**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

---------------o0o---------------



**ĐỒ ÁN 2**

**GVHD: TS. NGUYỄN VĨNH HẢO**

**NGUYỄN VÕ HỒNG MỸ HIỀN MSSV: 2010260**

**DƯƠNG NGỌC HOÀN MSSV: 2010020**

**TP. HỒ CHÍ MINH, THÁNG 1 NĂM 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH  **KHOA ĐIỆN – ĐIỆN TỬ**  **BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG** | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP. HCM, ngày 30 tháng 12 năm 2023* |

## ĐỀ CƯƠNG CHI TIẾT

|  |
| --- |
| **TÊN LUẬN VĂN:** |
| **Cán bộ hướng dẫn: TS. NGUYỄN VĨNH HẢO** |
| **Thời gian thực hiện:** Từ ngày 30/8 đến ngày 30/12 |
| **Sinh viên thực hiện:**  **Nguyễn Võ Hồng Mỹ Hiền - 2010260**  **Dương Ngọc Hoàn - 2010020** |
| **Nội dung đề tài:**   * Thiết kế mô hình xe tự hành cho môi trường trong nhà * Nghiên cứu xây dựng ứng dụng trên nền tảng ROS cho robot * Ứng dụng cảm biến Lidar cho robot xây dựng bản đồ |
| **Kế hoạch thực hiện:**  Các công việc của nhóm được hai thành viên phân chia như sau: |

|  |  |  |
| --- | --- | --- |
| **Giai đoạn** | **Dương Ngọc Hoàn** | **Nguyễn Võ Hồng Mỹ Hiền** |
| **Giai đoạn 1** | - Vẽ mô hình robot trên SolidWorks và đặt cắt mica.  - Xây dựng mô hình mô phỏng trên ROS2. | - Thiết kế mạch Driver cho 4 motor DC.  - Thiết kết mạch điều khiển chính.  - Tính toán vấn đề công suất. |
| **Giai đoạn 2** | - Hàn mạch và lắp ráp robot. | - Hàn mạch và lắp ráp robot.  - Test các module phần cứng. |
| **Giai đoạn 3** | - Tìm hiểu giao tiếp uROS.  - Lập trình Firmware uROS giao tiếp giữ Jetson Nano và Raspberry Pico. | - Tìm hiểu mối liên hệ vận tốc bánh xe và vận tốc của robot.  - Lập trình Firmware đọc cảm biến encoder và điều khiển PID cho động cơ. |
| **Giai đoạn 5** | - Tìm hiểu thuật toán Iterative Closets Point (IPC).  - Sử dụng Lidar để mapping không gian. | - Lập trình Firmware đọc adc giám sát pin của robot.  - Lập trình Firmware module scheduler. |
| **Giai đoạn 6** | - Tiến hành viết báo cáo. | - Tiến hành viết báo cáo. |

# **LỜI CẢM ƠN**

Đầu tiên, chúng em xin gửi lời cảm ơn tới Ban giám hiệu nhà trường, đoàn hội, khoa, tập thể giảng viên và công nhân viên chức nhà trường Đại học Bách Khoa – Đại học Quốc gia TP.HCM đã xây dựng cho chúng em một môi trường học tập và phát triển tốt. Đây là nền tảng cơ sở vững chắc cho chúng em có tiếp cận và rèn luyện được nhiều kỹ năng, kiến thức mới.

Tiếp theo, chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Vĩnh Hảo, người đã tận tình hướng dẫn chúng em thực hiện luận văn. Thầy không chỉ là điểm tựa tinh thần, giải đáp các vấn đề chuyên ngành và đưa ra những lời khuyên hết sức quý giá, mà còn tạo điều kiện vật chất, môi trường làm việc nghiên cứu thuận lợi nhất giúp nhóm hoàn thành luận văn với những mục tiêu đã đề ra.

Chúng em cũng chân thành cảm ơn các thầy cô Khoa Điện-Điện Tử, đặc biệt là những thầy cô trong bộ môn Điều Khiển và Tự Động Hóa. Các thầy cô luôn hết sức giảng dạy cho sinh viên một cách chu đáo, truyền đạt kiến thức nền tảng đến chuyên sâu cũng như những kinh nghiệm thực tế của bản thân, góp phần giúp chúng em nắm vững và áp dụng kiến thức vào luận văn, tạo động lực học tập nghiên cứu lâu dài.

Sau cùng, nhóm gửi lời cảm ơn đến các bạn cùng chuyên ngành, đặc biệt là các anh chị, các bạn ở phòng thí nghiệm 207B3 đã nhiệt tình giúp đỡ và đóng góp ý kiến để nhóm hoàn thành luận văn một cách hoàn thiện.

**TÓM TẮT ĐỒ ÁN 2**

Đề tài này thực hiện thiết kế thi công phần cứng và thiết kế xây dựng phần mềm cho robot tự hành môi trường trong nhà. Robot thực hiện các tác vụ như: vẽ bản đồ, định vị, điều hướng và tránh vật cản. Bộ não của robot là một máy tính nhúng Jetson Nano, các cảm biến được sử dụng cho robot là Lidar 2D, Real Sense Depth Camera, IMU và Encoder. Dựa vào các dữ liệu cảm biến này mà bản đồ 2D hoặc 3D có thể được xây dựng bởi thuật toán RTAB-Map, từ đó thưc hiện định vị và điều hướng cho robot (sẽ phát triển trong luận văn). Máy tính nhúng giao tiếp với bảng mạch nhúng có vi điều khiển Raspberry Pico để nhận dữ liệu Odometry từ Encoder và IMU, cũng như gửi tín hiệu điều khiển robot. Dữ liệu bản đồ vẽ trên máy tính nhúng có thể được hiển thị trên PC bất kỳ có cài đặt hệ điều hành ROS2 và phần mềm RVIZ thông qua chuẩn UDP không dây. UDP cũng giúp gửi các lệnh điều khiển trên PC xuống Jetson để hỗ trợ quá trình vận hành robot.

Odometry sử dụng cho việc vẽ bản đồ được lấy từ một trong hai nguồn cảm biến: Lidar hoặc IMU và Encoder. Kết quả thực hiện khi sử dụng hai nguồn trên được so sánh và đánh giá về khả năng cũng như giới hạn.

**CHƯƠNG 1. GIỚI THIỆU**

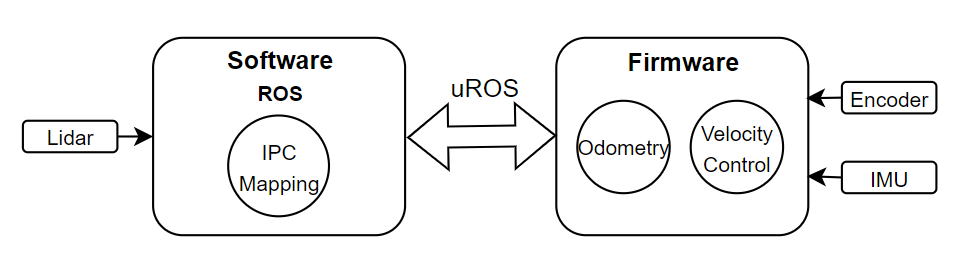
* 1. **Tổng quan đề tài nghiên cứu**

Hiện nay, công nghệ liên quan đến robot di động (Mobile Robot) như phương tiện tự hành (autonomous vehicles) hoặc robot phục vụ trong nhà đang được đưa rất nhiều vào trong nghiên cứu. Một thành phần thiết yếu trong ứng dụng robot là hệ thống điều hướng (navigation), nó giúp robot nhận dạng và tái tạo môi trường để việc di chuyển trở nên hiệu quả hơn. Một robot tự hành cần đáp ứng được các yêu cầu tất yếu: thứ nhất, cần phải xác định được vị trí của của bản thân trong hệ tọa độ tham chiếu; thứ hai, robot phải lập cho mình một kế hoạch tìm đường và tranh vật cản tự động; cuối cùng là robot phải nhận biết được môi trường xung quanh thông qua các cảm biến (nhận thức). SLAM (Simultaneous Localization and Mapping) là một hệ thống định vị và lập bản đồ trực quan thời gian thực giúp robot hoạt động ưu việt hơn.

