TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG

……o0o……

ĐỒ ÁN 2

XE TỰ HÀNH DÙNG LIDAR
NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **NGUYỄN PHÚC TOÀN**
MSSV: 19119139
VŨ THÙY PHƯƠNG TRINH
MSSV: 19119140

Hướng dẫn: **PGS.TS. TRƯƠNG NGỌC SƠN**

TP. HỒ CHÍ MINH 12/2024

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG

……o0o……

HCMUTE

ĐỒ ÁN 2

XE TỰ HÀNH DÙNG LIDAR
NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **NGUYỄN PHÚC TOÀN**
MSSV: 19119139
VŨ THÙY PHƯƠNG TRINH
MSSV: 19119140

Hướng dẫn: **PGS.TS. TRƯƠNG NGỌC SƠN**

TP. HỒ CHÍ MINH 12/2024

MỤC LỤC

DANH MỤC HÌNH

CHƯƠNG 1: TỔNG QUAN

1.1. ĐẶT VẤN ĐỀ

Tự động hóa vẫn luôn là chủ đề nghiên cứu thu hút nhiều sự quan tâm trong thời đại ngày nay. Những chiếc xe hay robot tự điều khiển mà không cần đến tác động của con người. Đây cũng là đề tài có tiềm năng rất lớn trong nhiều lĩnh vực khác nhau, là tiền đề cho các thiết bị hỗ trợ trong sinh hoạt sống của con người. Trên thực tế khi muốn lấy dữ liệu từ các địa hình hiểm trở khác nhau, đôi khi tồn tại một số môi trường không thuận lợi cho con người thực hiện việc khảo sát như những nơi chật hẹp hay cần ghi nhận từ vị trí cao. Vì thế các robot địa hình hay drone ra đời để giải đáp những khó khăn về mặt địa lý khi cần thu thập dữ liệu về môi trường. Các robot trên đều có đặc điểm chung là có thể ghi nhận lại những gì mà nó "nhìn thấy", trên hết là có thể đáp ứng được các vấn đề về tự động hóa, tức là có thể tránh các vật cản một cách chủ động mà không cần con người điều khiển. Nhóm thực hiện đã dựa vào ý tưởng này để đề xuất mục tiêu cho đồ án 2, đó là hệ thống xe tự hành sử dụng LiDAR.

Ý tưởng của đề tài chính là thiết kế một chiếc xe tự hành sử dụng cảm biến LiDAR. LiDAR (Light Detection and Ranging) là công nghệ đo khoảng cách bằng cách sử dụng laser để tạo ra bản đồ của môi trường xung quanh. Thiết bị này có thể vẽ lại bản đồ của khu vực để xác định phương hướng thích hợp cho việc di chuyển, và nhờ sử dụng những chùm tia laser mà thiết bị có thể đo được khoảng cách đến chướng ngại vật để né vật cản một cách chủ động.

1.2. MỤC TIÊU

Hệ thống xe tự hành là đề tài được nhiều nghiên cứu sinh quan tâm bởi tiềm năng phát triển của chúng. Mục tiêu của sản phẩm hệ thống xe tự hành cần phải đạt được một số yêu cầu của người dùng cũng như yêu cầu hệ thống. Xe tự hành có thể vận hành tự động mà không cần điều khiển. Thực hiện quét xung quanh để tạo bản đồ 2D của khu vực, và từ bản đồ đó di chuyển né các chướng ngại vật trên đường đi. Hệ thống cần đạt đến độ chính xác cao, thời gian quét môi trường thấp và tối ưu trong việc di chuyển.

1.3. NỘI DUNG NGHIÊN CỨU

- Tổng quan về thuật toán SfM.
- Tổng quan về thuật toán SLAM cũng như các dạng khác nhau của thuật toán.
- Giới thiệu về LiDAR A1M8 và Jetson Nano.
- Thiết kế hệ thống xe tự hành có thể vẽ lại bản đồ 2D.
- Hoàn thiện hệ thống và tiến hành thử nghiệm.
- Đánh giá kết quả.

1.4. GIỚI HẠN ĐỀ TÀI

Ngoài các chức năng mà xe tự hành có thể thực hiện, hệ thống vẫn còn một số nhược điểm chưa thể được tối ưu liên quan đến giá thành của từng bộ phận, các hạn chế về kiến thức chuyên môn cũng như thời gian thực hiện.

- Vì LiDAR có giá thành khá cao so với các cảm biến dò đường thông thường nên cảm biến LiDAR A1M8 mà nhóm sử dụng chỉ có thể quét trong phạm vi tối đa 12 mét.

- Laser từ cảm biến khi gặp một số môi trường phản xạ kém có thể làm sai lệch về độ chính xác gây nhiễu khi vẽ bản đồ.

- Hệ thống xe vẫn chưa điều chỉnh được cùng với tốc độ quét bản đồ.

- LiDAR chỉ đọc và vẽ được bản đồ 2D nên không thể nhận diện các vật cản khác tầm xoay của cảm biến, nếu vật ở quá thấp hay quá cao sẽ không thể né được.

1.5. BỐ CỤC

Chương 1: Tổng quan

Đặt vấn đề và giới thiệu tổng quát về đề tài, mục tiêu cũng như giới hạn đặt ra.

Chương 2: Cơ sở lý thuyết

Đưa ra một số lý thuyết cần thiết sử dụng trong đề tài bao gồm thuật toán SfM, SLAM, LiDAR, động cơ encoder và giao thức I2C.

Các lựa chọn phần cứng sao cho phù hợp với yêu cầu của đề tài.

Chương 3: Thiết kế xe tự hành

Trình bày sơ đồ hệ thống và giải thích cơ chế hoạt động của hệ thống.

Chương 4: Thuật toán quét LiDAR 2D

Đi sâu vào chi tiết của thuật toán và cách áp dụng thuật toán vào thiết kế.

Chương 5: Kiểm tra và đánh giá

Đưa ra các kết quả thực nghiệm và đánh giá hệ thống.

Chương 6: Hướng phát triển

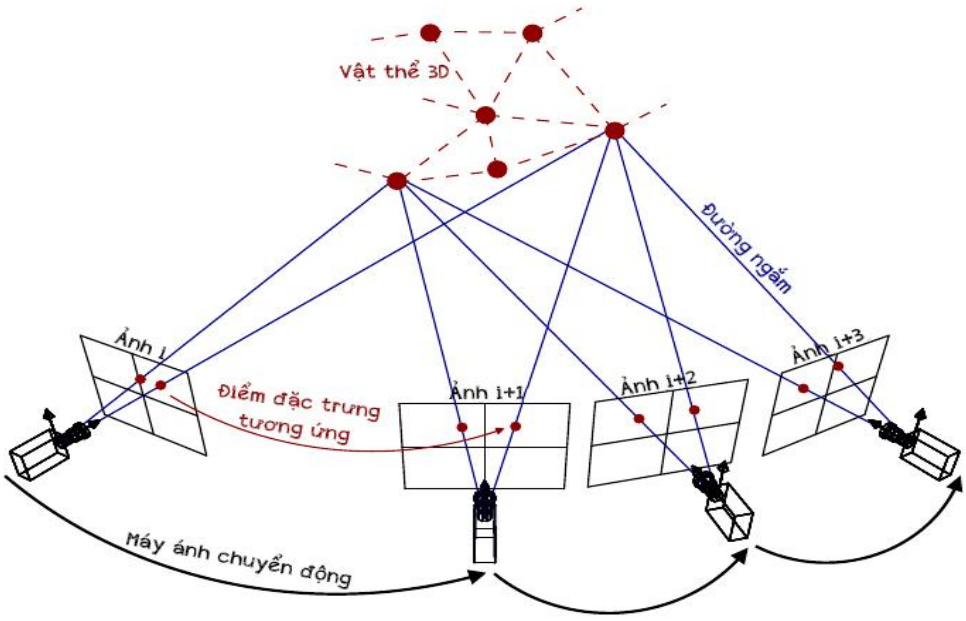
Kết luận về hệ thống và hướng phát triển.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. TỔNG QUAN VỀ THUẬT TOÁN SFM

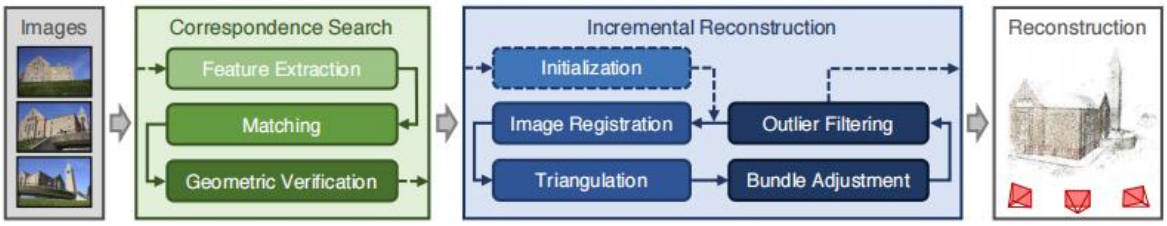
2.1.1. Khái niệm

SFM (Structure from Motion) là một lĩnh vực trong thị giác máy tính hướng đến việc tái tạo lại cấu trúc ba chiều của môi trường từ một chuỗi hình ảnh hai chiều. Công nghệ này có thể được ứng dụng trong đa dạng các lĩnh vực như robot, khảo cổ học và quy hoạch đô thị. Trọng tâm của SfM là một tập hợp các thuật toán phức tạp cho phép trích xuất thông tin không gian từ một loạt các hình ảnh.



Hình 2.1. Minh họa thuật toán SfM

Thuật toán SfM cho phép tính toán đồng thời các ma trận chiếu và các điểm 3D bằng cách sử dụng các điểm tương ứng trong mỗi khung nhìn. Đây là quy trình tái cấu trúc của thuật toán SfM:



Hình 2.2. Quy trình thuật toán SfM

+ Phân loại đặc trưng (Feature Extraction): Từ mỗi hình ảnh, SfM nhận diện các đặc trưng tại vị trí nhất định. Các đặc điểm phải bất biến dưới những thay đổi về bức xạ và hình học để SfM có thể nhận ra chúng trong nhiều hình ảnh khác nhau. Thuật toán SIFT và các dạng tương đương của nó được xem là tiêu chuẩn chuẩn mực về độ chính xác.

+ Nối (Matching): SfM phát hiện những hình ảnh có nội dung giống nhau phần cảnh bằng cách tận dụng các đặc trưng làm diện mạo mô tả của các hình ảnh. Nó tìm kiếm các đặc điểm tương ứng bằng cách tìm điểm giống nhất trong hình ảnh A cho mọi đặc điểm trong ảnh B. Cách tiếp cận này có độ phức tạp tính toán cao và cực kỳ hạn chế đối với các tập hình ảnh lớn. Đầu ra là một tập hợp các cặp hình ảnh có khả năng chồng chéo và các tính năng tương đồng liên quan của chúng.

+ Xác minh hình học (Geometric Verification): Ở giai đoạn thứ ba, thuật toán xác minh các cặp hình ảnh có khả năng chồng lấp. Vì bước matching chỉ dựa vào ngoại hình, không đảm bảo rằng các đặc điểm tương ứng thực sự ánh xạ tới cùng một điểm cảnh. Do đó SfM xác minh sự trùng khớp bằng cách ước tính một phép biến đổi ánh xạ các điểm đặc trưng giữa các hình ảnh bằng hình học xạ ảnh. Hình học Epipolar mô tả mối quan hệ cho một camera chuyển động thông qua ma trận thiết yếu E (đã hiệu chuẩn) hoặc ma trận cơ bản F (chưa hiệu chuẩn), và có thể được mở rộng thành ba khung nhìn bằng cách sử dụng tensor ba tiêu điểm 3x3x3. Nếu phép biến đổi hợp lệ ánh xạ một số đủ các đặc điểm giữa các hình ảnh, chúng được coi là đã được xác minh về mặt hình học. Ngõ ra của giai đoạn này được gọi là biểu đồ cảnh với các hình ảnh là các nút và các cặp hình ảnh đã được xác minh là cạnh.

+ Khởi tạo (Initialization): SfM khởi tạo mô hình bằng cách tái cấu trúc hai chế độ xem được lựa chọn cẩn thận. Việc lựa chọn một cặp ban đầu phù hợp rất quan trọng vì việc tái thiết có thể không bao giờ phục hồi sau một lần khởi tạo xấu. Khởi tạo từ vị trí dày đặc trong biểu đồ hình ảnh với nhiều camera chồng chéo thường mang lại kết quả tốt hơn và tái thiết chính xác hơn.

+ Thiết lập hình ảnh (Image Registration): Bắt đầu từ việc xây dựng lại số liệu, hình ảnh mới có thể được thêm vào mô hình hiện tại bằng cách giải bài toán phối cảnh n điểm bằng cách sử dụng tính năng tương ứng với các điểm tam giác đã có hình ảnh thiết lập trước (2D-3D tương đồng).

+ Tam giác đạc (Triangulation): Hình ảnh mới thiết lập phải tuân thủ điểm cảnh hiện có. Ngoài ra, nó còn có thể làm tăng phạm vi cảnh bằng cách mở rộng tập hợp các điểm thông qua tam giác đạc. Đây là một bước quan trọng trong SfM vì nó làm tăng tính ổn định của mô hình hiện có thông qua tính dư thừa và cho phép thiết lập hình ảnh mới bằng cách cung cấp thêm các 2D-3D tương đồng.

+ Điều chỉnh gói (Bundle Adjustment): Thiết lập hình ảnh và tam giác đạc là hai hoạt động riêng biệt, mặc dù sản phẩm có mối tương quan cao – sự không chắc chắn trong tư thế máy ảnh lan truyền đến các điểm tam giác và ngược lại, đồng thời các tam giác bổ sung có thể cải thiện tư thế máy ảnh ban đầu thông qua việc tăng độ dư thừa.

2.1.2. Ứng dụng

Thuật toán SfM đã và đang được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau với các ứng dụng như:

+ Trò chơi điện tử: Thuật toán được sử dụng để tạo môi trường 3D và các đối tượng trong thế giới ảo, trò chơi điện tử và ứng dụng thực tế ảo. Bằng cách ước lượng cấu trúc 3D từ các hình ảnh, SfM giúp xây dựng môi trường 3D chân thực và sống động.

+ Địa lý và bản đồ số: Trong lĩnh vực địa lý, SfM được sử dụng để tạo ra bản đồ 3D của môi trường từ dữ liệu hình ảnh thu thập từ máy bay không người lái (UAV) kết hợp với xử lý ảnh.

+ Điều khiển thiết bị tự hành: SfM được sử dụng để hỗ trợ việc định vị và xây dựng bản đồ cho robot tự động và xe tự hành.

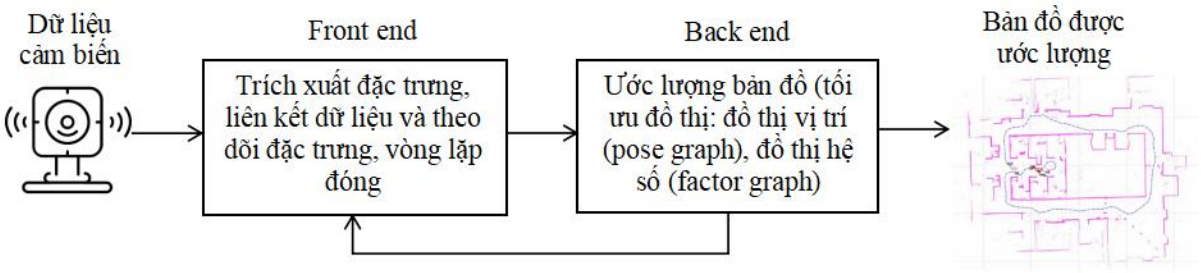
+ Giám sát và an ninh: SfM có thể được áp dụng trong giám sát và an ninh để theo dõi và phát hiện chuyển động, xây dựng mô hình 3D của một khu vực và phân tích hành vi.

