بسم الله الرحمن الرحيم

An-Najah National University
Faculty of Engineering and Information Technology
Computer Engineering Department

Graduation Project 2

Wall-coating Robot

Done by:
Rand Johari, Asmaa Lahlabat

Supervisor: **Dr. Aladdin Masri**

Submitted partial fulfillment of the requirements for the bachelor’s degree in computer engineering

2025-2026

Acknowledgment

We would like to begin by expressing our deepest thanks to the Almighty. Without His blessings and guidance, we would not have been able to reach this stage and complete our Graduation Project 2, *“Wall-coating Robot”*.

We are sincerely grateful to our parents and families for their constant support, patience, and encouragement throughout this journey. Your belief in us has given us the motivation and strength to continue, especially during the most challenging phases of the project.

We would also like to extend our heartfelt thanks to our supervisor, **Dr. Aladdin Masri**, for his continuous guidance, valuable feedback, and insightful suggestions. His support and direction greatly contributed to shaping and improving this project.

Finally, we would like to thank the **Computer Engineering Department** at the **Faculty of Engineering and Information Technology, An-Najah National University**, for providing the knowledge, facilities, and environment that made this work possible.

Disclaimer

The two members of this team, **Rand Johari** and **Asmaa Lahlabat**, are Computer Engineering students at **An-Najah National University**. We have worked together to complete this report, which is submitted as part of the requirements for **Graduation Project 2**, titled *“Wall-coating Robot”*.
Any mistakes or inaccuracies contained in this work are solely our responsibility as the authors. The ideas, analyses, recommendations, conclusions, and viewpoints expressed in this report are entirely our own and do not necessarily represent the views of **An-Najah National University**, the **Faculty of Engineering and Information Technology**, or the **Computer Engineering Department**.

Table of Contents

Table of Figures:

Abstract

This project presents the design and development of a wall-painting system that aims to improve efficiency, consistency, and safety in wall-coating tasks. The robot automates key parts of the process by using sensors and control logic to detect paintable surfaces, plan coverage paths, and apply paint with minimal human intervention. The work focuses on achieving reliable and repeatable motion, reducing manual effort and rework, and maintaining good coating quality. The system is tested and tuned to evaluate coverage uniformity, operational stability, and ease of use. Although commercial robotic painting solutions exist, this project represents a novel and practical student implementation in our department and serves as a step toward smarter and more automated painting workflows.

1 Introduction

1.1 Problem Statement

Wall-painting and wall-coating are still largely manual processes that depend on skilled workers using rollers, brushes, and ladders or scaffolding. This work is physically demanding, time-consuming, and often exposes workers to dust, fumes, and the risk of falling from heights. In addition, achieving uniform coverage over large surfaces is difficult, and differences between workers can lead to visible inconsistencies or rework. Existing robotic painting systems are typically expensive, designed for large contractors or industrial environments, and require complex setup and calibration. There is a lack of affordable, semi-autonomous solutions that can assist painters in small to medium projects while improving consistency, safety, and efficiency.

1.2 Problem Objective

The objective of this project is to design and implement a prototype **wall-coating robot** that can move along interior walls, map its surroundings, and provide a foundation for automated paint application. The robot is built on an omni-wheel mobile base with encoders, ultrasonic sensors for wall detection, and wireless communication with a PC for live visualization and control. It is also equipped with a vertical linear axis and a high-torque servo arm intended to carry a future painting tool. On the software side, the system performs room perimeter mapping, computes a scan center, and integrates with phone-based 3D scanning (Scaniverse) and Cloudcompare to generate a 2D representation of walls and no-paint areas such as windows and doors. The overall goal is to reduce manual effort, improve coating consistency, and increase safety by automating the most repetitive and geometric parts of the wall-coating process.

1.3 Scope of work

This project focuses on building the **hardware and software foundation** for a wall-coating system rather than a complete commercial painting solution. The scope includes:

- Designing and implementing an omni-wheel mobile robot with four DC gear motors, encoders, and motor drivers.
- Developing a room perimeter mapping algorithm using ultrasonic sensors and encoder-based odometry, with live visualization on a PC application over Wi-Fi.
- Calculating a scan center point within the mapped room and enabling the robot to navigate from its start corner to this center using its movement functions.
- Integrating a workflow that uses Scaniverse and CloudCompare to convert 3D wall scans into a 2D wall coordinate system and to define configurable no-paint zones.
- Designing and testing a vertical linear axis driven by a stepper motor and a high-torque servo arm to carry the future painting tool.

