Automating Indoor Wall Painting

Author: Roshan Jude Cardoza

Contents

1 Problem Statement

The manual painting process is often inefficient, both in terms of time and cost, and exposes workers to significant health risks from prolonged paint fume exposure. While automation presents a safer and more efficient alternative, it introduces several key challenges. Precise surface detection is crucial to identify edges, corners, and varying geometries, ensuring accuracy and preventing overspray. Uniform paint application must be maintained to achieve consistent quality without defects, requiring precise control over the paint delivery system. Furthermore, navigating complex indoor environments with obstacles and tight spaces demands advanced manoeuvring capabilities. Addressing these challenges is critical to developing a reliable autonomous solution that improves productivity, reduces costs, and eliminates health hazards associated with traditional painting methods.

2 Solution Overview

The objective of this project is to have an autonomously working vehicle capable of selecting the walls to be painted, ensuring that the paint has been applied uniformly as per the paint used as well as provide diagnostic message in case of failures due to layouts or otherwise.

There are 4 different categories that are to be considered when designing a project such as this.

1. Autonomous functioning vehicle: For this project, a robot will be used. This vehicle will be programmed to move the paint brush as well as the bucket towards the wall that needs to be painted.
2. Motor control for Paint brush movement against walls: The paintbrush (or roller depending on the job) would be attached to a mechanism which would move the paintbrush upwards and downwards.
3. Servo Motor to Control the Dipping of Paintbrush into appropriate paint: To ensure an even coat, the paintbrush must have approximately the same amount of paint between each section of walls.
4. Number of Walls that need to be painted: For the purposes of this project, the number of walls has been hard coded to 4 each having a 90-degree offset between them.
5. Object Detection and Door Detection: When the robot encounters an object within the wall such as a light switch, the robot ignores the light switch and waits until it has been cleared to resume painting. When a door has been detected (whether it has been closed or left open), the robot waits for the wall to appear again to resume painting.

Once these 4 parameters have been identified and defined, the robot must have some logic to start the process. But first, there are some assumptions that have been made to make this project more feasible.

2.1 Assumptions

1. It is assumed that every room, irrespective of its use case and size, as 4 adjoining walls. These walls do not have any space between them, such as an entry way to a balcony.
2. It is assumed that there exists a device that has been manufactured that ensures that when the brush or roller has been dipped, it is made to slide off any excess paint to avoid smearing of the floor or walls.
3. It is assumed that there exists a mount for both the motors, the servo as well as a position within the robot to place a paint canister.
4. It is assumed that the ceiling of the room would not be painted by the robot. This assumption has been made as it would increase the total complexity of the robot and might lead to more mechanical challenges to ensure that the robot has a fixed centre of mass with added paint.
5. It is assumed that there would be human intervention whenever needed. This would occur when: an error has occurred, a power supply issue and/or damage to robot mechanisms, the room needs to be changed and/or termination of painting due to completion of all rooms.
6. It is assumed that all rooms that are to be painted are free of all obstacles. Although this is not very realistic, it is a fair assumption to design a more basic robot. This assumption can be challenged later, upon which obstacle avoidance can be added to make the project more robust.
7. It is assumed that the height of each wall would be the same and that is assumed to be 240 cm. This assumption can be made editable on further iterations of the product.
8. It is assumed that doorways do not occur at the corner of the room and have a minimum of 30cm spacing between the corner of the wall and the doorway.

2.2 Method Employed

The methodology employed is shown in the Flow-Chart shown in Figure 2.1. Besides the more obvious methodology, there are subsystems at play. Firstly, there exists an infrared sensor that is used to determine the level of paint available in the canister. When the paint level is low (determined by the size of the paintbrush), an error message is generated, and the robot is brought to a “freeze” state. Secondly, there exists a mechanism which allows the paintbrush to be dipped in the paint. This mechanism consists of a Servo motor turning the brush 90 degrees towards the ground, allowing the vertical motor to dip the brush and retrieve it and correcting itself to 0 degrees or parallel with the ground. Lastly, after completion of a single wall, a flag will be generated to keep the count of the number of walls that have been completed within the room. When this flag reaches 4, the robot has been instructed to generate a message to confirm the painting of the room. The robot is then instructed to stop and wait for further instructions.