2.2. TỔNG QUAN VỀ THUẬT TOÁN SLAM

2.2.1. Khái niệm

SLAM (Simultaneous Localization And Mapping) là thuật toán ứng dụng cho các thiết bị xe tự hành, thuật toán có thể xây dựng bản đồ cũng như xác định vị trí của thiết bị trong cùng một lúc. Thuật toán này cũng cho phép các thiết bị tự hành nhận biết bản đồ trong các môi trường khác nhau.

SLAM bao gồm hai bước front end và back end để tạo được bản đồ cuối cùng. Tại bước front end là xử lý tín hiệu cảm biến, phụ thuộc rất nhiều vào cảm biến được sử dụng. Bước tiếp sau là để tối ưu hóa bản đồ, bước này không còn phụ thuộc vào dữ liệu cảm biến nữa.



Hình 2.3 Phương thức hoạt động của SLAM

Trên lý thuyết, SLAM sau khi ước lượng được bản đồ sẽ di chuyển xung quanh để tạo thành một bản đồ hoàn chỉnh. Trên thực tế, dữ liệu từ cảm biến chứa nhiều nhiễu, hoặc cảm biến di chuyển chệch hướng so với vị trí được ước tính dẫn đến sai lệch trên kết quả ngày càng lớn. Biểu đồ vị trí giúp kết nối các điểm vị trí của robot lại với nhau và nối điểm đầu và cuối trong khi vẫn giữ nguyên khoảng cách di chuyển thực tế của robot. Bằng cách này có thể "kéo" các sai lệch về đúng hoặc gần đúng với vị trí ước tính, tăng chính xác tương đối của bản đồ thực tế. Ngoài ra, SLAM sử dụng thuật toán vòng lặp đóng để ghi nhận các đặc trưng đã gặp qua nhằm sửa các lỗi tích lũy của vị trí khác nhau từ robot.

Thuật toán vòng lặp đóng là một thuật toán phụ của SLAM nhằm xác định các vị trí đã truy cập trước đó và sử dụng chúng để sửa các lỗi tích lũy trong ước lượng vị trí của robot. Thuật toán có thể được sử dụng bởi các phương pháp khác nhau, bao gồm các ràng buộc hình học, phương pháp dựa trên bề ngoài và phương pháp dựa trên

thời gian. Các sai lệch trong kết quả của vòng lặp đóng có thể do vị trí trông giống nhau, thay đổi ánh sáng, nhiễu cảm biến hoặc các yếu tố khác. Thuật toàn được sử dụng hầu hết trong các điểm ảnh từ LiDAR.

SLAM đã trở thành đề tài nghiên cứu trong nhiều năm qua. Với những cải tiến về tốc độ xử lý của máy tính và sự sẵn có của các cảm biến giá rẻ như máy ảnh và máy đo phạm vi như laser, thuật toán SLAM hiện được ứng dụng trong ngày càng nhiều lĩnh vực khác nhau.

Quy trình hoạt động của SLAM gồm hai bước chính:

+ Định vị (Localization): Dựa trên laser của cảm biến để xác định vị trí của thiết bị theo thời gian.

+ Lập bản đồ (Mapping): Nhờ vào cảm biến để thu được môi trường xung quanh, sau đó di chuyển để lấy được thông tin trực quan hơn, trích xuất các đặc điểm để có thể xây dựng thành một bản đồ.

Ứng dụng: Thuật toán SLAM hiện được ứng dụng trong một số thiết bị tự vận hành như robot quét dọn, xe hơi tự động hay robot tự hành.

Một số nhược điểm của thuật toán:

+ Lỗi sai lệch định vị gây ra chênh lệch đáng kể so với giá trị thực tế. Thuật toán SLAM chuyển động tuần tự bao gồm những vị trí sai số. Lỗi này tích lũy theo thời gian, gây ra sai lệch đáng kể so với giá trị thực tế. Nó cũng có thể khiến dữ liệu bị biến dạng gây khó khăn cho việc xác định bản đồ.

+ Chi phí tính toán cao trong việc xử lý hình ảnh và tối ưu hóa.

+ Chi phí các cảm biến như LiDAR, camera phân giải tốt có giá thành cao.

2.2.2. Các biến thể SLAM

SLAM trực quan (Visual SLAM hoặc vSLAM) sử dụng hình ảnh thu được từ máy ảnh và các cảm biến hình ảnh khác. Visual SLAM có thể được sử dụng bởi các loại máy ảnh đơn giản (máy ảnh góc rộng, mắt cá và hình cầu), máy ảnh mắt ghép (âm thanh nổi và đa máy ảnh), máy ảnh RGB-D (máy ảnh độ sâu và máy ảnh thời gian bay ToF). Visual SLAM có thể được triển khai với chi phí tương đối thấp. Ngoài ra, vì camera cung cấp một lượng lớn thông tin nên chúng có thể được sử dụng để phát hiện các điểm mốc (các vị trí đã đo trước đó). Việc phát hiện cột mốc cũng có thể được kết hợp với tối ưu hóa dựa trên biểu đồ, mang lại sự linh hoạt trong việc triển khai SLAM.

SLAM một mắt (Monocular SLAM) là một thuật toán vSLAM sử dụng một camera duy nhất làm cảm biến, điều này khiến việc xác định độ sâu trở nên khó khăn. Vấn đề này có thể được giải quyết bằng cách phát hiện điểm đánh dấu AR, bàn cờ hoặc các vật thể đã biết khác trong ảnh để định vị hoặc bằng cách kết hợp thông tin camera với một cảm biến khác như đơn vị đo quán tính (IMU), có thể đo các đại lượng vật lý như vận tốc và hướng. Công nghệ liên quan đến vSLAM bao gồm cấu trúc từ chuyển động (SfM), đo hình ảnh trực quan và điều chỉnh bó.

LiDAR SLAM (Light Detection And Ranging SLAM) là phương pháp chủ yếu sử dụng cảm biến laser (hoặc cảm biến khoảng cách). So với máy ảnh bình thường, máy ảnh ToF và các cảm biến khác, tia laser có độ chính xác cao hơn và được sử dụng cho các ứng dụng có phương tiện di chuyển tốc độ cao như ô tô tự lái và máy bay không người lái. Các giá trị đầu ra từ cảm biến laser thường là dữ liệu điểm 2D (x, y) hoặc 3D (x, y, z). Cảm biến laser cung cấp các phép đo khoảng cách có độ chính xác cao và hoạt động hiệu quả trong việc xây dựng bản đồ bằng thuật toán SLAM. Tính toán chuyển động (quãng đường đã di chuyển) được sử dụng để định vị phương tiện. Ngoài ra có thể sử dụng phương pháp tiếp cận dựa trên tính năng như Lidar Odometry And Mapping (LOAM) hoặc Fast Global Registration (FGR), dựa trên các tính năng của FPFH. Bản đồ đám mây điểm 2D hoặc 3D có thể được biểu diễn dưới dạng bản đồ lưới hoặc bản đồ ba chiều voxel. Đối với các ứng dụng robot trong nhà, SLAM lidar 2D thường được sử dụng, trong khi SLAM 3D thường được sử dụng cho phương tiện không người lái (drone, flycam,...) và xe tự hành.

2.3. TỔNG QUAN VỀ LIDAR VÀ ỨNG DỤNG

2.3.1. Giới thiệu chung về LiDAR

LiDAR (Light Detection And Ranging hay Phát hiện ánh sáng và đo khoảng cách) là công nghệ viễn thám dựa trên laser. Nguyên lý của LiDAR chính là hướng một tia laser nhỏ vào một bề mặt và đo thời gian để tia laser quay trở lại nguồn phát. Với tốc độ ánh sáng biết trước, dựa vào thời gian chùm tia sáng quay lại để tính được khoảng cách giữa LiDAR và vật thể, áp dụng công thức sau:

$$d = \frac{1}{2} t.c$$

Với d là khoảng cách từ máy phát và vật thể hay vật cản, t là thời gian tia sáng quay về và c là tốc độ ánh sáng.

LiDAR là hệ thống có thể đo một triệu khoảng cách từ cảm biến đến các điểm trên bề mặt. Công nghệ tiên tiến này hoạt động rất nhanh vì có thể tính toán khoảng cách giữa các cảm biến LiDAR và mục tiêu với tốc độ ánh sáng (300.000km/s). Hệ thống tích hợp 3 thành phần chính:

+ Máy quét laser: Hệ thống LiDAR phát xung ánh sáng laser từ nhiều hệ thống di động khác nhau (ô tô, máy bay không người lái…) xuyên qua không khí và thảm thực vật (Laser trên không) và thậm chí cả nước (Laser đo độ sâu). Máy quét nhận được ánh sáng dội trở lại để đo khoảng cách và góc. Việc lựa chọn máy quét ảnh hưởng rất lớn đến độ phân giải và phạm vi vận hành của hệ thống LiDAR.

+ Hệ thống dẫn đường và định vị: Cho dù cảm biến LiDAR được gắn trên máy bay, ô tô hay UAS (hệ thống máy bay không người lái), điều quan trọng là phải xác định vị trí và hướng tuyệt đối của cảm biến để đảm bảo dữ liệu thu được là dữ liệu có thể sử dụng được. Hệ thống Vệ tinh Định vị Toàn cầu (GNSS) cung cấp thông tin địa lý chính xác về vị trí của cảm biến (vĩ độ, kinh độ, độ cao) và Đơn vị Đo lường Quán tính (IMU) xác định tại vị trí này hướng chính xác của cảm biến. Dữ liệu được ghi lại bởi 2 thiết bị này sau đó được sử dụng để tạo dữ liệu thành các điểm tĩnh là nền tảng của đám mây điểm ánh xạ 3D.

+ Công nghệ máy tính: Để tận dụng tối đa dữ liệu, cần phải tính toán để hệ thống LiDAR hoạt động bằng cách xác định vị trí ánh xạ chính xác. Nó cũng cần thiết cho việc trực quan hóa dữ liệu trên chuyến bay hoặc hậu xử lý dữ liệu để tăng độ hội tụ và độ chính xác được cung cấp trong đám mây điểm ánh xạ 3D.

2.3.2. Ứng dụng

LiDAR đóng vai trò như một thiết bị tìm kiếm vật thể hay vẽ lại bản đồ trong không gian ba chiều. Tùy vào giá thành và mục đích sử dụng, LiDAR có thể ứng dụng vào nhiều thiết bị điện tử khác nhau từ những thiết bị sử dụng trong nhà đến những thiết bị quan trắc thực địa trong không gian lớn.

Đối với ngành công nghiệp oto và robot, LiDAR đóng vai trò quan trọng như một "giác quan" giúp hệ thống nhận diện môi trường và cảm nhận các vật cản xung quanh, giúp dự đoán hướng đi chính xác để né các chướng ngại vật trên đường.

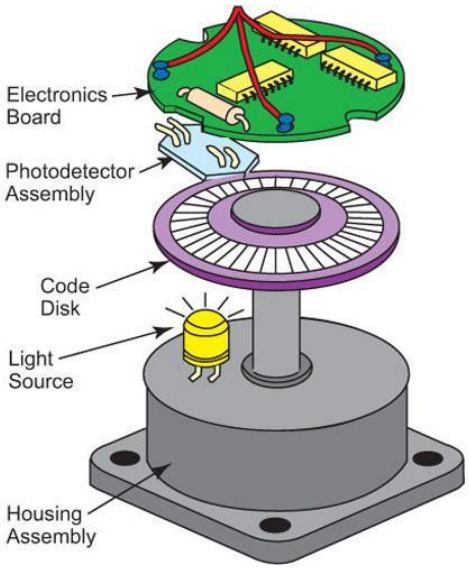
Đối với các ngành cần quan trắc thực địa và vẽ bản đồ, LiDAR có thể được sử dụng trên các con drone, xe địa hình hay tàu thuyền để đo đạc, thăm dò hay vẽ lại bản đồ. Trong những ứng dụng thuộc quan trắc địa, LiDAR có vai trò vô cùng quan trọng, bởi lẽ các thiết bị được sử dụng để thực hiện khảo sát tại những nơi mà con người không thể đặt chân đến.

2.4. TỔNG QUAN VỀ ĐỘNG CƠ ENCODER

2.4.1. Khái niệm về động cơ DC

Động cơ DC (direct current motor) là một động cơ có cấu tạo đơn giản từ 2 bộ phận trục quay động cơ (Rotor) và lớp vỏ tĩnh (Sator) bao quanh Rotor. Sator có vai trò bộ phận được cố định trong một động cơ để từ đó rotor tạo ra mô men xoắn từ đó có thể quay.

Với việc động cơ DC chỉ chạy theo chiều và cường độ của dòng điện nên việc xác định tốc độ quay (RPM) rất khó khăn là việc dừng motor đúng thời điểm hay ở khoảng cách cố định là gần như không thể, dẫn đến việc điều khiển chính xác trong các ứng dụng cách tay robot hay servo trở nên không khả thi, như nếu người dùng có thế xác định được góc quay của rotor thì những việc trên trở nên khả thi hơn.



Hình 2.4. Thiết kế cơ bản của bộ encoder cho động cơ DC

Encoder thường được có 2 loại chính:

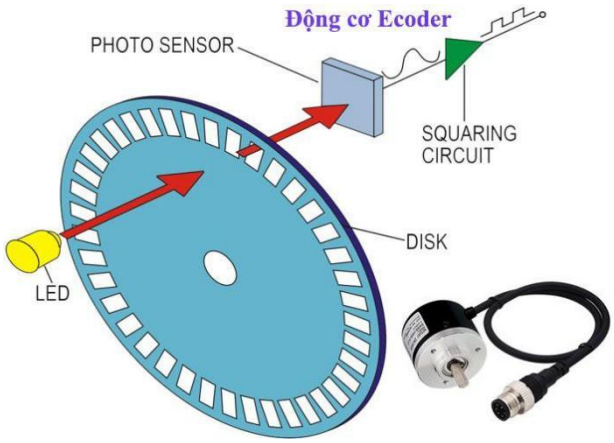
- Encoder trường của rotor để xác định tốc độ của động cơ với độ bền cao có thể hoạt động trong các môi trường khắc nghiệt những khi hoạt động trong gần các thiết bị điện tử thì từ trường sẽ gây nhiễu có thiết bị điện tử thậm chí là hư hỏng linh kiện điện tử.

- Encoder dùng đèn led thường là led hồng ngại và quang trở với thiết kế đơn giản giá thành rẻ độ chính xác cao cấu tạo 4 bộ phận chính:

+ Led và quang trở (Photodetector Assembly) với vai trờ như một nút nhấn kéo lên (pull-up button) và khi có vật cản đi qua thì mất đi nguồn sáng của led lúc này giá trị quang trở tăng làm cho dòng điện đi qua yếu hoặc ngắt.

+ Đĩa quay (Code disk) đóng vai trò tạo nguồn xung cho quang trở bằng việc thực hiện cho để cho ánh sáng đi đến quang trở thông qua các rảnh được khắc sẵn trên đĩa. Đĩa được nối với trục quay của rotor nên số vòng quay giữa bánh đĩa và trục xoay (bánh xe) sẽ hoàn toàn giống nhau nếu độc cơ không phải đi qua hộp số, nếu có thì người dùng cần tìm hiểu tỷ lệ giảm tốc của hộp số.

+ Mạch điện (Electronics Board) với vai trò là chuyền năng lượng đến led cũng như thực hiện giải mã tính hiệu dòng điện thông qua quang trở từ tính hiệu Analog sang tín hiệu số để tiện cho vi điều khiển xử lý.