Full implementation of a complete paint delivery system, outdoor operation, and advanced localization in highly cluttered or non-flat environments are considered outside the scope of this project and are left as future work.

2 Constraints, Standards and Codes and Earlier Work

2.1 Constraints

- **Limited hardware budget and sensing precision:** The system relies on low-cost components such as HC-SR04 ultrasonic sensors, DC gear motors with simple encoders, and hobby-grade drivers (L298N, DRV8825). This limits measurement accuracy and motion precision compared to industrial solutions that use LiDAR, high-resolution encoders, or professional motor drivers.

- **Limited availability of components in local stores:** Some specific components, such as the air compressor used for painting, are difficult to obtain locally. This forces the use of alternative parts, compromises on certain aspects of the design, or long waiting periods for the required components.

- **Power and safety limitations**: The robot is powered from a battery and DC-DC converters, which constrain the maximum current available for motors and future paint hardware. High currents can damage drivers or converters if not carefully tuned, so long tests and full-load experiments must be performed cautiously.

2.2 Standards / Codes

- **Programming Environment**: Arduino IDE was used for writing and deploying firmware on the Arduino Mega 2560 and ESP32 microcontroller boards, using Arduino C/C++. Python 3 was used on the PC side for the live mapping application and for processing 3D scan data.

- **Programming Languages**: Arduino C/C++ for the Arduino Mega 2560 (motor control, encoder reading, ultrasonic sensors, servo and stepper control, main state machine).
  Arduino **C/C++** for the ESP32 (Wi-Fi initialization and TCP–Serial bridge).
  Python for the PC application that receives telemetry, visualizes the room perimeter, computes the scan center, and converts wall point-cloud data into 2D no-paint rectangles.

- **Communication Protocols**: Serial Communication (UART**)** Used between the Arduino Mega and ESP32 for exchanging text-based commands and telemetry (e.g., START, POS, SONARS, DONE).
  TCP/IP over Wi-Fi The ESP32 exposes a TCP server over IEEE 802.11 wireless standards, allowing the PC application to connect, send commands, and receive mapping data.

- **WIFI Connectivity**: The ESP32 module operates according to IEEE 802.11 b/g/n Wi-Fi standards to establish a wireless connection with the PC. This link is used for sending robot positions, sensor readings, and for receiving control commands such as map start and scan start.

- **Motor Control**: DC Gear Motors Controlled via PWM signals generated by the Arduino Mega and driven through L298N motor drivers, allowing speed and direction control for the four omni wheels of the X-drive base.
  Stepper Motor Driven through a DRV8825 driver using standard step/direction signals from the Arduino Mega to control the vertical linear axis.
  High-Torque Servo Controlled using standard PWM servo pulses from the Arduino Mega to position the arm that will carry the future painting tool or scanning device.

- **Sensor Integration**: Ultrasonic Sensors (HC-SR04) Integrated using standard digital trigger and echo pins. The Arduino measures echo pulse duration to estimate the distance to walls and obstacles.
  Other digital/analog I/O pins on the Arduino Mega are reserved for future sensors related to coating control and safety.

- **Data Formats and Tools**: PLY point-cloud format exported from Scaniverse is used as input to CloudCompare and Python scripts.
  CSV and simple text-based formats are used for exchanging reference points, window/door corner coordinates, and no-paint rectangles between tools and the robot, following a consistent internal structure defined by the project.

2.3 Earlier work:

Previous work in the area of automated wall painting has mainly focused on **large, specialized industrial systems** or **rail-mounted sprayers** designed for factories, tunnels, or shipyards. These solutions are typically expensive, require detailed CAD models or precise manual setup, and are operated by trained technicians. They are not intended for small projects, frequent relocation, or student-level implementation, which limits their use in everyday indoor environments.

On the other side, several **low-cost and DIY projects** have used microcontrollers such as Arduino to build simple painting or wall-following robots. These prototypes usually rely on basic distance sensors and fixed motion patterns to follow a single wall, or they use simple wheeled platforms without full room mapping. While they demonstrate the feasibility of low-cost automation, they often lack:

- a reliable method to map the entire room,
- integration with modern 3D scanning tools, and
- a structured way to define and avoid no-paint areas like windows and doors.

There is also earlier work on **room-mapping mobile robots** that use ultrasonic sensors and encoders to trace the perimeter of a room. However, these robots are usually designed for navigation or cleaning purposes rather than wall coating, and they do not combine mapping with vertical motion hardware for painting or with an external 3D scan workflow.