Nhiều hệ thống SLAM khác nhau đang được nghiên cứu phát triển, hầu hết chúng sử dụng cảm biến quang, nổi bật trong số đó là Visual SLAM (camera-based) sử dụng thông tin ảnh thu được từ camera và Lidar SLAM (lidar-based) sử dụng thông tin laser từ cảm biến lidar. Với mục tiêu hoạt động ở môi trường trong nhà, đề tài nghiên cứu được nhóm lựa chọn sử dụng cảm biến Lidar và Depth Camera phục vụ nhiệm vụ SLAM thông qua một thuật toán mã nguồn mở giúp xây dựng bản đồ 2D và 3D với sự kết hợp của hai loại cảm biến trên.

Để xây dựng một hệ thống hoàn thiện thì đây là một đề tài khá rộng và cần có những chuyên gia có kiến thức, kinh nghiệm để có thể phát triển một hệ thống ổn định, hoạt động trôi chảy trên nhiều điều kiện môi trường khác nhau. Vì tính ứng dụng rộng lớn và tính phức tạp của vấn đề, luận văn chỉ ở mức tìm hiểu thuật toán và ứng dụng các thuật toán cơ bản để có thể xây dựng được mô hình hoạt động hoàn thiện.

* 1. **Mục tiêu đề tài**



Hình . Tổng quan hệ thống đề tài đồ án

Đồ án 2 với mục tiêu thiết kế phần cứng, điều khiển vận tốc robot và sử dụng Lidar để mapping. Hình 1 thể hiện tổng quan hệ thống cần xây dựng của đề tài. ROS sẽ đọc cảm biến và nhận odometry từ Firmware để xử lí dữ liệu thông tin về môi trường. Các khối trong ROS có chức năng nhận dữ liệu từ ROS để đưa ra xử lí cho ra đầu ra mong muốn và truyền tín hiệu điều khiển xuống Firmware. Đồng thời, giao tiếp uROS liên kết giữa Jetson Nano và Raspberry Pico để nhận dữ liệu odometry về cũng như gửi tín hiệu điều khiển vận tốc xuống để robot di chuyển. Từ đó, nhóm đặt ra mục tiêu cần hoàn thành như sau:

Thiết kế mô hình robot và mạch điều khiển.

Xây dựng mô hình robot nhận tín hiệu điều khiển vận tốc và thực thi đúng với yêu cầu đặt ra.

Xây dựng bản đồ 2D của môi trường trong nhà chưa biết trước bằng cảm biến Lidar.

* 1. **Phương pháp thực hiện**

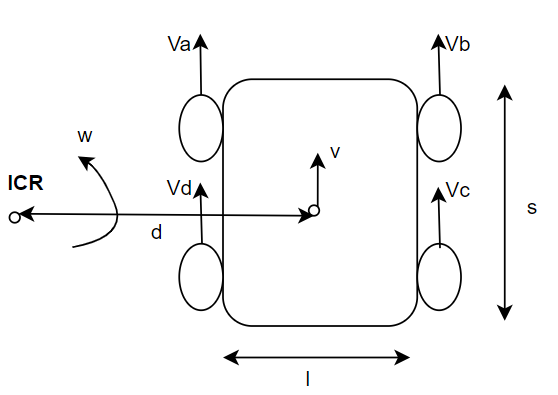
Dựa vào môi trường hoạt động của robot là ở trong nhà nên nhóm tự thiết kế và lắp ráp khung robot với động cơ, nguồn, vi điều khiển, cảm biến,… sao cho khung robot chịu được tải trọng của robot và có hình dáng giúp robot dễ dàng di chuyển trong nhà, phần này được trình bày ở mục 3.1. Quá trình thiết kế khung xe cùng mạch điều khiển và cảm biến được thực hiện thông qua phần mềm hỗ trợ là SolidWorks và Kicad, phần này được trình bày ở mục 3.3 và 3.4. Với nhiệm vụ đặt ra nhóm tiến hành tìm hiểu lí thuyết về các thuật toán liên quan, phần này được trình bày ở chương 2. Việc thiết kế phần mềm cho mạch điều khiển bao gồm điều khiển động cơ DC bằng thuật toán PID và đọc encoder, đồng thời tích hợp giao thức uROS để giao tiếp với nền tảng ROS được trình bày ở mục 3.5. Robot sử dụng máy tính nhúng Jetson Nano là bộ xử lí trung tâm, hoạt động với hệ điều hành Ubuntu có tích hợp ROS. Dựa trên đó, nhóm tiến hành thiết kế phần mềm xử lí chính trên ROS gồm việc đọc cảm biến Lidar và áp dụng thực toán Iterative Closets Point để xây dựng bản đồ 2D, được trình bày ở mục 3.6. Cuối cùng, nhóm thu thập kết quả tổng kết ở chương 4, từ đó đánh giá quá trình thực hiện, tìm hiểu cách khắc phục và hướng phát triển trong tương lai, được trình bày ở chương 5.

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1. Điều khiển tốc độ động cơ**

**2.1.1. Mô hình Differential Drive**

Khi robot được thiết kế theo mô hình Differential Drive, mối liên hệ giữa vận tốc của từng bánh xe 𝑣𝑙, 𝑣𝑟 với vận tốc xoay 𝜔 và vận tốc tịnh tiến 𝑣 của robot có thể được miêu tả rõ ràng bằng công thức.



Hình . Mô hình Differential Drive

Với ICR là tâm xoay tức thời khi robot di chuyển, l là khoảng cách giữa 2 bánh xe, s là chiều dài xe, d là khoảng cách từ trọng tâm xe đến ICR, 𝑣 và 𝜔 lần lượt là vận tốc trọng tâm robot và vận tốc quay quay của robot đối với tâm quay tức thời, va, vb, vc, vd là vận tốc của các bánh xe được xác định bằng công thức như sau:

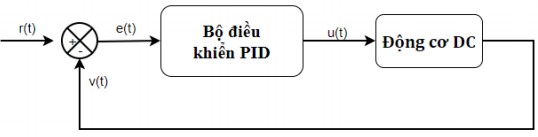


Phần chứng minh được trình bày ở phần phụ lục.

**2.1.2. Bộ điều khiển PID**

Bộ điều khiển PID là bộ điều khiển vòng kín có hồi tiếp được sử dụn phổ biến trong việc điều khiển động cơ DC. Với T là chu kì điều khiển, u(k) là tín hiệu điểu khiển, r(k) là tín hiệu đặt, v(k) là tín hiệu hồi tiếp, e(k) là sai số giữa tín hiệu đặt và tín hiệu hồi tiếp e(k) = u(k) – r(k) ở chu kì k. Rời rạc hóa ta được phương trình cập nhật tín hiệu điều khiển như sau:





Hình . Sơ đồ khối bộ điều khiển PID

Trong đó, Kp, Ki, Kd là các hệ số của 3 khâu (khâu tỉ lệ P, khâu tích phân I và khâu vi phân D) giúp tinh chỉnh bộ điều khiển và có ảnh hưởng lên bộ điều khiển như sau:



Hình . Ảnh hưởng của các hệ số trong bộ điều khiển PID[[1]](#footnote-1)

**CHƯƠNG 3. NỘI DUNG THỰC HIỆN**

**3.1. Các module phần cứng**

**3.1.1. RPLidar A1**

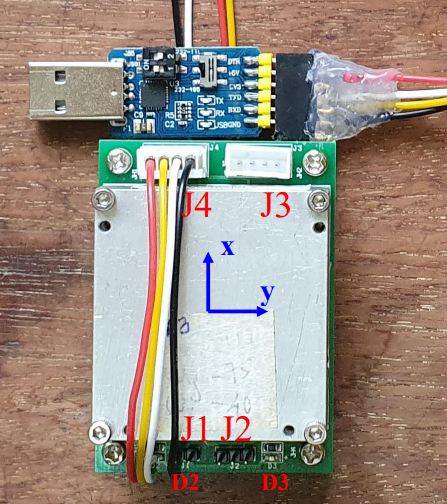
Cảm biến Laser Radar RPLIDAR A1 12m của hãng SLAMTEC sử dụng giao tiếp UART, kết nối máy tính qua mạch chuyển USB-UART và phần mềm đi kèm. Dữ liệu truyền về sẽ là các point clouds. Các point cloud này được sử dụng để vẽ bản đồ 2D.

Hình . RPLidar A1

**3.1.2. Cảm biến IMU ADIS16488 được nhúng trên board mạch chứa vi xử lí STM32F405RG**

16488 là một đơn vị đo quán tính (IMU) dựa trên công nghệ gia công vi mô (MEMS), bao gồm con quay hồi chuyển MEMS hiệu suất cao và gia tốc MEMS và đưa ra ba vận tốc góc và ba gia tốc.

