BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

BÁO CÁO TỔNG KẾT ĐỀ TÀI
NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN

NGHIÊN CỨU, THIẾT KẾ BỘ ĐIỀU KHIỂN ROBOT TỰ HÀNH DẪN ĐƯỜNG SỬ DỤNG CẢM BIẾN LIDAR

Sinh viên thực hiện: **Nguyễn Xuân Quang** **Điện 6, K12, khoa Điện**
Phạm Quang Linh TĐH 2, K13, khoa Điện
Nguyễn Mạnh Hùng TĐH 1, K12, khoa Điện
Đoàn Vượng Điện 6, K12, Khoa Điện

Người hướng dẫn: **TS. Lê Xuân Hải**

Hà Nội, 5/2021

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

BÁO CÁO TỔNG KẾT ĐỀ TÀI
NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN

NGHIÊN CỨU, THIẾT KẾ BỘ ĐIỀU KHIỂN ROBOT TỰ HÀNH DẪN ĐƯỜNG SỬ DỤNG CẢM BIẾN LIDAR

Sinh viên thực hiện:

Nguyễn Xuân Quang Nam/Nữ: **Nam** Dân tộc**: Kinh** Năm thứ**: 4/4**

Lớp, Khoa: **Điện 6,** khoa **Điện** Ngành học: **CN Kỹ thuật Điện, Điện tử**

Phạm Quang Linh Nam/Nữ: Nam Dân tộc: Kinh Năm thứ: 3/4

Lớp, Khoa: TĐH2, khoa Điện Ngành học: Tự động hóa

Nguyễn Mạnh Hùng Nam/Nữ: Nam Dân tộc: Kinh Năm thứ: 4/4

Lớp, Khoa: TĐH1, khoa Điện Ngành học: Tự động hóa

Đoàn Vượng Nam/Nữ: Nam Dân tộc: Kinh Năm thứ: 4/4

Lớp, Khoa: TĐH4, khoa Điện Ngành học: CN Kỹ thuật Điện, Điện tử

Người hướng dẫn: **TS. Lê Xuân Hải**

Hà Nội, 5/2021

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

LỜI CÁM ƠN

Được sự đồng ý và ủng hộ của Khoa Điện, Trường Đại học Công Nghiệp Hà Nội và dưới sự hướng dẫn của Thầy giáo T.S Lê Xuân Hải , chúng em đã thực hiện đề tài : **"*Nghiên cứu, thiết kế bộ điều khiển rô bốt tự hành dẫn đường sử dụng Lidar*"**

Để hoàn thành đề tài nghiên cứu này chúng em xin chân thành cảm ơn thầy giáohướng dẫn T.S Lê Xuân Hải đã hướng dẫn tận tình, chu đáo từ những ngày mới bắt đầu.Cô luôn nhiệt huyết và theo dõi mỗi bước đi của chúng em.

Cuối cùng chúng em cũng xin cảm ơn các thầy cô giáo trong khoa Điện đã luôn giúp đỡ, góp ý, ủng hộ và động viên để chúng em hoàn thành tốt nghiên cứu này.

Trong quá trình nghiên cứu chúng em đã vấp phải nhiều khó khăn, do hạn chế về kiến thức và kinh nghiệm nên chúng em không thể tránh khỏi những thiếu sót nhất định mà bản thân chưa thấy được.

Chúng em rất mong nhận được sự góp ý của quý thầy cô để đề tài này của chúng em được hoàn chỉnh hơn.

Chúng em xin chân thành cảm ơn!

Hà Nội, ngày… tháng … năm 2021

PHẦN MỞ ĐẦU

1.Lý do chọn đề tài

Trong thời đại công nghiệp 4.0, nhu cầu sử dụng robot cho công việc phức tạp thay thế cho con người càng lớn, một robot tự hành là một hệ tự trị, có khả năng lấy thông tin về môi trường của nó và làm việc trong một thời gian dài mà không cần sự can thiệp của con người.

Nghiên cứu, chế tạo và pháp triển robot tự hành, xe tự hành không người lái đang là một nghiên cứu tập trung của các nhà khoa học, công nghệ trên thế giới. Với việc phát triển nhanh của công nghệ xử lí tốc độ cao cũng như lưu trữ lượng data lớn, cộng với khả năng kết nối vạn vật, các robot tự hành xe tự hành ngày càng trở lên thông minh. Dựa trên các công cụ phần cứng mạnh, robot bây giờ được ứng dụng nhiều các hệ cảm biến mắt máy(camera 3D), Lidar, cảm biến quét laser,IMU, đặc biệt ứng dụng trí tuệ nhân tạo vào việc nhận dạng nhận biết và điều hướng thông qua các cảm biến đó. Với các công cụ lập trình truyền thống, việc xử lí các dữ liệu từ nhiều luồng khác nhau sẽ rất khó khăn. Từ đó, ROS được sinh ra để giải quyết những vấn đề này. ROS hiện nay đang được sử dụng rộng rãi và gần như là tối ưu cho việc xây dựng hệ thống điều khiển cho các loại robot, xe tự hành, xe tự lái trên thế giới.

Chính vì thế nhóm nghiên cứu chúng em đã lựa chọn đề tài về Robot tự hành để nghiên cứu và ứng dụng hệ điều hành ROS vào lập trình với mong muốn đem tới một giải pháp hữu ích cho xã hội.

2.Mục đich nghiên cứu

Tìm hiểu, xây dựng, thiết kế bộ điều khiển tự dẫn đường cho robot tự hành sử dụng cảm biến Lidar được lập trình thông qua phần mềm ROS và tích hợp bộ điều khiển PID để điều khiển vị trí cho xe tự hành trên vi điều khiển STM32F4 đồng thời giám sát tốc độ xe trong giao diện không dây HMI được thiết kế và lập trình trên module wifi Esp 8266. Quá trình thực hiện đề tài nghiên cứu, mục tiêu thực hiện đề tài được đề ra như sau :

- Tìm hiểu, nghiên cứu và ứng dụng được phần mềm ROS trong việc lập trình.
- Ứng dụng được các chức năng của cảm biến Lidar cho xe tự hành.
- Cài đặt được thuật toán PID cho xe tự hành trên nền vi điều khiển STM32F4 thông qua giao điện HMI.
- Thiết kế giao diện HMI không dây để giám sát tốc độ được tích hợp trên module wifi ESP 8266.

- Hiển thị bản đồ những nơi mà xe đi qua trên màn HDMI LCD
- Tích lũy kinh nghiệm và nâng cao khả năng phân tích và đánh giá chất lượng của một hệ thống điều khiển.

3.Đối tượng nghiên cứu

Đối tượng: Xe tự hành ba bánh sử dụng cảm biến Lidar.

4.Giả thiết nghiên cứu

Xe ba bánh tự hành chạy trên nền phẳng.

5.Phương pháp nghiên cứu

- Lý thuyết(trên các tài liệu đã công bố trông nước và quốc tế).
- Thực nhiệm: dựa trên xây dựng thiết kế phần cứng, phần mềm để cài đặt thuật toán điều khiển giao tiếp giữa máy tính nhúng (Pi 4 Model B) và STM32F411VET6 qua Websever. Cảm biến Lidar xử lí qua máy tính nhúng truyền cho STM32.

6.Ý nghĩa thực tiễn:

- Sử dụng công nghệ Lidar để thám hiểm, xây dựng bản đồ đồ môi trường ở những khu cách ly, có địa hình phức tạp, môi trường độc hại đối với con người.
- Giảm thiểu sức lực của con người. Phát triển xe tự hành, xe tự lái mang công nghệ Việt.

CHƯƠNG 1: TỔNG QUAN VỀ XE TỰ HÀNH

1.1 Giới thiệu xe tự hành

Khái niệm robot hay người máy là một cỗ máy cơ- điện tử được điều khiển bởi một chương trình máy tính hoặc một mạch điện tử, thực hiện công việc một cách tự động hoặc bán tự động, có khả năng xử lí những công việc phức tạp thay thế cho con người. Hiện nay có thể phân biệt các loại robot thành các mảng chính, được quan tâm nhiều nhất là: Các loại tay máy robot công nghiệp, robot di động, robot phòng sinh và robot cá nhân. Trong đó robot tự hành hay robot di động là một trong những lĩnh vực phát triển của robotics, trọng tâm là nghiên cứu chuyển động của robot trong một khoảng không gian nhất định. Robot tự hành thự sự là một lĩnh vực riêng biệt từ cuối năm 1960 bằng dự án Shakey tại SRI. Báo cáo của N.J.Nilsson " A Mobile Automation: An Application of Artificial Inteligence Techniques" tại IJCAI 1969 đã đưa ra các yếu tố " nhận thức", "lập bản đồ ", "lập kế hoạch đường đi" và các khái niệm về kiến trúc điều khiển.

Hình 1-1: Robot tự hành được ứng dụng trong quân sự và đời sống

Ngày nay với sự phát triển của máy tính và vi xử lý cũng như là các loại robot tự hành phát triển không ngừng. Robot có thể điều khiển bằng vi xử lý hoặc máy tính nhúng; một hệ thống robot có thể được kiểm soát được bởi một máy tính trung tâm quang mạng không dây. Việc tương tác với mô trường bên ngoài có thể được thự hiện bởi các cảm biến siêu âm, hồng ngoại hoặc camera. Nhớ tích hợp nhiều công nghệ tiên tiến mà tính năng của robot tự hành rất đa dạng: robot có thể duy chuyển trong môi trường độc hại, địa hình phức tạp, có thể vẽ lại bản đồ và truyền về bộ điều khiển trung tâm, vận chuyển hàng hóa,…

Như đã trình bày ở trên, có thể thấy xu hướng phát triển robot ngày càng có nhiều chức năng hơn cũng như tích hợp nhiều công nghệ hiện đại. Qua đó, nhóm nghiên cứu chúng em quyết định xây dựng một robot tự hành có các chwucs nặng như sau:

- Chuyển động theo hướng và vận tốc yêu cầu.
- Nhận biết được vật cản và tự động né vật cản.
- Tự động tìm đường đi thích hợp.
- Xây dựng bản đồ môi trường .
- Hiển thị thông tin về các thông số trạng thái của robot.
- Truyền dữ liệu tốc độ về máy chủ trên Websever.

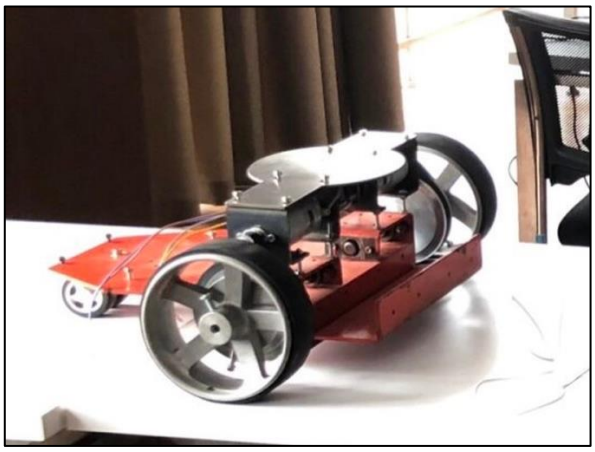
1.2 Các kết quả đạt được

- Xác định rõ hướng điêu trong khâu thiết kế cơ khí và bộ điều khiển cho robot.
- Hoàn thành xây dựng robot và mô hình điều khiển.
- Xây dựng và ghép nối các chức năng của từng thành viên thành công lên robot.
- Thử nghiệm thực tế thành công việc ghép nối cách thiết bị trong robot với nhau.

CHƯƠNG 2: XÂY DỰNG BỘ ĐIỀU KHIỂN CHO ROBOT TỰ HÀNH SỬ DỤNG CÔNG NGHỆ LIDAR

Trong quá trình nghiên cứu, sau khi xác rõ phương pháp nghiên cứu trên cơ sở lí thuyết về động lực học của xe tự hành, cùng các vấn đề đặt ra trong nghiên cứu. Việc đầu tiền nhóm chúng em đi xây dựng phần cứng cho robot tự hành.

2.1 Mô hình xe tự hành của nhóm nghiên cứu



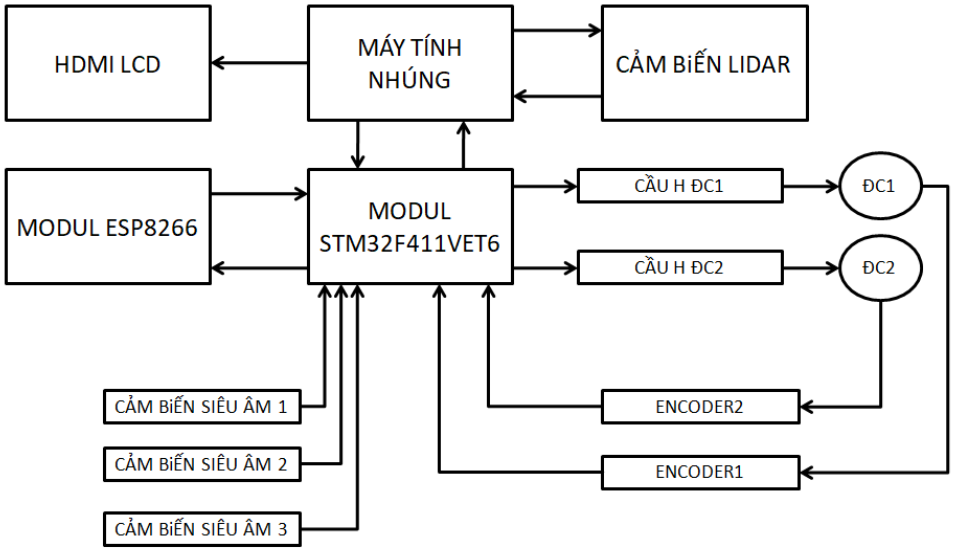
Hình 2.1: Mô hình xe tự hành

Phần cứng cơ khí của robot tự hành

Bảng 2-1: Bảng tên phần cứng của xe tự hành

Tên phần cứng		SL	Thông số, mô tả
Khung xe		1	Kích thướcAxBxZ cm,
Động cơ 775 Motor Dia.of Shaft 5mm		2	Công xuất max: 288W Điệnáp: 12 – 24VDC. Dòngđiện: 10 – 20A. Speed: 7600 – 12000 RPM. (vòng/phút)
Encoder Omron E6B2 – CWZ6C		2	Độ phân giải: 1000 xung/ vòng Số kênh: kênh A vàkênh B Điện áp hoạt động: 5 – 24VDC.
Bánh xe	Sau	2	Nhận động lực từ động cơ làm xe di chuyển.
	Trước	1	Bánh xe phụ, giúp xe di chuyển linh hoạt.
Bánh răng	ϕ10cm	2	Nhận đông lực từ động cơ xuống bánh xe.
	ϕ4cm	4	Truyền tốc độ từ động cơ sang Encoder. Tỉlệtruyềnlà 1:1.
	ϕ2cm	2	Truyền động lực từ động cơ xuống bánh xe.
Dây đai chuyền động	ϕ8cm	2	Truyền lực giữa 2 bánh răng Encoder – động cơ.
	ϕ15cm	2	Truyền lực giữa 2 bánh răng bánh xe – động cơ.