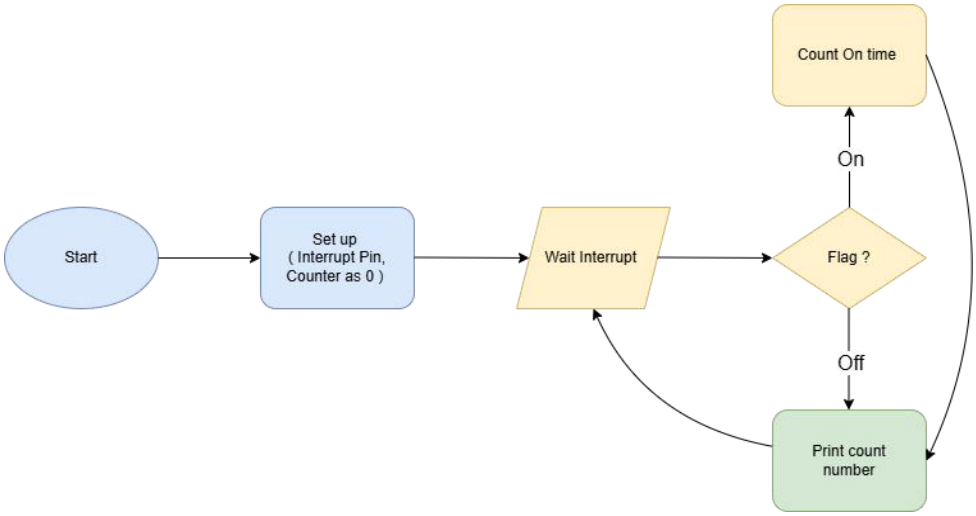


Hình 2.5. Sơ đồ hoạt động của Encoder

2.4.2. Các công thức tính và sơ đồ khối của động cơ

Một trong những số liệu quan trọng cho phép người dùng kiểm soát được góc quay của motor trong các ứng dụng yêu cầu động cơ phải chuyển động với độ chính xác cao đó tổng số xung cho một vòng quay. Từ tổng số xung cho một vòng quay thì người dùng có thế biết được trong tốc độ góc quay của động cơ là bao nhiêu (raido/s), cũng từ đây người dùng sẽ có thông tin về quảng đường mà động cơ có thế kéo được với một số lượng xung cố định.

Số xung cho 1 vòng quay của động cơ – độ phân giải tối thiểu (Pulse Per Revolution - PPR), để xác định được thì người dùng có thể đọc datasheet của nhà sản xuất về động cơ hoặc dùng vdk để đọc dữ liệu bằng việc xoay tay động cơ.



Hình 2.6. Sơ đồ khối thiết kế code đọc xung cho vi điều khiển

Góc quay nhỏ nhất của động cơ: thông số này tính toán góc quay nhỏ nhất của motor trong một ứng nhất định nào đó. Với việc động cơ quay trò, ta có.

$$N = \frac{360}{I}$$

N: số xung cho motor quay 360 độ (PPR)

I: độ quay nhỏ nhất (º , radian)

Trong trường hợp người dùng có một bộ giải mã động cơ với độ chính xác rất cao như cũng tại ra số lượng xung rất lớn cũng như cần được quay nhanh và chính xác.

Để giải quyết vấn đề trên thì nhà sản xuất có thể biến dạng dữ liệu vị trí quay của động cơ ở dạng analog, với các giá trị hiêu điện thế có thể hiện vị trí của động cơ từ đó người dùng có dùng giá trị hiệu thế để điều khiển động cơ đến vị trí chính xác.

$$M = \frac{360}{P}$$

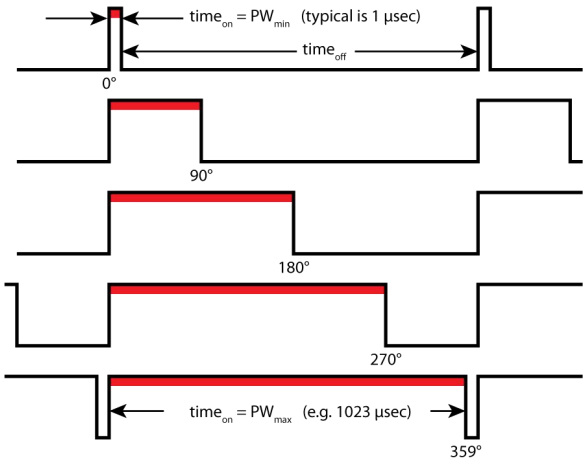
$$\log_2 M < X$$

M: số vị trí trên đĩa quay

P: khoảng cách giữa các vị trí (º , radian)

X: Số bit cần thiết để mã hóa dữ liệu theo hệ nhị phân

Để có được dữ liệu vị trí chính xác thì nhà sản xuất sẽ thực hiện mã hóa các vòng dữ liệu dựa trên các rảnh của đĩa che với số vòng có rảnh là số bit hay độ phân giải của encoder. Mỗi rảnh được đặt lệch một góc 90 so với rảnh đầu tiên.



Hình 2.7. Mã hóa dữ liệu bằng PWM

Những với việc hệ thống mã hóa vị trí rotor thì số lượng dây cần rất lớn để điều khiển được động cơ này, động cơ 8-bit mã hóa cần it nhất 12 chân để động cơ hoạt động (8 chân cho của vdk cho dữ liệu vị trí của rotor, 2 chân cung cấp nguồn cho bộ mã hóa, 2 chân để cơ động cơ hoạt động) để tiện cho việc giao tiếp với vi điều khiển

giao tiếp thì nhà sản xuất có thế đùng các bộ chuyển đổi dữ liệu sang dạng sóng răng cưa hay dạng xung pwm nhưng người dùng cần đọc datasheet.

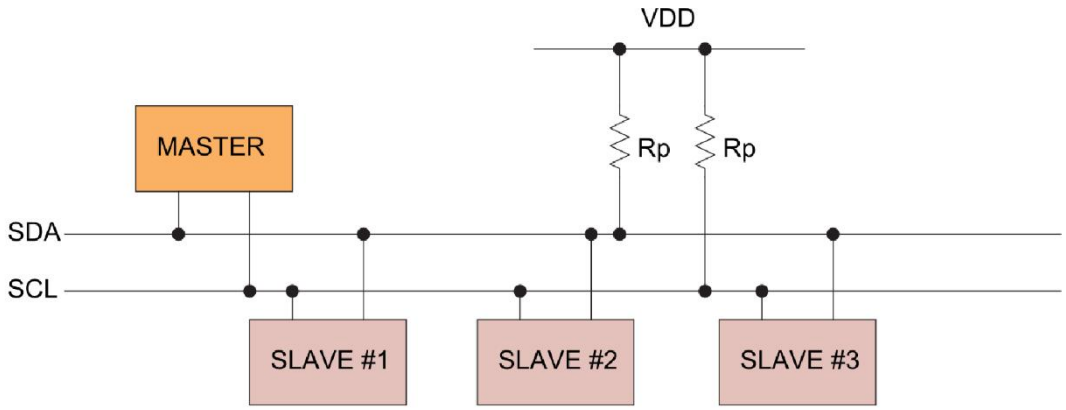
Các ánh sáng cần được cho qua một bộ lọc để chóng lóa để ngăn việc nhiễu từ đó có đảm bảo độ chính xác cho dữ liệu mã hóa vị trí của rotor. Đồng thời mã hóa dữ liệu theo 2 cách:

+ Mã hóa theo dữ liệu DAC với dữ liệu bit có thể biến đổi thành giá trị hiệu điện thế như một biến trở. Với ưu điểm có thế đưa thông tin vị trí nhanh nhưng số liệu dễ bị nhiễu nên cần vdk có thể xử lý cũng như phần cứng tương ứng cho việc xử lý dữ liệu analog và điều chỉnh cho phù hợp.

+ Mã hóa dữ liệu vị trí thông qua xung PWM với việc dữ liệu vị trí sẽ là biến thời gian cho độ rộng của xung pwm với việc mã hóa cho tính hiệu 10-bit thì người dùng chỉ cần một chu kỳ tính hiệu 1024us (2^{10}) với độ rộng xung tối đa là 1023us. Ưu điểm thiết kế và quá trình xử lý của vi điều khiển đơn giản chỉ cần một bộ đếm thời gian cho vi điều khiển cũng như hỗ trợ interupt, nhưng nhược điểm dữ liệu vị trí cần một khoản thời gian thì vdk mới có thể xác định được, độ phân giải càng cao thì thời gian chờ có thế lâu hơn.

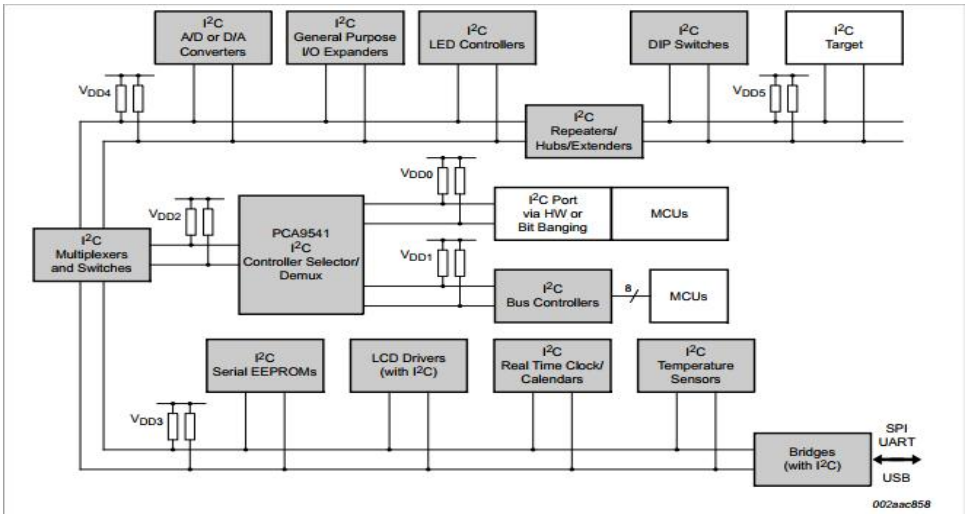
2.5. TỔNG QUAN VỀ GIAO THỨC I2C

2.5.1. Khái niệm về giao thức I2C



Hình 2.8. Sơ đồ kết nối chân cơ bản của giao thức I2C

Giao thức I2C (Inter-Integrated Circuit) một loại giao thức truyền dữ liệu đơn giản được phát triển bởi Philips Semiconductors vào năm 1982 với thiết kế đơn giản của nó chỉ dùng 2 dây cho việc giao tiếp và truyền dữ liệu theo hình thức Master-Slave thứ là tại một thời điểm chỉ có một thiết bị thực hiện truyền dữ liệu còn các thiết bị khác sẽ trong chế độ lắng nghe hoặc bỏ qua dữ liệu.

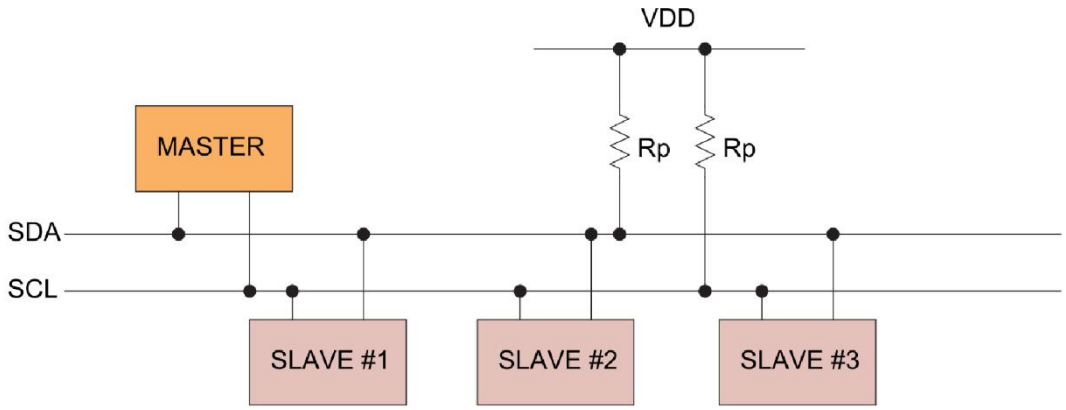


Hình 2.9. Sơ đồ kết nối I2C

+ Dây truyền dữ liệu chính của giao thức I2c chính là SDA, dữ liệu được truyền theo chuỗi 8 bit và khi nhận được dữ liệu thì Slave sẽ thực hiện tín hiệu ACK để cho master biết rằng là dữ liệu đã nhận đủ và thực hiện tiếp thông số dữ liệu chuỗi tiếp theo. Nhưng dữ liệu truyền lại không có thêm bất kỳ thông tin gì nên người dùng có thể thực hiện chỉnh sửa thêm các tính năng cho chuỗi dữ liệu này. Khi việc truyền dữ liệu bị lỗi do không nhận đủ 1 byte hay nếu người dùng thêm dữ liệu kiểm tra thì slave có thể gửi tín hiện NACK để cho master biết và thực hiện truyền lại dữ liệu.

+ Dây SCL có chức năng thực hiện truyền tần số xung cho các thiết bị trên đường truyền từ đó quyết định tốc độ truyền của dữ liệu giữa Master và Slave. Chuẩn I2C hỗ trợ các tốc độ như : 100Kbits/s (Standard-mode _ Sm), 400Kbits/s (Fast-mode _ Fm), 1Mbit/s (Fast-mode Plus _ Fm+), 3.4Mbits/s (High-speed mode _ Hs-mode).

2.5.2. Ứng dụng



Hình 2.10. Ứng dụng kết nối của I2C

Với việc I2C có thể đóng vai trò như một bộ mở rộng ngoại vi với khả năng dọc và ghi dữ liệu trên các chân bất kỳ và thiết kế đơn giản từ dó khiến cho ứng dụng sử dụng giao thức I2C được ứng dụng rất nhiều trong trong việc thiết kế cảm biến, mà mình hiển thị với tần số quét thấp cũng như các bộ chuyển đổi tín hiệu PWM, ADC, đồng thời bằng việc kế thêm bộ transitor nhận biết giá trị dòng điện cho các ngõ vào của IC nhà sản suất có thêm tính năng interupt từ đó cho phép người dùng áp dụng các tính năng có thời gian đọc dữ liệu không cố định.

2.6. TỔNG QUAN VỀ PHẦN CỨNG

2.6.1. NVDIA Jetson Nano Developer Kit tích hợp 16GB eMMC

Jetson Nano Developer Kit 16 Gb eMMC là board phát triển của nhãn hiệu NVDIA và Waveshare được nâng cấp từ Jetson Nano Developer Kit B01. Module được kết hợp để cung cấp đầy đủ các tính năng và tiêu chuẩn để giải quyết các dự án AI như xử lý giọng nói, phân loại hình ảnh, nhận diện, phát hiện đối tượng,…

So với kit B01, Jetson Nano Dev Kit đã hủy khe cắm thẻ nhớ TF trên board lõi Nano và thêm chip nhớ eMMC 16GB giúp ổn định khả năng đọc và viết, tuy nhiên vẫn mở rộng thêm khe thẻ nhớ TF để người dùng mở rộng bộ nhớ.

