robot có thể tự di chuyển đến vị trí mà ta mong muốn. Chế độ này được sử dụng để ứng dụng cho việc sạc pin tự động hoặc di chuyển hàng hóa đến địa điểm cố định.



Hình 8. 10 Chế độ di chuyển theo vị trí cố định

8.9. Thực nghiệm:

Giới hạn của việc thực nghiệm:

- Robot chỉ thực hiện công việc giám sát một khu vực hẹp.
- Robot còn nhiều xảy ra ở các cảm biến
- Chưa thực hiện được trên phần cứng

CHƯƠNG 9: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

9.1. Về lý thuyết

Về cơ bản, chúng em đã hoàn thành việc lựa chọn cấu hình robot, cơ cấu truyền động, cơ cấu tác động. Thiết kế được robot trên phần mềm Inventer.

Giải được bài toán động học của robot.

Viết được code điều khiển chuyển động robot, đọc các cảm biến hoạt động trên Robot.

9.2. Về thực nghiệm

Hoàn thành việc gia công các chi tiết và lắp ráp hoàn thiện robot.

Thu được các giá trị mong muốn từ các ngoại vi.

Chưa thực nghiệm điều khiển trên robot.

9.3. Ưu điểm, nhược điểm

Ưu điểm:

- Cơ cấu cứng vững, di chuyển tốt, đổi hướng ổn định.
- Hoạt động êm.
- Robot nhỏ gọn.

Nhược điểm:

- Độ chính xác chưa cao, trong quá trình di chuyển còn rung lắc
- Độ thẩm mỹ chưa cao.

9.4. Hướng phát triển

Đề tài có tính mở, có thể tiếp tục nghiên cứu phát triển để áp dụng vào thực tế, ứng dụng trong các môn khác khó hơn như phục vụ, chuyển hàng, ...

Có thể thiết kế gọn và đẹp hơn nữa.

Có thể trang bị thêm cho robot các cảm biến khác, hệ thống quan sát,.. để robot chạy tự động. Tối ưu hóa code điều khiển hoặc bộ vi xử lý, để robot hoạt động nhanh nhạy và linh hoạt hơn.

TÀI LIỆU THAM KHẢO

1. Mobile robot
https://en.wikipedia.org/wiki/Mobile_robot
2. Giới thiệu Mobile robot
https://baoanjsc.com.vn/tin-hang/mobile-robots-lagi_2_1_31453_vn.aspx
3. Bài tập lớn Tự động hóa: Mobile robot ba bánh - ba motor - CĐ Kỹ thuật Cao Thắng
<https://tailieu.vn/doc/bai-tap-lon-tu-dong-hoa-mobile-robot-ba-banh-ba-motor-cd-ky-thuat-cao-thang-1683320.html>
4. Đồ án tốt nghiệp: Mobile robot
<https://luanvan123.info/threads/mobile-robot-phuc-vu-ban.122472/>
5. Types and Applications of Autonomous Mobile Robots (AMRs)
<https://www.conveyco.com/types-and-applications-of-amrs/>
6. Tracked mobile robot:
<https://123doc.net/document/2320101-do-an-tot-nghiep-tracked-mobile-robot.htm>
7. The future of Smart Logistics Automation:
<https://www.roboticstomorrow.com/article/2020/10/the-future-of-smart-logistics-automation/15857/>
8. Analysis and Control of Mobile Robots in Various Environmental Conditions
<https://core.ac.uk/download/pdf/53189138.pdf>
9. A simulation and experimental study on wheeled mobile robot path control in road roundabout environment
<https://journals.sagepub.com/doi/pdf/10.1177/1729881419834778>
10. sách Wheeled Mobile Robot 1st Edition
11. Tài liệu bộ lọc Kalman
<https://tailieumienphi.vn/doc/bo-loc-kalman-3d9ptq.html>