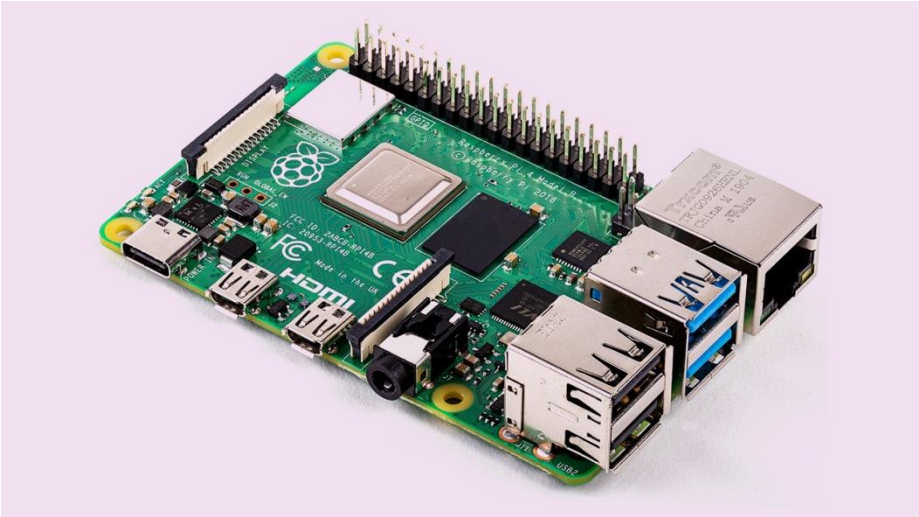
2.2 Phần cứng bộ điều khiển



Hình 2.2: Sơ đồ khối phần cứng điều khiển xe tự hành

Nhìn vào sơ đồ ta có thể thấy kết nối giữa các thành phần trong hệ thống của Robot tự hành. Trung tâm bộ điều khiển là Chip STM32F411VET6 cùng với máy tính nhúng chạy hệ điều hành ROS. Đầu tiên máy tính nhúng kết nối với cảm biến Lidar và màn hình HDMI LCD với nhiệm vụ xây dựng bản đồ môi trường, đồng thời cho biết được vị trí của robot. Chip STM32F411VET6 kết nối với cầu H, động cơ, encoder, cảm biến siêu âm cùng ESP8266 thực hiện nhiệm vụ tự động điều khiển xe khi chạy kết hợp với thuật toán PID, cùng với đó truyền tốc độ lên Websever qua ESP8266, để giám sát tốc độ xe.

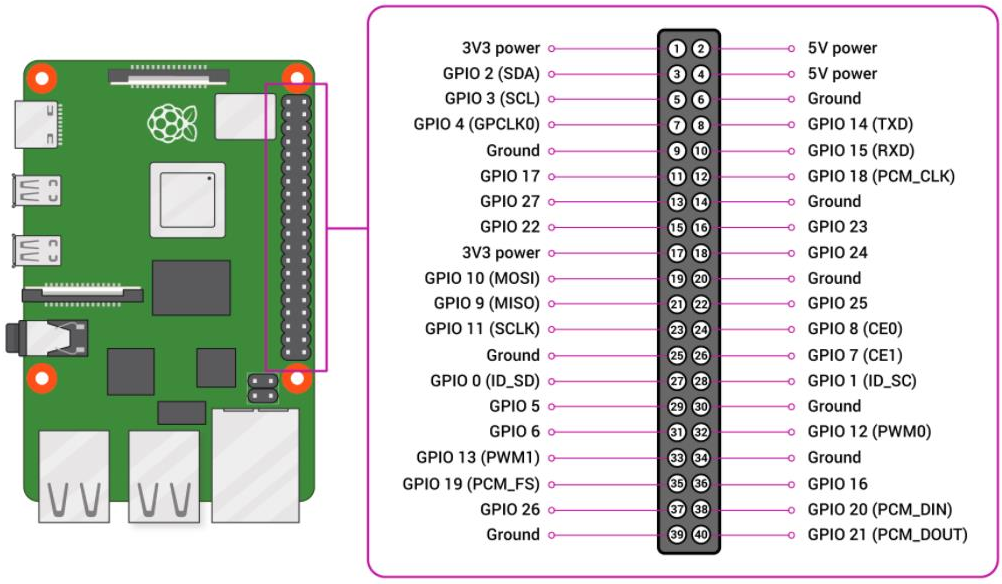
2.2.1 Raspberry Pi 4 Model B



Hình 2.3: Pi 4 Model B

Raspberry Pi là chiếc máy tính kích thước nhỏ được tích hợp nhiều phần cứng mạnh mẽ đủ khả năng chạy hệ điều hành và cài đặt được nhiều ứng dụng trên nó. Phiên bản Raspberry Pi đầu tiên được phát hành tháng 2 năm 2012, và tới nay đã có nhiều phiên bản khác nhau, với sự nâng cấp của phần cứng, cũng như hướng tới những mục tiêu khác nhau. Trong đề tài nghiên cứu lần này chúng em sử dụng Pi 4 Model B.

Thông số kĩ thuật của Pi 4 Model B:



Hình 2.4: Cấu tạocủa Pi 4 Model B

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB, 4GB hoặc 8GB LPDDR4-3200 SDRAM (tùy thuộc vào kiểu máy)

- 2,4 GHz và 5,0 GHz IEEE 802.11ac wireless
- Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 cổng USB 3.0, 2 cổng USB 2.0
- Đầu cắm GPIO 40 chân tiêu chuẩn Raspberry Pi
- 2 × cổng micro-HDMI (hỗ trợ lên đến 4kp60)
- Cổng hiển thị MIPI DSI 2 làn
- Cổng camera MIPI CSI 2 làn
- Cổng video tổng hợp và âm thanh stereo 4 cực
- 26 chân GPIO. Khi thiết lập là input, GPIO có thể được sử dụng như chân interupt, GPIO 14 & 15 được thiết lập sẵn là chân input.
- 1UART, 1 I2C, 2 SPI, 1 PWM (GPIO 4)
- 2 chân nguồn 5V, 2 chân nguồn 3.3V, 8 chân GND.

2.2.2 Cảm biến siêu âm HC-SR04

Để xe tự hành có thể tránh vật cản và di chuyển linh hoạt nhóm đồ án sử dụng 3 cảm biến siêu âmHC-SR04 để tính khoảng cách khi gặp vật cản.

Cảm biến siêu âm hoạt động dựa theo nguyên lý cho và nhận, tức là cảm biến sẽ phát ra 1 nguồn sóng liên tục với tốc độ của sóng siêu âm. Khi bước sóng này gặp vật cản thì sẽ phản hồi lại, cảm biến siêu âm sẽ nhận được bước sóng phản hồi này đồng thời sẽ tiến hành phân tích để biết được khoảng cách từ vật cản đến cảm biến. Nhờ đó, chúng ta sẽ biết được khoảng cách từ cảm biến cho tới mức chất lỏng hoặc chất rắn.

Hình 2.5: Cảm biến siêu âm HC - SR04

Thông số kỹ thuật module cảm biến siêu âm HC-SR04:

- Điện áp: 5V DC
- Dòng hoạt động: < 2mA
- Mức cao: 5V
- Mức thấp: 0V
- Góc tối đa: 15 độ
- Khoảng cách: 2cm – 450cm (4.5m)
- Độ chính xác: 3mm

2.2.3 HDMI LCD 7inch

Màn hình 7 inch có thiết kế nhỏ gọn, nhẹ nhàng, thường sử dụng cho camera quan sát, máy tính, Raspberry Pi, CCTV, XBOX, PS3, PS4, Android Box, máy công nghiệp... tiện lợi dễ dàng di chuyển và sử dụng vì kích thước nhỏ gọn, nhẹ nhàng.

Màn hình 7 inch HDMI này rất phù hợp với chiếc máy tính mini thông qua cáp kết nối HDMI

Hình 2.6: Màn hình HDMI LCD

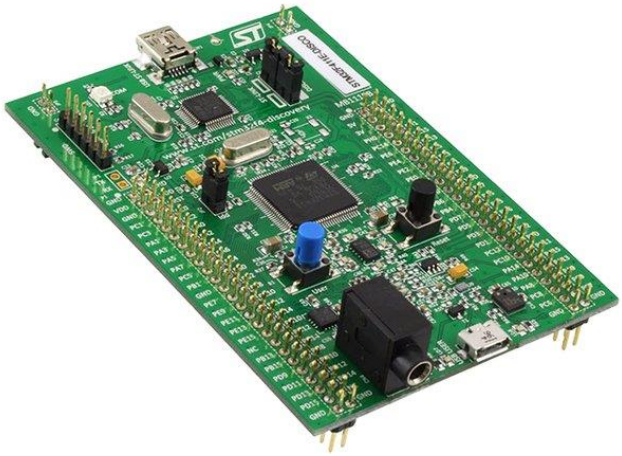
Trong đề tài nghiên cứu lần này, chúng em sử dụng màn hình HDMI LCD 7 inch để hiện thị địa hình xung quanh xe hay nhưng nơi mà xe đi qua. Dưới đây là một số thông số kĩ thuật đặc trung của màn.

Thông số kỹ thuật:

- Kích thước màn hình: 7 "IPS
- Đèn nền: LED
- Độ phân giải: 1920 × 1200 pixel
- Tỷ lệ khung hình: 16:10
- Độ sáng: 450cd / m²
- Tỷ lệ tương phản: 1200: 1
- Khối lượng: 120g
- Gócn hìn: 80 ° / 80 ° (L / R) 80 ° / 80 ° (U / D)
- Đầuvào 1 * HDMI

- Đầu ra 1 * HDMI
- Nguồn: DC 12V
- Công suất tiêu thụ: ≤12W

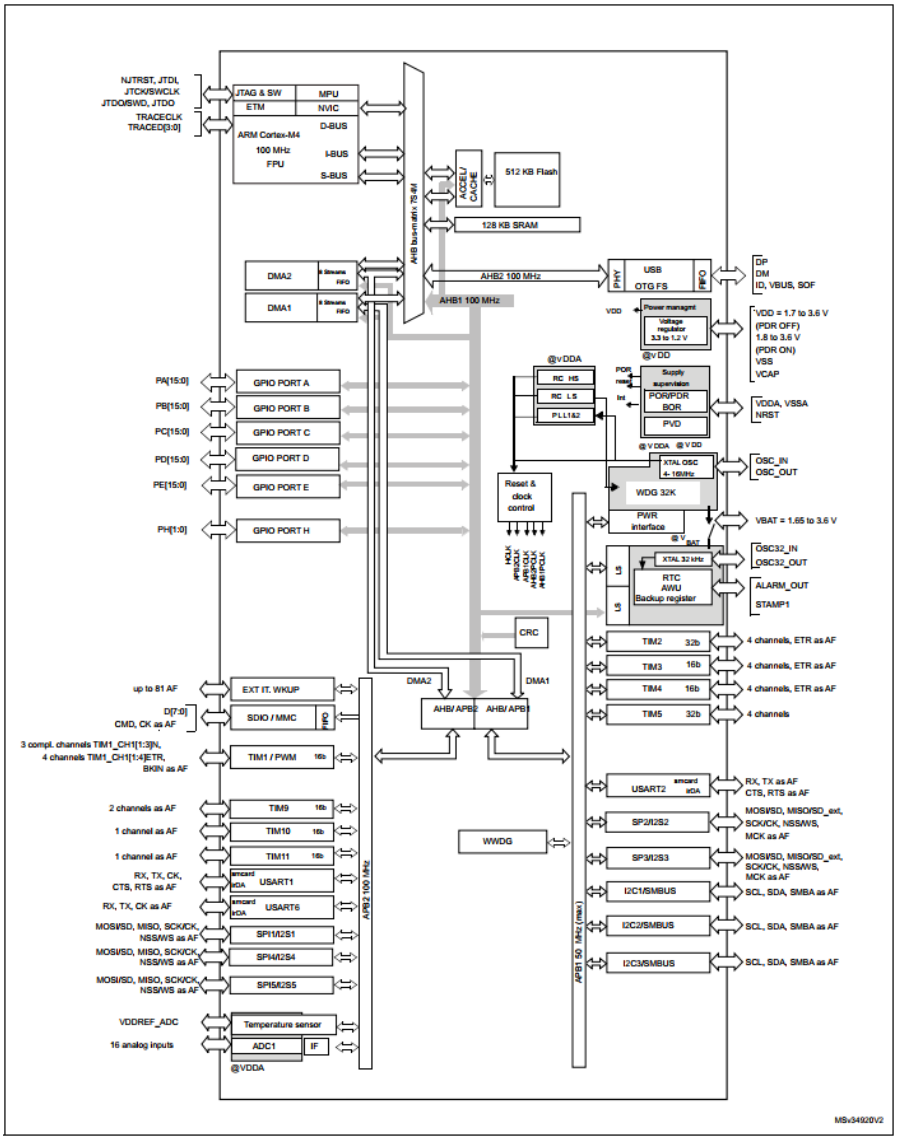
2.2.4 Kit STM32F411VET6 Discovery



Hình 2.7: Kit STM32F411 VET6

Kit STM32F411VET6 Discovery là một kit phát triển lập trình nhúng của hãng STMicrocontroller . Kít có thiết kế nhỏ gọn, nhiều ngoại vi vàkhả năngxử lý dữ liệu mạnh mẽ. Cùng với đó là một hệ sinh thái gồm các phần mềm và thư viện đa dạng được hãng ST phát triển cùng phần cứng để hỗ trợ người dùng lập trình, nhất là với sinh viên chúng em muốn học tập và làm quen với lập trình nhúng.