Cấu hình:

+ GPU: NVIDIA Maxwell™ architecture with 128 NVIDIA CUDA® cores and 0.5 TFLOPS (FP16)

+ CPU: Quad-core ARM® Cortex®-A57 MPCore processor

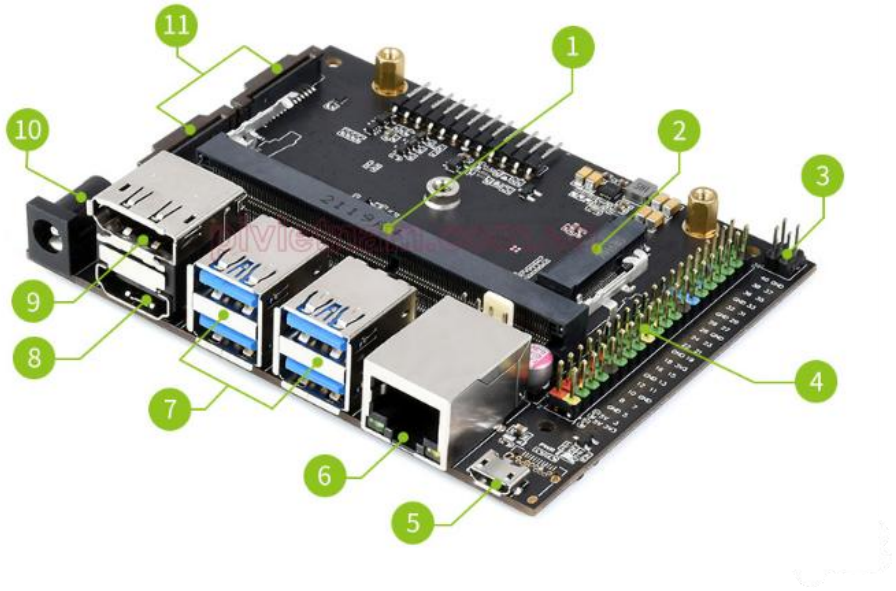
+ Bộ nhớ: 4 GB 64-bit LPDDR4 1600 MHz – 25.6 GB/s

+ Lưu trữ: 16 GB eMMC 5.1 Flash

+ Kết nối: Wifi yêu cầu chip ngoài, Ethernet Base-T 10/100/1000

+ IO: 3 x UART, 2 x SPI, 2 x I2S, 4 x I2C, nhiều đầu GPIO

Cấu tạo board:



Hình 2.11. Cấu tạo board Jetson Nano

1. Socket cắm Module

2. M.2 key connector

3. Chân PoE

4. 40 PIN GPIO

5. Cổng Micro USB

6. Cổng Gigabit Ethernet: 10/100/1000 Base-T

7. 4 cổng USB 3.0

8. Cổng HDMI

9. Cổng Display Port

10. Giắc cắm DC

11. Đầu nối camera 2x MIPI CSI

2.6.2. RPLiDAR A1M8

Cảm biến khoảng cách Lidar RPLIDAR A1M8 360° Laser Range Scanner được sản xuất bởi hãng SLAMTEC sử dụng cho các ứng dụng đo khoảng cách, phát hiện vật cản, lập bản đồ bằng tia Laser trong xe, robot tự hành, hệ thống chống trộm, ..., cảm biến có độ ổn định và độ chính xác cao.

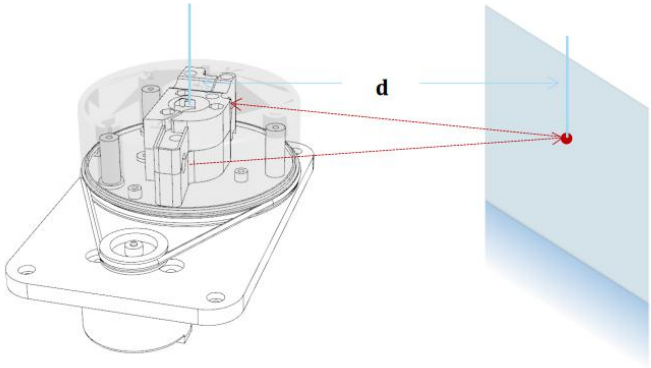
Cảm biến khoảng cách Lidar RPLIDAR A1M8 360° Laser Range Scanner sử dụng giao tiếp UART nên có thể dễ dàng giao tiếp với Vi điều khiển, Máy tính nhúng hoặc kết nối máy tính qua mạch chuyển USB-UART và phần mềm đi kèm, cảm biến có khả năng quét xa với khoảng cách 0.15~12m, tốc độ quay 5.5Hz và tần số lấy mẫu lên đến 8000 điểm trên 1 giây.

Hình 2.12. LiDAR A1M8

Thông số kỹ thuật:

- Phương pháp đo: Tam giác
- Điện áp hệ thống: 5VDC
- Dòng điện hệ thống: 100mA
- Tiêu thụ năng lượng: 0.5W
- Đầu ra: UART (mức điện áp 3.3)
- Tầm đo: 0.15~12m
- Tần số lấy mẫu: 8K
- Tốc độ quay: 5.5Hz
- Độ phân giải góc: ≤1°
- Phạm vi: 360°

Nguyên lý hoạt động: RPLiDAR dựa trên nguyên lý đo khoảng cách tam giác bằng laser và sử dụng phần cứng tốc độ cao để thu thập và xử lý tầm nhìn xung quanh. Hệ thống đo dữ liệu khoảng cách với tốc độ hơn 8000 lần mỗi giây và với đầu ra có độ phân giải tốc độ cao. RPLiDAR phát ra tín hiệu laser hồng ngoại đã điều chế và tín hiệu laser sau đó được phản hồi bởi vật cản được phát hiện. Tín hiệu trở lại được lấy mẫu bằng thị giác hệ thống thu nhận trong RPLIDAR A1 và DSP được nhúng trong RPLIDAR A1 bắt đầu xử lý dữ liệu mẫu và giá trị khoảng cách đầu ra và giá trị góc giữa vật thể và RPLIDAR A1 thông qua giao thức truyền thông. Hệ thống quét tốc độ cao được gắn với bộ xoay 360 độ giúp hệ thống ghi lại được toàn bộ môi trường xung quanh theo thời gian thực trong phạm vi cho phép.



Hình 2.13. Sơ đồ nguyên lý hoạt động RPLiDAR A1

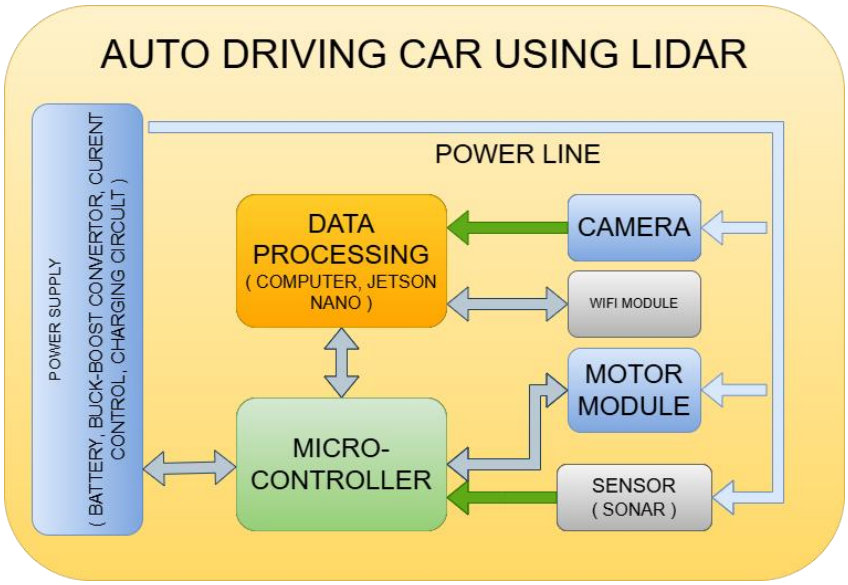
CHƯƠNG 3: THIẾT KẾ XE TỰ HÀNH

3.1. GIỚI THIỆU VỀ XE TỰ HÀNH

Xe tự hành một trong những lĩnh vực đang phát triển với tốc độ chóng mặt nhưng hiện nay và một trong những yếu tố giúp cho bộ xử lý bên trong xe tự hành có thể thu thập dữ liệu cho việc học hỏi của model A.I bên trong xe.

Nhóm sẽ tập trung vào thiết kế mô hình có khả năng:

+ Vẻ bản đồ 2D và phát họa bản đồ 3D.
+ Tự di chuyển đến khi bao gọn không gian kép kín.
+ Tự động nén vật cản.
+ Có thể tự điều chỉnh tốc độ của motor.
+ Dùng camera nhận diện vật cản, thực hiện né vật cản.



Hình 3.1. Sơ đồ khối cấu tạo của xe

Xe sẽ được thiết kế theo các khối module để khi thực hiện thay đổi bất kỳ bộ phận nào thì nhóm có chỉ cần cập nhật lại phần mềm cho hệ thống là xe có thể hoạt động bình thường. Xe được cấu tạo từ 4 khối chính:

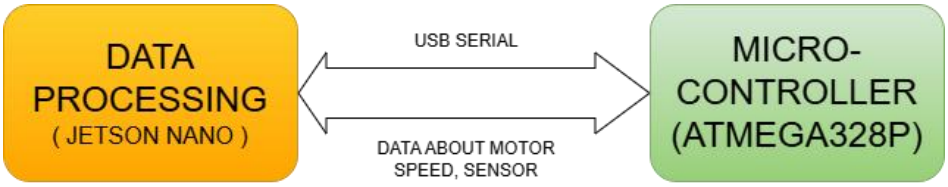
+ Khối nguồn: được thiết kế bằng pin litium với mạch buck-boost giúp cố định dòng điện ở một giá trị hiệu điện thế ổn định trong suốt quá trình hoạt động. Cung cấp 2 giá trị điện áp riêng biệt +5V cho vi điều khiển và khối xử lý (máy tính nhúng raserry hoặc jetson nano) đến 6-12V cho động cơ DC hoạt động.

+ Khối vi điều khiển (vi điều khiển ATMEGA328P) có chức năng thực hiện thu thập số liệu, từ khối xử lý vận tốc của động cơ, thực hiện đo đạt giá trị dung lượng pin còn sót lại và thực hiện điều chỉnh cần thiết để đảm bảo tốc độ di chuyển của xe không thay đổi trong suất quá trình hoạt động.

+ Khối xử lý dữ liệu: thu thập dữ liệu từ cảm biến lidar với hình ảnh từ camera để từ đó có thể vẻ một bản đồ 2D, cũng như thực hiện phát họa bản đồ 3D của những khung vực đã đi qua, xác định vị trí của mô hình xe trong bản đồ, tính toán ra được tốc độ và hướng đi của động cơ.

+ Module: động cơ xe lúc này được đóng gói tích hợp với diver điều khiển chiều xoay của động cơ, được tích hợp với IC mở rộng GPIO và PWM giao tiếp I2C từ đó cho phép người dùng có thể điều khiển thông qua BUS dữ liệu.

3.2. SƠ ĐỒ KHỐI ĐIỀU KHIỂN ĐỘNG CƠ



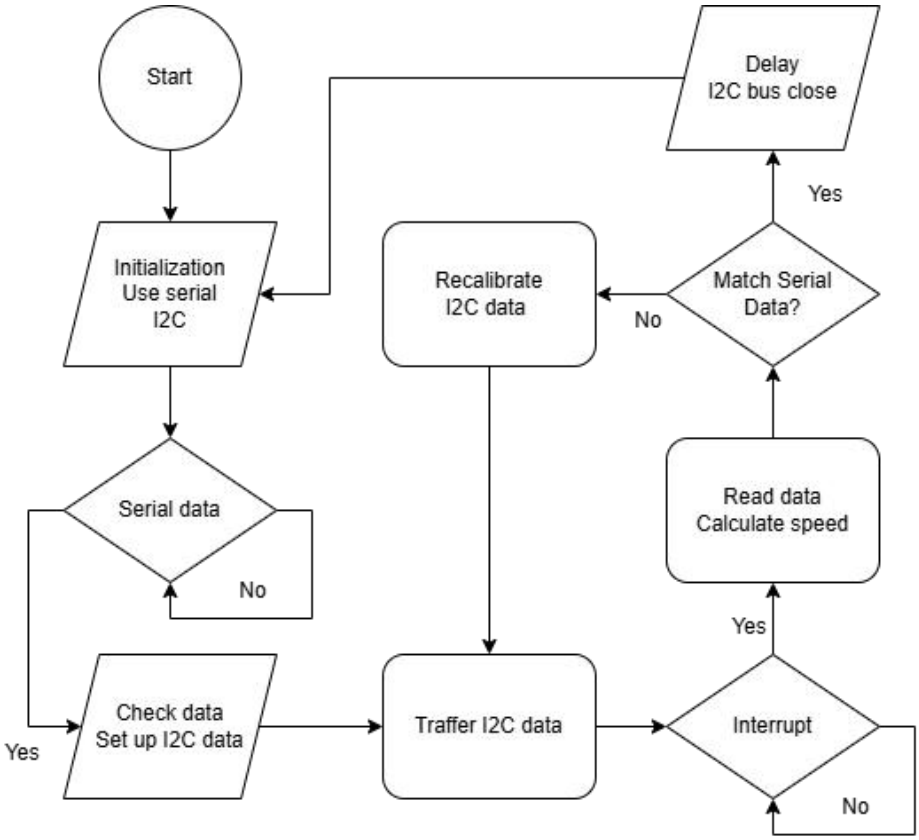
Hình 3.2. Sơ đồ truyền nhận dữ liệu khối xử lý và khối điều khiển

Để tiện cho việc phát triển lập trình phần cứng và thực hiện nâng cấp sửa chữa thì nhóm em đã thực hiện tách bộ phận xử lý thành 2 khối:

+ Khối xử lý: tập chung vào việc xử lý các tác vụ phức tạp như render hình ảnh điểm tọa độ vẽ bản đồ, tính toán tọa độ của xe tại một thời điểm, tính toán tốc độ vị trí hướng đi của xe để né vật cản.

+ Khối điều khiển: thực hiện điều khiển động cơ, điều chỉnh chiều xoay của động cơ theo hướng xoay, thực hiện thu thập dữ liệu của động cơ để xác định tốc độ, thu thập thêm bất kỳ dữ liệu cảm biến đơn giản như siêu âm, cảm biến hồng ngại…

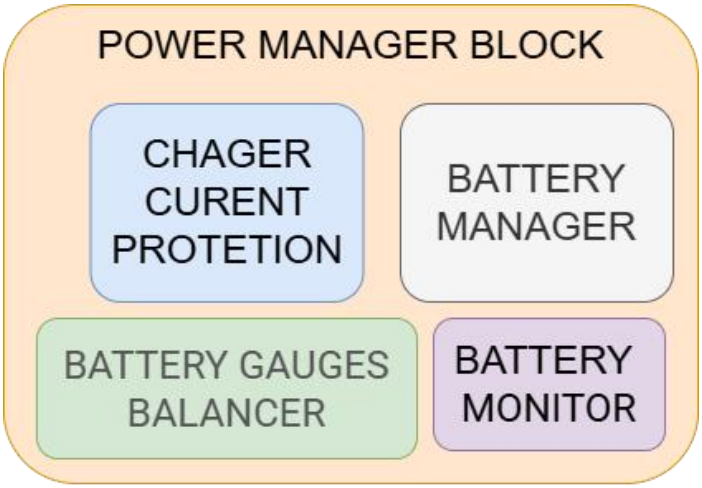
+ Khối điều khiển cũng đóng vai trờ như một bộ tiền xử lý giúp tính toán xử lý dữ liệu thô thông số tốc độ của motor và thực hiện các điều chỉnh để cho động cơ chạy đúng theo như yêu cầu của khối xử lý.



