ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Robot Vacuum

Team #9

TIANYU ZHANG
(tianyu7@illinois.edu)
LONG CHANG
(longc2@illinois.edu)
ZHEYI HANG
(zheyih2@illinois.edu)
KAILONG JIN
(kailong3@illinois.edu)

TA: Tielong Cai
Mentor: Meng Zhang

May 23, 2023

Abstract

Household robot vacuums have revolutionized floor cleaning by providing convenient and automated solutions. However, these devices are not flawless and encounter various issues. In large houses with multiple floors, robots falling from high places or getting stuck on small obstacles can be problematic. To address these challenges, our proposed solution enhances robot vacuum functionality. By incorporating anti-falling wheels, a suspension system, and an efficient path-finding algorithm, our robot vacuum can prevent falls and pass over low obstacles efficiently. Additionally, we have implemented lift interaction capabilities, enabling the robot to clean multiple floors seamlessly.

With these enhancements, our robot vacuum offers improved safety, efficiency, and adaptability. It ensures stability, avoids falls, and follows optimal cleaning paths. The lift interaction feature allows it to clean multiple floors effectively. Our solution aims to provide a comprehensive and reliable cleaning experience for users in multi-floor households, eliminating common issues encountered by traditional robot vacuums.

Contents

1 Introduction

1.1 Purpose

Robot vacuums have become increasingly popular due to technological advancements, evolving from basic models to intelligent devices with a range of features. However, there are still several challenges that need to be addressed. Firstly, there is a risk of the robot vacuum falling from high surfaces and getting damaged. Secondly, it can get stuck on uneven terrain, which hampers its cleaning efficiency. Additionally, the robot vacuum lacks the ability to clean multiple floors without manual intervention. Lastly, the existing path-finding algorithm can be further optimized to improve performance.

To overcome these issues, we propose an advanced robot vacuum solution that integrates four subsystems into the existing design. The first subsystem is an anti-fall steering system that uses a mechanical structure to prevent falls near stairs. The second subsystem, known as the low obstacle passing subsystem, enhances the suspension structure to adapt to complex and uneven surfaces. It automatically raises the vacuum's chassis when the laser distance sensor detects low obstacles. The third subsystem, the elevator interaction subsystem, enables the robot vacuum to call and use elevators, simplifying multi-floor cleaning. Lastly, we employ 2D vision to find a better cleaning path, enhancing efficiency and forming an effective path-finding subsystem.

By integrating these four subsystems, we enhance the reliability and efficiency of the robot vacuum while reducing the workload for users. This solution ensures safety, improved adaptability to different surfaces, and efficient cleaning across multiple floors. It empowers users with a more reliable and efficient cleaning experience.

1.2 Functionality

1. The robot vacuum should possess the capability to traverse thresholds or obstacles up to a height of 4 cm seamlessly, ensuring uninterrupted cleaning.
2. The signal transmission between the robot vacuum and the elevator should be robust, guaranteeing reliable communication for the vacuum to request elevator services after completing its cleaning tasks on one floor, allowing it to move autonomously to the next floor for continued operation.
3. The adopted algorithm should effectively handle scenarios involving low obstacles, ensuring a minimum room coverage of 90 percent during the cleaning process.

The first requirement emphasizes the functionality of the low obstacle passing subsystem. This subsystem enables the robot vacuum to smoothly navigate over low obstacles such as carpets, exposed wires, and room thresholds. In our design, we incorporate an inclined plane that allows the robot vacuum to climb over these obstacles effortlessly.

The second requirement highlights the significance of the elevator interaction subsystem. This subsystem enables the robot vacuum to perform multi-floor cleaning autonomously.

Our objective is to ensure a fully automated process without the need for human intervention.

The final requirement pertains to the effective path-finding subsystem. This subsystem ensures that the robot vacuum follows a comprehensive cleaning path based on a 2D map of the room. We have made changes to the design by eliminating the marking of low obstacles to the map due to hardware constraints, which is elaborated upon in Section 2.5.5 of our design.

1.3 Subsystem Overview

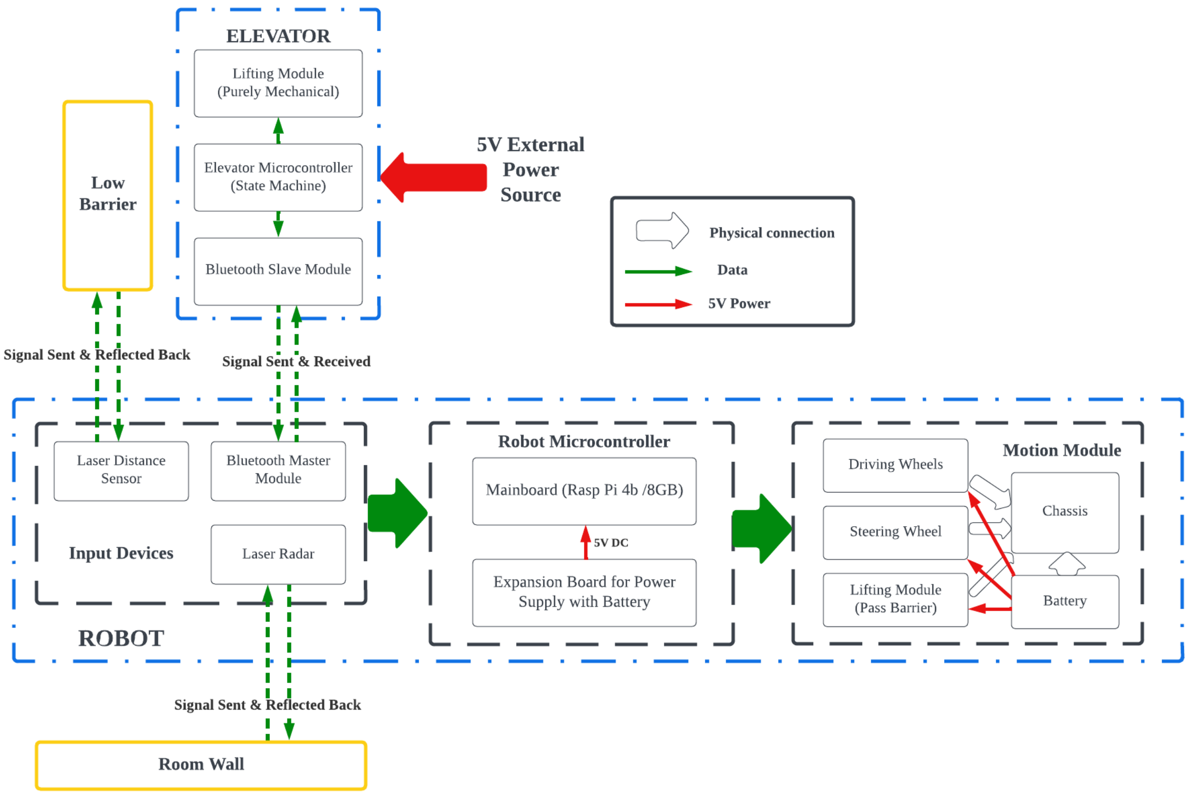


Figure 1: Overall Block Diagram

Figure 1 shows the overall block diagram of our robot vacuum. The operation of the robot vacuum involves three primary steps. Firstly, it receives input information from various devices. For instance, it obtains 2D environmental data from the laser radar (Lidar) and detects low obstacles using the laser distance sensor (TOF sensor). Secondly, the robot vacuum processes this information using its microcontroller. This includes tasks such as calling the elevator, constructing 2D maps, and implementing obstacle avoidance algorithms. Finally, the robot vacuum controls its wheels and suspension system to ensure appropriate motion and fulfill the requirements of the subsystems. These requirements encompass obstacle traversal, fall prevention, elevator interaction, and adherence to the generated cleaning path. In contrast, the elevator is relatively simple and dedicated solely to the elevator interaction subsystem. When the robot vacuum enters the elevator, the elevator automatically ascends and transports the robot vacuum to the next floor.

2 Design

2.1 Cleaning Subsystem

As a robot vacuum, it should have a cleaning capacity. This project designs a theoretically feasible cleaning system including an edge brush, a dust collecting box and a fan. The dust collecting box is located in the center of the robot vacuum, and the vacuum fan is located behind the dust collecting box, forming a front-to-back air duct. A pair of three-leaf brushes are arranged in front of the robot vacuum chassis to clean the ground.

2.1.1 Design of Edge Brush

Figure 2 shows the design diagram where the robot vacuum has two edge brushes driven with two rubber bands and one motor.