Kiến trúc ARM của dòng chip Cortex M4 là dạng kiến trúc RISC dành cho các vi điều khiển, nó có cấu trúc rất phức tạp. Với con chíp STM32F411 thuộc dòng Cortex M4 là dòng chíp 32 bit được hãng ST thiết kế có tốc độ lến tới 100Mhz, bộ nhớ chương trình lớn 512KB (512x8) và một loạt các ngoại vi phổ biến nhất hiện nay như DMA, I²S, I2C, POR, PWM, USART… Với nhiều ưu điểm như vậy khi được tích hợp lên Kít STM32F411VET6 Discovery hãng ST đã mang đến cho người dùng một công cụ học tập cực kỳ hiệu quả áp dụng lên các dự án, đề tài cho sinh viên, người dùng mới làm quen với vi điều khiển.



Hình 2.8: Cấu trúc STM32F411VET6

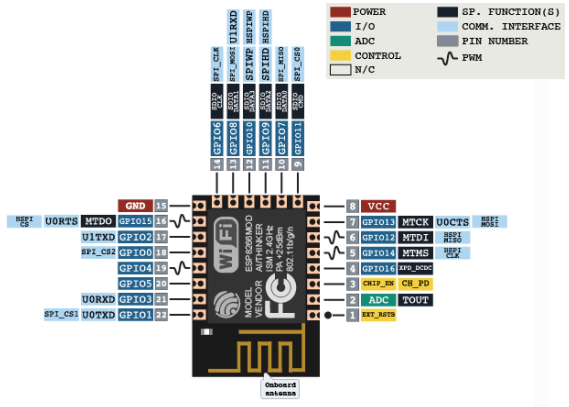
Các đặc điểm nổi bật của KIT STM32F411VET6 Discovery:

- Điện áp hoạt động 5VDC có thể lấy nguồn từ kít để cấp nguồn cho cảm biến, ngoại vi khác..
- Tích hợp thạnh anh ngoài cho phép tốc độ xử lý 0 – 100MHz và có thể thay đổi tùy theo mục đích của người lập trình.
- Kích thước bộ nhớ lên tới 512KB giúp vi điều khiển lưu trữ được lượng lớn chương trình.
- Hơn 64 pin GPIO có thể tùy biến chức năng của từng chân.
- 3 Bộ giao tiếp USART giúp kít có thể giao tiếp truyền dữ liệu với nhiều vi điều khiển khác.

- 8 Timer tùy chỉnh được chế độ hoạt động như Encoder Mode, PWM, sử dụng làm ngắt, và bộ đếm thời gian thực RTC.
- 3 giao tiếp I2C cho phép giao tiếp với nhiều cảm biến, các thiết bị Master/Slave.
- 9 Cổng ADC cho phép vi điều khiển tiếp nhận và xử lý được nhiều thông tin.
- Phần mềm STM32CubeIDE và thư viện HAL hỗ trợ lập trình rất hiệu quả và mạnh mẽ.
- Cấu hình chip ngay trên giao diện rất trực quan và dễ dàng.
- Tích hợp khối DMA giúp cho việc truyền và xử lý dữ liệu nhanh hiệu quả giúp giảm thời gian xử lý của vi điều khiển.
- Tích hợp con quay hồi chuyển trên bo mạch.
- Tích hợp nút nhấn và 4 LED đơn trên bo mạch.
- Tích hợp sẵn mạch nạp STLink dễ dàng nạp chương trình và debug….

2.2.5 Module Wifi ESP8266

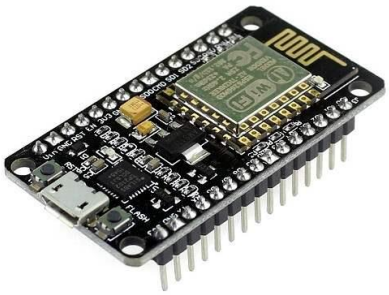
ESP 8266 là một vi điều khiển tích hợp sẵn module wifi trên cùng một con chip vật lý.Vì là một vi điều khiển nên ESP 8266 có thể lập trình và giao tiếp được với các cảm biến và vi điều khiển khác đồng thời kết nối mạng wifi giúp ESP 8266 có thể truyền nhận dữ liệu không dây giúp kết nối giao diện Webserver với vi điều khiển, từ đó người dùng có thể điều khiển thiết bị, giám sát và thu thập dữ liệu từ xa.



Hình 2.9: Sơ đồ chân ESP 8266

Với nhiều ưu điểm và có áp dụng được vào nhiều ứng dụng vào nhiều mô hình tự động hóa vì thế dòng chíp này được nhiều nhà sản suất phát triển với nhiều module với

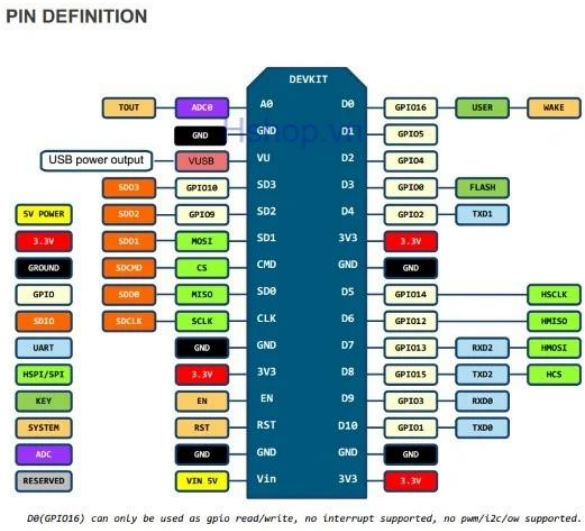
tính năng và nhù cầu khác nhau. Trong đề tài nghiên cứu nhóm đồ án chúng em sử dụng "Kit RF Thu Phát Wifi ESP8266 NodeMCU Lua CP2102" được rất nhiều kỹ sư và các bạn sinh viên sử tin dùng.



Hình 2.10: Module ESP 8266 NodeMCU Lua CP2102

Đặc điểm Kit:

- Sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code.
- Được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT.
- Module sử dụng chip nạp và giao tiếp UART mới và ổn định nhất là CP2102 có khả năng tự nhận Driver trên tất cả các hệ điều hành Window.



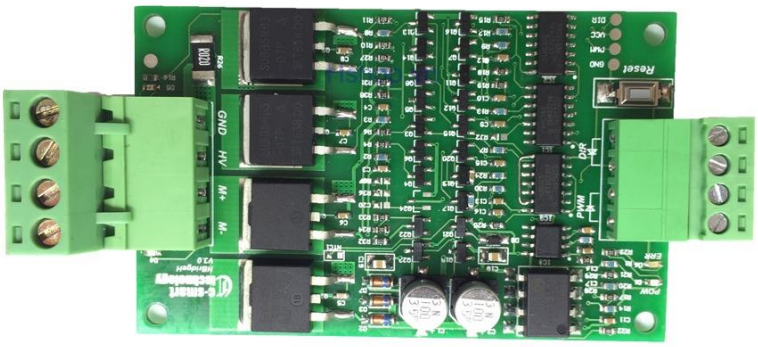
Hình 2.11: Sơ đồ chân của Module ESP 8266 NodeMCU Lua CP2102

Thông số kỹ thuật:

- IC chính: ESP8266
- Phiên bản firmware: NodeMCU Lua
- Chip nạp và giao tiếp UART: CP2102.
- GPIO tương thích hoàn toàn với firmware Node MCU.
- Cấp nguồn: 5VDC MicroUSB hoặc Vin.
- GIPO giao tiếp mức 3.3VDC
- Tích hợp Led báo trạng thái, nút Reset, Flash.
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Kích thước: 25 x 50 mm

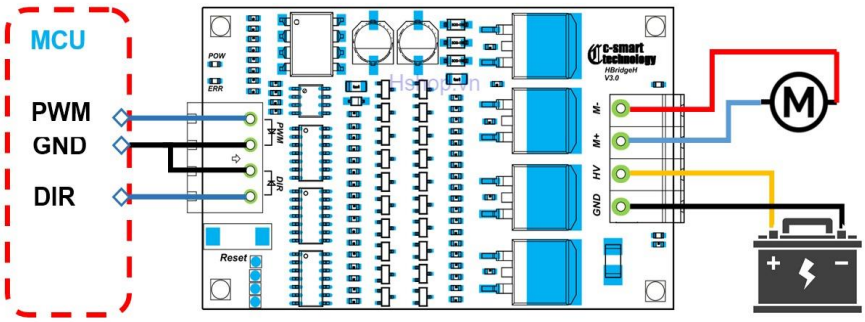
2.2.6 Mạch điều khiển động cơ

Mạch điều khiển động cơ hay còn gọi là mạch cầu H là một trong những mạch công suất được sử dụng trong việc điều khiển động cơ một chiều, động cơ bước. Trong thực tế có nhiều loại mạch cầu Hcó kiểu dáng, kích thước khác nhau nhưng điểm chung là chúng đều đảm nhiệm nhiệm vụ chính là điều khiển tốc độ, điều khiển vị trí và đảo chiều động cơ theo mục đích của người lập trình.



Hình 2.12: Mạch điều khiển động cơ

Trong mô hình xe tự hành, động cơ nhóm chúng em sử dụng là loại động cơ một chiều có công xuất tối đa lên tới 288W vì thế chúng em lựa chọn “Mạch Điều Khiển Động Cơ DC H-Bridge H”.



Hình 2.13: Sơ đồ đấu dây mạch điều khiển động cơ

Thông số kỹ thuật:

- Điện áp vận hành 8~32VDC.
- Công suất 500W, tần số PWM lên tới 20 KHz.
- Dòng điện liên tục 16A.
- Ngõ vào điều khiển cách ly quang tần số cao.
- Led báo hiệu chiều quay, báo hiệu nguồn.
- Bảo vệ ngắn mạch, quá tải, qúa nhiệt.
- Cho phép độ rộng xung trên toàn giải 0-100%.

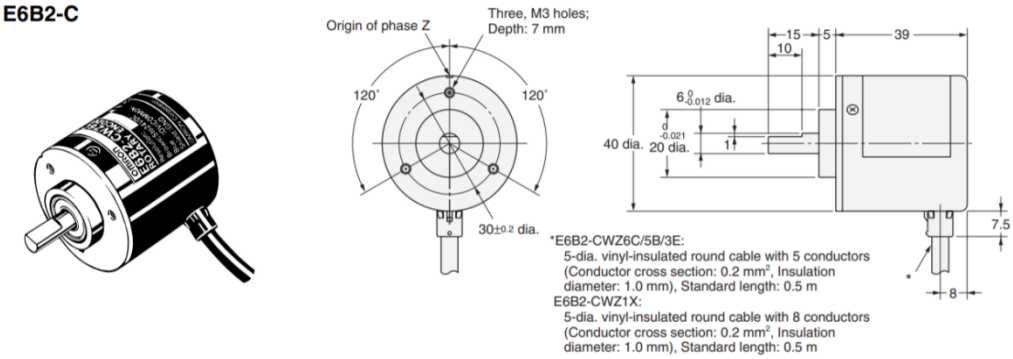
2.2.7 Encoder Omron E6B2 – CWZ6C

Encoder hay còn gọi là bộ mã hóa quay hoặc bộ mã hóa trục, là một thiết bị cơ điện chuyển đổi vị trí góc hoặc chuyển động của trục hoặc trục thành tín hiệu đầu ra analog hoặc kỹ thuật số. Encoder được dùng để phát hiện vị trí, hướng di chuyển, tốc độ… của động cơ bằng cách đếm số vòng quay được của trục.

Hình 2.14: Encoder Omron E6B2 - CWZ6C

Để đo tốc độ động cơ nhóm sử dụng Encoder Omron E6B2 – CWZ6C để đọc xung sau đó từ số xung đọc được tính toán ra tốc độ động cơ theo công thức:

“”



Hình 2.15: Kích thước và thống số kỹ thuật của Encoder

Thông số kỹ thuật:

- Model: Omron E6B2-CWZ6C 1000 p/r
- Điện áp sử dụng: 5~24VDC.
- Dòng tiêu thụ: max 80mA
- Số xung: 1000 xung / 1 vòng (1000 p/r)
- Số kênh xung: 3 kênh xung riêng biệt A, B, Z.

- Tần số đáp ứng tối đa: 100Khz
- Dạng ngõ ra xung: NPN cực thu hở (cần mắc trở treo lên VCC để tạo mức cao (High))
- Đường kính trục: 6mm

2.2.8 Cảm biến Lidar

Mục đích chính trong đề tài nghiên cứu này chính là việc sử dụng cảm biến Lidar để xây dựng bản đồ môi trường do hãng SLAMTEC. Để làm được điều này cảm biến Lidar được thiết kế rất chi tiết, hiện đại. Dưới đây là nhưng thông số kĩ thuật đặc trưng của cảm biến Lidar.

Hình 2.16: Items của cảm biến RPLIDAR A1

Như có thể quan sát hình bên trên, cảm biến Lidar A1 được trang bị:

- USB Adapter để kết nối với các thiết bị khác như máy tính nhúng, vi điều khiển,…
- RPLIDAR A1 communication cable là cáp giao tiếp của Lidar.
- Module RPLIDAR để quét và xây dựng bản đồ.

Hình 2.17: Cảm biến RPLIDAR A1

Cảm biến Laser Radar RPLIDAR A1 12m sử dụng giao tiếp UART nên có thể dễ dàng giao tiếp với Vi điều khiển, máy tính nhúng hoặc kết nối máy tính qua mạch chuyển USB-UART và phần mềm đi kèm, cảm biến có khả năng quét xa với khoảng cách lên đến 12m, tần số tối đa 10Hz với 8000 samples per time, phù hợp cho vô số các ứng dụng khác nhau.