Hình 3.3. Sơ đồ khối truyền và xử lý dữ liệu motor của khối điều khiển

3.3. THIẾT KẾ PHẦN CỨNG

3.3.1. Nguồn



Hình 3.4. Sơ đồ khối các tính năng bên trong khối pin

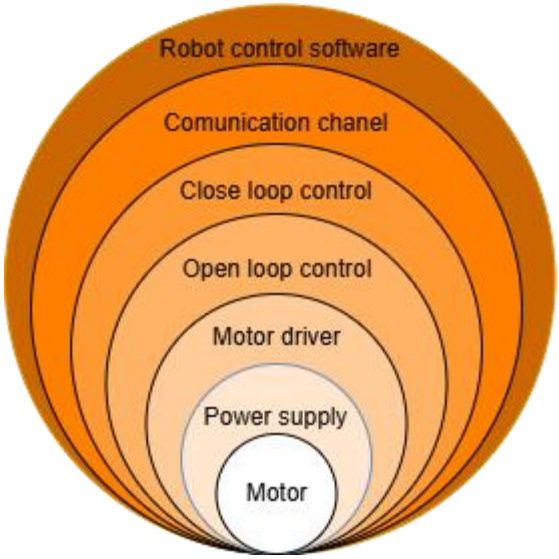
Khối pin sẽ bao gồm 4 chức năng chính:

+ Chức năng quản lý pin: dòng IC với với khả năng thực hiện xử lý dữ liệu của pin (điện thế, cường độ dòng điện) từ đó sẽ kích hoạt các tính năng cần thiết như bảo vệ pin khi tải hoặc điện áp đầu vào quán lớn thì mạch sẽ kích hoạt tính năng bảo vệ pin thực hiện ngắt nguồn pin.

+ Thực hiện giám sát pin: sử dụng IC hay mạch có chức năng thực thu thập dữ liệu dung dung lượng điện áp ra của pin chuẩn đổi những dữ liệu này sang dạng số rồi thực hiện truyền đến vi điều khiển thông qua các giao thức như I2C, SPI, UART, … cũng như cho phép vdk thực hiện điều chỉnh khi cần.

+ Cân bằng pin: đây là mạch thụ động chủ yếu cân bằng pin trong suất quá trình sử dụng sạc vì pin sẽ bị chai không đồng đều dẫn đến hiện tượng ngắn mạch dẫn đến pin không thế dùng được nữa nếu như không được cân bằng đúng.

3.3.2. Khối điều khiển động cơ



Hình 3.5. Sơ đồ các lớp điều khiển cho động cơ

Việc điều khiển động cơ có thể được chia thành các chức năng chính:

+ Mạch quản lý nguồn động cơ: có chức năng cung cấp một điện áp ổn định với động cơ hoạt động lâu dài.

+ Motor driver: với việc động cơ thường yêu cầu công suất tải rất lớn khiến cho vi điều khiển khi tiếp xúc với động cơ thường bị dòng đánh ngược do quá trình quay của động cơ làm cháy mạch, IC driver dùng để tạo một lớp bảo vệ đồng thời dùng tín hiệu từ vi điều khiển để thực hiện thay đổi chiều quay và tốc độ cho động cơ.

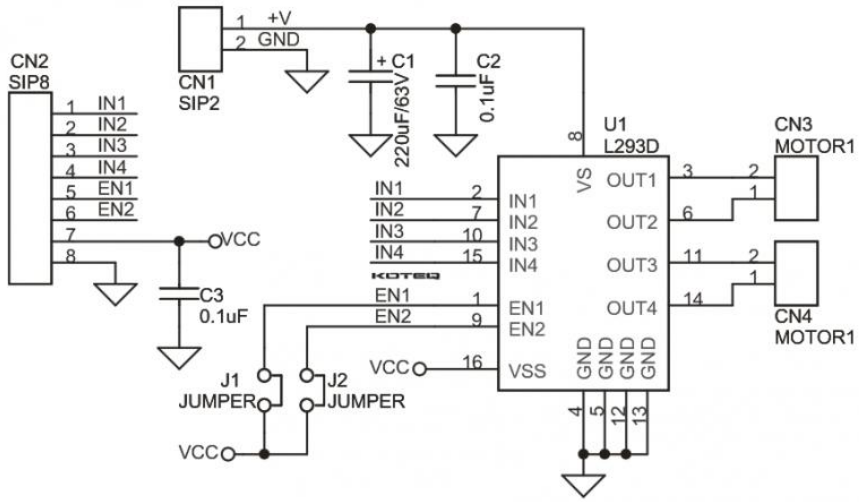
+ Kiểm soát động cơ vòng mở: bằng cách sử dụng động cơ với một encoder một bộ giải mã vị trí hay tốc độ quay của động cơ.

+ Kiểm soát động cơ vòng đóng: vi điều khiển có thể dùng số liệu tốc độ của động cơ (số xung trong 1s) từ dó có thế thay đổi thông tin tốc độ của động cơ nhanh hơn hay chậm hơn theo nhu cầu của người dùng.

+ Thiết lập một kênh giao tiếp: Cho phép vi điều khiển có thế điều khiển động cơ như một module với chuẩn giao tiếp như SPI hoặc I2C.

+ Cổng giao tiếp cho vi điều khiển: thực hiện viết một chương trình cho phép người dùng hay các chương trình khác thực hiện giao tiếp với vi điều khiển cho phép vi điều khiển thực hiện thay đổi tốc độ chiều quay của từng bánh xe.

3.3.3. Cơ chế điều khiển



Hình 3.6. Schematic của motor driver L293D

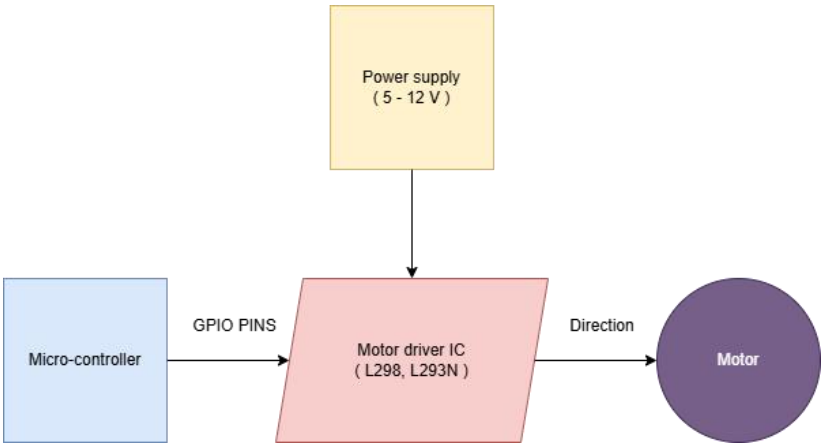
IC L293D được thiết kế như một bộ buffer cho các tín hiệu của vi điều khiển bằng việc sửa dụng CMOS từ đó cho phép IC có thể cung cấp nguồn 0.6A ổn định cho bất kỳ động cơ nào với giá trị hiệu điện thế từ 4.5V đến 36V. Với thiết kế CMOS có cực máng mở cho phép IC không chỉ cung nguồn cũng như thực hiện điều hướng dòng ra của động cơ DC.

Chức năng của các chân IC:

+ Chân IN1-IN4 có tính năng là ngõ vào cho phép người dùng đều khiển động cơ thông qua mức logic, với mức giá trị mức 0 thì ngõ ra tương ứng sẽ có tự động thành một công tắc nối đất từ đó giúp cho điện áp ngõ ra là 0V.

+ Chân EN1-EN2 cho phép driver thực hiện nhận dữ liệu ngõ vào theo cặp EN1 điều khiển chân IN1 – IN2 và EN2 điều khiển IN3 – IN4, với mức logic 0 thì ngõ nào chân IN1 và IN2 sẽ không còn có khả năng điều khiển được động cơ nứa lúc này động cơ có chiều quay giống như chiều quay lúc chân EN1 có mức logic 1 trước đó.

a. Điều khiển trực tiếp



Hình 3.7. Sơ đồ khối điều khiển động cơ trực tiếp

Để thực hiện điều khiển trực tiếp người dùng phải dùng dùng jumper để nối chân EN1 với EN2 với một nguồn 5V. Từ đó giúp cho IC hoạt động bằng cách nhận ngõ vào thông qua chân IN1-IN4.

Lúc này vi điều khiển có thế dùng các cặp IN1-IN2 và IN3-IN4 để thực hiện điều khiển 2 động cơ, thông qua việc nối chân dương (+) của động cơ vào cổng OUT1 hoặc OUT3, chân âm (-) được nối với ngõ ra OUT3 hoặc OUT4 với 4 chế độ khác nhau thông qua 2 bit điều khiển:

+ 0x0: Động cơ ngắt hoàn toàn do không có bất kỳ gí trị điện áp nào vào động cơ.

+ 0x1: Động cơ quay ngược chiều kim đồng hồ.

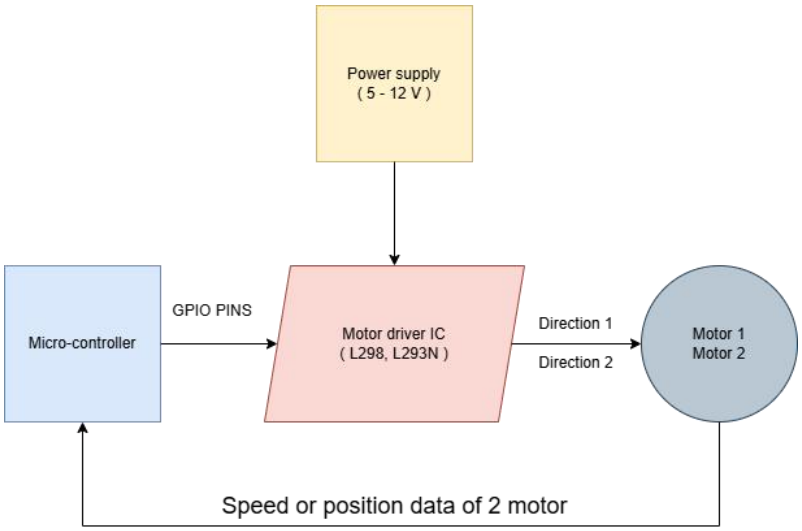
+ 0x2: Động cơ quay theo chiều kim đồng hồ.

+ 0x3: Lúc này với cùng một giá trị điện áp khiến cho động cơ và phần lõi mất đi hết từ tính từ đay giúp cho động cơ dừng hòa toàn.

Ưu điểm:

+ Thiết kế đơn giản.
+ Có thể thay đổi chế độ quay với tốc độ cao.
+ Một driver có thể điều khiển 2 động cơ

Nhược điểm: Người dùng không thể biết được tốc độ động cơ, nên phụ thuộc nhiều chế độ dừng khẩn cấp rất nhiều, làm giảm độ bền của động cơ.



Hình 3.8. Sơ đồ khối cho điều khiển 2 động cơ trực tiếp

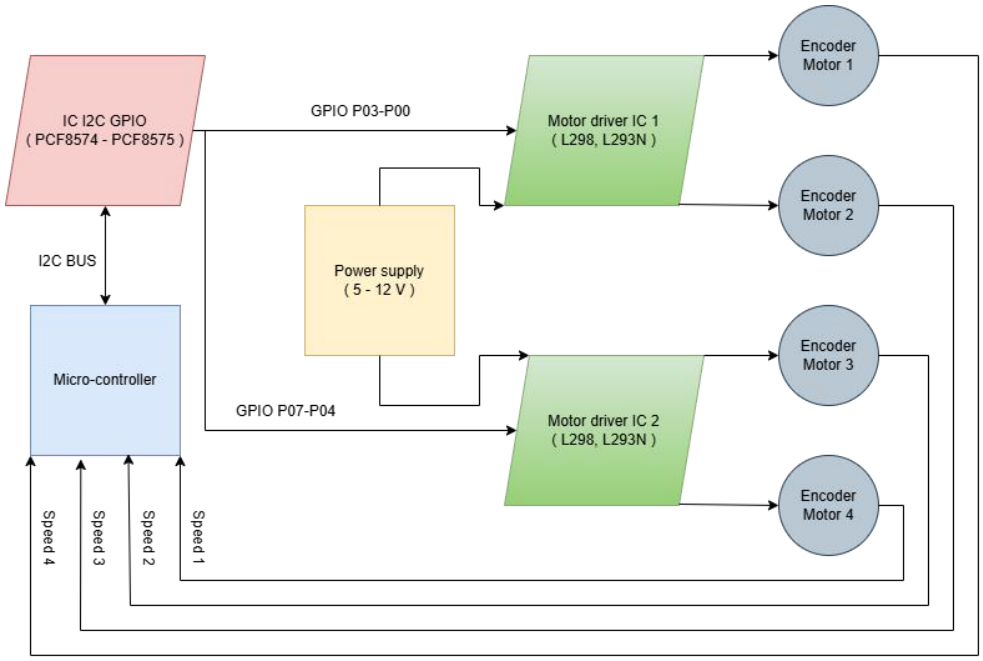
Với việc thêm chân mã hóa kênh A và kênh B người dúng có thể xác định được chiều quay cả động cơ thông qua việc xét xung interrupt tại thời điểm t. Interrupt bị kích hoạt thông qua xung đi lên của tín hiệu A, từ đó mỗi lần vi điều khiển phát hiện interrupt thì sẽ kiểm tra coi xung tín hiệu B đang ở mức cao (quay theo chiều kim đồng hồ) hay mức thấp (quay ngược chiều kim đồng hồ).

Ưu điểm: Người dùng điều chỉnh được chiều quay trong quá trình chạy.

Nhược điểm:

+ Tuy có thể điều chỉnh được tốc độ quay của rotor bằng pwm nhưng yêu cầu vi điều khiển phải có thiết kế thêm timer interrupt tăng độ phức tạp cho phần mềm.
+ Thiết này xử dụng nhiều chân của vi điều khiển, một motor dùng đến 4 chân của vdk.

b. Điều khiển gián tiếp thông qua PCF8574



Hình 3.9. Sơ đồ khối điều khiển động cơ bằng PCF8574 và I2C

Bằng việc xử dụng IC pcf8574 thì người dùng có thế điều khiển chân IN của ic l2893 thông qua giao thức I2C.

Ưu điểm:

+ Giảm thiểu được số lượng chân IO cần dùng xuống 1 nữa.

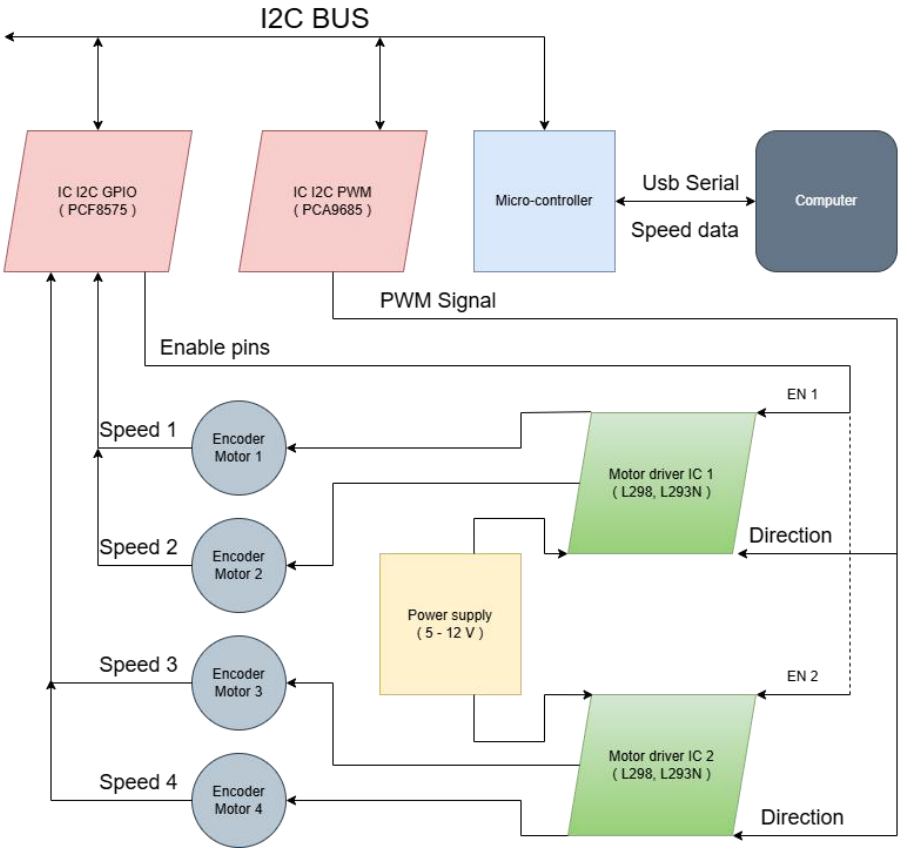
+ Giúp tinh gọn mà module khóa động cơ, cho phép điều khiển qua I2C.