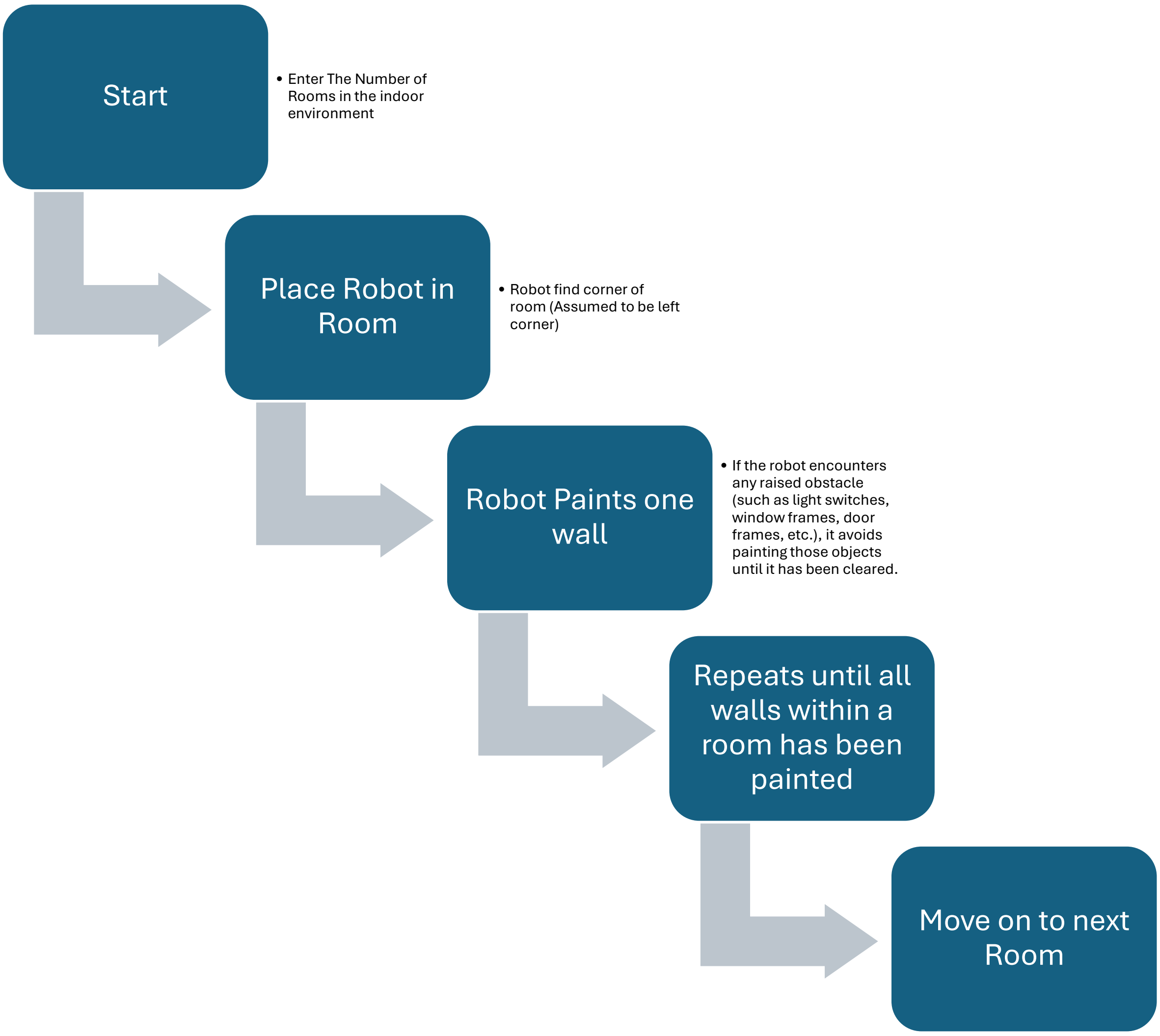

Figure 2.1 Flowchart of Autonomous Painting Robot

3 Technical Details

3.1 Electronic Hardware Components

1. Arduino Uno – Microcontroller used to control the different mechanisms
2. Ultrasonic Sensors – To measure the front, right and left side distances to find the corner of the room, ensure max height has been painted as well to move between walls and sections of walls being painted.
3. Servo Motor – To control the dipping action of the paintbrush.
4. L293D Motor Driver – To control vehicle movement.
5. Induction Motors – Controlling driving action of the robot.
6. DC Motors – Controls the vertical action of the paintbrush, which are connected using gear-drives and chain-gears (depending on mechanical specifications). Another set of motors controls the horizontal action of paintbrush to avoid obstacles on walls as well as to retract or extend the brush during the dipping process.
7. Infrared Sensors – Used to check the level of paint in the paint canister. Once paint level is deemed to be low, an error message is generated.