Thông số kỹ thuật:

- Điện áp sử dụng: 5VDC
- Chuẩn giao tiếp: UART
- Phương pháp phát hiện vật cản: Laser
- Khoảng cách phát hiện vật cản tối đa: 12m
- Góc quay: 360 độ.
- Tốc độ lấy mẫu tối đa: 8000 Samples per time.
- Tần số quét tối đa: 10Hz
- Kích thước: 71 x 97mm

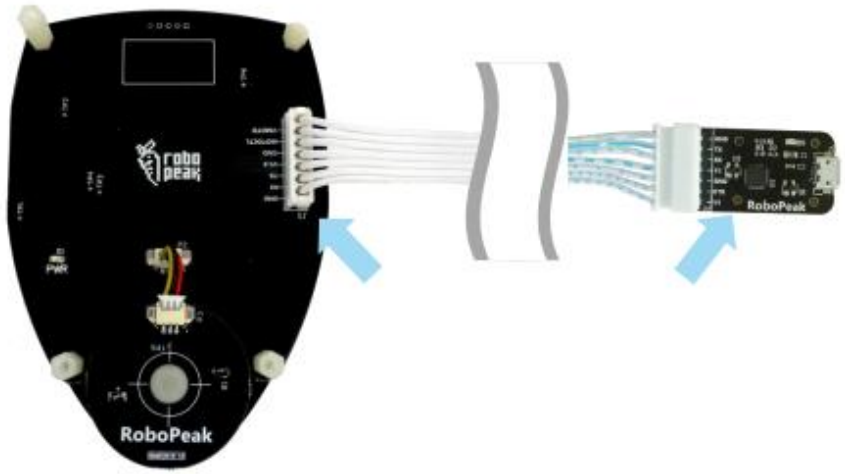
2.3 Hệ điều hành ROS và cảm biến Lidar

Trong đề tài nghiên cứu lần này, chúng em chú trong xây dựng mô hình điều khiển trên hệ điều hành ROS kết hợp cảm biến Lidar, sau đó áp dụng bộ điều khiên PID sử dụng chip STM32f411VET6. Mục đích là để phát triển tư duy với các phần mềm lập trình mới, thông minh hơn, tiếp cần dần với AI

2.3.1 Cách kết nối và sử dụng cảm biến Lidar

Dưới đây cách cách chúng em kết nối cũng như sử dụng cảm biến Lidar trong đề tài nghiên cứu.

- Bước 1: Kết nối dây RPLIDAR A1 với USB adapter bằng cáp giao tiếp. Mô tả như hình dưới.

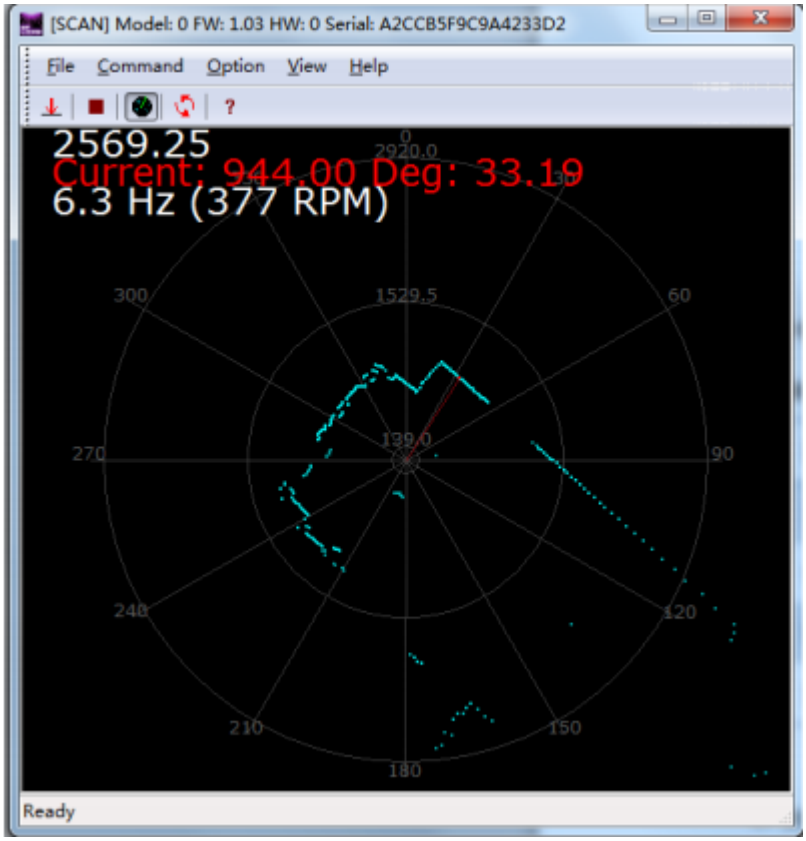


Hình 2.18:Kết nối giữa Lidar với USB adapter

- Bước 2: Kết nối giữa USB adapter với máy tính nhúng sử dụng cap Micro-USB. Sau khi kết nối mắt quét của Lidar sáng lên, Lidar bắt đầu quét.

Hình 2.19: Kết nối USB adapter với máy tính nhúng

Sau khi kết nối Lidar với máy tính nhúng, chúng em sử dụng phần mềm "Frame_grabbe" là phần mềm demo khi sử dụng cảm biến Lidar do hãng SLAMTEC.



Hình 2.20: Xây dựng mô ohngr môi trương với phần mềm “Frame_grabbe”

2.3.2 Tổng quan về hệ điều hành ROS

Hệ điều hành ROS – Robot operation system là một nền tảng mã nguồn mở dành cho việc thiết kế, xây dựng các ứng dụng phần mềm robot, cung cấp một hệ thống điều khiển robot phân tán trên cụm máy tính không đồng nhất.

Hình 2.21: ROS

ROS ra đời với mục đích cho phép các nhà nghiên cứu tăng tốc phát triển các hện thống robotic mới, hỗ trợ sử dụng các mã lệnh thông qua các công cụ và giao diện tiêu chuẩn. ROS có những đặc điểm khiến nó trở thành hệ điều hành vô cùng thích hợp đối với robot như:

- Hỗ trợ lập trình không đồng bộ do lập trình theo hướng call- back.

- Hệ điều hành mang tính chất phân tán nên các tiến trình nằm riêng biệt nhưng vẫn có thể kết nối thông qua messagge(thông điệp).
- Tránh được việc phụ thuộc phần cứng do sử dụng phương thức truyền tin kiểu mesagge.

Về cấu trúc, hệ điều hành ROS gồm ba tầng: Filesystem Level, Computation Graph Level, Community Level. Các tầng có liên quan vô cùng mật thiết với nhau và trong mỗi tầng lại chứa đựng những định nghĩa, khái niệm về các thành phần tạo nên kiến trúc cũng như cách hoạt động của hệ thống.

2.3.3 Tầng ROS Filesystem Level

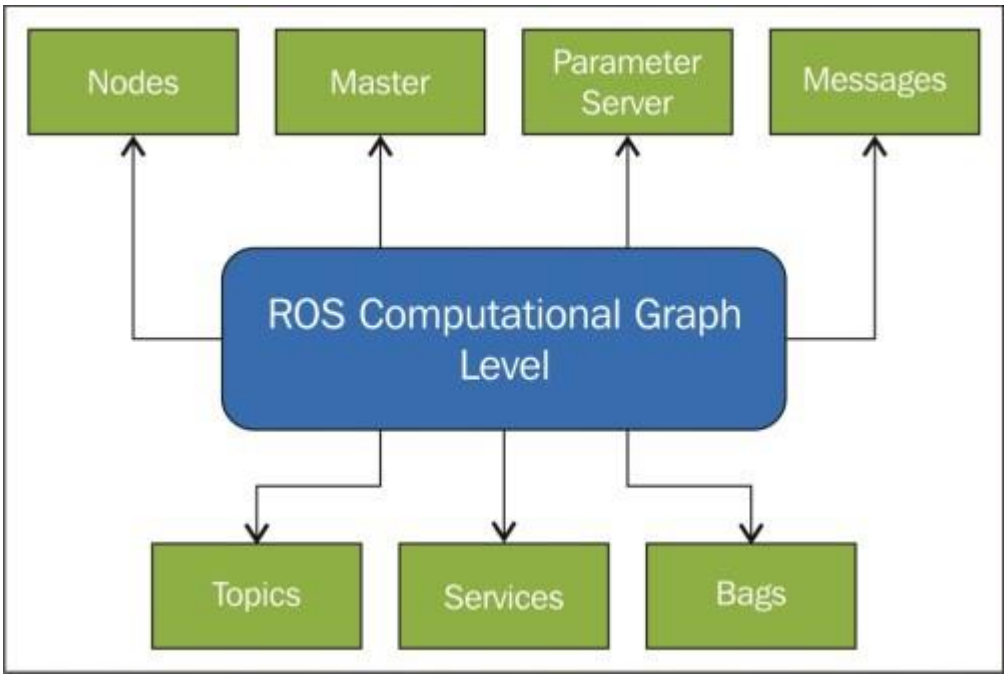
Filesystem là nguồn tài nguyên ROS được lưu trữ trên bộ nhớ hệ thống bao gồm những thành phần sau:

- Packages: Gói dữ liệu là đơn vị chính trong tổ chức phần mềm của hệ điều hành ROS. Một packages có thể chứa các lệnh thực thi của ROS(nodes), một thư viện dành cho ROS, tập dữ liệu, các file cấu hình, hoặc các dữ liệu cần thiết khác trong hệ thống. Packages chính là thành phần nguyên tử nhỏ nhất được xaay dựng và đưa vào sử dụng trong ROS. Có nghĩa rằng hạt nhân nhỏ nhất mà ta có thể xây dụng và đưa vào sử dụng trong ROS chính là một packages.
- Metapackages: Là các packages cụ thể chỉ làm nhiệm vụ đại diện cho nhóm các packages khác có liên quan với nhau. Hầu hết các Metapackages đều được sử dụng giống như việc đảm báo tính tương thích sau này khi chuyển đổi sang rosbuild stack.
- Package Manifests: bảng kê khai một thông tin dữ liệu của packages(package.xml), cung cấp siêu dữ liệu về packages đó bao gồm các tên gọi, phiên bản, thông tin bản quyền(license) và những yếu tố phụ thuộc của gói dữ liệu đó. Manifests cón chứa thông tin đặc trưng của ngôn ngữ lập trình ví dụ như các cờ báo (flags) của trình biên dịch.
- Stacks: là một tập hợp các packages phối hợp với nhau để thực hiện các chứu năng cụ thể, chẳng hạn như “navigation stack” là tập hợp các packages dẫn đường cho robot.

- Stacks Manifests: (stack.xml) cung cấp thông tin về một stack, bao gồm thông tin bản quyền (license) và các thông số phụ thuộc vào các stack khác.
- Message (msg) Types: thông tin mô tả message, được lưu trữ trong my_package/msg/MyMessageType.msg, định nghĩa cấu trúc dữ liệu cho các messages được gửi trong ROS.
- Service (srv) types: thông tin mô tả các services, được lưu trữ trong my_package/srv/MyServiceType.srv, định nghĩa các cấu trúc dữ liệu cho các lệnh truy cập(request) và các phản hồi(response) của các services trong ROS.

2.3.3.1 Tầng Computation Graph Level

"Computation Graph" (lược đồ tính toán) là một mạng peer-to-peer của các tiến trình ROS cùng thực hiện xử lý với nhau. Computation Graph cơ bản gồm các thành phần: các nút (nodes), Master, Parameter Server, messages, servicer, topics, và bags. Tất cả các thành phần này đều cung cấp dữ liệu cho Graph bằng những phương thức khác nhau.



Hình 2.22: Tầng Computation Graph Level

- Node: Là các nút thắt thực hiện quá trình tính toán. Mỗi nút ROS có thể viết bằng nhiều ngôn ngữ lập trình khác nhau ví dụ như cpp hay

python… Với việc sử dụng hệ thống thư viện đa dạng ta có thể sử dụng nhiều phương thức giáo tiếp để trao đổi dữ liệu giữa các nút. Mỗi nút ROS đảm nhiệm một vai trò khác nhau.

- Master: ROS Master cung cấp tên cho các nút, nhờ vậy mà các nút có thể tìm thấy và trao đổi thông tin cho nhau.
- Parameter Server:Là một phần của ROS Master cho phép lưu trữ dữ liệu, tất cả các nút có thể truy cập và sửa đổi các giá trị này.
- Messages: hay tin nhắnlà công cụ để các nút giao tiếp với nhau. Tin nhắngồm các dữ liệu được gửi đi được mã hóa. Các kiểu dữ liệu phổ biến như: kiểu nguyên, kiểu thực, mảng, kí tự…
- Topics: hay chủ đềlà phương tiện để truyền các tin nhắn. Mỗi chủ đề có một tên duy nhất và bất kỳ nút nào cũng có thể lên dùng chủ đề này để gửi dữ liệu qua đó.
- Service: Là giao tiếp 2 chiều trong ROS bao gồm: gửi đi và phản hồi.
- Bags: Là một định dạng đảm nhiệm vấn đề lưu và phát lại tin nhắn trong ROS. Bags là một cơ chế quan trọng để lưu trữ dữ liệu cẳng hạn như dữ liệu của cảm biến.

2.3.3.2 Tầng ROS Community Level

Các khái niệm tầng kết nối của ROS được hiểu là những tài nguyên ROS mà cộng đồng người sử dụng có thể trao đổi với nhau. Đó có thể là phần mềm hoặc những kiến thức liên quan. Các loại tài nguyên nay bao gồm:

- Distribute: ROS Distribution là tập hợp các phiên bản mà ta có thể cài đặt. Các phiên bản này có vai trò tương tự như các bản phân phối của Linux.
- Repositories: ROS dựa vào một mạng liên kết các kho lưu trữ code, nơi mà các tổ chức khác nhau có thể phát triển và cho xuất bản các thành phần phần mềm robot của riêng mình.
- Trang ROS Wiki: Cộng đồng ROS Wiki là diễn đàn chính thống chuyên cung cấp các tài liệu thông tin liên quan tới ROS. Bất cứ ai cũng có thể đăng ký một tài khoản và đóng góp tài liệu của riêng họ mà có thể chỉnh

sửa hoặc cập nhật. Hoặc họ có thể viết các bài hướng dẫn, và nhiều hơn thế nữa.

- Blog: Trang blog Willow Garage luôn cung cấp thông tin cập nhật thường xuyên, bao gồm cả hình ảnh và video liên quan tới ROS.