Nhược điểm:

+ Làm tăng độ phức tạp của thiết kế với nhiều IC.

+ Làm cho việc điều khiển bằng PWM khó khăn năng hơn do phải tín toán chung với thời gian truyền I2C.

c. Điều khiển gián tiếp thông qua PCA9685 và thu thập dữ liệu bằng PCF8575



Hình 3.10. Sơ đồ khối động cơ encoder bằng IC PWM PCA9685 và PCF8574

Thông qua IC PCA9685 người dùng có thể điều khiển động cơ với mức công suất chạy tùy ý thông qua việc bật tắc động cơ nhanh trong 1 giây.

Ưu điểm:

+ Tất cả động cơ mà xe cần có thể được điều khiển thông qua I2C.
+ Giảm thiểu số chân cần giao tiếp xuống 2 chân.

Nhược điểm:

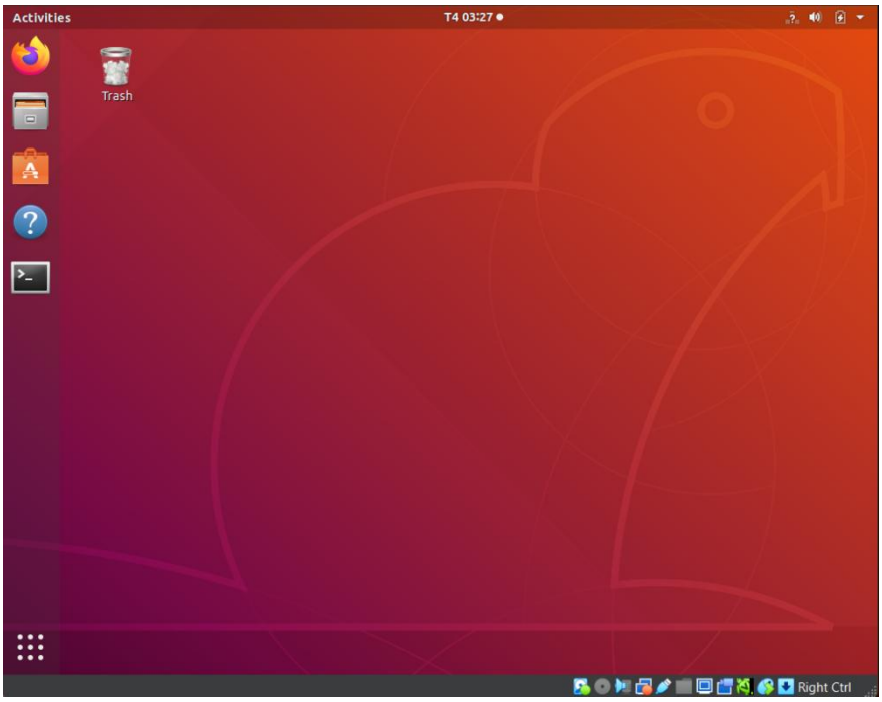
+ Thiết kế rất phức tạp phần cứng lẫn phần mềm.
+ IC PCF8575 không hỗ trợ đọc mức logic của ngõ nào.

CHƯƠNG 4: THUẬT TOÁN QUÉT 2D

4.1. XÂY DỰNG MÔI TRƯỜNG

LiDAR A1M8 được sử dụng trên nền tảng ROS. ROS (Robot Operating System, hay hệ điều hành Robot) là bộ phần mềm mã nguồn mở. Mặc dù ROS không phải là một hệ điều hành (OS) mà là một tập hợp các khung phần mềm để phát triển phần mềm robot, nó cung cấp các dịch vụ được thiết kế cho một cụm máy tính không đồng nhất như trừu tượng hóa phần cứng, điều khiển thiết bị cấp thấp, triển khai chức năng thường được sử dụng, truyền tin nhắn. Các quy trình dựa trên ROS đang hoạt động có thể được biểu diễn thành biểu đồ. Dù cho độ thích ứng cao cũng như độ trễ thấp trong việc đi khiển robot, ROS không phải là hệ điều hành thời gian thực (RTOS).

Để thiết lập môi trưởng sử dụng ROS, có thể thiết lập ROS Windows hay Ubuntu. Nhóm thực hiện sử dụng Ubuntu 18.04 LTS, khởi tạo máy ảo và cài đặt ROS.



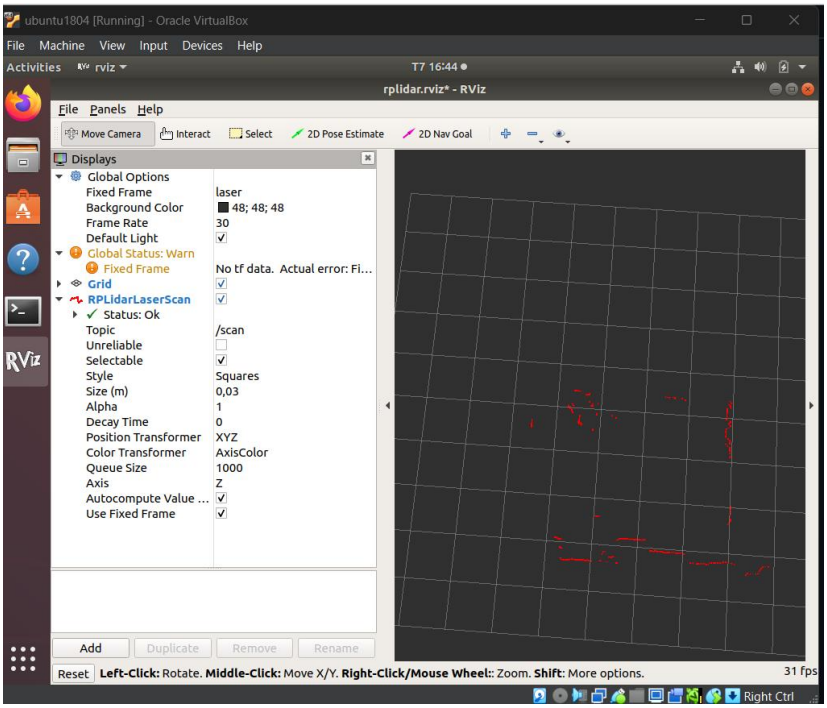
Hình 4.1. Giao diện Ubuntu phiên bản 18.04

Tùy vào phiên bản sử dụng để cài đặt ROS thông qua Terminal, nhóm đã sử dụng ROS Melodic, sử dụng cú pháp:

sudo apt install ros-melodic-desktop-full

4.2. KẾT NỐI LIDAR VÀ CÀI ĐẶT THƯ VIỆN

Sử dụng dây nối được nhà sản xuất cung cấp sẵn để kết nối với máy ảo thông qua cổng USB. Tạo thư mục làm việc và sử dụng các thư viện có sẵn của RP LiDAR để kiểm tra kết nối giữa LiDAR và máy ảo. Kết quả thu được từ laser được hiển thị dưới dạng điểm trên rviz. Rviz (ROS Visualization) là phần mềm trực quan hóa 3D giúp theo dõi trạng thái robot từ các dữ liệu cảm biến (hình ảnh, pointcloud,…) hoặc từ các tọa độ khác nhau. Rviz có thể được sử dụng cho các cảm biến thật và robot mô phỏng.



Hình 4.2. Kiểm tra kết nối LiDAR, hiển thị trên rviz

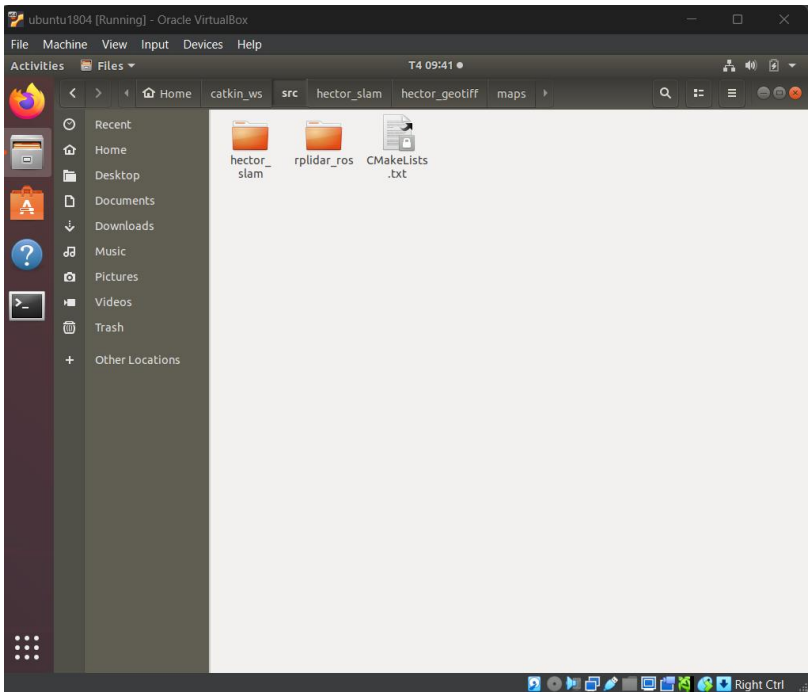
Để có được bản đồ hiển thị trên rviz thì không thể chỉ dựa vào dữ liệu pointcloud như trên hình 4.2, con người cần một cái nhìn trực quan hơn để khảo sát bản đồ, vì thế thuật toán Hector SLAM ra đời. Như trên lý thuyết của LiDAR, khi laser từ cảm biến nhận được tín hiệu vật cản và truyền trở về, tại đấy đánh dấu các pointcloud chính là viền của vật cản. Giả sử xét trên ma trận pixel, dữ liệu thu về được chia thành 3 loại giá trị:

+ 1 là giá trị khi laser gặp điểm dừng và dội ngược về cảm biến, hay nói cách khác chính là vật cản từ môi trường xung quanh.

+ 0 là giá trị khi laser vượt thẳng qua mà không gặp trở ngại nào nhưng vẫn nằm trong tầm quét của LiDAR.

+ -1 là giá trị chưa được khám phá, là môi trường khi cảm biến không hoặc chưa thể quét đến.

Thuật toán Hector SLAM phụ thuộc rất nhiều vào việc quét khung liên tiếp để xây dựng bản đồ. Tuy thuật toán không phụ thuộc vào các thông số đo lường nhưng rất cần thông tin truyền về từ laser, vậy nên Hector phù hợp với các dạng robot di chuyển bước tiến hoặc xoay. Nó thực hiện quét tại hai mốc thời gian liên tiếp và sử dụng các pointcloud để so sánh. Vị trí của cảm biến được cập nhật dựa trên vị trí trước đó và sự thay đổi của vị trí. Việc di chuyển LiDAR sang vị trí mới giúp cảm biến ghi lại và thực hiện xây dựng bản đồ tiếp theo. Cảm biến không xác nhận giá trị điểm ngay từ một lần quét duy nhất vì có nguy cơ chứa nhiễu, mỗi điểm còn chứa giá trị nhị phân khác. Giá trị này là hàm số của số lần điểm được quét qua là có phải vật cản hay không và xác suất dự đoán là chính xác. Sau nhiều lần quét để cảm biến xác định về độ chính xác giá trị tại điểm ảnh, khi đó mới hiển thị tại vị trí đó có phải vật cản hay không.