Đồng thời board nhúng hỗ trợ thuật toán ước lượng góc xây dựng trên cảm biến có hoặc không sử dụng từ trường, tự ước lượng giá trị nhiễu nền của cảm biến.

****

Hình . Cảm biến IMU ADIS16488

**3.1.3. Máy tính nhúng Jetson Nano**

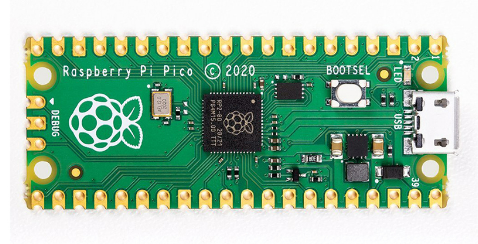
Các giải thuật SLAM vẽ bản đồ và định vị cho robot được thực hiện trên ROS, vì vậy cần có một máy tính chạy hệ điều hành này gắn trên robot. Bộ kit phát triển NVIDIA Jetson Nano là một máy tính nhúng nhỏ nhưng rất mạnh mẽ, cho khả năng tính toán cao, phù hợp với các ứng dụng nhúng. Máy tính nhúng có tích hợp nhiều cổng đọc được dữ liệu từ camera, lidar, USB UART, … và sử dụng nguồn 5V – 4A.

****

Hình . NVIDIA Jetson Nano

**3.1.4. Vi điều khiển Raspberry Pico**

Raspberry Pico là mạch vi điều khiển hiệu năng cao, chi phí thấp được xây dựng dựa trên chip RP2040-chip vi điều khiển được thiết kế bởi chính Raspberry Pi. Nó có thể được lập trình lại dễ dàng qua USB từ Raspberry Pi hoặc máy tính khác sử dụng C/C++ hoặc MicroPython.



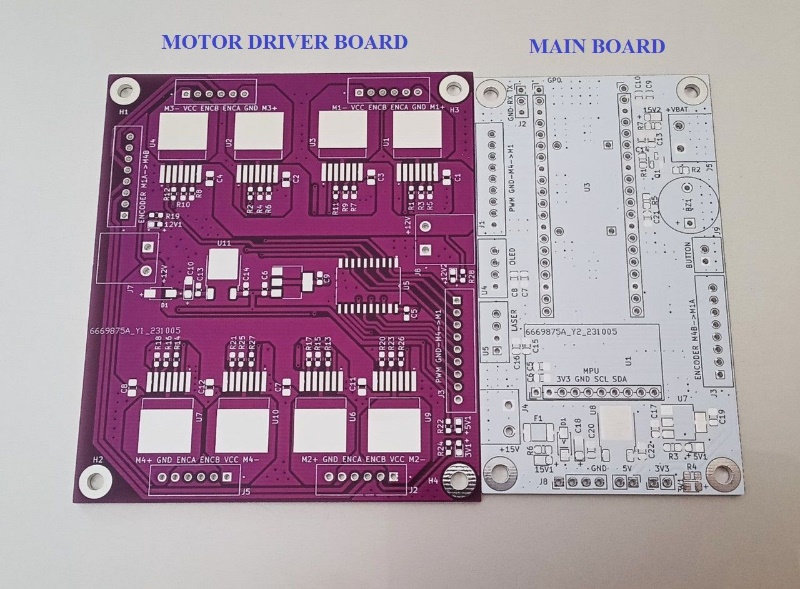
Hình . Raspberry Pico

**3.1.5. Driver motor, board mạch chính và DC Servo JGB37-520 có Encoder**

Board mạch cầu H dùng IC BTS7960, driver sử dụng nguồn 12V-4A, dùng để điều khiển 4 động cơ DC. Driver nhận đầu vào là 8 xung điều khiển PWM, đầu ra là 8 chân nguồn cấp cho 4 động cơ.

Board mạch chính chứa vi điều khiển Raspberry Pico, 2 module đọc adc, nguồn LDO, và các header để kết nối cảm biến. Sử dụng nguồn 15V-1A.

DC Servo JGB37-520 sử dụng nguồn 12V-1A (max) 333RPM



Hình . Driver và board mạch chính



Hình . DC Servo JBG37-520

**3.1.6. Pin sạc 18650 Li-ion**

Pin sạc Li-ion 18650 2000mAh 3.7V 3C thường được sử dụng để làm sạc dự phòng, cấp nguồn cho mạch điện hoặc các cấu trúc robot đơn giản.



Hình . Pin sạc Li-ion 18650

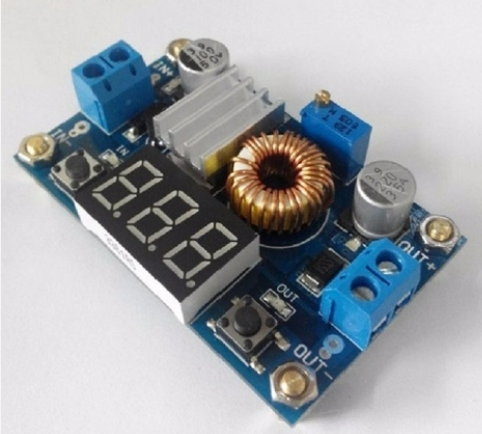
**3.1.7. LM2596S mạch giảm áp 3A và XL4015 mạch giảm áp 5A**

Mạch giảm áp DC-DC LM2596S 3A có kích thước nhỏ gọn có khả năng giảm áp từ 30VDC xuống còn 1.5VDC mà vẫn đạt hiệu suất cao (92%), thích hợp cho các ứng dụng chia nguồn, hạ áp,…

Mạch giảm áp XL4015 5A mạch có tích hợp đồng hồ Led và phím chức năng chọn hiển thị áp đầu vào (4-38VDC) và đầu ra (1.25-36VDC) để tiện theo dõi. Dễ dàng điều khiển điện áp bằng biến trở tinh chỉnh để thông số chính xác.



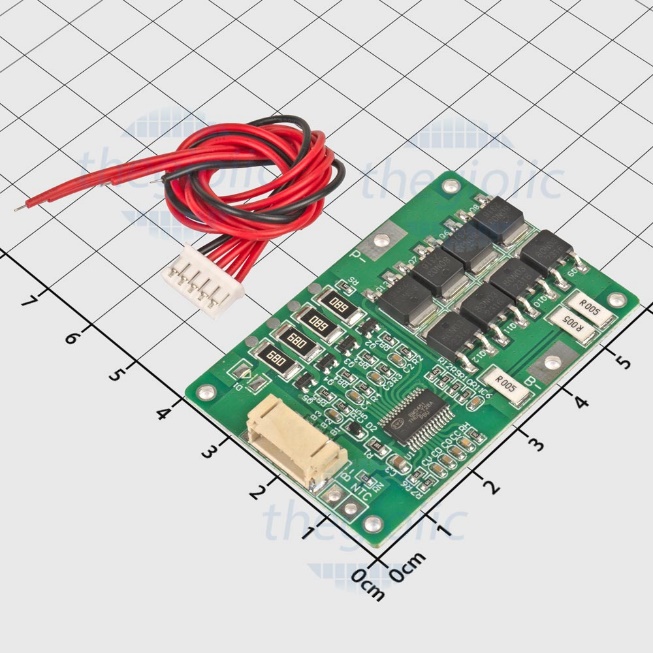
Hình . Mạch giảm áp DC LM2596S 3A



Hình . Mạch giảm áp DC XL4015 5A

**3.1.8. Mạch sạc pin 18650 4S**

JH-996040 8 MOS mạch sạc pin có điện áp sạc là 16.8V có thể chịu được dòng điện tức thời lên tới 40A, bảo vệ an toàn hiệu quả cho pin với thiết kế 4s.