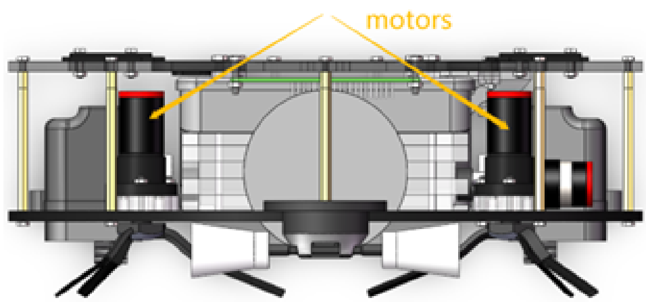


Figure 2: 3D diagram of the side brush driving system

2.1.2 Design of Air Duct and Dust Collector

When the robot begins to work, the dust port and brushes begin to rotate, thereby drawing dust and debris from the floor into the robot's dust container or bag. The dust container is located at the center of the robot and can be easily removed and cleaned. Figure 3 shows the effect drawing of the whole dust collecting system, which mainly includes a trapezoidal air intake channel, dust collecting box, and a fan bin. The dust collection box and the air intake channel are fixed on the bottom plate through the card slot for easy disassembly and cleaning. It is also noted that the air intake channel is formed by two side panels, four beams, and an upper beam.

2.1.3 Design of Fan

Since our design focuses on path planning and multi-terrain adaptability, we do not focus on the cleaning effect. This robot vacuum is able to suck up Styrofoam balls. As shown in Figure 4, we choose a common pipeline fan on the market to provide suction, with a power of 6 W and 12 V voltage supply. Because our steering wheels also use 12 V power, they can share one power supply and avoid the hassle of a voltage-lifting module. This project uses 3D-printed parts to assemble the purchased fan as part of the air duct.

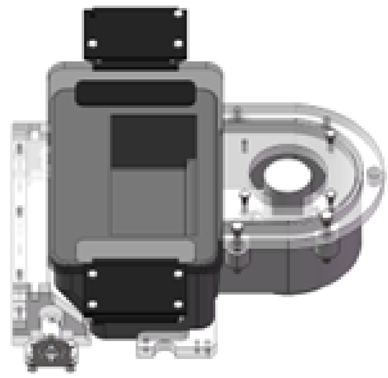


Figure 3: 3D diagram of air duct and dust collector

(a) Complete appearance of the fan (b) Disassembled picture of the fan

Figure 4: Fan Pictures

2.1.4 Design Alternatives

1. The driving mode of the edge brush is changed from two motors respectively to one motor and two rubber bands, which simplifies the structure and reduces energy consumption.
2. The fan position is changed from vertical to horizontal, which reduces the volume and increases the suction force.

2.2 Anti-fall Steering Subsystem

The anti-fall unit allows the robot to automatically turn when approaching the edge of stairs, it is a critical safety feature that prevents it from falling and causing damage.

2.2.1 Mechanical Part

As shown in Figure 5, the mechanical structure consists of a steering wheel (5a) that is positioned perpendicular to the forward direction and two tapered front wheels (5b). When the robot vacuum moves normally, the steering wheel does not contact with the ground. When the robot moves to the edge of the platform and the conical front wheel protrudes out of the platform, its chassis height decreases, causing the steering wheel to contact the ground, so as to realize steering and avoid falling off the platform.

(a) Steering wheel 3D model (b) Front wheel 3D model

Figure 5: Anti-fall subsystem mechanical structure 3D models

2.2.2 Sensor Part

The sensor structure mainly relies on a laser distance sensor (TOF sensor). In fact, this sensor is also used to detect obstacles in front of the target. The sensor is placed on the head of the upper plate of the vacuum robot (see Figure 6). If the ground is more than 20 cm away from the sensor, it means the nose has leaned out of the platform and is about to fall. At this point, the sweeping robot stands back 5 cm and turns 90 degrees in place to avoid falling. When the sensor fails or is too late to respond, the steering wheel comes into play and forces the steering to avoid a fall.

Figure 6: Sensor placed on robot head (circled in green)

2.2.3 Design Alternatives

The feature was firstly designed to be purely mechanical, requiring no maintenance or software updates to avoid the possibility of electronic failure. But in the actual implementation process, we decided to use the model of "mechanical structure + sensor". This is because the width of the rear wheel of the sweeping robot is obviously greater than the width of the front wheel, so when the angle between the direction of the sweeping robot and the platform boundary is less than or equal to 12 degrees, it can not continue to complete the steering. When the robot vacuum gets close to the edge of the platform by a small degree, the rear wheels fall before the front wheels, and the steering fails. The critical angle is calculated as the angle between the connection between the front wheel and the rear wheel and the forward direction. The full calculation is shown in Appendix C.

2.3 Low Obstacle Passing Subsystem

The low obstacle passing subsystem of our robot vacuum is designed to allow it to navigate and pass over low obstacles such as thresholds or small bumps on the floor. This subsystem is made up of several key components including a TOF sensor, a circuit board, several electric actuators, wheels, and 3D-printed fixings.

2.3.1 Electrical Part

To satisfy our expectations for the low obstacle passing subsystem, we design a PCB board to achieve all the following specifications:

1. The measured distance data from the TOF sensor is real-time transferred to the mainboard of the robot vacuum through USB.
2. If the TOF sensor detects that there exists a change in the distance towards the ground larger than a threshold, the PCB board outputs signals to control the electric linear actuators to stretch out to their limits.
3. After reaching their limits, the actuators remain for a fixed time and then the PCB board outputs signals to control actuators to return back to their original statuses.

To meet these specifications, we design the PCB board. The RESET module introduces a button to generate a reset signal for the STM32 chip to realize the function of reset and the 5 V-3.3 V module is the 5 V DC to 3.3 V DC transformer circuit. The lower-left module is the Micro USB serial port circuit. To be specific, the CH340N chip is used to realize signal conversion between USB and TTL. In the upper-middle part of the figure, one 12 V DC to 5 V DC transformer circuit together with two electric linear actuator drive circuits are shown. Here 78L05 chip is used to realize the voltage step-down while L298N chips are chosen to control the actuators. Specifically, P11, P12, P19, and P20 are four actuator drive ports but only three of them are used. In the lower-middle part of the figure, the STM32F103C8T6 chip together with its minimum system circuit is shown. The two connected circuits on their right are used to generate clock signals. The SWD module is used to download programs to the STM32 chip while the BOOT module determines which mode the chip boots in. In our design, ports BOOT0 and BOOT1 are always connected to the ground since the STM32 chip is designed to always boot from the main flash memory. Furthermore, the PCB board schematic diagram, printed circuit diagram and the front of the PCB board after soldering are listed in Appendix B.

2.3.2 Mechanical Part

The mechanical part of our low obstacle passing subsystem consists of electric actuators and wheels. The electric actuators are responsible for lifting the main body of the robot vacuum, while the wheels are used to maintain balance and move the robot forward.

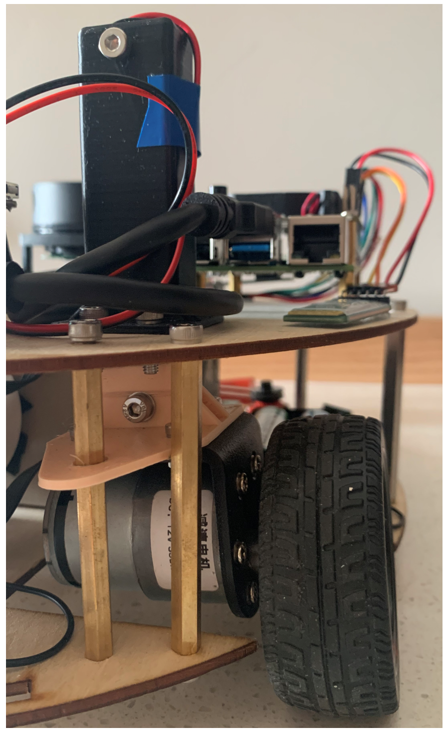


Figure 7: Subsystem mechanical assemble diagram

Figure 7 shows how we assemble the mechanical part of the robot. The electric actuators (the upper part with the cover) are connected to the main body of the robot vacuum and are controlled by the circuit board. The wheels are mounted on 3D-printed fixings that are designed to hold them securely in place. The fixings are designed to securely attach the electric actuators to the frame of the robot vacuum, while also providing enough clearance for the wheels to move up and down. Refer to Appendix B for the 3D models of the fixings.