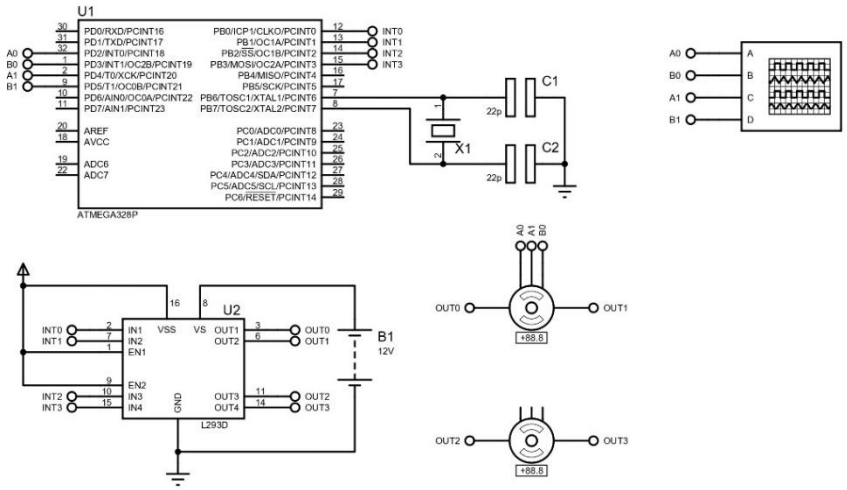


Hình 4.3. Thêm thư viện của Hector SLAM cùng thư mục làm việc với Lidar

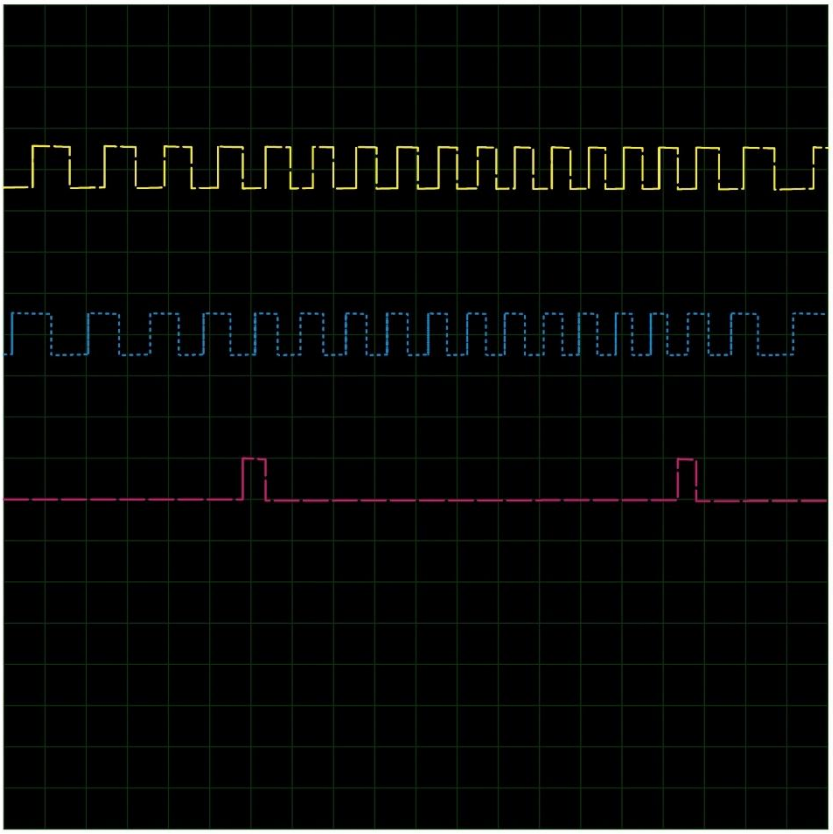
CHƯƠNG 5: KIỂM TRA VÀ ĐÁNH GIÁ

5.1. KẾT QUẢ ĐẠT ĐƯỢC

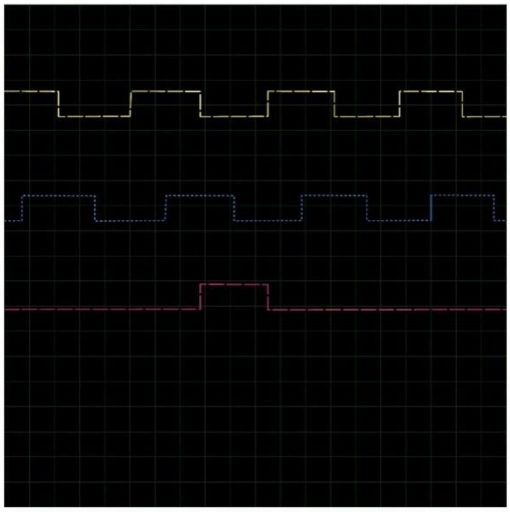
5.1.1. Mô hình điều khiển trực tiếp

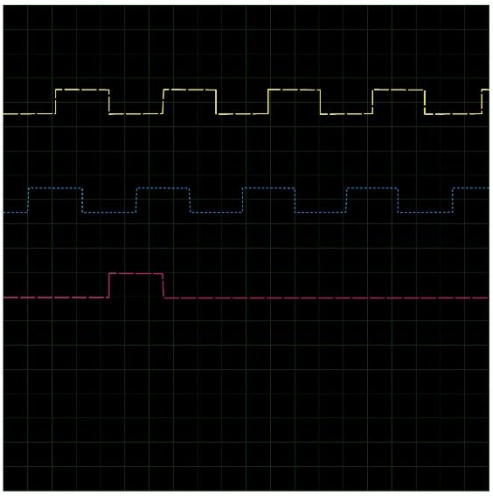


Hình 5.1. Sơ đồ mô phỏng của motor encoder

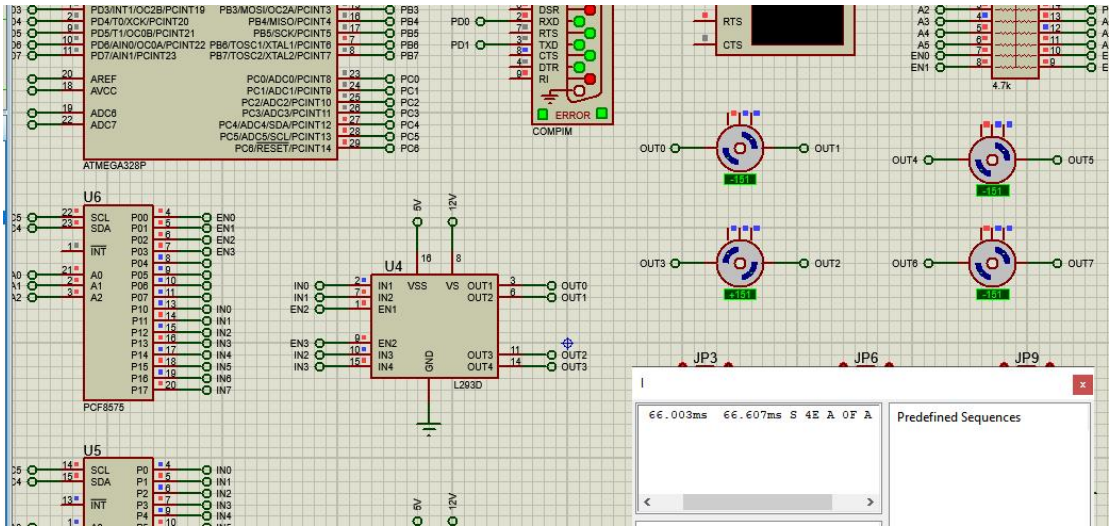


Hình 5.2. Sóng xung thu được của động cơ encoder



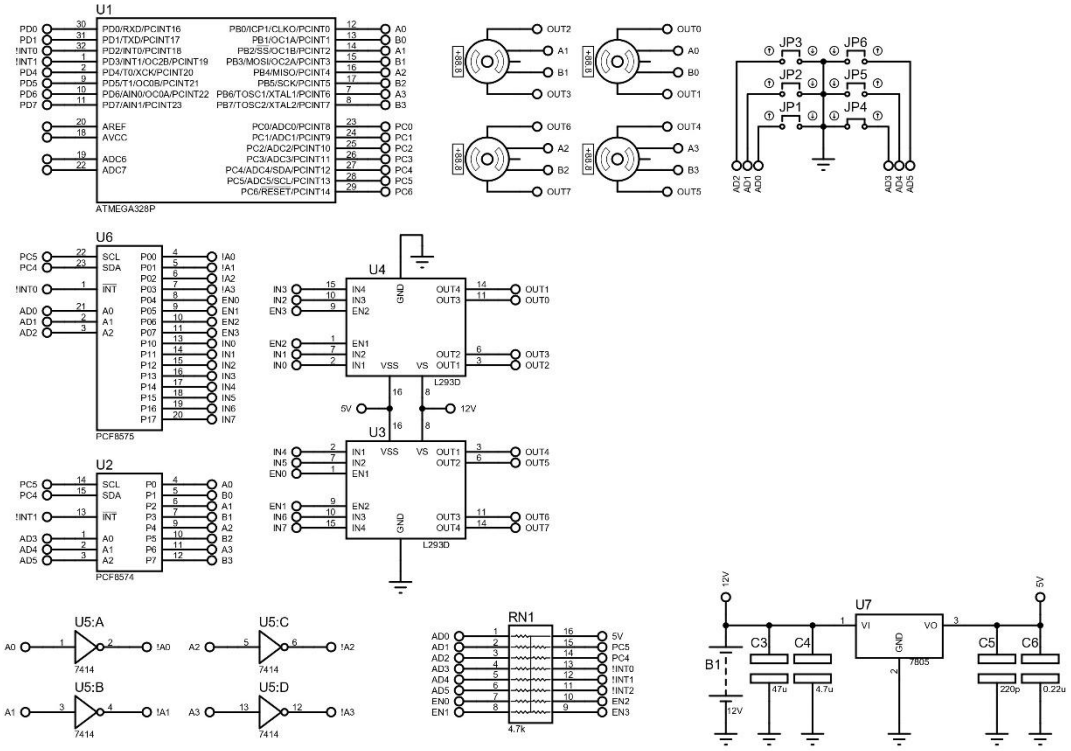


Hình 5.3 và 5.4. Xung tín hiệu của động cơ khi quay thuận/ ngược chiều kim đồng hồ

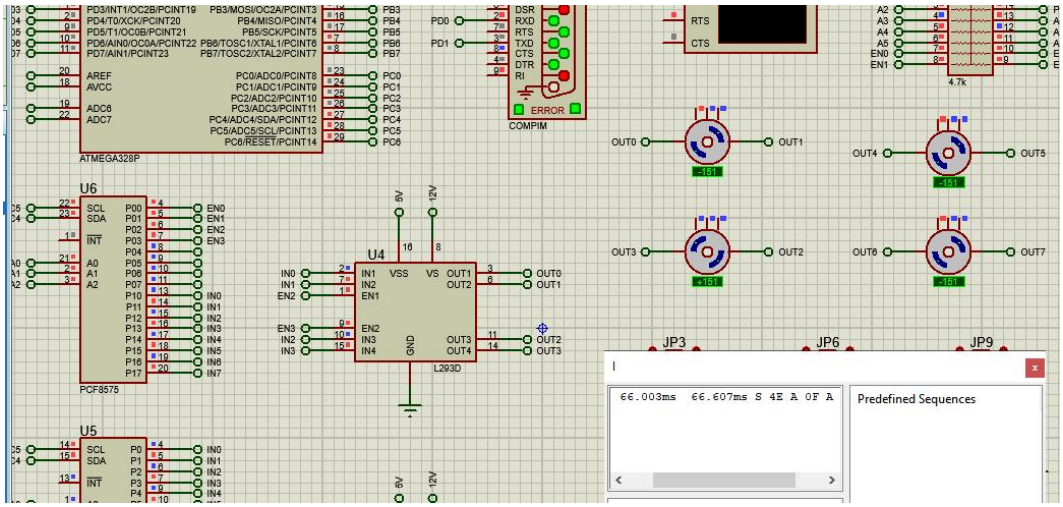


Hình 5.5. Tốc độ mô phỏng của motor

5.1.2. Mô hình điều khiển qua I2C

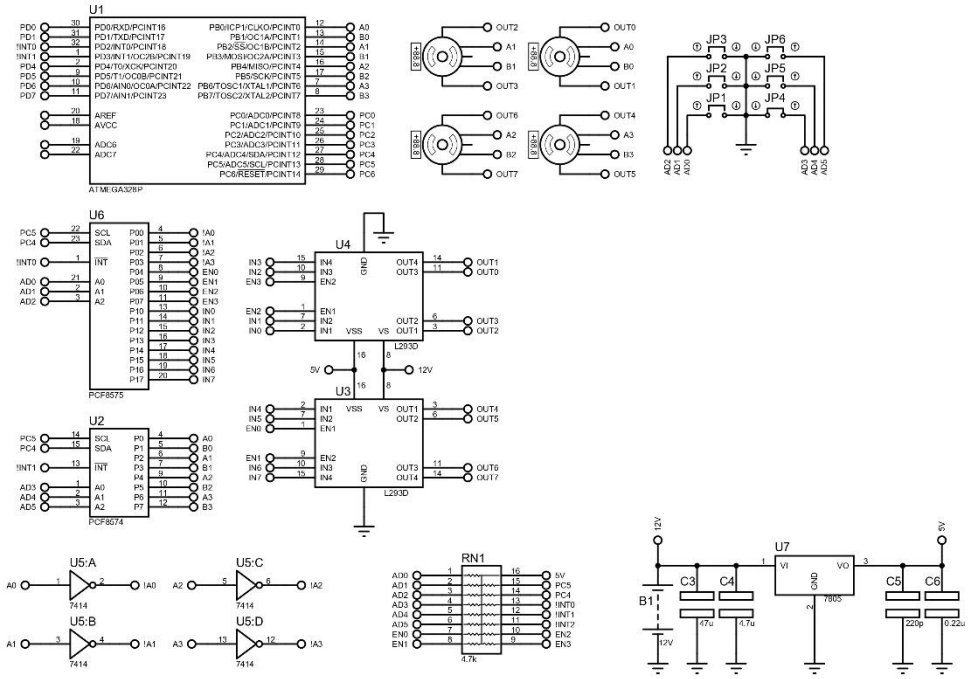


Hình 5.6. Schematic thực hiện mô phỏng module động cơ I2C

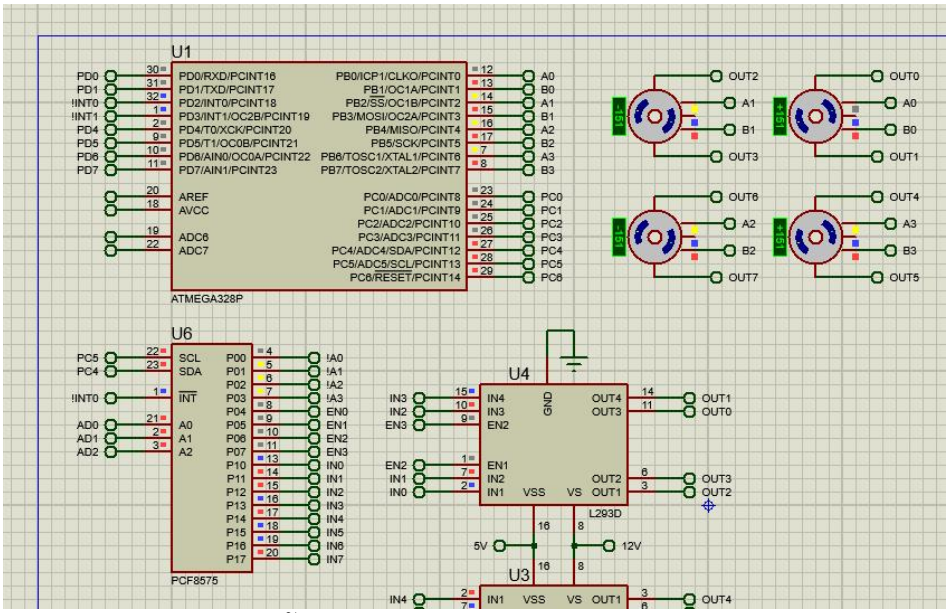


Hình 5.7. Ảnh điều khiển động cơ chiều bằng I2C

5.1.3. Mô hình điều khiển PID qua I2C

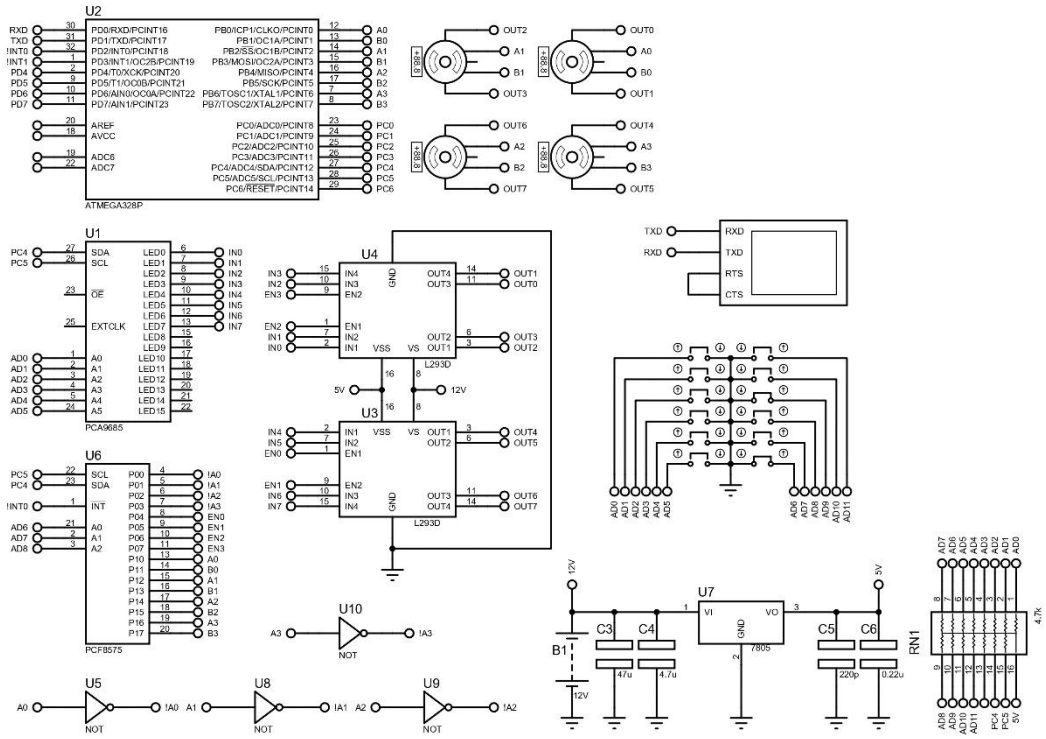


Hình 5.8. Schematic đọc dữ liệu xung encoder thông qua PCF8575



Hình 5.9. Lỗi quá trình đọc ngõ ra encoder motor

Với việc sóng tín hiệu ngõ của encoder dùng cạnh lên của xung tính hiệu để ghi dữ liệu từ đó khi tín hiệu đi qua IC cổng đảo làm delay tín hiệu ban đầu dẫn đến việc đọc ghi tín hiệu của encoder bị khó khăn hơn.