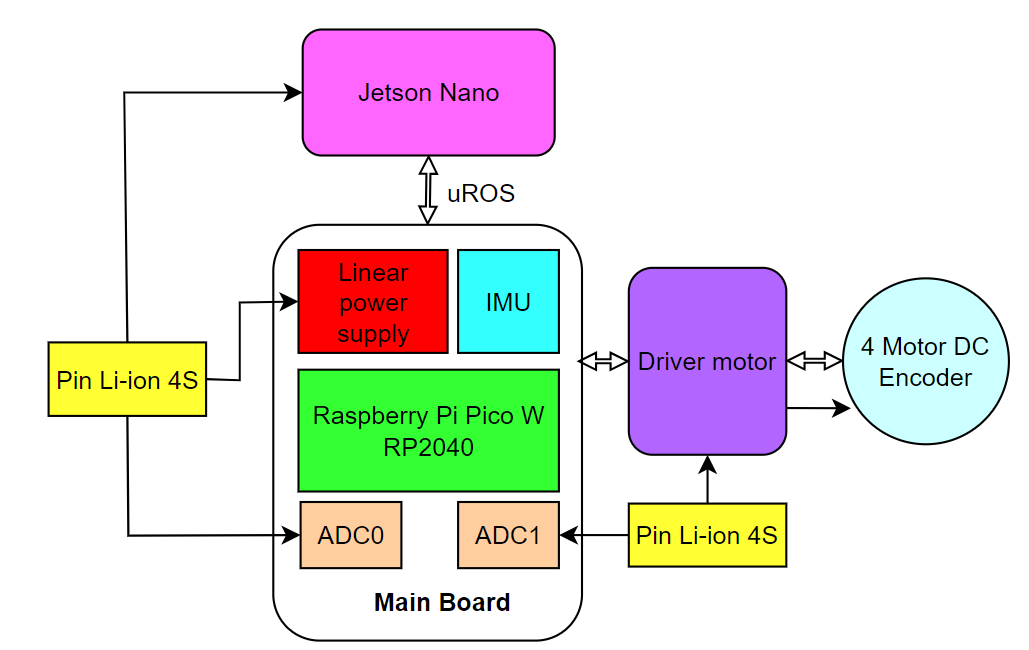


Hình . Mạch sạc pin 18650 4S

**3.2. Kết nối phần cứng**

**3.2.1. Khối vi điều khiển**

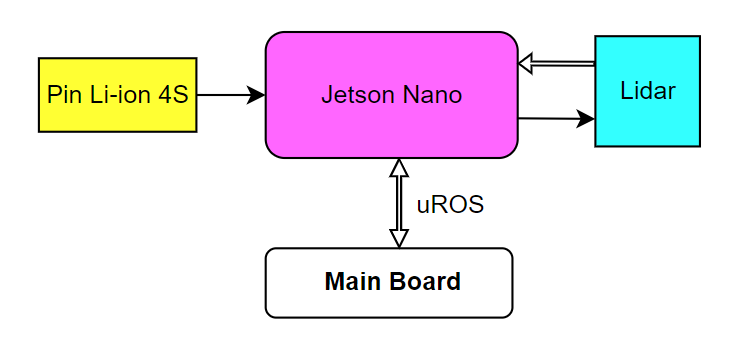
Main board được cấp nguồn bởi pin Li-ion 4S 15V đây cũng là nguồn cấp cho Jetson Nano. Khi nhận được thông tin điều khiển từ Jetson Nano (ROS) thông qua giao tiếp uROS, vi điều khiển dùng bộ điều khiển PID để xuất xung PWM cho mạch Driver điểu khiển động cơ. Driver được cấp bằng nguồn Li-ion 4S 15V khác, nhằm tách riêng nguồn cho mạch công suất và nguồn cho mạch điều khiển.



Hình . Sơ đồ khối tổng quan mạch điều khiển

**3.2.2. Khối máy tính nhúng**

Hệ điều hành robot (ROS) được cài đặt trên máy tính nhúng Jetson Nano. Máy tính nhúng được cấp nguồn bởi pin Li-ion 4S 15V, qua mạch buck DC-DC XL4015 để có được nguồn 5V-4A. Cảm biến Lidar sẽ lấy nguồn từ máy tính nhúng thông qua cổng USB.

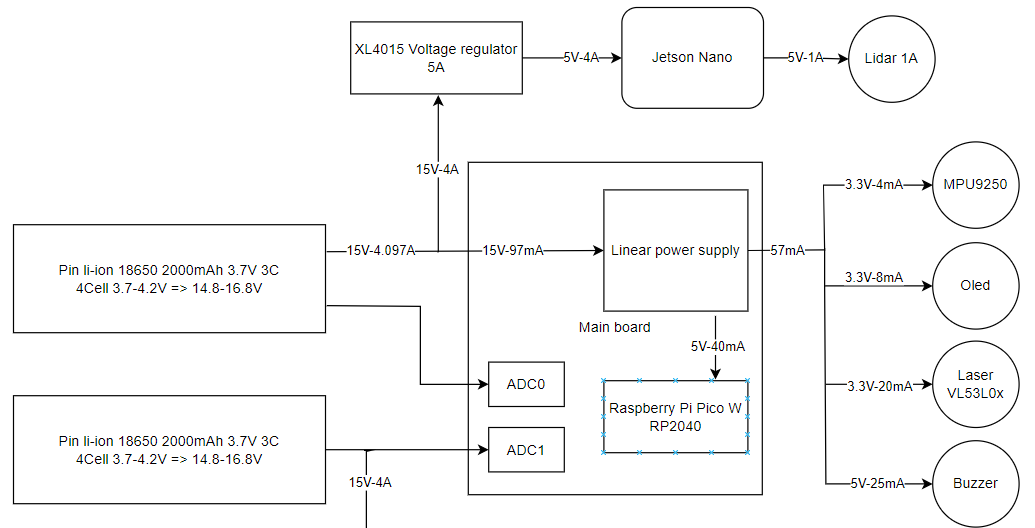


Hình . Sơ đồ khối máy tính nhúng

**3.3. Thi công mạch điều khiển**

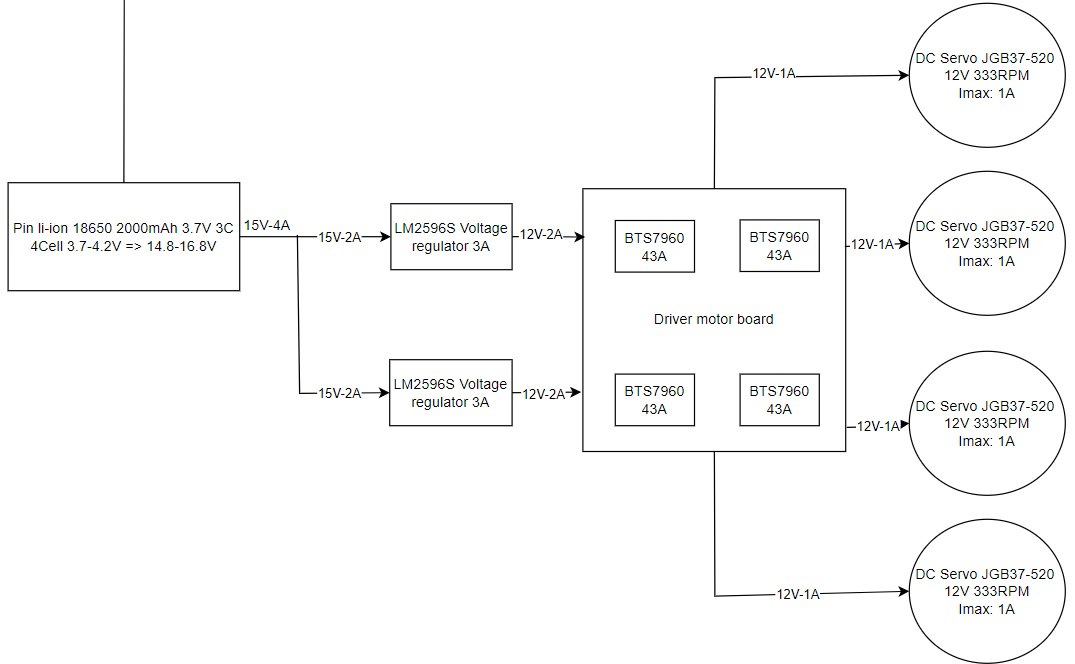
**3.3.1. Power diagram của toàn hệ thống**