2.3.3 Design Alternatives

Electric actuators are preferred over rack and pinion systems in the low obstacle passing subsystem due to their compact size, precise control, and versatility. Electric actuators, such as linear motors or solenoids, can efficiently convert electrical energy into linear mechanical motion, allowing the robot vacuum to raise or lower its chassis when encountering low obstacles. Unlike rack and pinion systems, which require additional mechanical components and space, electric actuators offer a more streamlined and integrated solution. Additionally, electric actuators provide finer control and adjustability, allowing for optimized obstacle detection and navigation. Overall, electric actuators offer a reliable and efficient mechanism for the low obstacle passing subsystem, enhancing the robot vacuum's ability to smoothly maneuver around obstacles.

2.4 Elevator Interaction Subsystem

Nowadays, there are robots in the hotel that can deliver takeaways or service products across multiple floors, all of which take elevators by themselves. We would like to extend this functionality to our robot vacuum, which is useful when there is a high stair that separates the room into two parts, or there are multiple floors to be cleaned. By letting the robot vacuum automatically interacts with the elevator or left, the robot can do multi-level cleaning in one launch. A more detailed block diagram is shown in Figure 8.

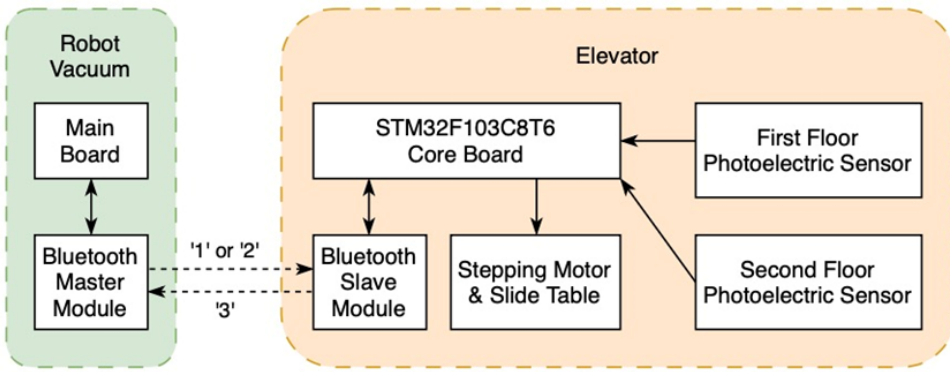


Figure 8: Elevator Interaction Subsystem Detailed Block Diagram

2.4.1 Electrical Part

The signal transmission between the robot vacuum and the elevator is realized by Bluetooth modules. The elevator control circuit is based on one STM32F103C8T6 core board. When the elevator is supposed to move between the two floors, the elevator core board outputs corresponding forward or reverse rotation signals to the stepping motor which further drives the slide table. On the heights of both the first and the second floor, two photoelectric sensors are placed to detect whether the elevator has exactly reached the corresponding floor or not and this information tells the core board when to stop the stepping motor. To satisfy our expectations for this elevator interaction subsystem, our design eventually meets all the following specifications.

1. The robot vacuum can automatically call the elevator to move to its current floor.
2. Every time the robot vacuum sends out a character '1' or '2', the elevator eventually reaches the corresponding first or second floor.
3. After the elevator reaches the desired floor, it sends out a character '3' in response.

2.4.2 Mechanical Part

The elevator is simulated by a small lifting device which is built mainly based on a sliding table controlled by a stepping motor. When the elevator is supposed to move between the two floors, the elevator core board outputs corresponding forward or reverse rotation signals to the stepping motor which further drives the slide table.

Meanwhile, the electric slide table is supplemented with several aluminum profiles to construct the lifting device. The 12 aluminum profiles are arranged in a square shape and connected with the corner pieces, forming the frame of the elevator. Figure 9 shows the diagrams of the designed 3D-printed corner pieces. These corner pieces are made of ABS plastic, which is lightweight and has good mechanical properties. They serve as connectors between the aluminum profiles, ensuring that the frame remains in place.

(a) Corner pieces 3D diagram 1 (b) Corner pieces 3D diagram 2

Figure 9: Elevator corner pieces 3D diagrams

2.4.3 Design Alternatives

In the final design, we decide to use Bluetooth modules instead of infrared devices which are in our preliminary design to realize signal transmission between the robot vacuum and the elevator because the use of Bluetooth modules is more reliable and could better simulate the realistic application scenarios.

2.5 Effective Path-Finding Subsystem

This subsystem is the central processing unit of our robot vacuum. Driven by Robot Operating System (ROS) [1], this subsystem controls the robot's motion behavior and activates other subsystems. By delivering control signals from the processor, our robot vacuum can patrol around the room to build maps, determine the most efficient cleaning route, pass low obstacles, interact with the elevator, etc. Figure 10 illustrates the subsystem block diagram.

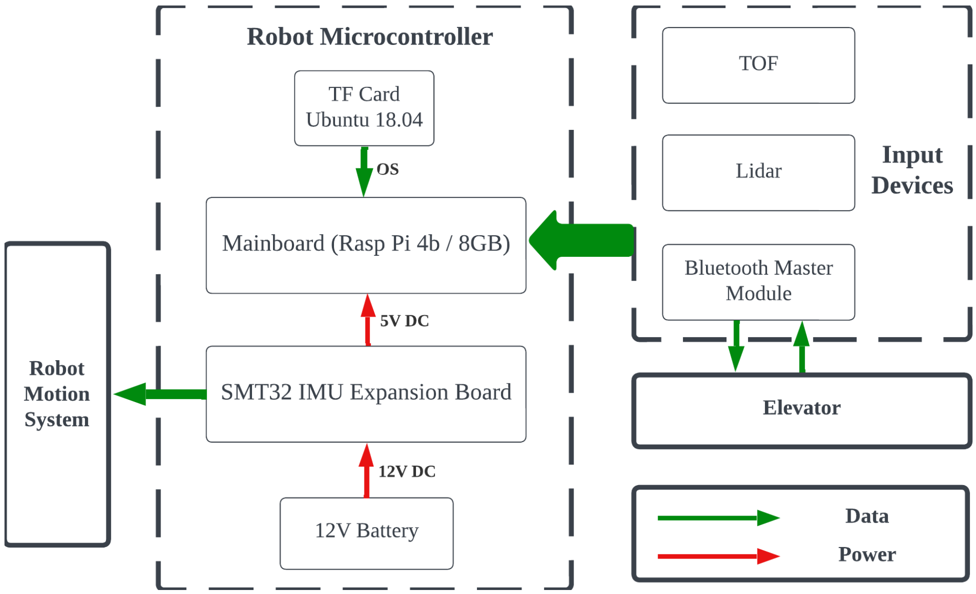


Figure 10: Block diagram of effective path-finding subsystem

2.5.1 Hardware Layout and Software Specifics

Figure 11 illustrates the hardware layout of this subsystem. In the top middle, we choose a Rasp Pi 4b (8GB RAM) [2] as our mainboard. The left part is the laser radar (Lidar) with type RPLIDAR A1M8 [3], which is used for map building and robot pose detection and calibration. There are two boards shown on the right (there is one on top of the other). The bottom one is the USB hub for more USB port devices such as tty-USB and Lidar. The top one is one off-the-shelf robot expansion board [4] with the following functionalities. The hardware physical diagrams are listed in Appendix B.

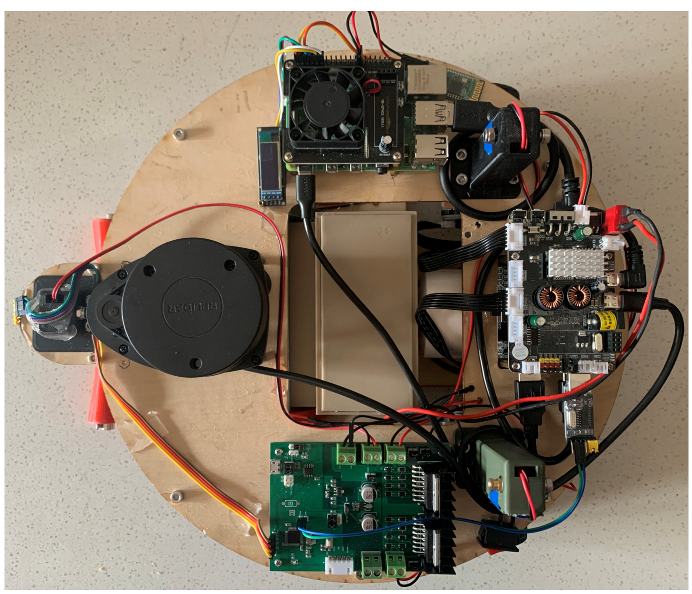


Figure 11: Hardware layout of the robot vacuum

- Provide stable 5 V and 12 V power sources.
- Control the motors with SMT32 [5].
- Use MPU9250 [6] for robot motion tracking and orientation sensing.