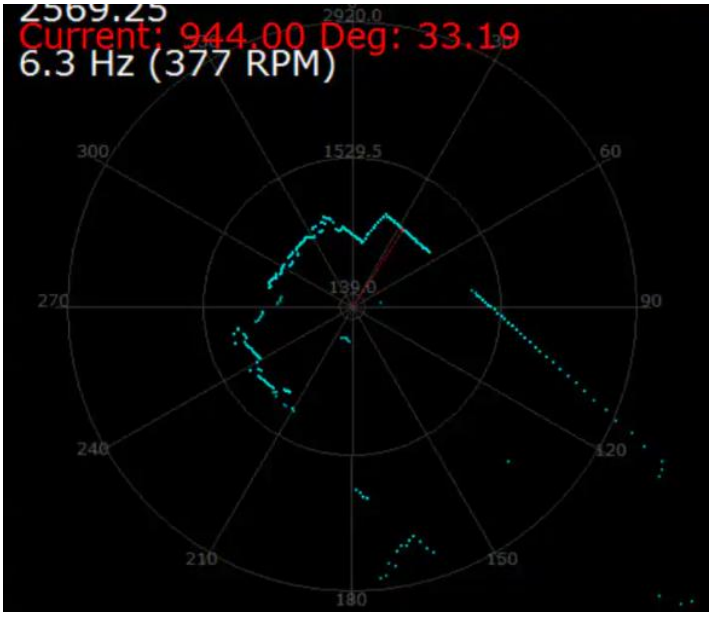
2.3.4 Thuật toán xử lí ảnh

SLAM (Simultaneous Localization And Mapping) là công nghệ giám sát vị trí và dựng lại bản đồ môi trường trong công nghệ xe tự hành sử dụng thuật toán xử lí ảnh. SLAM xây dựng lại bản đồ bằng cách thu thập các tín hiệu từ cảm biến được gắn trên robot như Lidar, IMU, Camera 3D…

Để thực hiện SLAM chúng em sử dụng một gói(package) là Hector SLAM cho WMR Robot bằng việc thu thập các dữ liệu từ RPLidar. Hector SLAM sử dụng hector_mapping node để tìm hiểu bản đồ môi trường đồng thời ước tính tư thế 2D của robot với tốc độ khung hình quét của RPLidar.Thuật toán xử lí ảnh của Hector Slam tôi chia thành hai phần chính: thứ nhất là thuật toán vẽ bản đồ(Mapping), thứ hai là thuật toán xác định sự thay đổi của vị trí theo thời gian(Odometry).

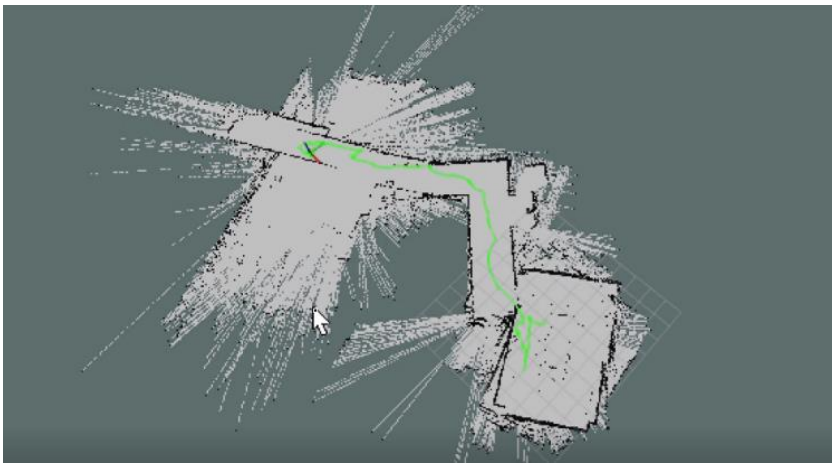
2.3.4.1 Thuật toán vẽ bản đồ(Mapping)

Để thực hiện vẽ lại bản đồ, gói Hector Slam sẽ lấy dữ liệu từ cảm biến RPLidarA1. Để lấy được dữ liệu này trong môi trường ROS thì Node RPLidarA1 phải đóng gói các dữ liệu: khoảng cách, góc.Sau đó dữ liệu đã đóng gói được gửi qua dưới dạng tin nhắn(Messange) về Parameter Server.



Hình 2.23: Tổng hợp dữ liệu cảm biến Lidar

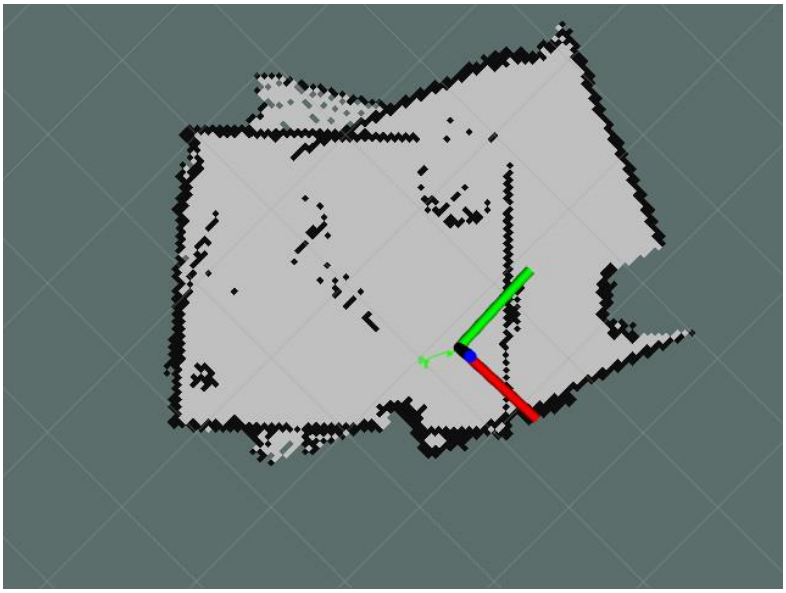
Các dữ liệu được tổng hợp lại thành các điểm . Thuật toán xây dựng bản đồ như sau: Sử dụng dữ liệu khoảng các và góc từ Lidar vẽ các đường thẳng màu trắng từ vị trí của Lidar đến vật cản, kết thúc tại vật cản là một điểm chấm màu đen. Mật độ các điểm và đường thẳng sau mỗi lần quét sẽ ngày càng dày đặc, như ta thấy trên hình 2-23 các vùng màu trắng(tập hợp của các đường kẻ trắng dày đặc) sẽ thể hiện khu vực trống đã quét tới của Lidar, các đường màu đen(tập hợp các điểm tại vật cản) thể hiện vật cản(tường).Thuật toán này càng đúng đắn khi phân biệt được cả hai vật cản cố định và vật cản không cố định.



Hình 2.24: Quá trình thực nghiệm thuật toán bản đồ

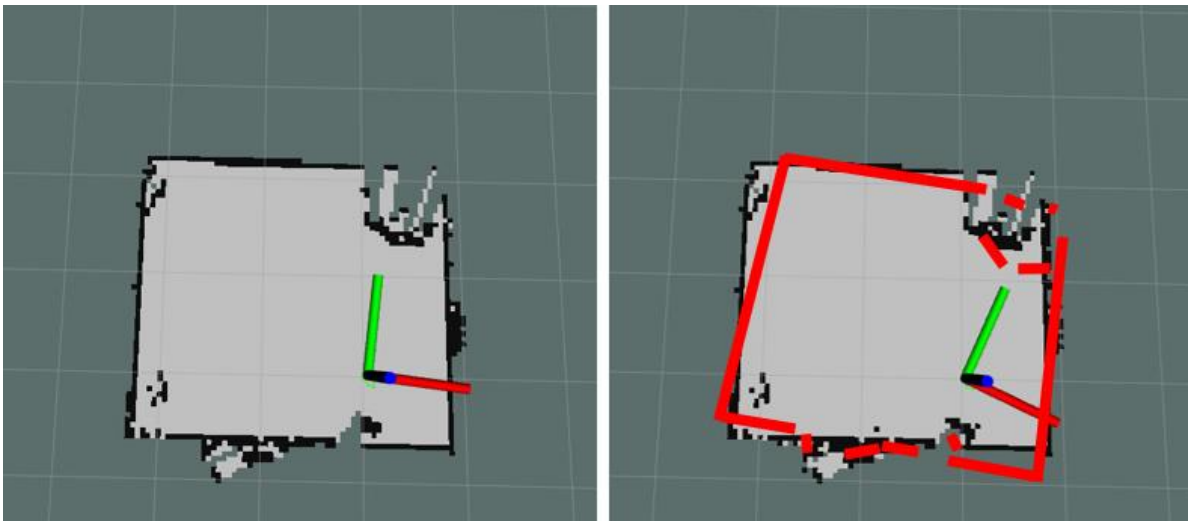
2.3.4.2 Thuật toán xác định sự thay đổi của vị trí theo thời gian(Odometry)

Khi sử dụng cảm biến Lidar, chúng em đã gặp một số vấn đề liên quan đến xây dựng môi trường như sau: Khi robot quay một góc, đi chuyển một đoạn ngắn đồng nghĩa với việc Lidar cũng bị thay đổi góc quét và môi trường quét đồng nghĩa với việc xây dựng lên bản đồ mới. Việc thay đổi góc như vậy làm cho bản đồ mới sẽ bị đè hay bị trùng lên bản đồ cũ. Hiện tượng này được gọi là “MappingOverlay”, gây ảnh hưởng nghiêm trọng, gây sai lệch trong quá trình xây dựng bản đồ. Vì chúng em đã sử dụng thuật toán “xác định sự thay đổi của vị trí theo thời gian(Odometry)” để khắc phục vấn đề nêu trên.



Hình 2-25: Hiện tượng MappingOverlay

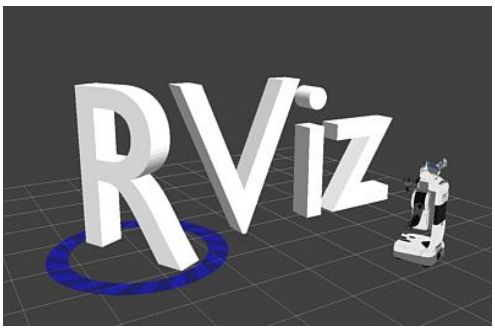
Trong gói Hector Slam, thuật toán (Odometry) là thuật toán xử lí ảnh có thể sử dụng toàn toàn dữ liệu chỉ từ cảm biến Lidar mà không cần thêm bất kì cảm biến nào khác để xác định góc quay, vị trí của robot. Thuật toán này sử dụng thư viện TensorFlow so sánh sự khác nhau giữa hai khung hình liền kề để tìm ra những điểm chung giữa các khung hình trong chu kì quét của Lidar. Sau đó tập hợp các điểm chung gần nhau nhất và xây dựng lên bản đồ môi trường.



Hình 2.25: So sánh hai khung hình liền kề trong chu kì quet của Lidar

Trong kh robot xây dựng bản đồ, chúng em có sử dụng phần mềm RVIZ để theo dõi , giám sát trực tiếp, trực quan ba chiều của các loại cảm biến và hoạt động của robot tự hành. RVIZ truy xuất các tín hiệu trong được cung cấp trong ROS ví dụ như cảm biến Lidar, IMU, GPS,… Không chỉ thế, RVIZ sử dụng thông tin từ thư viện TensorFlow để hiện thị các thông tin trong khung tọa độ chung trong không gian 3 chiều. Từ đó quan

sát được hướng đi của robot, các dữ liệu từ cảm biến, biết được những sai lệch, từ đó chỉnh định lại hệ thống.

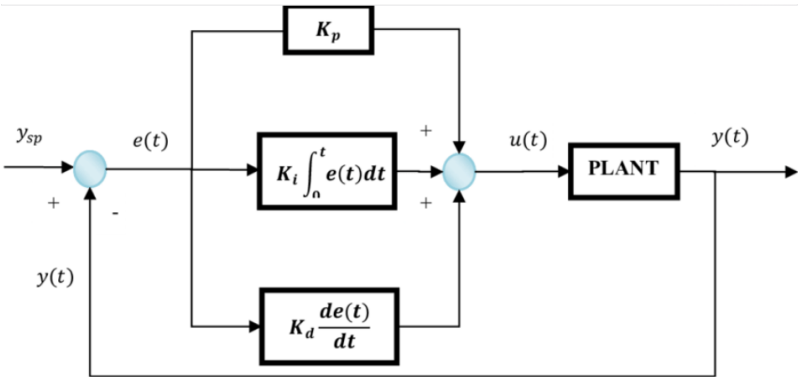


Hình 2.26: RVIZ

2.4 Xây dựng bộ điều khiển PID và lưu đồ thuật toán

Sau khi xây dựng xong việc xây dựng bản đồ môi trường qua hệ điều hành ROS. Tiếp theo chúng em đi xây dựng bộ điều khiển PID cho robot tự hành.

2.4.1 Khái quát về bộ điều khiển PID

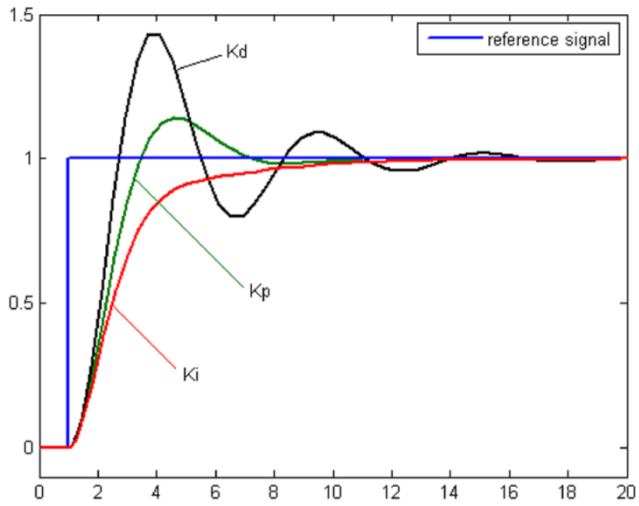


Hình 2.27: Sơ đồ bộ điều khiển PID

PID **(**Proportional Integral Derivative**)** là một cơ chế phản hồi vòng điều khiển được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp. Bộ điều khiển PID được sử dụng nhiều nhất trong các hệ thống điều khiển vòng kín (có tín hiệu phản hồi). Bộ điều khiển PID sẽ tính toán giá trị sai số là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào. Để đạt được kết quả tốt nhất, các thông số PID sử dụng trong tính toán phải điều chỉnh theo tính chất của hệ thống-trong khi kiểu điều khiển là giống nhau, các thông số phải phụ thuộc vào đặc thù của hệ thống.