Công suất phần khối điều khiển chính



Hình . Power diagram của khối điều khiển

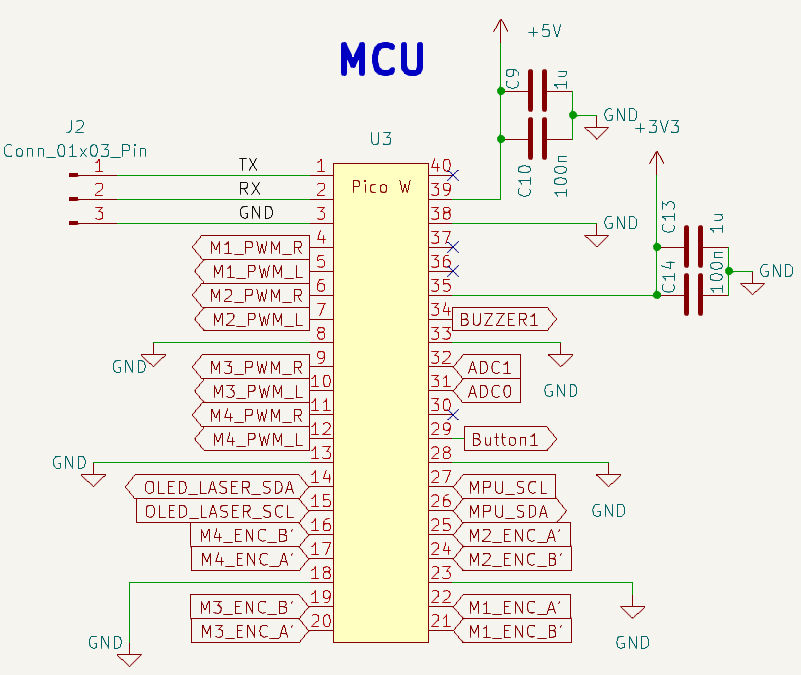
Công suất phần động lực



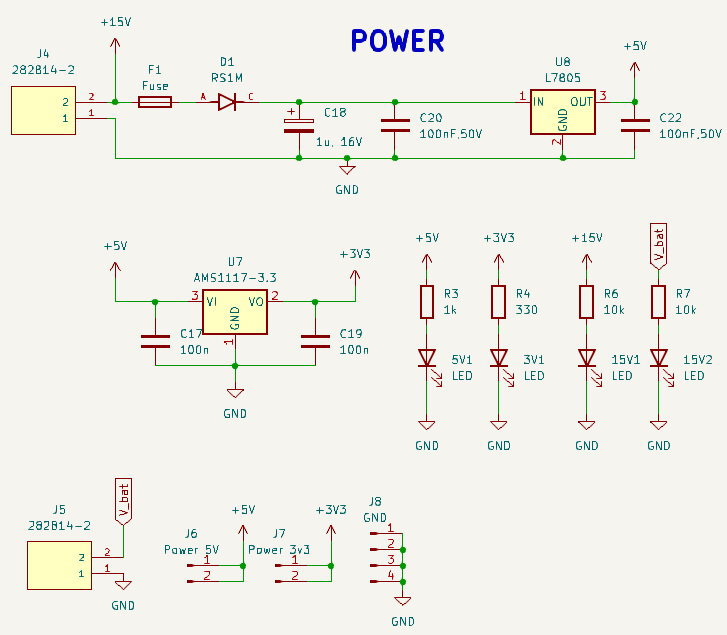
Hình . Power diagram phần động lực

**3.3.2. Mạch điều khiển chính chứa vi điều khiển Raspberry Pico**

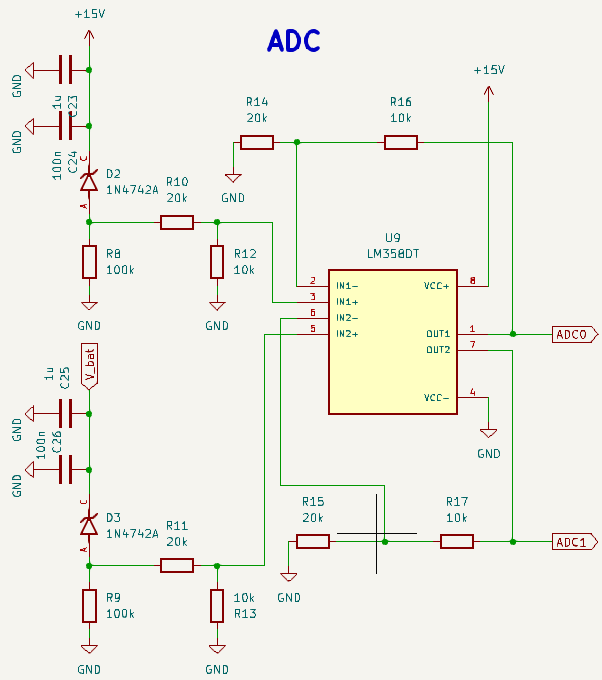
Schematic của từng khối trong main board



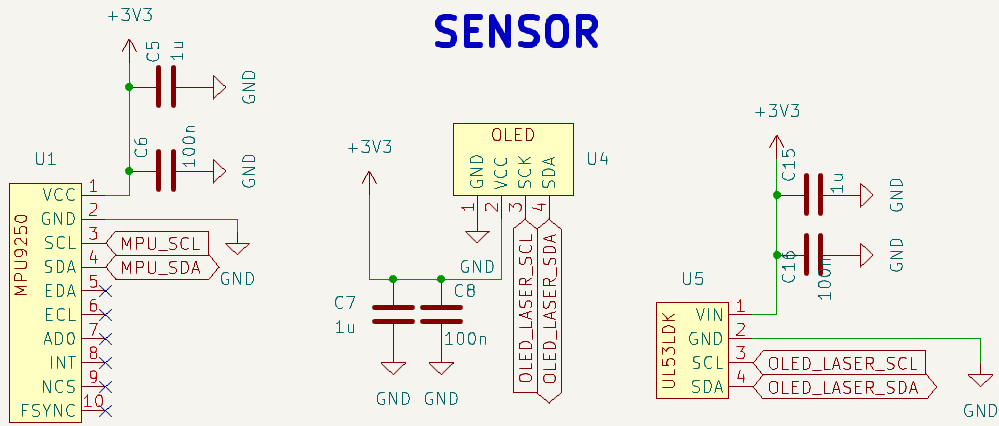
Hình . Schematic của khối vi điều khiển



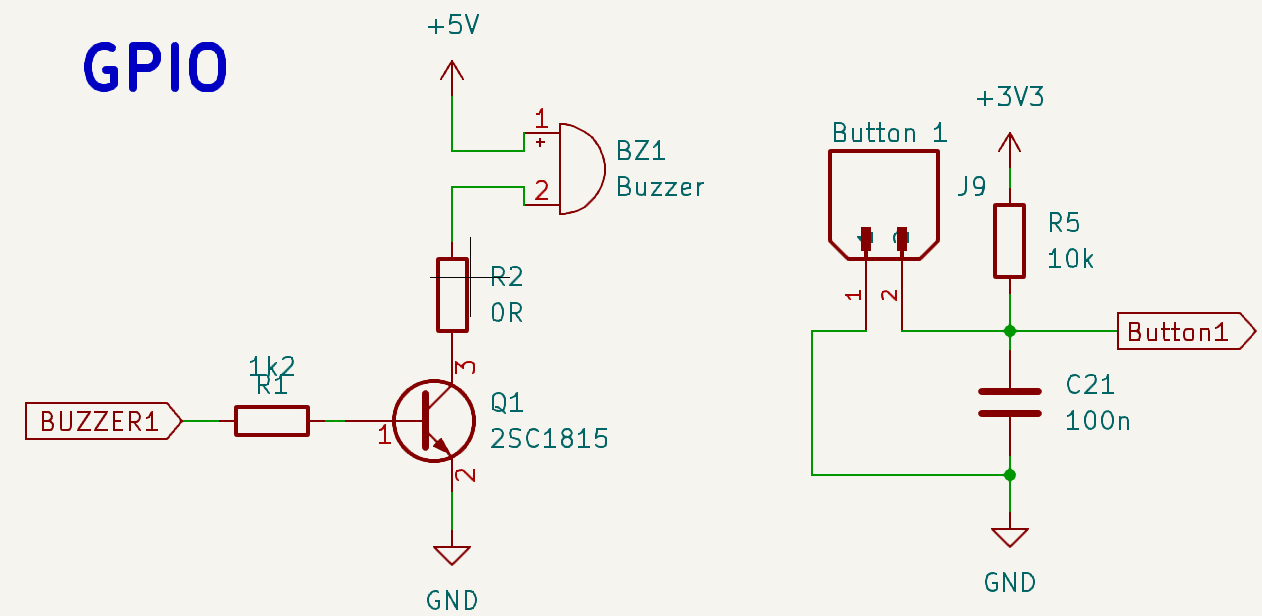
Hình . Schematic khối nguồn của main board