For software, we run Ubuntu 18.04 operating system on the Rasp Pi with ROS melodic version installed. We use ROS package [7] to run the Lidar. We use Rviz (ROS 3D Robot Visualizer Platform) [8] for any visualization work in our project.

2.5.2 Subsystem Workflow

The complete workflow of our robot vacuum consists of two phases:

- **Patrol Phase:** The robot vacuum automatically patrols the room while building the 2D map simultaneously. When patrolling, it tries to avoid approaching barriers like walls. When the robot vacuum goes back to the original point, it stops and saves the map.
- **Clean Phase:** The robot generates the cleaning route and starts following that route, trying to cover the ground space as much as possible.

2.5.3 Patrol Phase

The robot vacuum starts the general workflow by patrolling the room to capture the environment data with its input devices. When patrolling, the Lidar continuously scans the shape of the room. After collecting the environment data, the mainboard builds the 2D map of the room with SLAM *Gmapping* algorithm [9].

The SLAM *Gmapping* algorithm is used in robotics to create a map of an unknown environment while simultaneously estimating the robot's location within that environment by processing Lidar range measurements to generate a grid-based map representation. This algorithm can handle complex and dynamic environments and robustness against Lidar noise. Also, it does not require perfect Lidar scanning performance, which is interesting to us as RPLIDAR A1M8 is the cheapest Lidar (around ¥500) we can find. An example 2D map with this algorithm is shown in Figure 12.

In this phase, our robot vacuum also uses the Lidar scanned data to avoid approaching barriers, which is shown in Algorithm 1. After initialization (Line 1-6), the robot checks the Lidar scanned data set to identify the relative position of the barrier (Line 8-20). Then, the robot sends commands to the motor to avoid the barrier (Line 21-35). We update the data at the end and perform the next check (Line 36-37) until the robot moves back to the starting point (Line 7).

2.5.4 Clean Phase

With the built map in the Patrol Phase, the robot vacuum builds the cleaning route using a full coverage path planner [10] with Backtracking Spiral Algorithm (BSA) described in [11]. BSA is a path-finding algorithm that systematically explores neighboring cells in a

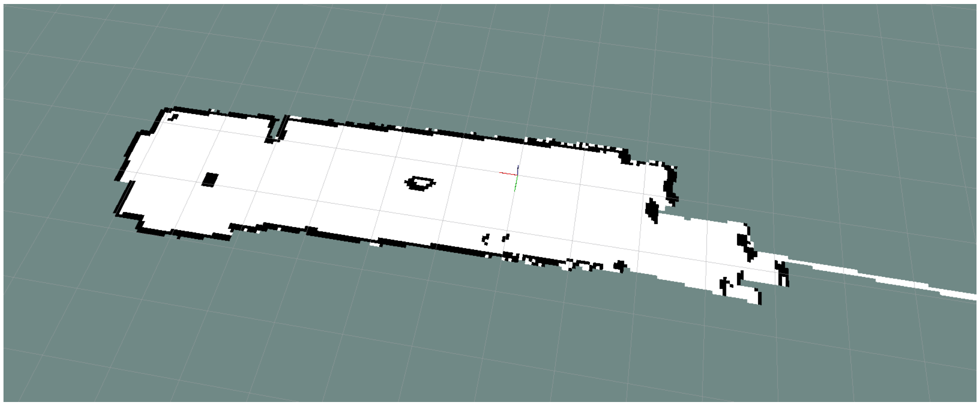


Figure 12: SLAM *Gmapping* algorithm map illustration

spiral pattern starting from the initial position in a grid. It also uses a stack to track visited cells and backtracks when necessary (DFS based). With the given radius of the robot vacuum as the input, the algorithm tries to cover the ground base as much as possible. As shown in Figure 13, the green lines are the generated path. Note that the example map is not entirely enclosed, so the paths are expanded to the whole screen.

We also add trajectory following [12] to this algorithm. When the robot vacuum starts cleaning, there is a fake point moving along the generated route, and the robot vacuum continuously follows that point behind. In this case, our robot vacuum cleans the room following the desired route, which satisfies the path-following functionality. As shown in Figure 13, the blue lines are the trajectory of the robot vacuum. Note that we expect the simulation (13a) to have a better performance than the demonstration (13b).

2.5.5 Design Alternatives

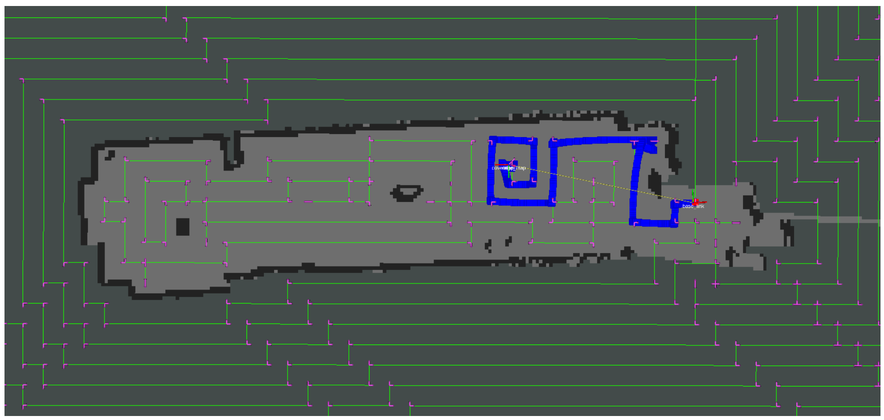
- Previously, we used the expansion board with two 3.7 V Lithium batteries [13] as the power source. However, the total power of all compartments on the robot vacuum is about 50 W, so two 3.7 V batteries are not enough for it to run for several hours. So we change the source battery to a 6000 mAh, 12 V Lithium battery, connecting to the SMT32 IMU robot expansion board [4] for stable power supplies.
- We deprecated the initial design to mark the position of the low obstacles on the 2D map. This is due to Lidar hardware properties. To have a stable 2D map, the Lidar cannot move along the z-axis. However, when we pass the obstacles, the Lidar rises along with the chassis, which confuses the Lidar about its exact position. As a result, the Lidar keeps re-sampling and the built map is be no longer usable. We cannot deal with this problem in our current design, so we discard that system property.

Algorithm 1: Algorithm for Laser Avoidance

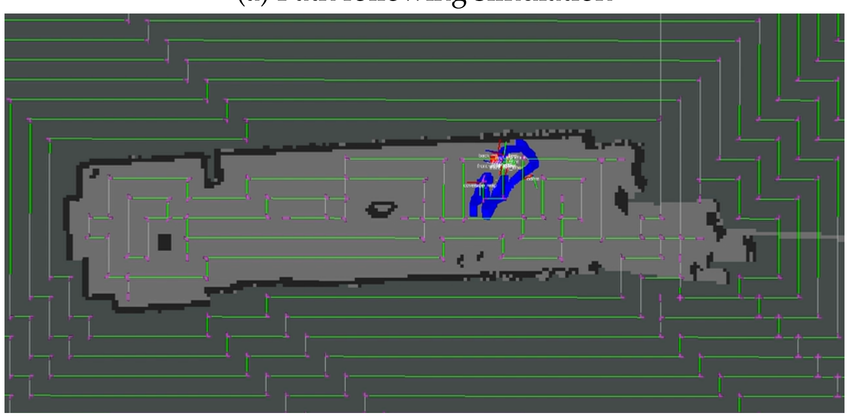
Data: T: Distance Tolerance, D: Lidar Response Distance, θ: Lidar Response Angle, (x_0, y_0): Starting Position

Result: cmd: Command Controlling the Robot

```
frontWarning ← 0;
rightWarning ← 0;
leftWarning ← 0;
cmd ← None;
(x, y) ← getCurrentPosition();
S ← getLidarScannedDataSet();
while sqrt((x − x0)^2 + (y − y0)^2) ≥ T do
    foreach s in S do
        if s.distance < D then
            if |s.angle| < θ then
                frontWarning ← frontWarning + 1;
            end
            else if s.angle > θ then
                rightWarning ← rightWarning + 1;
            end
            else if s.angle < −θ then
                leftWarning ← leftWarning + 1;
            end
        end
    end
    if frontWarning > 10 then
        cmd ← "goback";
        continue;
    end
    else if rightWarning > 10 then
        cmd ← "turn left";
        continue;
    end
    else if leftWarning > 10 then
        cmd ← "turn right";
        continue;
    end
    else
        cmd ← "go straight";
    end
    (x, y) ← getCurrentPosition();
    S ← getLidarScannedDataSet();
end
```



(a) Path following simulation



(b) Path following demonstration

Figure 13: Path following diagrams in Rviz

3 Requirements and Verification

3.1 Cleaning Subsystem

3.1.1 Requirements

1. The cleaning unit should have enough suction power to pick up debris from the floor.
2. The filter of the vacuum cleaner module should avoid releasing tiny particles into the indoor air.