Trong đó:

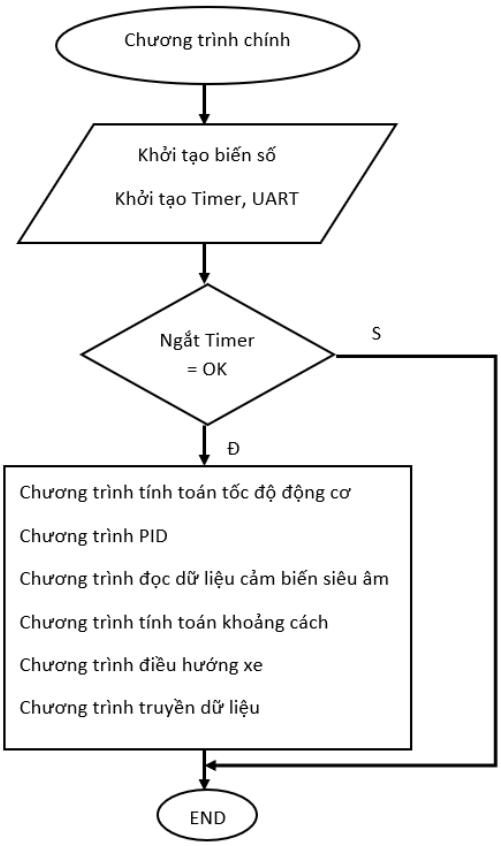
- P (Proportional): là phương pháp điều chỉnh tỉ lệ, giúp tạo ra tín hiệu điều chỉnh tỉ lệ với sai lệch đầu vào theo thời gian lấy mẫu.
- I (Integral): là tích phân của sai lệch theo thời gian lấy mẫu. Điều khiển tích phân là phương pháp điều chỉnh để tạo ra các tín hiệu điều chỉnh sao cho độ sai lệch giảm về 0. Từ đó cho ta biết tổng sai số tức thời theo thời gian hay sai số tích lũy trong quá khứ. Khi thời gian càng nhỏ thể hiện tác động điều chỉnh tích phân càng mạnh, tương ứng với độ lệch càng nhỏ.
- D (Derivative): là vi phân của sai lệch. Điều khiển vi phân tạo ra tín hiệu điều chỉnh sao cho tí lệ với tốc dộ thay đổi sai lệch đầu vào. Thời gian càng lớn thì phạm vi điều chỉnh vi phân càng mạnh, tương ứng với bộ điều chỉnh đáp ứng với thay đổi đầu vào càng nhanh.



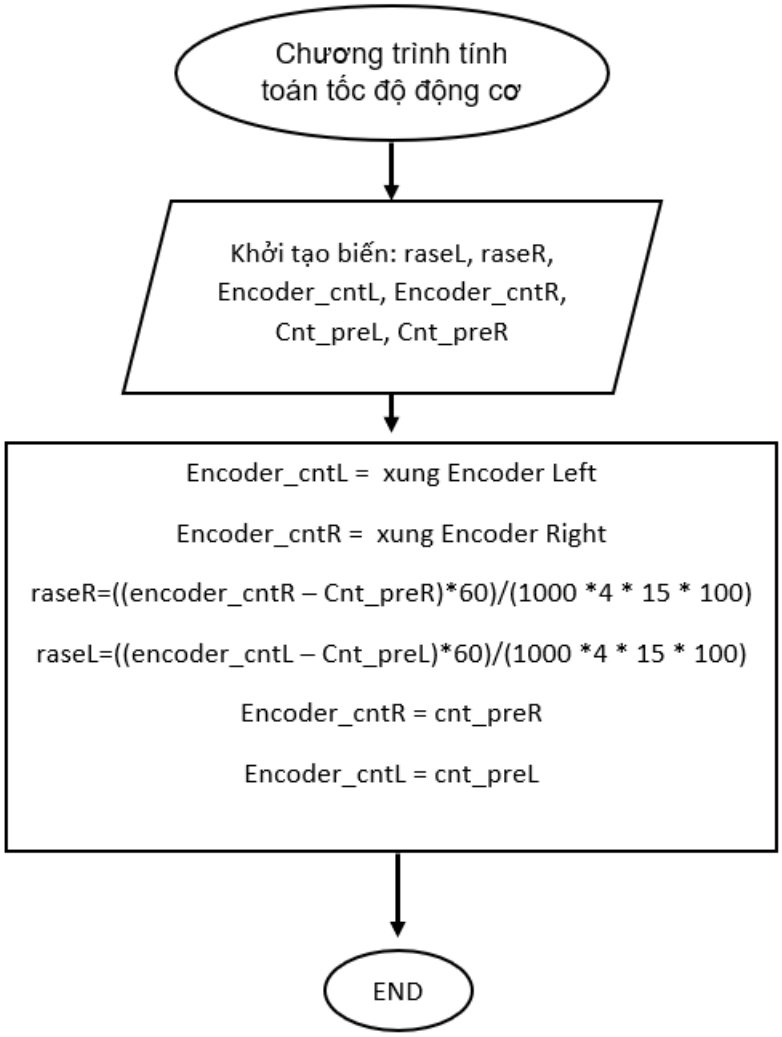
Hình 2.28: Đồ thị bộ điều khiển PID

- Bộ điều khiển tỉ lệ – P (Proportional Controller).
- PI (Proportinal and Integral Controller) gọi là bộ điều khiển tỉ lệ và tích phân.
- PD (Proportional and Derivative (PD) Controller) gọi là bộ điều khiển đạo hàm.
- PID (Proportional, Integral, and Derivative (PID) Controller) là bộ điều khiển tỉ lệ – tích phân- đạo hàm (vi phân).

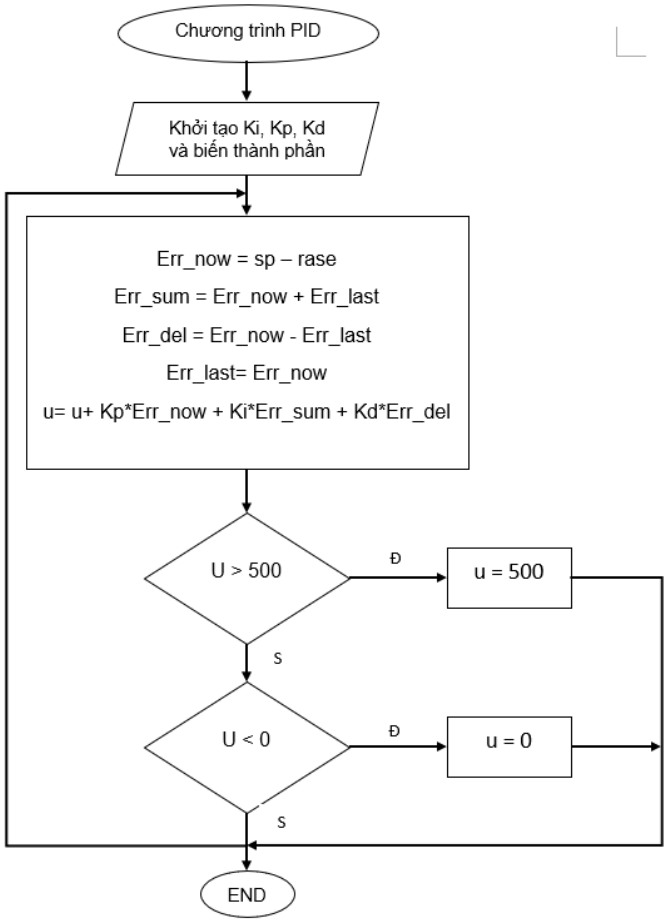
Sau khi nghiên cứu về lí thuyết, tiếp theo chúng em đi xây dựng lưu đồ thuật toán điều khiển lập trình trên chip STM32.



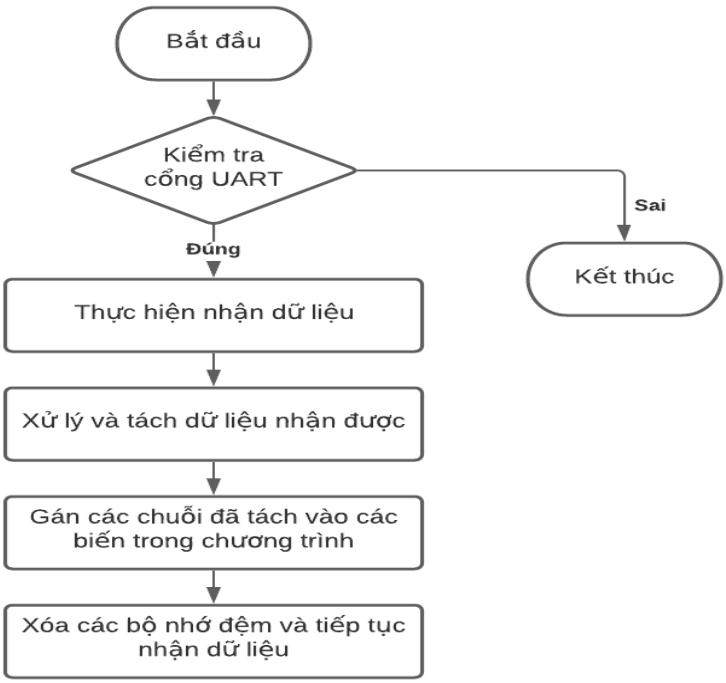
Hình 2.29: Chương trình chính của bộ điều khiển



Hình 2.30: Chương trình tính tốc độ

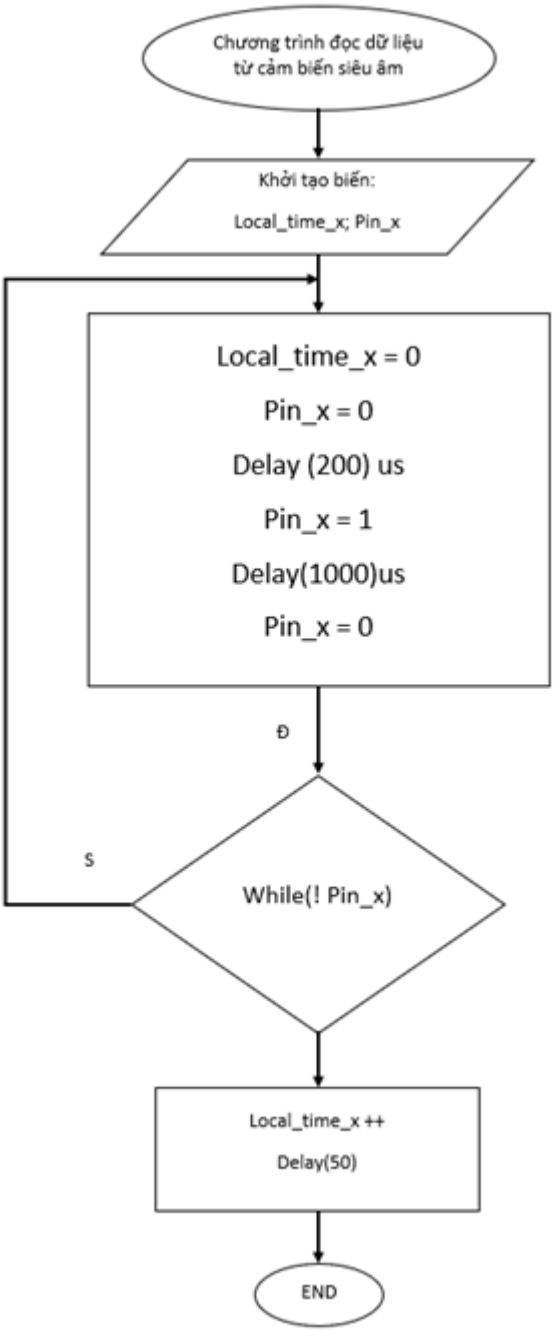


Hình 2.31: Chương trình PID

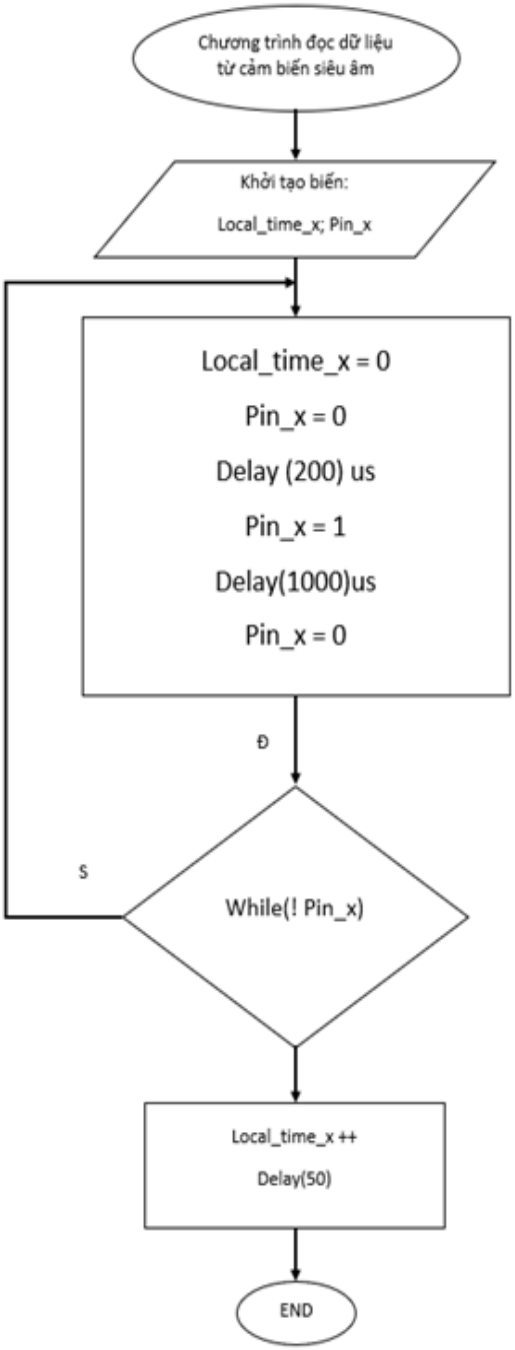


Hình 2.32: Thuật toán lí dữ liệu truyền thông lên Web

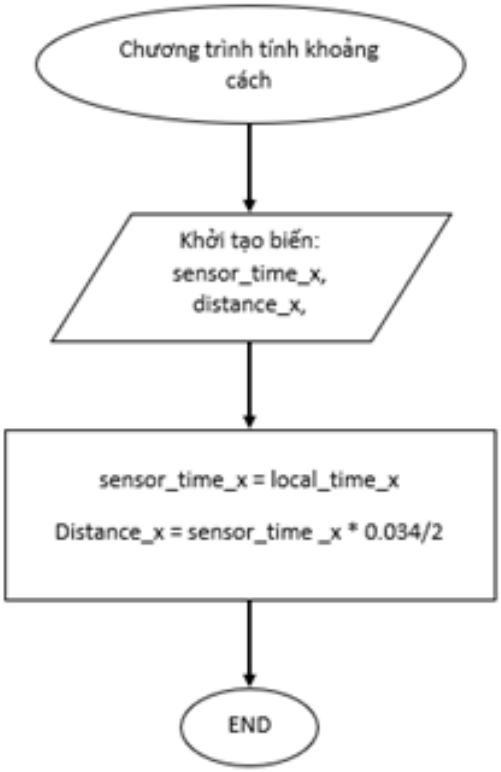
Tiếp theo là chúng em xây dựng thuật toán né vật cản sử dụng cảm biến siêu âm. Rồi lập trình trên chip SMT32F411VET6 cài đặt xuống xe.