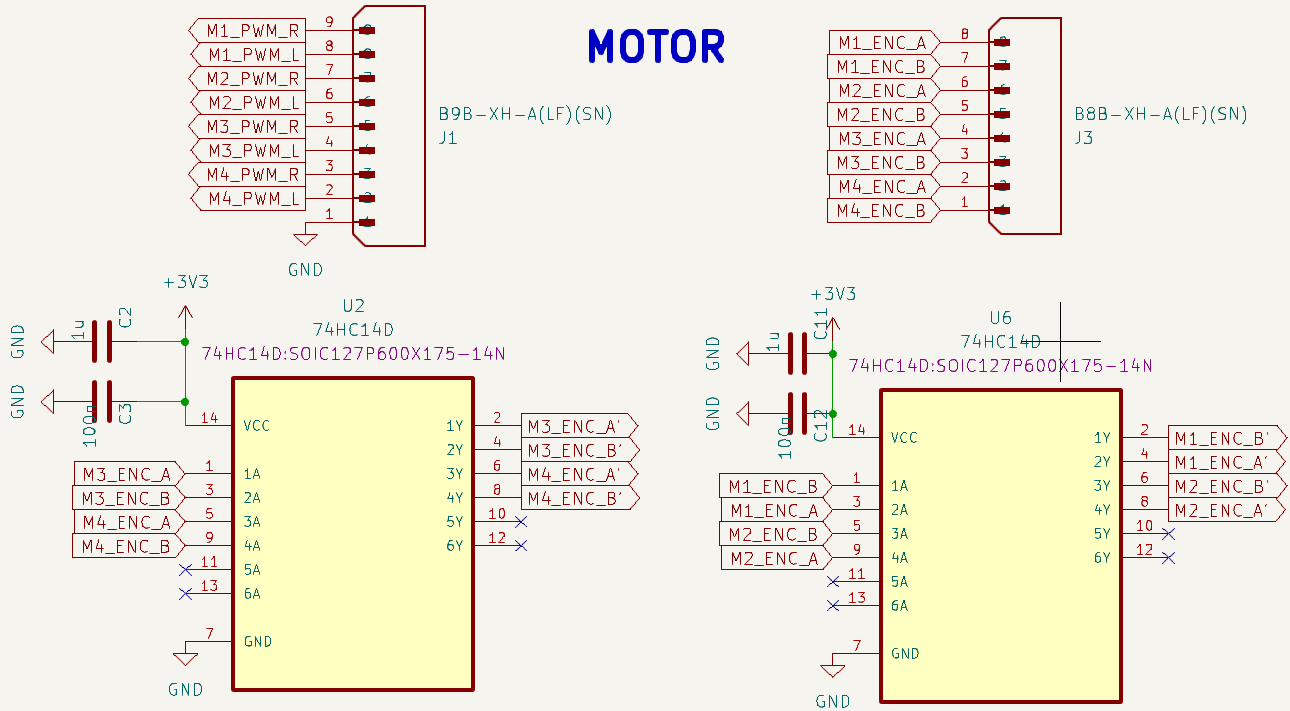
Hình . Schematic của khối adc



Hình . Schematic của khối cảm biến

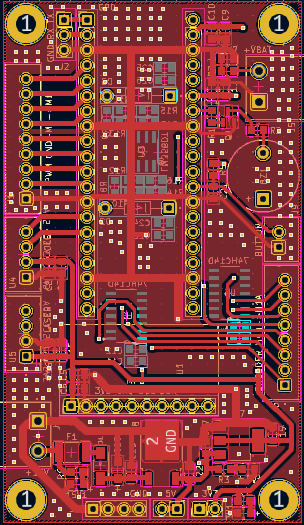


Hình . Schematic của khối GPIO

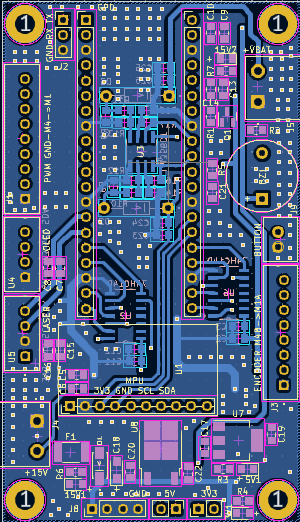


Hình . Schematic của khối motor

Layout của main board



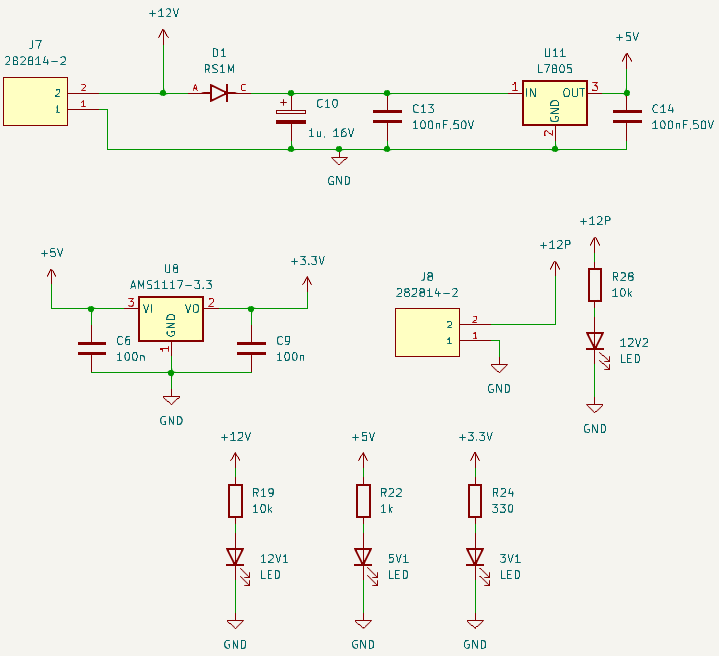
Hình . PCB top layer của main board



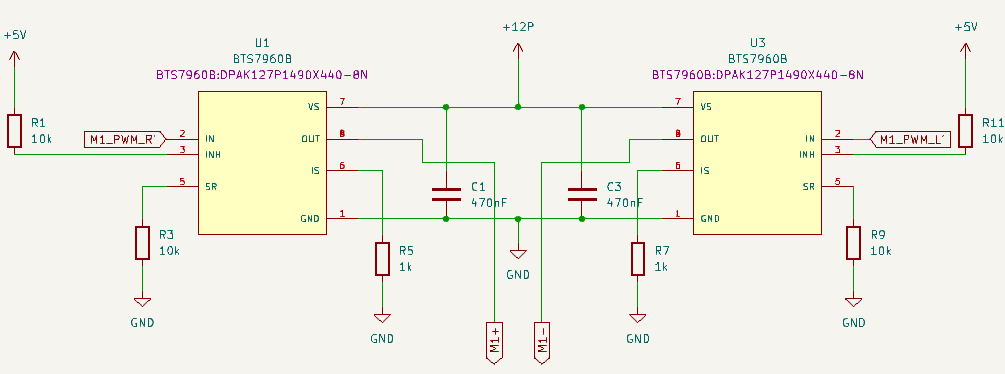
Hình . PCB bottom layer của main board

**3.3.3. Mạch Driver điều khiển motor DC**

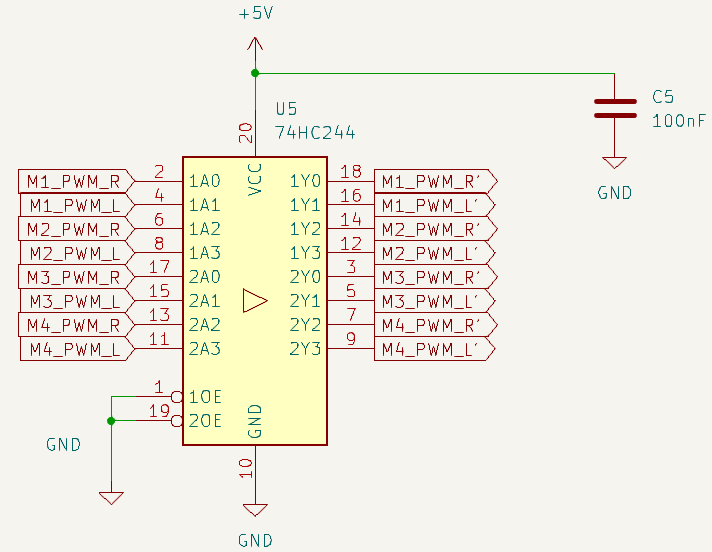
Shematic của mạch Driver điều khiển motor DC