3.1.2 Verification

We scattered many Styrofoam balls on the floor and controlled the robot vacuum to pass over with the cleaning subsystem on. We examined if we can suck the balls into the dustbin and if the balls were blown away.

3.1.3 Results

The robot vacuum could pick up all the Styrofoam balls on the floor and no Styrofoam balls were released into the air.

3.2 Anti-fall Steering Subsystem

3.2.1 Requirements

1. When the TOF sensor fails to respond, the steering wheel should turn the robot vacuum around with a weight of at least 4 kg.
2. When the TOF sensor is on and the angle between the edge and the robot vacuum ranges from 45°to 135°, the robot does not fall.

3.2.2 Verification

1. We tested with a 5 kg robot vacuum. We turned off the sensor and checked if the steering wheel helped the robot turn around.
2. We let the robot approach the edge from different angles.

3.2.3 Quantitative Results

1. The steering wheel prevented a 5 kg robot vacuum from falling when the sensor was off, but the robot speed could not be too fast (less than 10 cm/s).
2. All the angles from 15°to 165°passed the test. We proved our critical angle theorem in Appendix C.

3.3 Low Obstacle Passing Subsystem

3.3.1 Requirements

1. The measured distance data from the laser distance sensor is real-time transferred to the mainboard.
2. The robot vacuum should be able to climb over fixed thresholds with a height of at most 4 cm and a width of at least longer than the width of the robot without getting stuck.

3.3.2 Verification

1. We examined the received frequency of measured distance data from the sensor.
2. We tested the robot vacuum's ability to climb over thresholds (with an inclined plan placed) of varying heights ranging from 0.5 cm and 1 cm to 3 cm and 4 cm.

3.3.3 Quantitative Results

1. The frequency of data receipt was 0.5 s/time, which met our requirement quite well.
2. The robot vacuum could navigate the thresholds up to 4 cm height smoothly without encountering any difficulties.

3.4 Elevator Interaction Subsystem

3.4.1 Requirements

1. The robot vacuum should reach the designated position in the elevator within 60 s from the time the vacuum arrives at the front of the elevator.
2. The robot vacuum should thoroughly leave the elevator within 10 s from the time the elevator reaches the desired floor.

3.4.2 Verification

1. We measured the time taken by the robot vacuum to reach the designated position inside the elevator from the moment it arrived at the front of the elevator several times. We then calculated the average time.
2. We measured the time taken by the robot vacuum to fully exit the elevator from the moment the elevator reached the desired floor several times. We then calculated the average time.

3.4.3 Quantitative Results

1. The average time to enter the elevator was 37.68 s, which is shorter than 60 s.

2. The average time to exit the elevator was 6.24 s, which is shorter than 10 s.

3.5 Effective Path-Finding Subsystem

3.5.1 Requirements

1. The robot vacuum should autonomously patrol an unfamiliar environment and create a robust 2D map capable of accommodating dynamic changes in the surroundings.
2. When operating within a confined space, the robot vacuum should generate a path that covers a minimum of 90 percent of the floor area.
3. The robot vacuum should be able to follow the generated path with a maximum deviation of 10 cm.

3.5.2 Verification

1. A room measuring 3 m x 8 m was selected, and the robot vacuum was tasked with patrolling the area. The generated map was examined to assess its accuracy and ability to adapt to dynamic changes, such as the addition or removal of obstacles during the patrol.
2. Using the generated map, the coverage rate of the room was calculated and compared to the desired threshold of 90 percent.
3. The robot vacuum was instructed to follow the generated path and the maximum deviation from the intended path was measured.

3.5.3 Quantitative Results

1. The robot vacuum successfully generated a comprehensive and clear map (see Figure 12) of the 3 m x 8 m room. It demonstrated the capability to update the map dynamically when encountering changes in the environment.
2. The average coverage rate achieved by the robot vacuum was approximately 92 percent (see Figure 13a), surpassing the minimum requirement of 90 percent.
3. Although the robot vacuum successfully followed the generated path, it exhibited a maximum deviation of up to 50 cm (see Figure 13b). This deviation can be attributed to hardware limitations, specifically the Lidar and MPU9250 on the expansion board, which prevent precise localization and orientation sensing on the map.

4 Cost and Schedule

4.1 Cost

All units on cost are RMB (¥) by default.

4.1.1 Labor

According to a report by Chinese Education Online [14], fresh graduates in computer science earn around ¥6,800 per month (¥42.5 per hour), while electrical engineering graduates earn about ¥6,300 per month (¥39.4 per hour), and mechanical engineering graduates earn approximately ¥6,000 per month (¥37.5 per hour). Considering that our project spans 14 weeks and each person contributes 10 hours per week, we will accumulate a total of 140 hours. The labor cost breakdown is provided in Table 1.

Table 1: Labor Cost

Name	**Major**	**Hourly Salary** (¥)	**Hours Needed** (hr)	**Total Cost** (¥)	**Total Cost x 2.5** (¥)
Tianyu Zhang	ECE	42.5	140	5,950	14,875
Long Chang	ME	37.5	140	5,250	13,125
Zheyi Hang	ME	37.5	140	5,250	13,125
Kailong Jin	EE	39.4	140	5,516	13,790
Total (¥)					54,915

4.1.2 Parts

Table 2: Parts Cost

Description	Quantity	Manufacturer	Vendor	Cost/Unit (¥)	Total Cost (¥)
Aluminum Profile (20*20*400)	8	Zhejiang Bangli Hardware Products Co., Ltd.	1688	7.4	59.2
Aluminum Profile (20*20*800)	4	Zhejiang Bangli Hardware Products Co., Ltd.	1688	14.8	59.2
Motor-25GA-370 (0.45 A, 6 V)	2	Ningbo Zhenhaigewa Transmission Equipment Co., Ltd.	1688	23	46
Electric Linear Actuator (30 mm, 60 N)	3	Taihengli Technology Co.	TaoBao	149	447
TOF050C Laser Distance Sensor (50 cm)	1	Xintai Microelectronics Technology Co.	TaoBao	17	17
STM32F103C8T6 Core Board	1	Xintai Microelectronics Technology Co.	TaoBao	15	15
Slide Table & 57 Stepping Motor	1	Meike Chuandong Technology Co.	TaoBao	310	310
HC-05 Bluetooth Module	2	Xintai Microelectronics Technology Co.	TaoBao	14	28
Rasp Pi 4b/8GB	1	Raspberry Pi (Trading) Ltd.	Yahboom	1,429	1,429
Rplidar A1M8	1	Shanghai Slamtec Co., Ltd.	Yahboom	498	498
Total (¥)					2908.4

4.1.3 Grand Total

During our design, we used various machines and services in the lab. We add the estimated expenses to the grand total cost.

1. Estimated 3D printing cost: ¥50
2. Estimated laser cutting cost: ¥50
3. Estimated mechanical components cost: ¥20
4. Other possible costs: ¥20

So the grand total cost would be: $¥54,915[Labor] + ¥2908.4[Parts] + ¥140 = ¥57,963.4$

4.2 Schedule

Table 3: Project Schedule

Week	Task
03/20/2023	Finish Design Document
	Design low obstacle passing subsystem circuit part
	Study existing path-finding algorithms
	Assemble our robot vacuum
03/27/2023	Make the circuit board for the suspension system
	Make the robot vacuum work based on existing algorithms
	Design anti-fall steering system mechanical part
	Design low obstacle passing subsystem mechanical part
04/03/2023	Integrate the improved suspension system to robot vacuum
	Find how to improve path-finding algorithms
	Make the mechanical structure for the anti-fall steering system
	Make the mechanical structure for low obstacle passing subsystem
	Design the signal transmission between the robot and elevator
04/10/2023	Start programming our modified path-finding algorithm
	Integrate anti-fall steering system to robot vacuum
	Integrate the improved suspension system into the robot vacuum
	Design the circuit board for the elevator

Continued on next page

Table 3 – Project Schedule Continued

Week	**Task**
04/17/2023	Design the elevator mechanical part
	Finish programming the path-finding algorithm
04/24/2023	Implement the elevator interaction subsystem
	Implement the effective path-finding subsystem
	Implement the elevator interaction subsystem
05/01/2023	Fix bugs & Integration testing on the elevator interaction subsystem
	Fix bugs & Integration testing on the effective path-finding subsystem
	Fix bugs & Integration testing on the anti-fall steering subsystem
	Fix bugs & Integration testing on the low obstacle passing subsystem
05/08/2023	Prepare final presentation
	Working on the final presentation
05/15/2023	Finish final report & Team Evaluation

5 Conclusion

5.1 Accomplishments

Throughout our project, we achieved significant accomplishments in developing an advanced adaptive robot vacuum for complex terrain and multiple floors. We integrated four subsystems to create a reliable, efficient, and versatile vacuum.