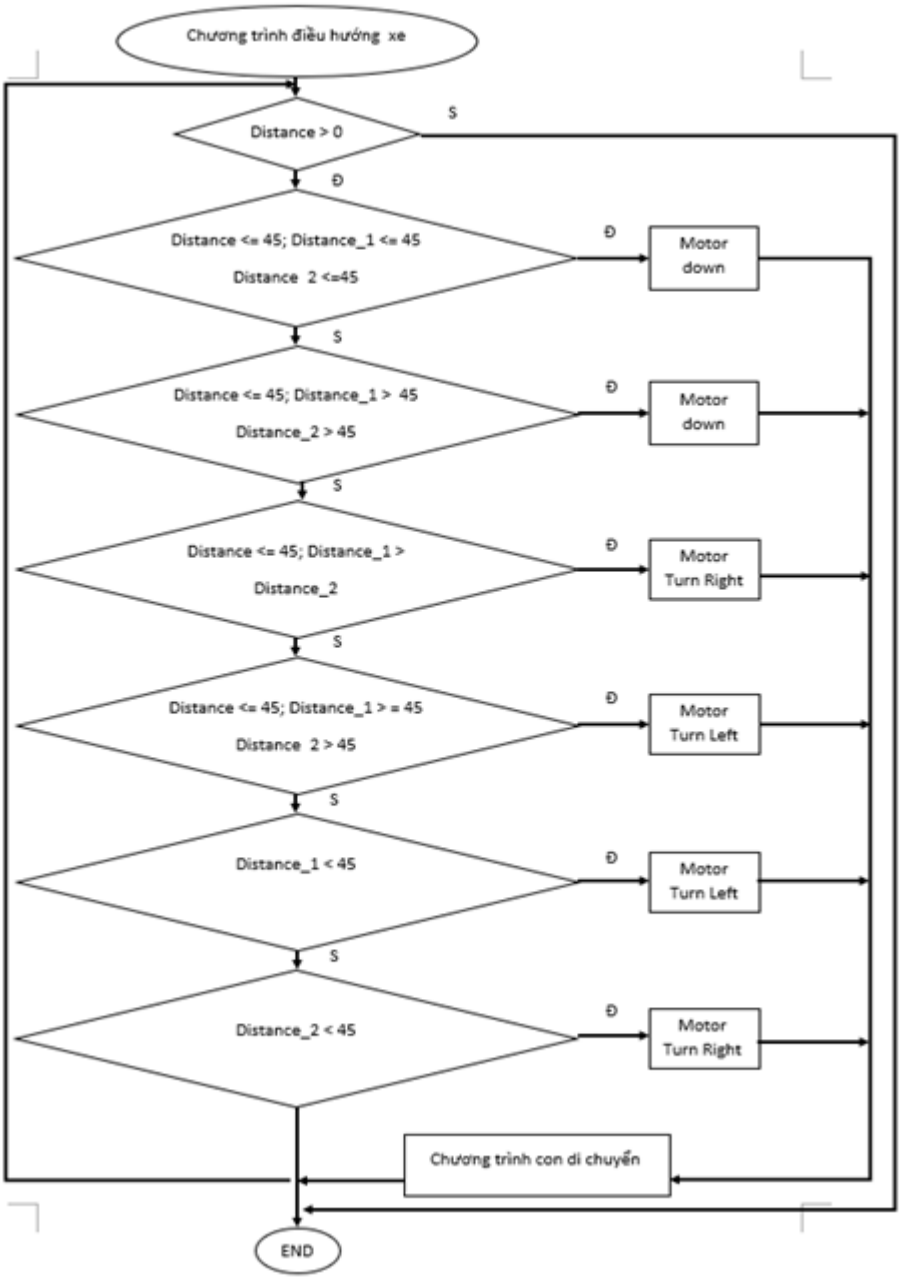
Hình 2.33: Chương trình đọc dữ liệu từ cảm biến siêu âm



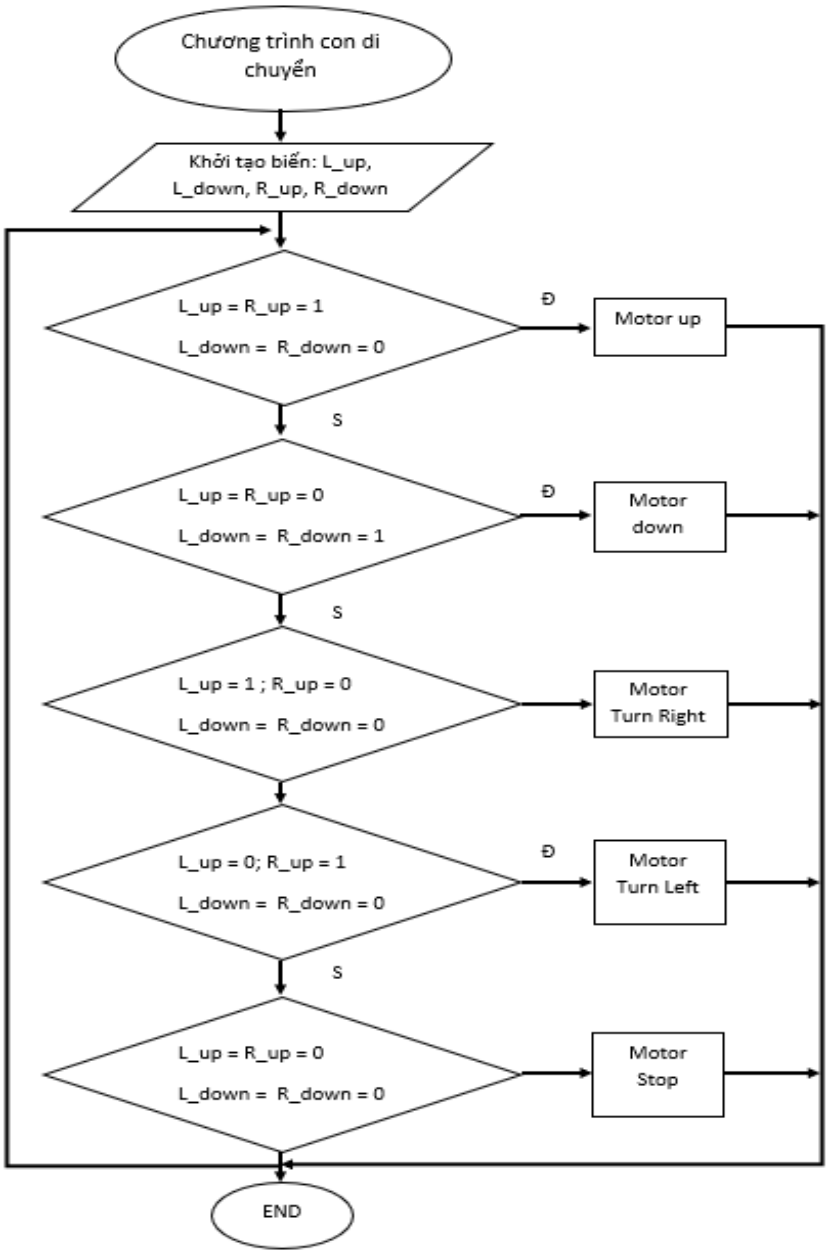
Hình 2.34: Chương trình điều hướng xe



Hình 2.35: Chương trình tính toán khoảng cách



Hình 2.36: Chương trình điều hướng xe



Hình 2.37: Chương trình con di chuyển

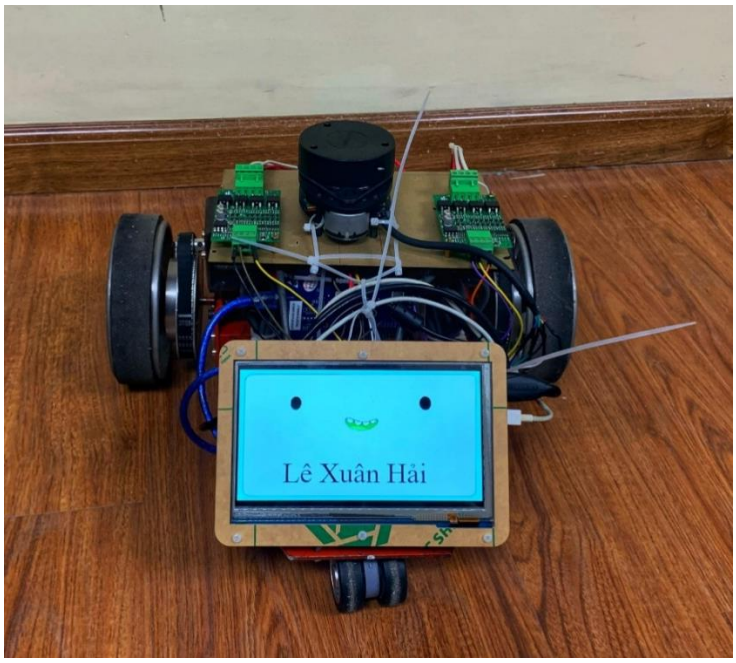
2.5 Kết quả đạt được

- Xây dựng thành công bản đồ môi trường dựa trên công nghệ Lidar.
- Giao tiếp giữa máy tính nhúng(Pi 4 Model B) với STM32 xây dựng hướng di chuyển cho robot.
- Xây dựng thành công bộ điều khiển PID cài đặt lên robot tự hành.
- Robot nhận diện và né được vật cản.
- Hiển thị được tốc độ lên Websever.

CHƯƠNG 3: KẾT QUẢ NGHIÊN CỨU

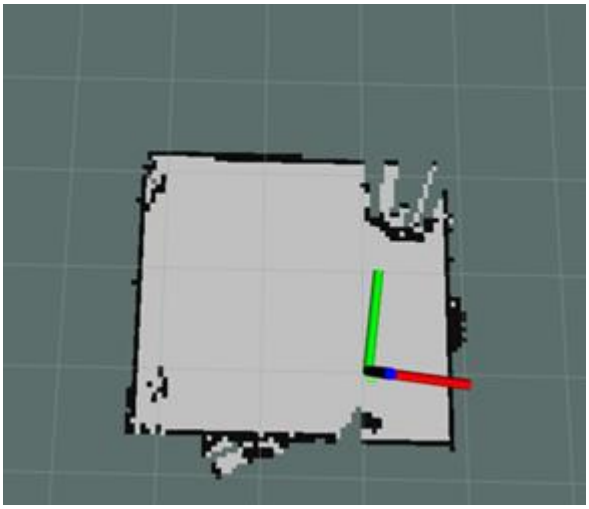
Trong quá trình thực nghiệm, chúng em cho xe chạy tại tần 10 nhà A1, trên địa hình phẳng.

3.1 Chương trình xây dựng bản đồ môi trường.



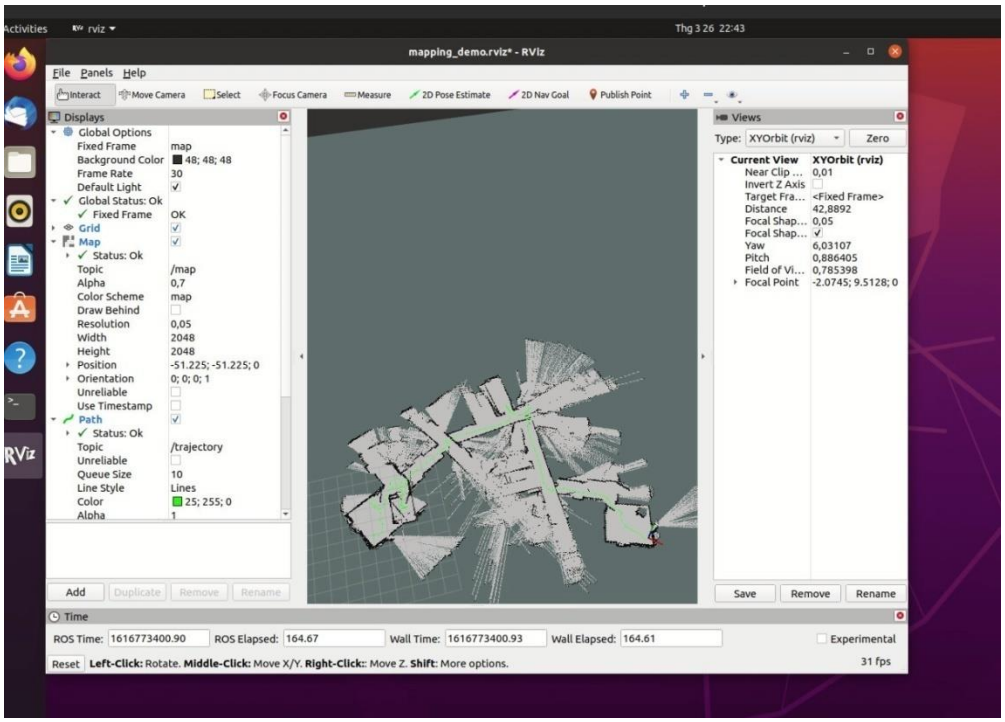
Hình 3.1: Hình ảnh xe thực tế

Đầu tiền chúng em cho cảm biến Lidar quét và xây dựng bản đồ tại không gian nhỏ, cho Lidar đứng tại một chỗ. Với mục đích xcc định được các vật cản xung quanh cảm biến, xây dựng được bản đồ một cách nhanh nhất và chính xác nhất. Về việc xây dựng bản đồ chúng em dùng thuật toán vẽ bản đồ(Mapping)(mục 2.3.2.1). Trong quá trình xây dựng bản đồ chúng em gặp một vấn đề là khi thay đổi góc quét của cảm biến Lidar tại một địa điểm, 2 khung hình bản đồ liền kề nhau sẽ bị đè lên nhau.. Chúng em đã sử dụng thuật toán Odometry(mục 2.3.2.2)



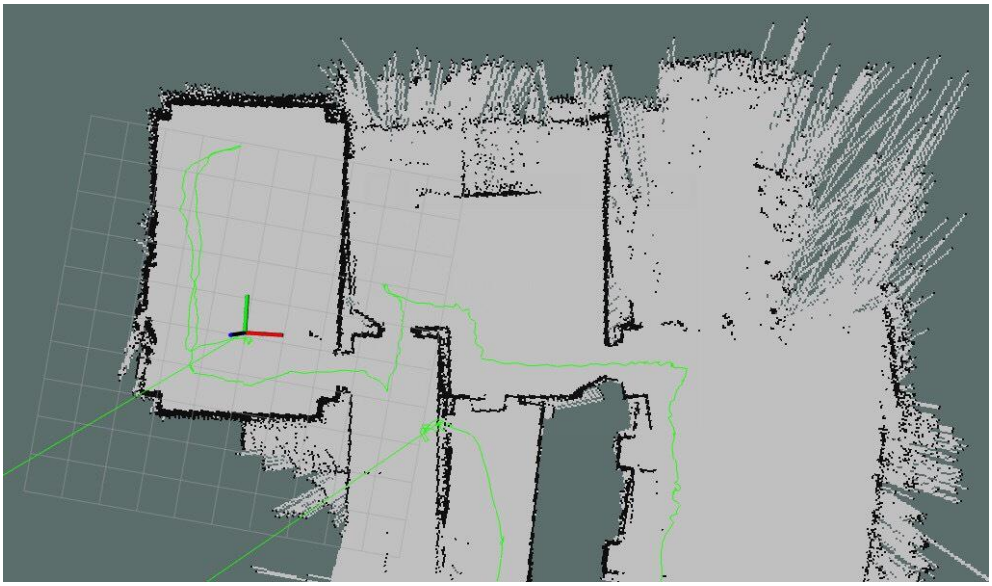
Hình 3.2: Xây dựng bản đồ tại một không gian nhỏ

Sau khi xây dựng thành công bản đồ tại một không gian nhỏ. Chúng em đặt cảm biến lên xe cùng việc điều khiển bán tự độ cho xe chạy để xây dựng bản đồ.Bước đầu thì vây xây dựng bản đồ vẫn còn chưa được chính xác, sau nhiều lần chỉnh định, lập trình lại hệ thống bản đồ cũng được xây dựng lại hoàn chỉnh và hoàn thiện hơn.