Name	Quantity	Component
U1	1	Arduino Uno R3
M1, M2	2	Hobby Gearmotor
M3, M4, M5, M6	4	DC Motor
U2	1	H-bridge Motor Driver
DIST1, DIST2, DIST3	3	Ultrasonic Distance Sensor (4-pin)
SERVO1	1	Positional Micro Servo
U3	1	IR sensor

3.2 Software Used

1. Arduino IDE
2. TinkerCAD

3.3 Libraries Used

1. Servo.h – Arduino IDE

4 Diagrams and Simulations

4.1 Circuit Diagram

The circuit diagram for the project is shown in Figure 4.1.

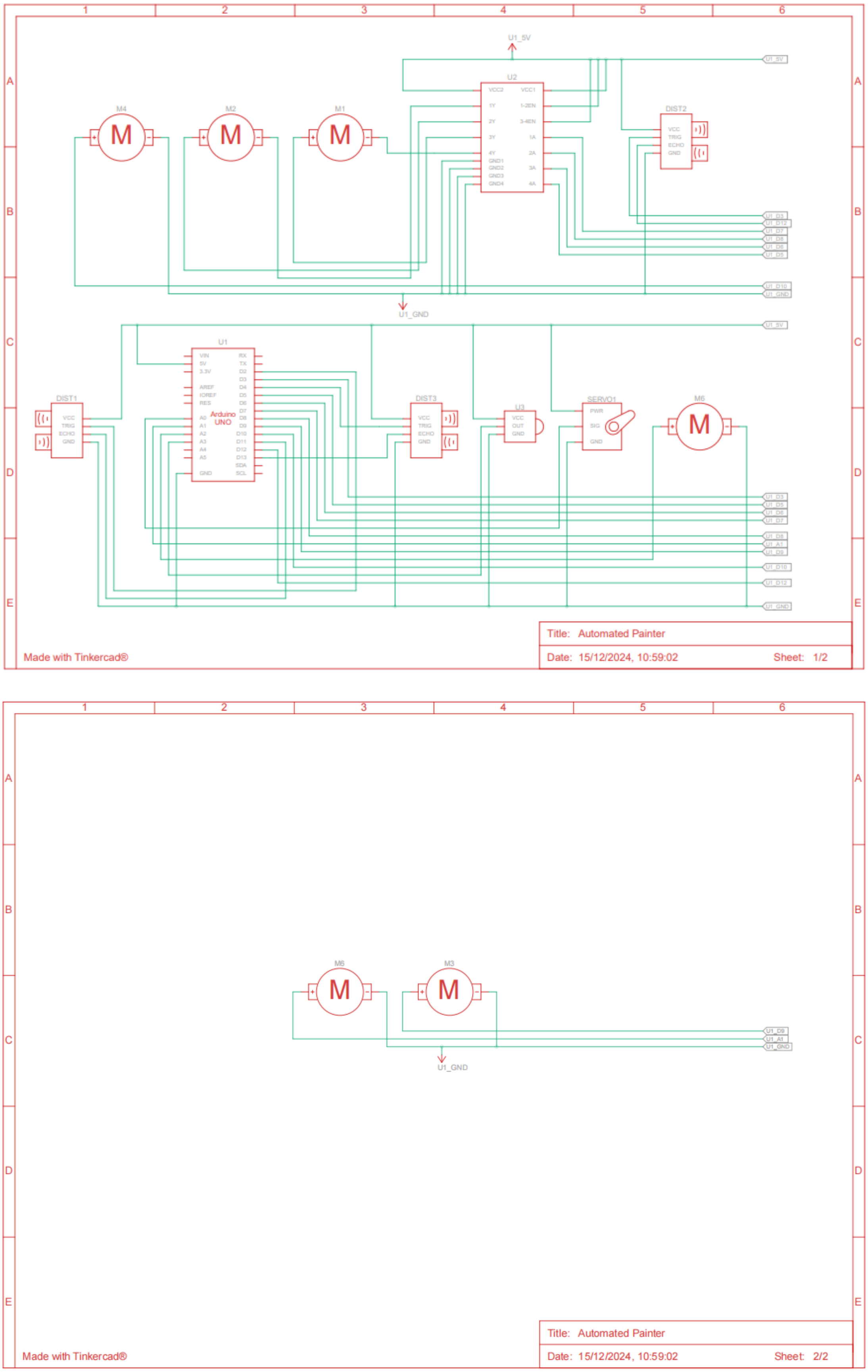

Figure 4.1 Circuit Diagram of Autonomous Painting Vehicle

4.2 Side-Profile View of Proposed Solution

The side profile view is shown in Figure 4.2.

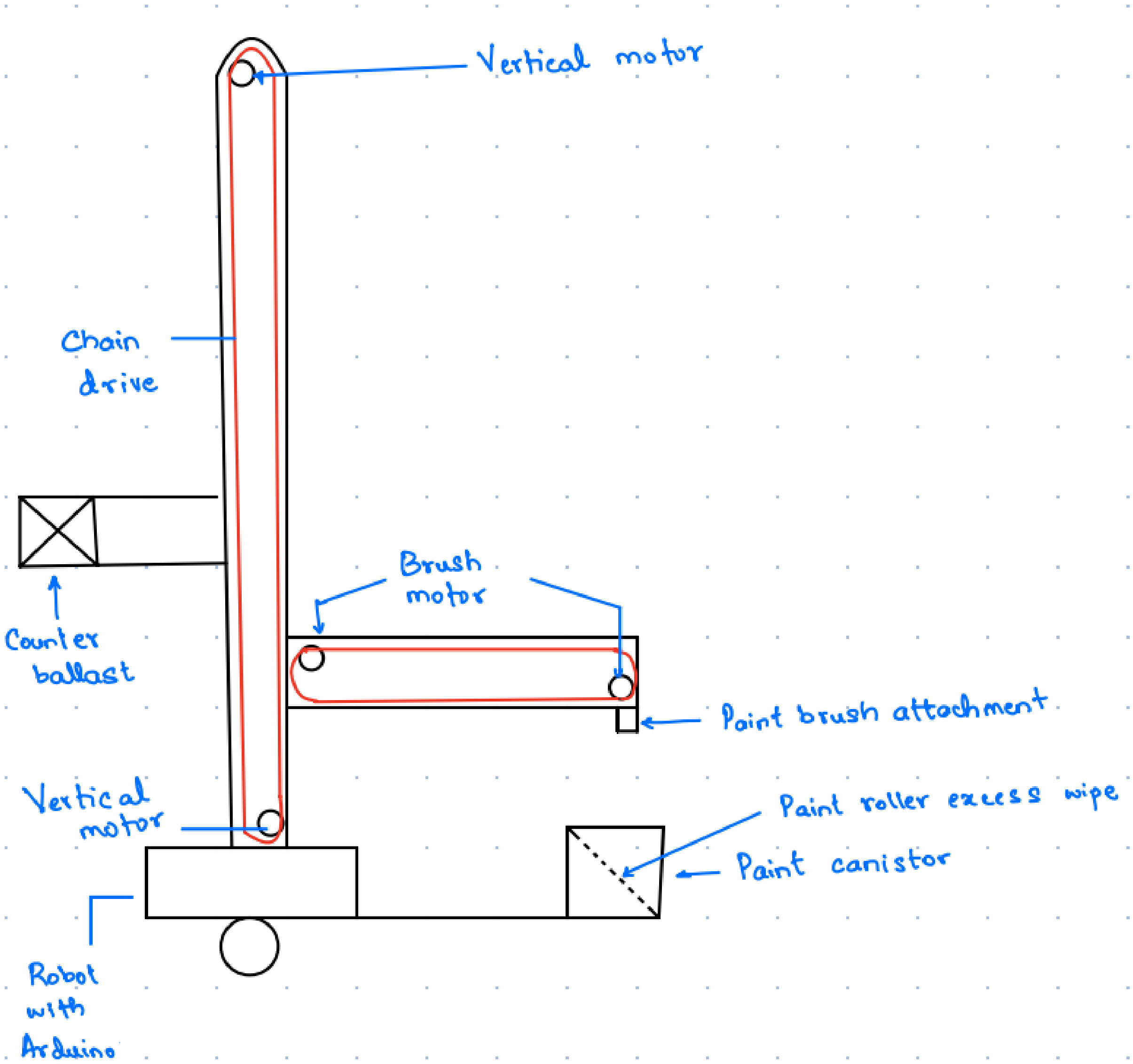

Figure 4.2 Side Profile view of Proposed Solution

4.3 Simulation Results

While a physical model would lead to more tangible results, a more preliminary study has been conducted in which manual parameters were set to test functionality.