The robot effectively cleans foam balls with its side brushes and fan. It detects platform edges and avoids falls using a combination of a laser sensor and a vertical steering wheel structure. With the laser sensor and improved suspension, our robot vacuum automatically raises its chassis for smoother cleaning on a complex terrain with a height difference of at most 4 cm. Signal transmission via Bluetooth modules to the elevator enables seamless multi-floor cleaning without manual intervention, enhancing automation and convenience. Our path-finding algorithm autonomously generates maps and plans routes. The robot autonomously patrols, generates 2D maps, and covers over 90 percent of the ground area with reasonable position deviations.

Overall, our project delivers a highly capable robot vacuum that excels in cleaning complex terrain, navigating multiple floors, and providing a user-friendly experience.

5.2 Uncertainties

Firstly, our robot vacuum features an open structure without a shell, chosen for ease of assembly during production. However, this design leaves a significant number of exposed wires and electric components, which increases the risk of breakage or short circuits.

Secondly, we have identified an issue with the width of the rear wheels, which may cause the robot vacuum to fall when approaching edges at horizontal angles less than 12 degrees (refer to Section 2.2.3 and Appendix C).

Thirdly, when activating the low-obstacle passing functionality during the patrol phase, the Lidar's movement along the z-axis with the chassis can disrupt its precise positioning and hinder accurate map building.

Lastly, it's important to note that our robot vacuum's performance can be influenced by environmental variations and unforeseen conditions during the cleaning phase. Factors such as changes in lighting, surface textures, or the presence of unexpected obstacles in different cleaning environments may affect the vacuum's performance and navigation accuracy. Addressing these uncertainties is crucial to ensure consistent and reliable operation.

5.3 Future Work

While our project has made significant advancements in developing an advanced adaptive robot vacuum for complex terrain and multiple floors, there are still areas for future work and improvements. The following actions can be considered to further enhance both the hardware and software aspects of the project.

Regarding the hardware, it is important to design and install a protective shell for the robot vacuum. This will safeguard the internal wires and electric components while also improving the overall aesthetics of the device. Additionally, efforts can be made to optimize the low obstacle passing subsystem by arranging the electric components on the printed circuit board in a more compact manner, thus reducing its size and improving efficiency. Moreover, considering the integration of better components, such as a Lidar with higher scanning capability, would significantly enhance the robot vacuum's ability to accurately locate itself within the mapped environment.

On the software side, there is ample room for improvement. Firstly, exploring the implementation of software control for the actuators and steering wheel, instead of relying solely on electrical control via the PCB board and mechanical switches, would allow for more flexible on/off control and enhance automation capabilities. Finally, utilizing advanced algorithms can further enhance the robot vacuum's performance. For instance, incorporating real-time responsiveness in the cleaning phase to adapt to changes in the environment would greatly improve efficiency. Additionally, integrating cameras and leveraging graphics recognition technologies like OpenCV could enhance the functionality and capabilities of the robot vacuum.

5.4 Ethical Considerations

Safety: Robot vacuums should prioritize safety by incorporating safety features and undergoing rigorous testing to prevent accidents and minimize risks to individuals and property.

Accessibility: Design robot vacuums with universal accessibility in mind, ensuring they are user-friendly and accessible to individuals of all abilities, aligning with the ACM Code of Ethics and Professional Conduct Clause 1.4 [15].

Collision Prevention: Despite their ability to navigate obstacles, robot vacuums may unintentionally collide with objects or people. To mitigate this risk, incorporate sensors and safety features to prevent collisions.

Entanglement Risk: Robot vacuums' brushes and mechanisms can become entangled in cords or rugs, leading to device malfunction or injury. Implement safety features like automatic shutoffs or sensors to detect entanglement and prevent accidents.

Electric Shock/Fire Hazard: Proper adherence to OSHA electrical safety standards [16], such as designing electrical systems and wiring methods in compliance with regulations, reduces the risk of electric shock or fire hazards.

In addition to these principles, a code of ethics for robot vacuums should encompass guidelines for testing, certification, accountability, and responsibility. This ensures manufacturers and users are held accountable for ethical considerations.

References

[1] R. Team. "ROS:Home." (2023), [Online]. Available: https://www.ros.org/ (visited on 05/22/2023).

[2] *Raspberry Pi 4 Computer Model B*, 4b/8GB, Raspberry Pi Trading Ltd., May 2020. [Online]. Available: https://static.raspberrypi.org/files/product-briefs/200521+Raspberry+Pi+4+Product+Brief.pdf (visited on 03/22/2023).

[3] *RPLIDAR A1-Low Cost 360 Degree Laser Range Scanner-Introduction and Datasheet*, A1M8, Rev. 3.0, Shanghai Slamtec Co., Ltd., Oct. 2020. [Online]. Available: https://bucket-download.slamtec.com/d1e428e7efbdcd65a8ea111061794fb8d4ccd3a0/LD108_SLAMTEC_rplidar_datasheet_A1M8_v3.0_en.pdf (visited on 03/22/2023).

[4] Yahboom. "Yahboom ROS robot control board with STM32F103RCT6 IMU for ROS Raspberry Pi." (2023), [Online]. Available: https://category.yahboom.net/collections/a-expansion-board/products/ros-driver-board (visited on 04/11/2023).

[5] *STM32F103xC, STM32F103xD, STM32F103xE Datasheet-Production Data*, STM32F103RCT6, Rev. 13, STMicroelectronics, Jul. 2018. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f103rc.pdf (visited on 05/22/2023).

[6] *MPU-9250 Product Specification*, MPU-9250A, Rev. 1.1, InvenSense, Jun. 2016. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf (visited on 05/22/2023).

[7] Slamtec. "RPLIDAR ROS package." (2022), [Online]. Available: https://github.com/Slamtec/rplidar_ros (visited on 04/11/2023).

[8] Ros2. "RViz." (2023), [Online]. Available: http://wiki.ros.org/rviz (visited on 04/11/2023).

[9] B. Gerkey. "slam_gmapping." (2017), [Online]. Available: https://wiki.ros.org/slam_gmapping (visited on 05/22/2023).

[10] R.-I. Q.-A. R. S. Components. "Full Coverage Path Planner (FCPP)." (2023), [Online]. Available: https://github.com/nobleo/full_coverage_path_planner (visited on 04/11/2023).

[11] E. Gonzalez, A. Pérez, and A. R. Cortés, "Bsa: A complete coverage algorithm," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 2040–2044.

[12] R.-I. Q.-A. R. S. Components. "Tracking_pid." (2023), [Online]. Available: https://github.com/nobleo/tracking_pid (visited on 05/22/2023).

[13] MakerFocus. "MakerFocus Raspberry Pi Expansion Board UPSPack Standard Power Supply -RPi V3P." (2023), [Online]. Available: https://www.makerfocus.com/products/raspberry-pi-expansion-board-ups-pack-standard-power-supply (visited on 03/22/2023).

[14] C. E. O. (EOL). "The 2022 Blue Book of Employment: Revealing the Average Monthly Income of College Graduates." (Jun. 2022), [Online]. Available: https://www.eol.cn/news/yaowen/202206/t20220614_2231533.shtml (visited on 03/21/2023).

[15] ACM. "ACM Code of Ethics and Professional Conduct." (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/08/2023).

[16] Osha.gov. "Law and Regulations." (2023), [Online]. Available: https://www.osha.gov/laws-regs (visited on 03/21/2023).