Hình 3.3: Bản đồ tầng 10 nhà A1 ở chế độ bán tự động

Sau khi điều khiển bán tự động, tiếp theo chúng em quyết định ghép nối thêm cảm biến siêu âm vào xe, cho xe chạy ở chế độ tự độ đồng thời truyền dữ liệu tốc độ lên tốc độ lên Websever chạy theo thuật toán PID.

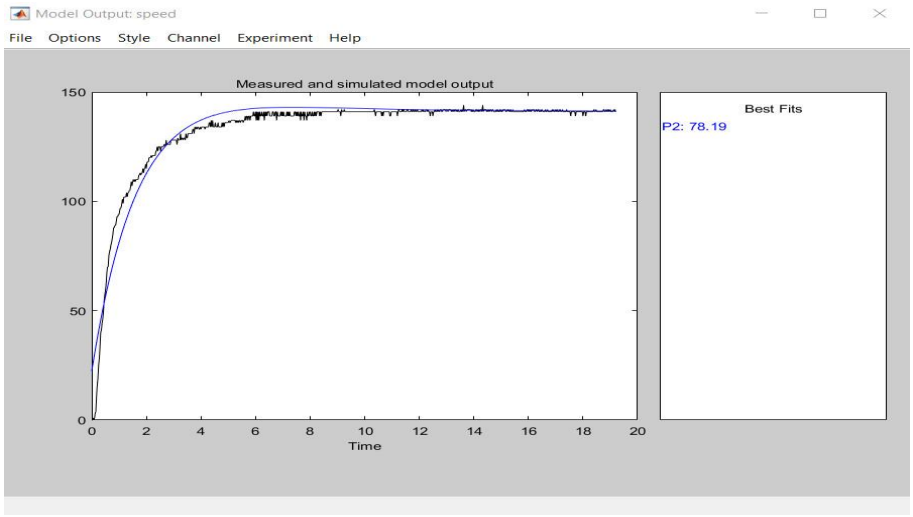


Hình 3.4: Quá trình xây dựng bản đồ tâng 10 nhà A1 ở chết độ tự động

Như có thể quan sát trên hình vẽ, việc xây dựng bản đồ môi trường tâng 10 nhà A1 rất rõ nét. Đường màu xanh trên hình là những đường robot tự hành đã đi qua. Các chấm đen thể hiện vật cản, tường xung tại môi trường xung quanh xe được vẽ lại. Tổ hợp ba vector đỏ xanh dương xanh lá thể hiện vị trí trí và hướng của robot.

3.2 Thuật toán PID

Dưới đây là kết quả mô phỏng PID trên phần mềm Matlap của nhóm nghiên cứu chúng em.

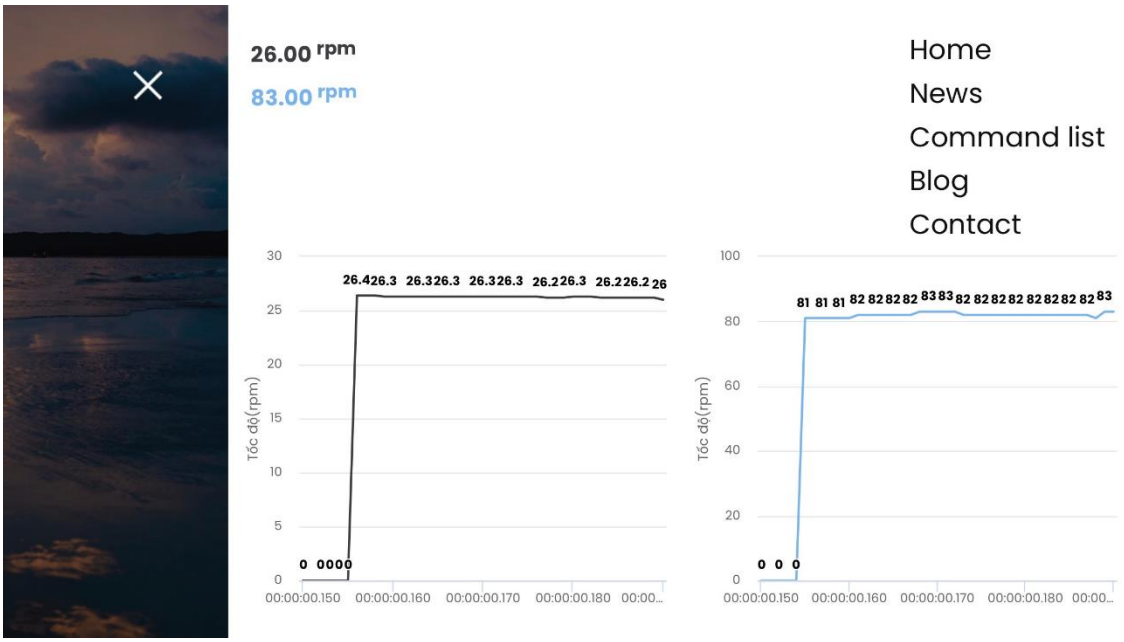


Hình 3.5: Kết quả mô phỏng đối tượng trên Matlab

Biểu đồ thể hiện tốc độ theo thời gian mô tả đúng 78,19% so với đối tượng thực với hàm truyền được tính ra như sau:

$$G(s) = \frac{K_0}{(1 + T_{M1} * s)(1 + T_{M2} * s)} = \frac{11.769}{(1 + 1.84 * s)(1 + 2.71 * s)}$$

Sau khi mô phỏng thuật toán chúng em tiếp tục cài đặt thực lên xe cho xe chạy với thuật toán PID, đồng thời truyền dữ liệu lên Web để theo dõi tốc độ của xe.



Hình 3.6: Xe khi chạy với thuật toán PID

3.3 Kết luận

Trong quá trình nghiên cứu và thực hiện, dưới chỉ bảo tận tình của TS. Lê Xuân Hải. Nhóm chúng em đã đạt được một số kết quả như sau:

- Đề tài nghiên cứu đã hoàn thành đúng tiến độ đã đề ra.
- Thành công trong việc xây dựng bản đồ môi trường sử dụng cảm biến Lidar chạy trên hệ điều hành ROS.
- Tích lũy được nhiều kiến thức về hệ điều hành ROS và cảm biến thông minh Lidar.
- Thành công trong việc cài đặt thuật toán PID giám sát tốc độ qua Websever thông qua STM32 và ESP8266.
- Bổ sung và củng cố thêm các kiến thức về lập trình, lối tư duy để giải quyết một vấn đề.
- Nâng cao khả năng làm việc nhóm và trách nhiệm công việc đối với từng thành viên trong nhóm.

3.4 Hướng phát triển nghiên cứu

- Ứng dụng được nhiều chứ năng của cảm biến Lidar cùng với hệ điều hành ROS vào cho robot tự hành.
- Kết hợp trí tuệ nhân tạo AI vào thêm cho xe giúp nhận biến khuôn mặt, và các vật thể giống như con người.
- Cải tiến thiết bị cùng với cơ khí của xe để điều khiển với khoảng cách xa và có thể áp dụng được vào thực tiễn.
- Áp dụng được các bộ điều khiển như trượt, PIDm lên robot tự hành.

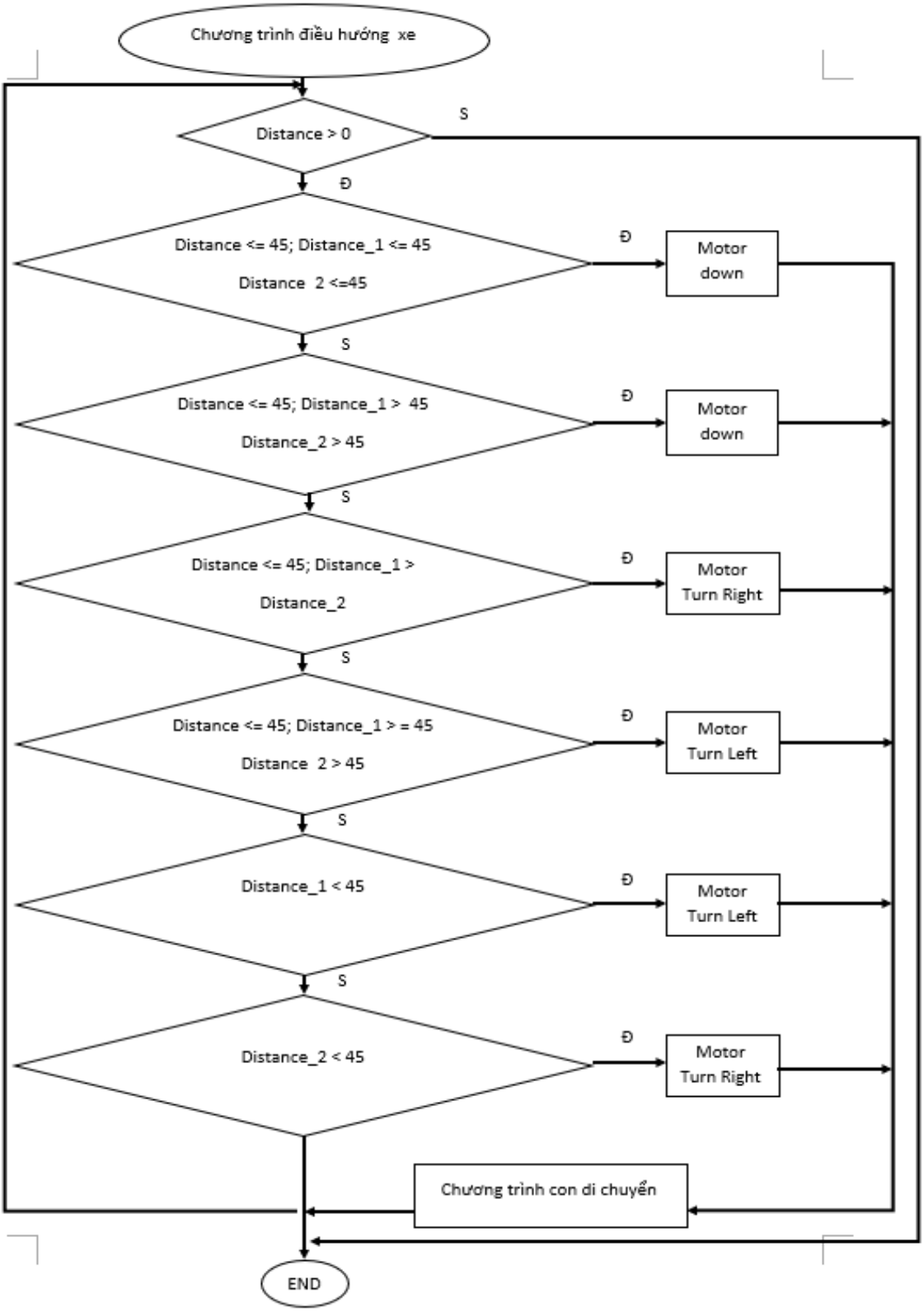
Tài liệu tham khảo

[1] Le Xuan Hai, Nguyen Van Thai , Bui Trong Duong, Vu Thi Thuy Nga, Thai Huu Nguyen , Phan Xuan Minh : IMPLEMENTATION OF A LABORATORY OVERHEAD CRANE CONTROL SYSTEM. Tạp chí Nghiên cứu KH&CN quân sự, Số 44, 08 – 2016.

[2] Đỗ Duy Phú – Nguyễn Thu Hà : Giáo trình KỸ THUẬT VI XỬ LÝ VÀ VI ĐIỀU KHIỂN. Nhà xuất bản Khoa Học và Kỹ Thuật. năm 2016

[3] Bùi Văn Huy :Giáo trình LÝ THUYẾT ĐIỀU KHIỂN TỰ ĐỘNG Đại học Công Nghiệp Hà Nội.năm 2018.

[4] ROS.org và RPLIDAR A1 datasheet



Hình 3-6.1: Chương trình điều hướng xe

Dựa vào phần cứng là mạch điều khiển động cơ với mỗi đông cơ sử dụng một cầu H có đầu vào là tín hiệu logic. Từ nguyên lý đó để điều khiển xe tiến lùi hay quay trái quay phải ta chỉ cần thay đổi mức logic cấp cho các chân tín hiệu của mạch điều khiển động cơ. Việc cấu hình cũng như thay đổi mức logic này được thực hiện dễ dàng trong bước cấu hình vi điều khiển và có thể thay đổi theo biến số trong chương trình chính.