To ensure proper working, each subsystem was tested individually.

4.3.1 Corner Detection

The first step in building this project was to detect the corner of a given room. Since there are 4 walls within a room, the robot can start at any given corner and works its way to others.

Here is the Serial Monitor output that is generated when tweaking the Ultrasonic Sensors manually.

Serial Monitor

```
Front Distance: 154.56 cm, Left Distance: 165.85 cm, Right Distance: 176.29 cm
Moving forward.
Front Distance: 21.59 cm, Left Distance: 167.42 cm, Right Distance: 175.07 cm
Turning left to align with the wall.
Aligning alongside the wall.
Front Distance: 22.30 cm, Left Distance: 167.45 cm, Right Distance: 168.91 cm
Turning left to align with the wall.
Aligning alongside the wall.
Front Distance: 22.30 cm, Left Distance: 166.06 cm, Right Distance: 27.91 cm
Left corner detected!
Corner detection complete. Ready for the next operation.
```

Once the corner has been detected, the Paint levels are to be checked. An IR sensor is used to check whether light penetrates the paint canister. It is assumed that the paint canister is transparent, and the IR sensor would be placed alongside the normal paint level.

Serial Monitor

```
Front Distance: 22.10 cm, Left Distance: 167.25 cm, Right Distance: 27.23 cm
Left corner detected!
Corner detection complete. Ready for the next operation.
PAINT LEVELS NORMINAL
```

4.3.2 Paining Process

The paintbrush has been coded to move upwards and downwards until the section has been completed. Once the section has been completed, the robot positions itself to the next section depending on the size of the brush.

```
Extending BRUSH

PAINTING WALL UPWARDS
PAINTING WALL UPWARDS
PAINTING WALL UPWARDS
PAINTING WALL DOWNWARDS
 PAINTING WALL DOWNWARDS
 PAINTING WALL DOWNWARDS
 Completed one up and down stroke of the wall.
DIPPING BRUSH
Moving on to NEXT Section
PAINTING WALL UPWARDS
PAINTING WALL UPWARDS
PAINTING WALL UPWARDS
PAINTING WALL DOWNWARDS
 PAINTING WALL DOWNWARDS
 PAINTING WALL DOWNWARDS
 Completed one up and down stroke of the wall.
DIPPING BRUSH
Moving on to NEXT Section
End of wall detected. This wall completed
Moving to the next wall...
Corner detection complete. Ready for the next operation.
NEXT WALL FOUND
Number of Walls Painted = 1
PAINTING WALL UPWARDS
PAINTING WALL UPWARDS
PAINTING WALL UPWARDS
```

Once the wall has been fully painted, the robot backups and does the corner detection again after turning 90 degrees to the right. This cycle is repeated until all 4 walls have been painted. It should be noted that the paintbrush will be dipped in paint after completion of a single section. This is a design choice which can be altered later depending on the thickness of the paint, type of paint and how often does it dry up. These parameters are to be discussed with the client to make any changes.

4.3.3 Object within Wall Detection and Door Detection

Once the robot has detected a light switch or a window, an error message is generated. The robot will then keep scanning the distance to ensure that the object has been cleared and continues painting. A flag has been set to ensure that there is no runaway.

```
PAINTING WALL DOWNWARDS
 WALL OBJECT DETECTED. IGNORING FOR PAINTING
```

A similar syntax is written for door detection. If the sensor detects a larger distance than 30cm, then a door is detected. The brush is moved out the way and sensor wait for it to come back to its normal value before resuming to paint.