Appendix A Complete Requirement & Verification Tables

Table 4: Requirement & Verification of Cleaning Unit

Requirements	Verification	Status
The cleaning unit should have enough suction power to effectively pick up dust and other debris from the floor.	We choose a dirty room to clean and check the dust box after cleaning.	Satisfied
The filter of the vacuum cleaner module should avoid releasing dust into the indoor air.	We examine the cleaning process and see if we can feel the dust flowing into the air.	Satisfied

Table 5: Requirement & Verification of Anti-fall Steering Subsystem

Requirements	Verification	Status
The steering wheel should be able to turn the robot vacuum around with a weight of at least 4 kg.	Without the control from the microcontroller and input devices, we started with a 5 kg robot and let it move off the edge, examining whether it can successfully turn around.	Satisfied
When the angle between the edge and the robot vacuum ranges from 45°to 135°, the subsystem works normally and the robot does not fall.	We let the robot approach the edge in different directions, each time noting down the angle degree. The robot should not fall when the angle is between 45°and 135°.	Over-achieved

Table 6: Requirement & Verification of Low Obstacles Passing Subsystem

Requirements	Verification	Status
The measured distance data from the laser distance sensor is real-time transferred to the mainboard.	We examined the received frequency of measured distance data from the sensor.	Satisfied
The robot vacuum should be able to climb over **fixed thresholds** with a height of at most 4 cm and a width of at least longer than the width of the robot without getting stuck.	We tested the robot vacuum's ability to climb over thresholds of varying heights ranging from 0.5 cm and 1 cm to 3 cm and 4 cm.	Over-achieved

Table 7: Requirement & Verification of Elevator Interaction Subsystem

Requirements	Verification	Status
The robot vacuum should reach the designated position in the elevator within 60 s from the time the vacuum arrives at the front of the elevator.	We measured the time taken by the robot vacuum to reach the designated position inside the elevator from the moment it arrived at the front of the elevator several times. We then calculated the average time.	Satisfied
The robot vacuum should thoroughly leave the elevator within 10 s from the time the elevator reaches the desired floor.	We measured the time taken by the robot vacuum to fully exit the elevator from the moment the elevator reached the desired floor several times. We then calculated the average time.	Satisfied

Table 8: Requirement & Verification of Effective Path-Finding Subsystem

Requirements	Verification	Status
The robot vacuum should autonomously patrol an unfamiliar environment and create a robust 2D map capable of accommodating dynamic changes in the surroundings.	A room measuring 3 m x 8 m was selected, and the robot vacuum was tasked with patrolling the area. The generated map was examined to assess its accuracy and ability to adapt to dynamic changes, such as the addition or removal of obstacles during the patrol.	Satisfied
When operating within a confined space, the robot vacuum should generate a path that covers a minimum of 90 percent of the floor area.	Using the generated map, the coverage rate of the room was calculated and compared to the desired threshold of 90 percent.	Satisfied
The robot vacuum should be able to follow the generated path with a maximum deviation of 10 cm.	The robot vacuum was instructed to follow the generated path and the maximum deviation from the intended path was measured.	Partially satisfied

Appendix B Additional Hardware Figures

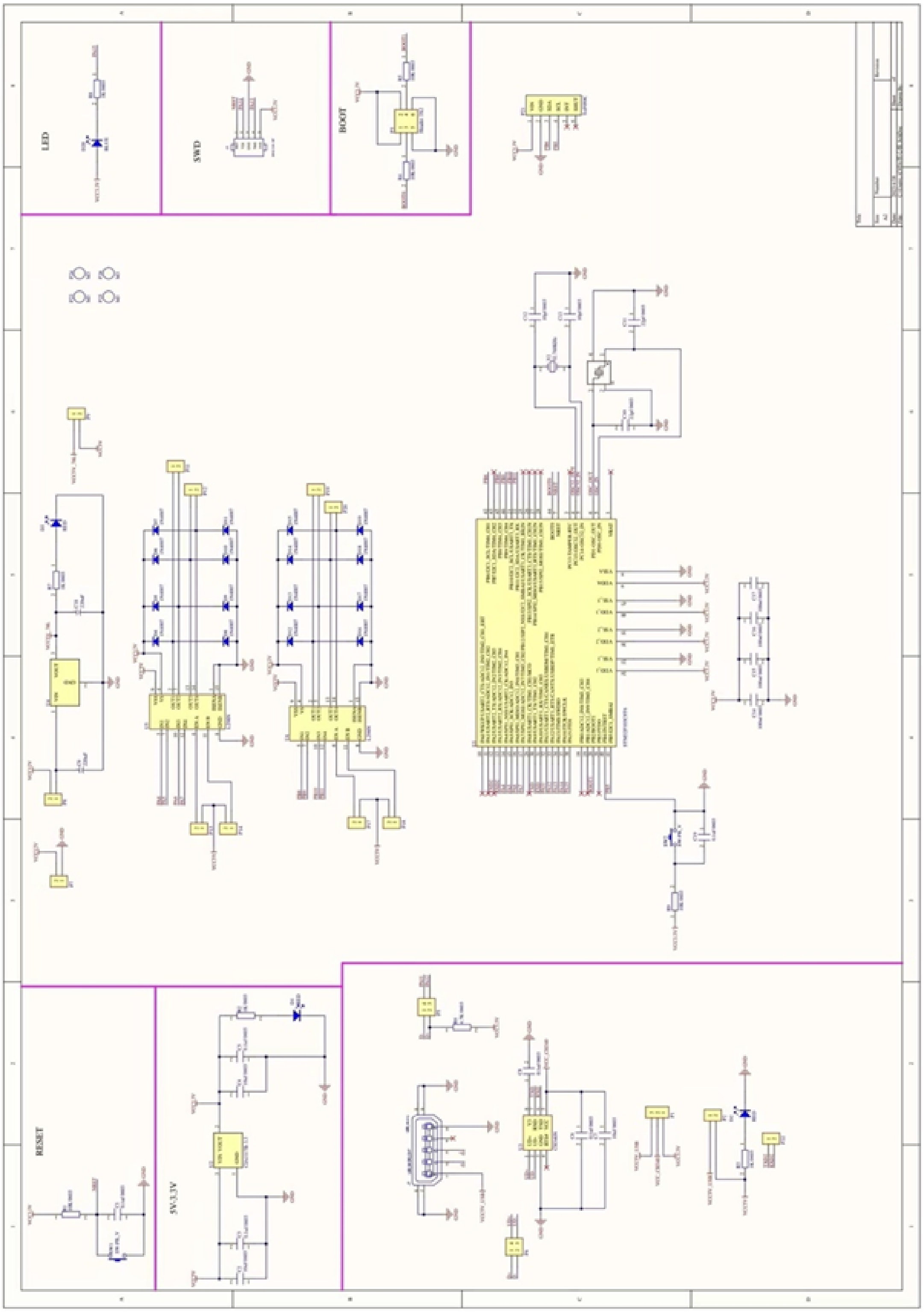
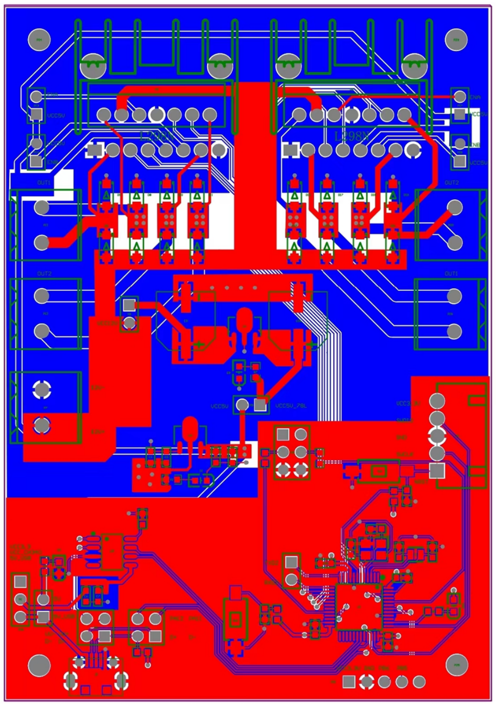
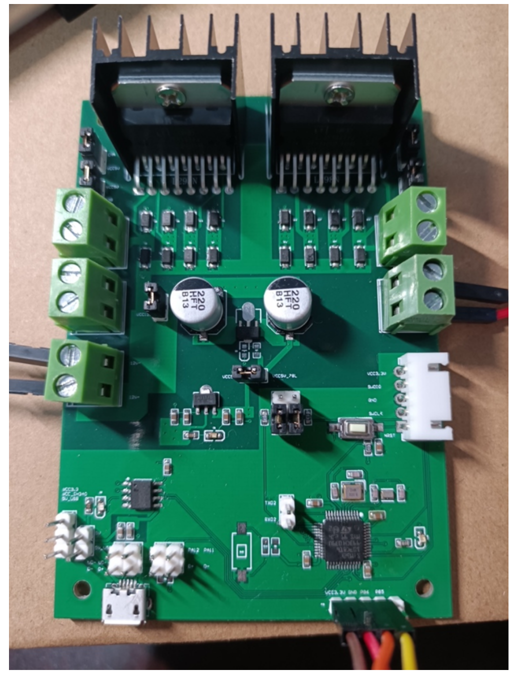