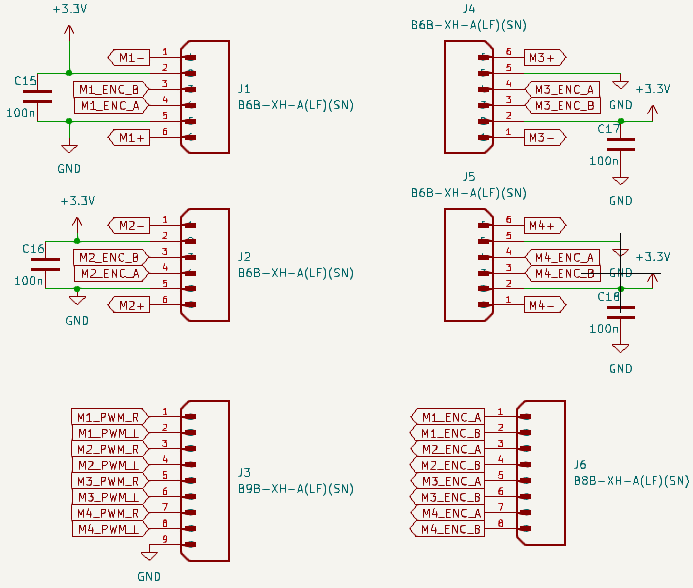
Hình . Schematic khối nguồn của Driver



Hình . Shematic của một cầu H (tương tự cho các mạch cầu còn lại)

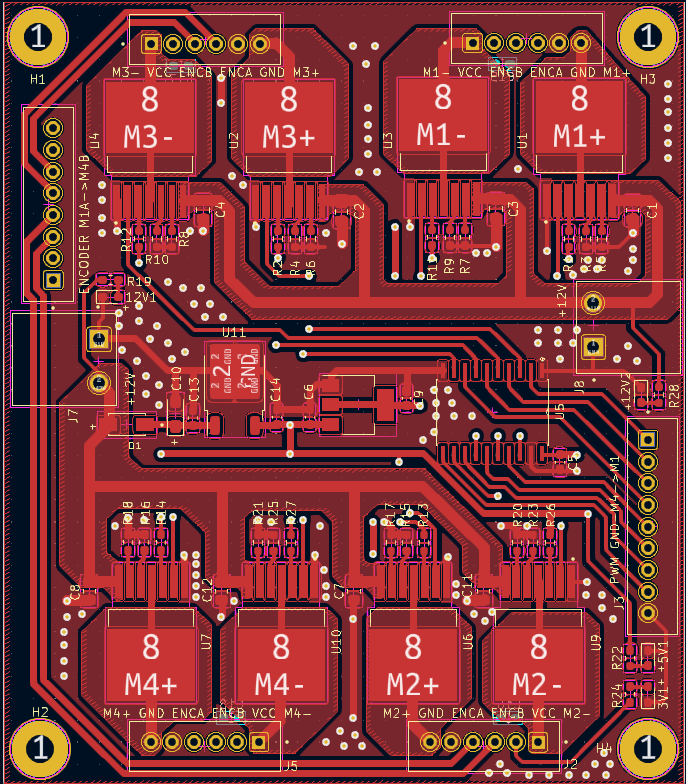


Hình . Schematic của IC đệm

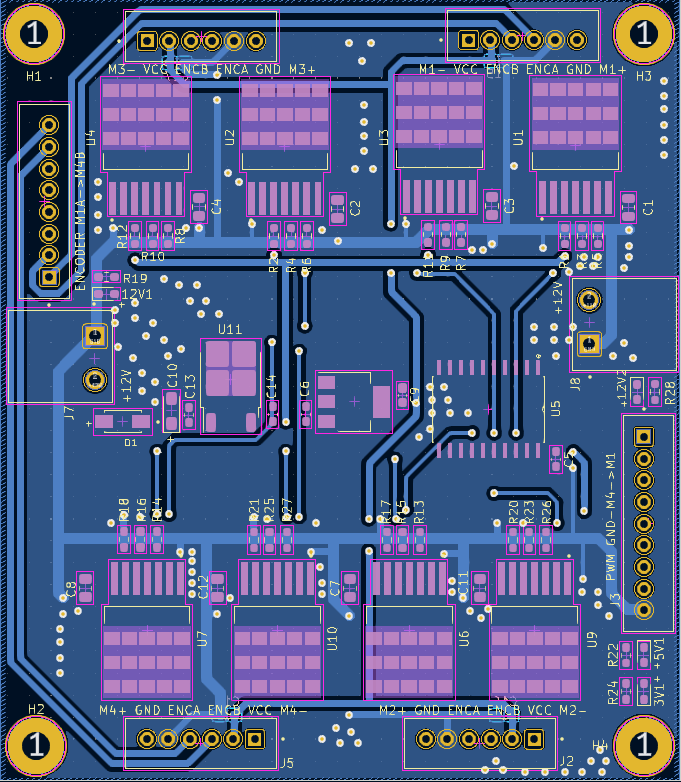


Hình . Schematic của khối connect

Layout của mạch Driver



Hình . PCB top layer của Driver



Hình . PCB bottom layer của Driver

**3.4. Thiết kế mô hình**

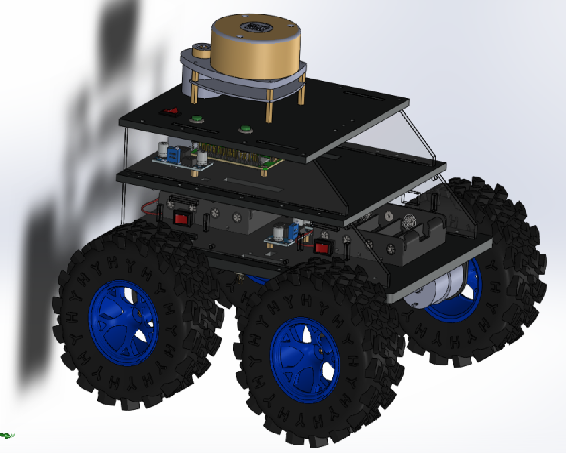
Nhóm sử dụng phần mềm vẽ 3D SolidWorks để vẽ mô hình robot, tự thiết kế các tầng robot đáp ứng các nhu cầu lắp ráp cần thiết cho các thành phần phần cứng được trình bày ở mục 3.1. Robot sẽ được vẽ với 3 tầng nhầm thực hiện 3 nhiệm vụ riêng biệt với nhau gồm:

Tầng dưới: sẽ được sử dụng để lắp 4 motor có bánh xe và driver điều khiển, đồng thời đây là nơi cố định nguồn điện và chứa mạch sạc. Nhiệm vụ của tầng này là cung cấp nguồn, sạc pin và nhận tín hiệu điều khiển motor để di chuyển robot.

Tầng giữa: sẽ lắp main board, máy tính nhúng Jetson Nano. Có thể nói nơi đây là bộ bão của robot. Nhiệm vụ của tầng này là thu thập thông tin từ các cảm biến và thực hiện các giải thuật để đưa ra các tín hiệu điều khiển.

Tầng trên: chứa Lidar và IMU, đây là nơi thu thập dữ liệu môi trường.

Sau quá trình tính toán kĩ lưỡng và vẽ nhiều mẫu thiết kế khác nhau, nhóm đã quyết định chọn thiết kế mô hình cuối cùng như hình dưới đây để có thể đáp ứng được các yêu cầu đặt ra, thực hiện các tác vụ mà robot cần làm.