Hình 5.10. Schematic điều khiển động cơ bằng PCA9685 và đọc ngõ vào PCF8575

5.1.2. Xây dựng bản đồ bằng LiDAR

Sau khi khởi động trình chạy của hector_slam_launch, kết quả xây dựng bản đồ thu được như sau.

```
[ INFO] [1735055315.702132825]: Geotiff node started
HectorSM map lvl 1: cellLength: 0.1 res x:1024 res y: 1024
[ INFO] [1735055315.834634507]: HectorSM p_base_frame_: base_link
[ INFO] [1735055315.836233149]: HectorSM p_map_frame_: map
[ INFO] [1735055315.836287150]: HectorSM p_odom_frame_: base_link
[ INFO] [1735055315.836301172]: HectorSM p_scan_topic_: scan
[ INFO] [1735055315.836319201]: HectorSM p_use_tf_scan_transformation_: true
[ INFO] [1735055315.836391392]: HectorSM p_pub_map_odom_transform_: true
[ INFO] [1735055315.836520741]: HectorSM p_scan_subscriber_queue_size_: 5
[ INFO] [1735055315.836618476]: HectorSM p_map_pub_period_: 2.000000
[ INFO] [1735055315.836700304]: HectorSM p_update_factor_free_: 0.400000
[ INFO] [1735055315.836763067]: HectorSM p_update_factor_occupied_: 0.900000
[ INFO] [1735055315.836859663]: HectorSM p_map_update_distance_threshold_: 0.400000
[ INFO] [1735055315.837136645]: HectorSM p_map_update_angle_threshold_: 0.060000
[ INFO] [1735055315.837248642]: HectorSM p_laser_z_min_value_: -1.000000
[ INFO] [1735055315.837382641]: HectorSM p_laser_z_max_value_: 1.000000
```

Hình 5.11. Một số tham số giá trị từ Hector SLAM

Trong đó:

Pub_map_odom_transform: Xác định có nên xuất giá trị bản đồ -> khoảng cách từ hệ thống hay không.

Map_pub_period: Khoảng thời gian xuất bản đồ (s).

Update_factor_free: Công cụ sửa đổi bản đồ để cập nhật các ô trống trong khoảng [0.0, 1.0]. Giá trị 0.5 nghĩa là không đổi.

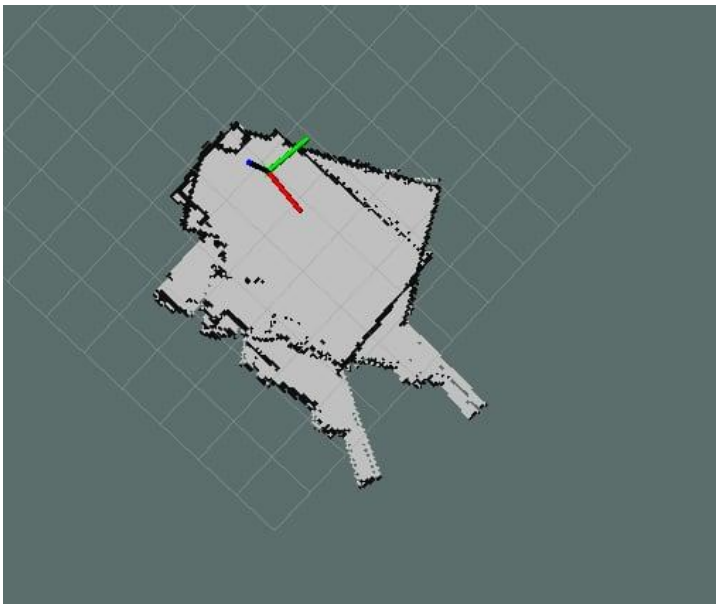
Update_factor_occupied: Công cụ sửa đổi bản đồ để cập nhật giá trị trong khoảng [0.0, 1.0]. Giá trị 0.5 nghĩa là không đổi.

Map_update_distance_threshold: Ngưỡng thực hiện cập nhật bản đồ theo mét (m). Hệ thống phải đạt được ngưỡng thay đổi trên hoặc gặp thay đổi góc được chỉ định bởi tham số map_update_angle_threshold để cập nhật lại bản đồ.

Map_update_angle_threshold: Ngưỡng thực hiện cập nhật bản đồ theo radian (rad). Hệ thống phải sử dụng ngưỡng thay đổi góc hoặc được chỉ định ở tham số map_update_ distance_threshold để cập nhật lại bản đồ.

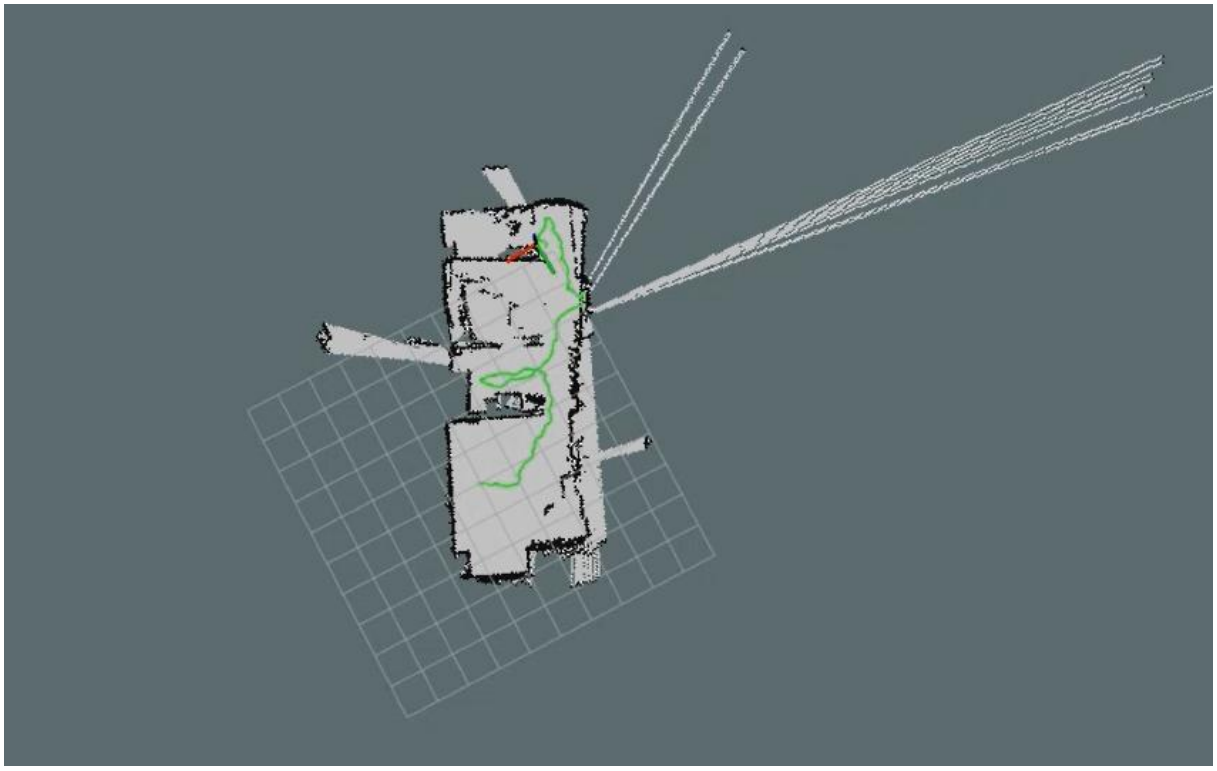
Laser_z_min_value: Chiều cao tối thiểu so với khung quét laser để hệ thống sử dụng các điểm cuối. Điểm cuối được quét có giá trị thấp hơn sẽ bỏ qua.

Laser_z_max_value: Chiều cao tối đa so với khung quét laser để hệ thống sử dụng các điểm cuối. Điểm cuối được quét có giá trị cao hơn sẽ bỏ qua.



Hình 5.12. Bản đồ khi xoay LiDAR vượt ngưỡng

Hình 5.12 là bản đồ bị vẽ chồng chéo sai lệch khi xoay LiDAR vượt ngưỡng chỉ số được thiết lập sẵn Map_update_angle_threshold. Lúc này LiDAR sẽ phát cảnh báo ở Terminal “angle change too large”. Những góc sai lệch đó chính là nhiễu chồng chéo nhau khi LiDAR vẫn chưa kịp xác nhận giá trị của điểm ảnh, việc xoay cảm biến quá nhanh làm góc phản xạ của ánh sáng dội về cũng sai dần dẫn đến bản đồ bị méo hoặc chồng lên nhau. Cách khắc phục lỗi này là di chuyển và LiDAR chậm xuống thấp hơn hoặc bằng ngưỡng đã đặt ra để đảm bảo giá trị đầu ra chính xác.



Hình 5.13. Bản đồ thu được từ cảm biến LiDAR A1M8

Từ bản đồ 5.13 có thể thấy phần màu trắng mang giá trị 0 là điểm được quét mà không gặp vật cản nào, phần màu đen mang giá trị 1 là điểm khi laser dội ngược về cảm biến khi gặp vật cản. Phần màu xám là phần mà cảm biến không hoặc chưa quét đến, có thể là bên trong vật cản hoặc bên kia bức tường. Vì LiDAR sử dụng laser để đo khoảng cách nên sẽ không tránh khỏi những vật liệu có phản chiếu khiến kết quả sai sót. Có thể thấy ở bên trái bản đồ, do laser bị dội ngược khi gặp gương nên kết quả cho ra một không gian không tồn tại. Việc này có thể gây sai lệch trầm trọng trong việc vẽ lại bản đồ.

5.2. NHẬN XÉT VÀ ĐÁNH GIÁ

Trong suốt quá trình nghiên cứu đồ án "Xe quét bản đồ sử dụng LiDAR" nhóm đã đạt được những mục tiêu như sau:

+ Học thêm về thuật toán xử lý tín hiệu.

+ Tìm hiểu về quá trình và phương pháp hiệu quả để chế tạo một robot.

+ Hiểu thêm về một số thuật toán liên quan đến xây dựng bản đồ từ môi trường.

Đi kèm theo những điều trên thì nhóm cũng gặp không ít khó khăn trong việc chế tạo vì hạn chế về thời gian cũng như kiến thức chuyên môn:

+ Chưa tối ưu thời gian xây dựng bản đồ.

+ Trong quá trình thiết kế khung xe do chọn vi điều khiển có sẵn dẫn đến khả năng phát triển cũng như thực hiện nhanh các chức năng gặp khó khăn. Như việc điều khiển xe với 4 bánh có tốc độ chạy khác nhau dùng động cơ encoder.

+ Phần cứng dùng nhiều kết nối làm cho việc sửa hay thay đổi mất nhiều thời gian, nhóm sử dùng máy tính nhúng Jetson Nano có bộ nhớ eMMC nhưng kích thước bộ nhớ nhỏ và máy không hỗ trợ kết giao thức SSH khiến cho việc xử dụng gặp nhiều khó khăn.

+ Tính toán nguồn pin chưa hợp lý do việc thiếu khối nguồn dẫn đến việc robot chưa tận dụng được triệt để công suất của pin.

+ Khả năng dùng máy tính Linux cho việc học còn kém trong việc phát hiện lỗi và thực hiện cài đặt cho máy tính Linux.

CHƯƠNG 6: HƯỚNG PHÁT TRIỂN

Đề tài vẫn còn nhiều thiếu sót cũng như tiềm năng phát triển rất lớn mà nhóm vẫn chưa thể thực hiện. Từ các nhược điểm đã được liệt kê của đề tài, nhóm muốn đưa ra một số định hướng phát triển để đề tài hoàn chỉnh hơn:

+ Rút ngắn chi phí thiết kế, sử dụng raspberry pi để tối ưu chi phí phát triển thành hệ thống hoàn chỉnh.

+ Phát triển thêm động cơ xoay để LiDAR có thể quét môi trường 3D.

+ Thay thế phần cứng phù hợp với nhiệm vụ của robot tự hành.

+ Có thể gắn thêm camera giám sát để thu được kết quả chính xác hơn.

+ Thiết kế mạch in cho các module và lập trình thư việc cho chắc chức năng của xe.

+ Học hỏi thêm về máy tính nhúng dùng hệ điều hành Linux.

Không gian làm việc của nhóm với mong muốn phát triển lên đồ án tốt nghiệp, bao gồm các tài liệu liên quan, các thử nghiệm mô phỏng:

https://github.com/Toan8561/DA2_2024-2025/

TÀI LIỆU KHAM KHẢO

[1] MathWorks, "What is SLAM?" https://www.mathworks.com/discovery/slam.html

[2] Henry Lim, "Introduction to SLAM (Simultaneous Localization and Mapping), https://ouster.com/insights/blog/introduction-to-slam-simultaneous-localization-and-mapping

[3] "Understanding Structure From Motion Algorithms", https://medium.com/@loboateresa/understanding-structure-from-motion-algorithms-fc034875fd0c

[4] Robotic, "Introduction to Loop Closure Detection in SLAM", https://www.thinkautonomous.ai/blog/loop-closure/#:~:text=Loop%20closure%20is%20a%20sub,in%20the%20robot's%20pose%20estimation.

[5] "What is LiDAR and how does it work?", https://www.yellowscan.com/knowledge/how-does-lidar-work/

[6] Rmlab, Jetson Nano Development Kit A tích hợp 16GB eMMC, https://pivietnam.com.vn/combo-jetson-nano-dev-kit-a-basic-tich-hop-16gb-emmc-phien-ban-thay-the-cho-nvidia-jetson-nano-b01-kit-pivietnam-com-vn.html

[7] Waveshare, "Jetson-Nano-Dev-Kit", https://www.waveshare.com/wiki/JETSON-NANO-DEV-KIT

[8] Hshop, "Cảm biến khoảng cách LidarRPLIDAR A1M8 360 Laser Range Scanner", https://hshop.vn/cam-bien-laser-radar-lidar-rplidar-a1

[9] Slamtec, RPLIDAR A1 Datasheet,

https://www.slamtec.ai/wp-content/uploads/2023/11/LD108_SLAMTEC_rplidar_datasheet_A1M8_v3.0_en.pdf

[10] Matlab, chuỗi video về xe tự định hướng,

https://www.youtube.com/playlist?list=PLn8PRpmsu08rLRGrnF-S6TyGrmcA2X7kg

[11] SciJoy, "How Encoders Monitor Position - Physics (Position Activity)",

https://www.youtube.com/watch?v=j8rG3bjqriI

[12] Wikipedia, "Rotaty encoder", https://en.wikipedia.org/wiki/Rotary_encoder

[13] Asahi Kasei, "Encoder types and mechanism",

https://www.akm.com/global/en/products/rotation-angle-sensor/tutorial/type-mechanism-1/

[14] USDIGITAL, "Absolute Encoder Interfaces",

https://www.usdigital.com/blog/absolute-encoder-interface-types/

[15] Microchip, “8Pulse Width Modulation (PWM) Background”, https://skills.microchip.com/programming-an-8-bit-pic-mcu-pulse-width-modulation-pwm-output-using-mcc/726696

[16] Electronic Lab, “Analog To Digital Conversion – Performance Criteria”, https://www.electronics-lab.com/article/analog-to-digital-conversion-performance-criteria/

[17] Chetan Shidling, “What Is ADC, Working Of ADC, Analog To Digital converter”, https://cselectricalandelectronics.com/what-is-adc-working-of-adc-analog-to-digital-converter/

[18] All About Circuits, “Flash ADC”, https://www.allaboutcircuits.com/textbook/digital/chpt-13/flash-adc/

[19] Dibyendu Biswas, “Hector SLAM”, https://dibyendu-biswas.medium.com/hector-slam-6ce73e3b372b

[20] Gökhan Atalı, “RPLidar ROS Hector SLAM”, https://www.youtube.com/watch?v=h16BGFK_V9Q

[21] Computer Vision Foundation, “Structure from Motion revisited”, https://openaccess.thecvf.com/content_cvpr_2016/papers/Schonberger_Structure-From-Motion_Revisited_CVPR_2016_paper.pdf