Figure 14: PCB board schematic diagram

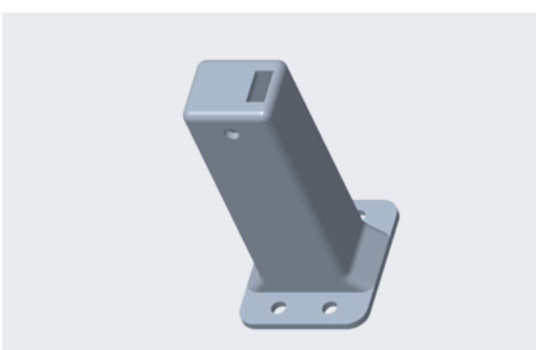


(a) PCB board printed circuit diagram

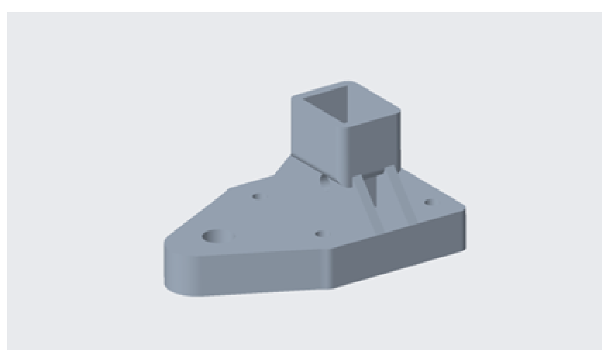


(b) Front of PCB after soldering

Figure 15: Low obstacle passing subsystem electrical diagrams

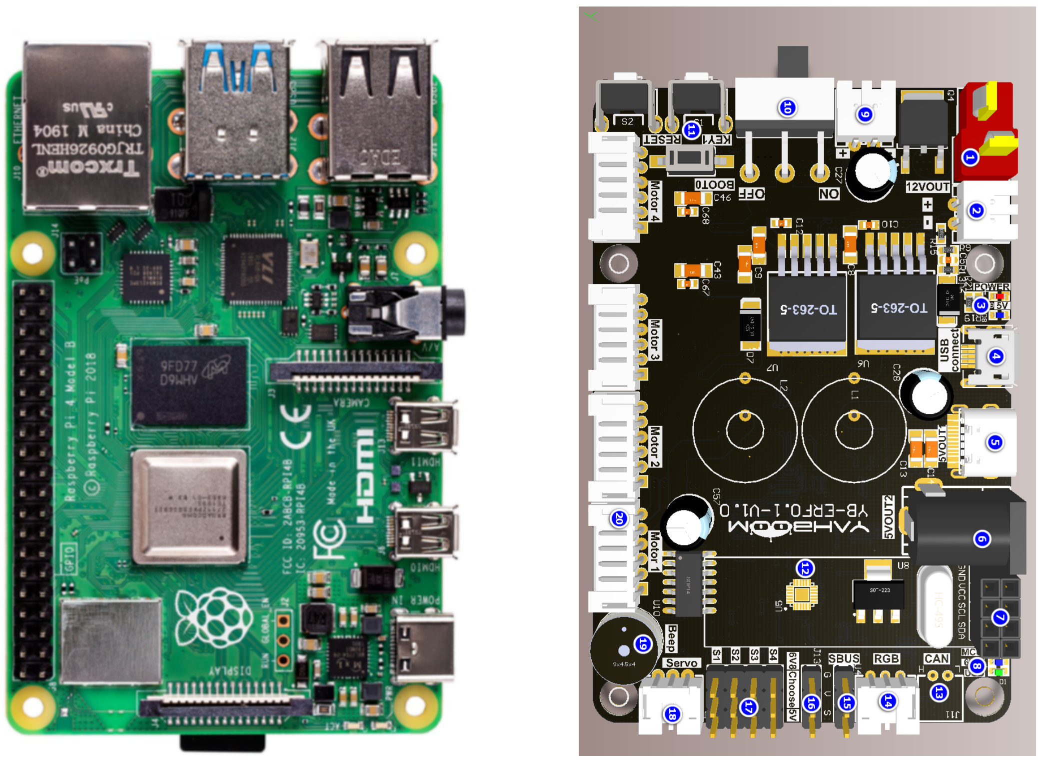


(a) Fixing upper part diagram



(b) Fixing lower part diagram

Figure 16: Low obstacle passing subsystem mechanical diagrams



(a) Rasp Pi 4b physical diagram (b) Expansion board physical diagram

Figure 17: Effective path-finding subsystem hardware physical diagrams

Appendix C Critical Angle Calculation

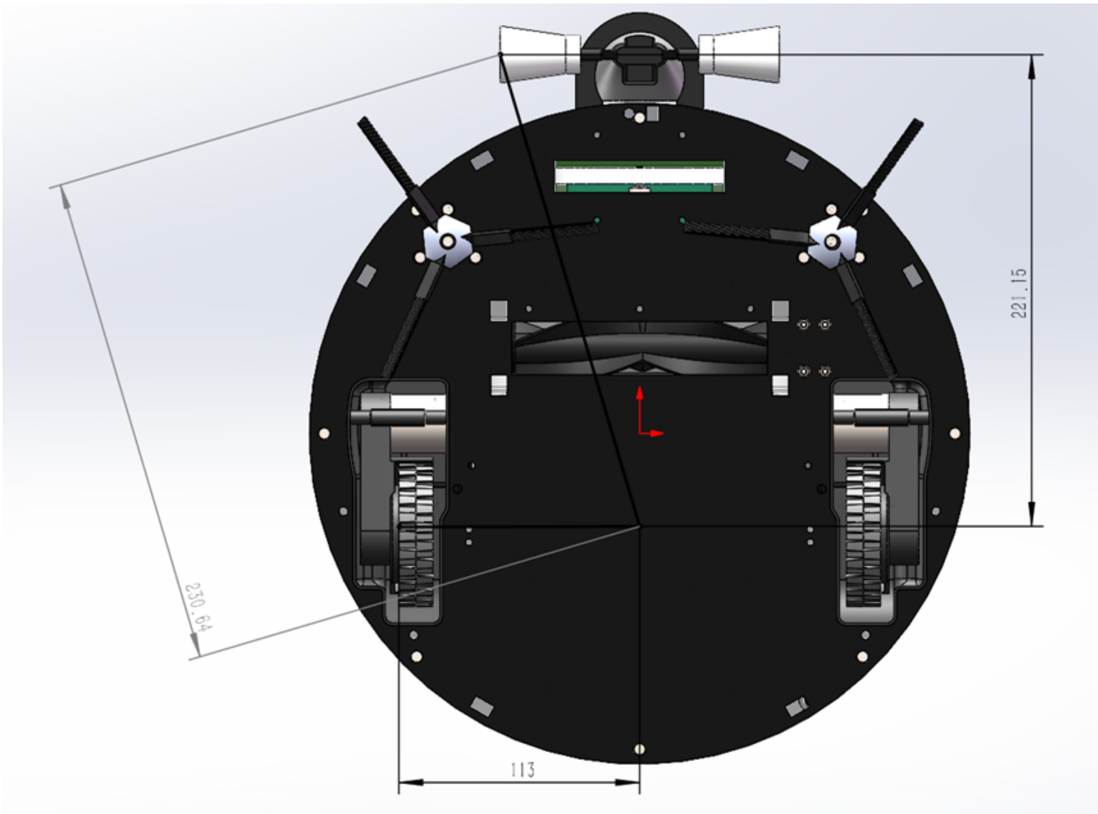


Figure 18: Anti-falling critical angle estimation diagram

As illustrated in Figure 18, the wide difference between the front and back wheels is:

$$d_1 = \sqrt{230.64^2 - 221.15^2} = 65.48mm \tag{1}$$

The horizontal distance of drive to the front wheel edge:

$$d_2 = 113 - d_1 = 47.52mm \tag{2}$$

The vertical distance of drive to the front wheel edge:

$$L_1 = 221.14mm \tag{3}$$

Straight line distance from drive to front wheel edge:

$$L_2 = \sqrt{L_1^2 + d_2^2} = 226.05mm \tag{4}$$

Critical angle:

$$\theta = \arccos\left(L_1/L_2\right) = 12.07° \tag{5}$$

Therefore, the mechanical steering function can only be used as an aid to prevent falling, and the complete steering function can be achieved with sensor control.