4.3.4 Termination after Completion of Room

Once the robot finishes painting an entire room of 4 walls, an error message is displayed indicating that all walls have been painted. The robot is then stopped, and no further actions are allowed. To enable multiple room operations, the robot must be reset after each room.

```
ALL WALLS HAVE BEEN PAINTED IN ROOM

ALL WALLS HAVE BEEN PAINTED IN ROOM
```

5 Conclusion and Future Improvements

The solution developed is both unique and innovative, automating the traditionally manual task of indoor wall painting through a cost-effective and efficient robotic system. Widely available components, such as Arduino Uno, DC motors, and sensors, were integrated to ensure precision in paint application, including corner detection, uniform distribution, and automated wall transitions. Health concerns related to paint fumes have been addressed by eliminating human exposure during the process. Usability is enhanced through features like obstacle detection and diagnostic error handling. A scalable and modular design allows customization for various room layouts, including dynamic wall height adjustments and potential future enhancements like LiDAR-based mapping and IoT integration for real-time monitoring. The forward-thinking approach ensures adaptability to complex environments while maintaining simplicity and practicality, making this a standout, safety-oriented solution for automating indoor wall painting.

5.1 Drawbacks and Design Considerations

1. It requires a constant supply of power, mostly DC power in the form of batteries. If the battery charge were to run out during the operation of a particular wall, the memory would be erased, and the robot would default to start the painting of walls that might have already been painted.
2. The walls have been set to a particular height. This design choice has been made for ease of use rather than practical applications. If the wall length were to change per room in the indoor environment, there would be drastic consequences.
3. The paint being used is assumed to be of a particular weight and does not discriminate against its own weight. This is not true. Although the paint weight is a

much smaller issue, there have been cases where the weight of the paint can change the centre of mass of the whole setup.

4. The use of both an Arduino Uno as well as an ESP32 module is electrically draining any DC power supply. Moreover, each power rails can only supply a certain amount of current before falling into a low energy state for certain devices.
5. Not all layouts within an indoor environment are the same. Some layouts have 5 walls and others have more. With addition of walls, the angle between each wall can no longer be assumed to be 90 degrees. This would change the way the robot switches between walls.

5.2 Future Improvements

1. Use of Lithium-Ion Batteries – These cells can be used as rechargeable batteries that are capable of delivery the designed output power required to power the chips. Moreover, these cells can be programmed to show battery levels within the product and can be recharged or batteries replaced in case of low power levels.
2. The heights of the walls can be entered using a Keypad and 16x2 LCD panel to show the values that are being stored. Using this dynamic system, the walls can be painted precisely and without the worry of fringe cases. Alternatively, a LiDAR system can be installed to completely map the room that needs to be painted but this would increase the total cost and complexity of the machine.
3. A counter-ballast can be added/removed depending on the weight of the paint canister. This needs to be calibrated before the paint has been used. This requires some research into the different types of paint.
4. To enable different layout configurations, a more dynamic model must be developed. For more complex layouts where angles between the walls is not 90 degrees, a flex sensor can be installed at the base of the robot to check the angle between the walls. This sensor would then feed the angular data to the microcontroller and a rotational code can be set to turn the robot to the direction such that it is perpendicular to the wall.
5. Ground obstacle avoidance can also be added to the product to make it more robot. However, implementation of these systems would make the coding more complex as there are multiple height considerations at play. Since the robot painting arms would be large, movement across objects such as tables with different heights than that of chairs would make it dangerous.

5.3 Feasibility and Scalability

Since all hardware components are freely available in the market, this product is ready to build with basic hardware implementation. The cost of each component is rather inexpensive, and the solutions provided do not require multiple microcontrollers. Multiple software solutions are also not needed as everything is implemented in Arduino IDE.

In terms of scalability, the design can be coded to fit larger layout options with non-traditional rooms. Although this would be a bit more complex, especially when room start getting into 6 walls and above, with non-standard rooms, it would still be feasible to make the robot work. Moreover, usage of a Wi-Fi module for IoT communications can be implemented by using a more advanced Arduino chip or Raspberry Pi integrations. Integrations with cloud interfaces such as Blynk would enable the user to receive live updates on their mobile devices. To enhance it further, multiple paint brushes can be attached to the same robot, with added stability due to weight and centre of mass. This would enable the robot to paint multiple walls at the same time. It would require multiple boards doing the same function but that would not increase the cost exponentially.

6 Resources

1. GitHub Repo Link: https://github.com/roshan22cardoza/Automating-Indoor-Wall-Painting