Hình . Mô hình robot thiết kế dùng SolidWorks

**3.5. Xây dựng Firmware**

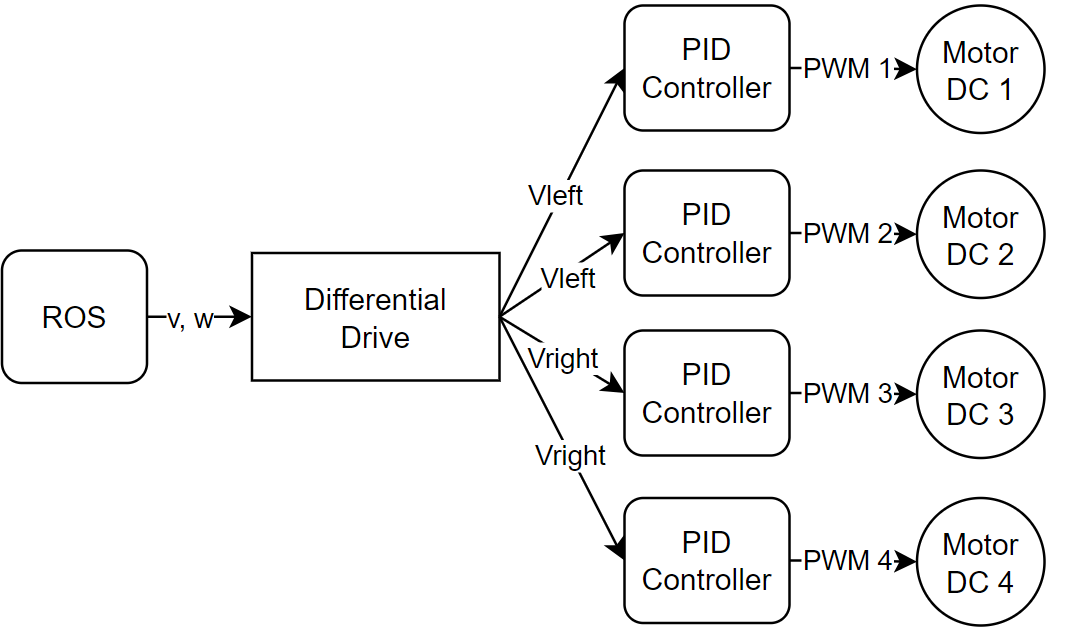
**3.5.1. Điều khiển tốc độ động cơ**

Việc điều khiển tay để robot di chuyển và mapping hay Navigation đều cần sự tương tác giữa máy tính nhúng và vi điều khiển. Máy tính nhúng sẽ gửi lại giá trị vận tốc tính tiến 𝑣 và vận tốc xoay 𝜔 qua cổng nối tiếp bằng giao tiếp uROS. Do robot được thiết kế theo mô hình Diferential Drive, ta sẽ tính toán được giá trị vận tốc riêng cho từng bánh vright, vleft để đạt được 𝑣 và 𝜔.

Khi có được vận tốc riêng cho từng bánh xe, đề tài sử dụng thuật toán PID để xuất xung PWM chính xác cho Driver , với mục tiêu chính là hạn chế tối thiểu sai số xác lập và độ vọt lố với thời gian xác lập ngắn.

|  |  |  |  |
| --- | --- | --- | --- |
| Tín hiệu đặt | 𝑟(𝑡) | 𝑣𝑟ight hoặc 𝑣𝑙eft | m/s |
| Tín hiệu hồi tiếp | 𝑣(𝑡) | 𝑣̅right hoặc 𝑣̅left | m/s |
| Tín hiệu điều khiển | 𝑢(𝑡) | Chu kỳ điều khiển  xung PWM | % |

Bảng . Bảng thông tín tín hiệu bộ điều khiển PID



Hình . Sơ đồ khối điều khiển vận tốc 4 bánh xe

Ta có số xung khi quay một vòng của mỗi kênh encoder là 11x30 = 330 xung khi đọc encoder ở mode x4 thì encoder sẽ xuất được 1320 xung/vòng.

Giả sử chu kì đọc encoder là Tenc(s), số xung đọc được trong lần lấy mẫu là p(xung/Tenc) và bán kính của bánh xe là R(m) ta được công thức tính vận tốc của bánh xe v(m/s) như sau:



Tốc độ di chuyển trung bình của robot khoảng 20cm/s và bán kính bánh xe là 6.25cm, vì vậy nhóm lựa chọn chu kì đọc encoder là Tenc = 100 ms để trong một chu kì lấy mẫu thì số xung đọc được khoảng 67 xung, nhờ đó gai nhiễu của encoder sẽ không làm thay đổi quá lớn giá trị vận tốc của motor khi tính toán.

Dựa vào sự ảnh hưởng của các thông số lên bộ điều khiển ta có các bước tinh chỉnh thông số như sau:

Bước 1: Đặt Ki, Kd ban đầu bằng 0. Tăng Kp đến khi POT đạt 10%.

Bước 2: Tăng Ki đến khi exl = 0.

Bước 3: Tăng Kd đến khi POT đạt nhỏ nhất.

|  |  |  |
| --- | --- | --- |
| **Chu kỳ điều khiển PID** | 𝑇 | 100 ms |
| **Chu kỳ đọc encoder** | 𝑇𝑒𝑛𝑐𝑜𝑑𝑒𝑟 | 100 ms |
| **Tần số xung PWM** | 𝑇𝑃𝑊𝑀 | 25 khz |
| **Hệ số khâu P** | Kp | 0.2 |
| **Hệ số khâu I** | Ki | 0.7 |
| **Hệ số khâu D** | Kd | 0 |

Hình . Bảng giá trị thông số điều khiển tốc độ motor

**3.5.2. Đọc adc và xây dựng module scheduler**

Raspberry Pico hỗ trợ đọc adc 12-bit từ 0-3.3V, giá trị adc đọc được từ vi điều khiển là n, điện áp tham chiếu là Vref (V), điện áp của diode zenner là Vzenner(V) và hệ số khuếch đại là K ta có công thức tính điện áp của pin như sau:

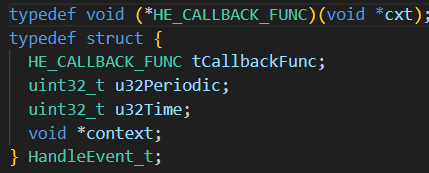


Với thiết kế mạch adc được trình bày ở mục 3.3.2 ta có bảng thông số như sau:

|  |  |  |
| --- | --- | --- |
| **Điện áp tham chiếu** | Vref | 3.27 V |
| **Hệ số khuếch đại** | K | 2 |
| **Điện áp diode Zenner** | Vzenner | 11.8 V |

Bảng . Bảng giá trị thông số tính điện áp của pin

Module scheduler là module để các function đăng kí sự kiện thực hiện với chu kì T. Cấu trúc của scheduler gồm pointer function, chu kì thực hiện sự kiện và các thông số của sự kiện được khai báo như sau:



Bảng . Struct của module scheduler

**3.5.3. Giao tiếp uROS**

**3.6. Xây dựng Software**

**CHƯƠNG 4. KẾT QUẢ THỰC HIỆN**

**4.1. Kết quả xây dựng mô hình robot**

**4.1.1. Mô hình robot thực tế**

**4.1.2. Thông số mô hình và nhận xét**

**4.2. Kết quả điều khiển tốc độ động cơ có tải**

**4.3. Kết quả đọc adc**

**CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

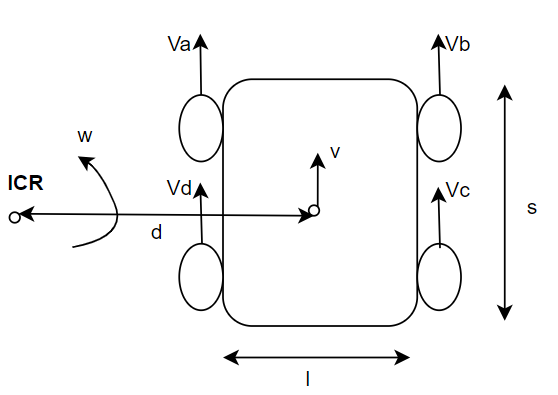
**5.1. Kết luận**

**5.2. Hướng phát triển**

**TÀI LIỆU THAM KHẢO**

[1] Lin Xu, Jiaqiang Du, Baoye Song & Maoyong Cao, “A combined backstepping and fractional-order PID controller to trajectory tracking of mobile robots”, Systems Science & Control Engineering, 2022.

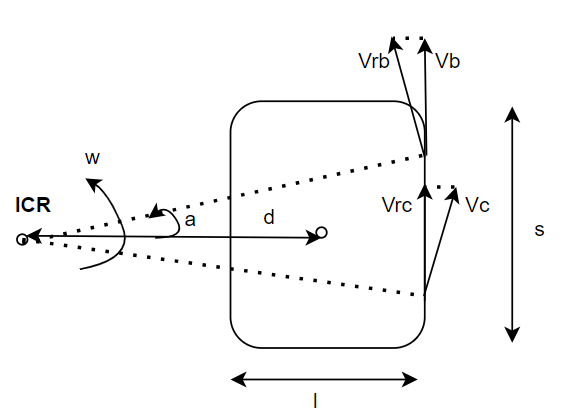
# **PHỤ LỤC**

**Differential Drive**

Hình . Mô hình Differential Drive

Ta có:





Do tính đối xứng của khung robot nên ta có:



Ta có:



Từ các phương trình trên ta được:



Chứng minh tương tự cho trường hợp 2 bánh xe bên trái ta được:



1. Nguồn: Tài liệu thí nghiệm môn Cơ sở Điều khiển tự động, khoa Điện-Điện tử, trường Đại học Bách Khoa Thành phố Hồ Chí Minh. [↑](#footnote-ref-1)