This project builds on these foundations by combining:

- an omni-wheel mobile base with encoder-based perimeter mapping,
- a Wi-Fi–connected PC application for live visualization and scan-center calculation,
- a workflow using Scaniverse and CloudCompare to convert 3D wall scans into 2D coordinates and configurable no-paint zones, and
- a vertical linear axis and high-torque servo arm prepared for future paint application.

To our knowledge, no similar graduation project in our department has integrated room mapping, phone-based 3D scanning, and planned wall coating in a single system, which makes this work a new and practical contribution.

3 literature Review

The development of autonomous robots for surface finishing and wall-painting has gained momentum in recent years, especially in the construction sector. Commercial systems such as Okibo’s EG7 robot, which performs AI-guided 3D scanning and autonomous plastering and painting on construction sites, demonstrate that automated wall finishing can already operate alongside human workers at scale [6]. Similarly, interior wall-painting robots like WALT and large autonomous latex-paint spraying robots from Legend Robot show that mobile wall-coating platforms can deliver consistent coverage, high productivity, and fully automated spraying of walls and ceilings in residential and commercial buildings [7], [8]. However, these solutions are heavy, expensive, and targeted at large contractors, leaving a gap for smaller, lower-cost systems suitable for research and educational use.

In parallel with these industrial systems, there is a long line of research on low-cost indoor mapping and localization using simple sensors. Barak et al. proposed a mobile robot equipped with a single HC-SR04 ultrasonic sensor that navigates along room boundaries and builds a two-dimensional map of an unknown indoor environment by merging sub-maps generated in MATLAB [1]. Kassem and Asem developed a low-cost SLAM system that combines a Kinect depth camera, Raspberry Pi, Arduino motor control and ROS to achieve simultaneous localization and mapping in indoor environments using off-the-shelf components [2]. More recent work investigates ultrasonic-based indoor localization systems using multiple beacons and time-of-flight trilateration, achieving centimetre-level position accuracy with simple microcontrollers and inexpensive ultrasonic sensors [3].

For capturing detailed wall and room geometry, both robot-mounted LiDAR and handheld or smartphone-based LiDAR have been explored. Hu et al. introduced a robot-assisted mobile laser scanning approach that integrates SLAM, robot motion control and path planning on a legged robot to reconstruct dense 3D point clouds of building interiors and to semantically segment walls, ceilings and other components [5]. At the same time, Catharia et al. evaluated new LiDAR sensors integrated into smartphones and showed that, with a suitable acquisition protocol, smartphone LiDAR can provide sufficiently accurate 3D point clouds for digitising complex interior buildings such as railway stations [4]. Consumer applications like Scaniverse make this kind of smartphone-based 3D scanning widely accessible and allow users to capture walls and rooms directly from a mobile phone [9].

Building on these works, our project aims to combine low-cost ultrasonic-based room perimeter mapping with smartphone-based 3D wall scanning to define no-paint regions, and a compact omni-drive wall-coating robot that can autonomously follow the mapped perimeter and apply coating while avoiding windows and other obstacles.

4 Methodology

4.1 Technologies – Components:

4.1.1 Arduino Mega 2560

The Arduino Mega 2560 is an ATmega2560-based microcontroller board that provides 54 digital input/output pins, 16 analog inputs, and multiple hardware serial ports. It is widely used in robotics and control projects due to its large number of I/O pins and strong community support.

In this project, the Arduino Mega 2560 serves as the main controller of the wall-coating robot. It is responsible for generating PWM signals for the DC gear motors, reading encoder feedback, interfacing with the ultrasonic sensors, controlling the NEMA17 stepper motor via the DRV8825 driver, and driving the high-torque servo motor. It also implements the main state machine of the system, including room perimeter mapping, scan-center navigation, and communication with the ESP32 module through a hardware serial port.

Figure 1: Arduino Mega 2560

4.1.2 ESP32

The ESP32 is a low-cost, low-power microcontroller with integrated Wi-Fi and Bluetooth connectivity. It features a dual-core processor, multiple UARTs, and sufficient memory to run network stacks and custom applications.

Within this project, the ESP32 is used as a Wi-Fi communication bridge between the Arduino Mega and a PC application. It opens a TCP server over Wi-Fi and forwards text-based commands and telemetry between the PC and the Mega using serial communication. This setup allows the robot to send real-time position and sensor data for live visualization, and to receive commands such as starting the mapping process or moving to the scan center, without requiring a wired connection to the robot.

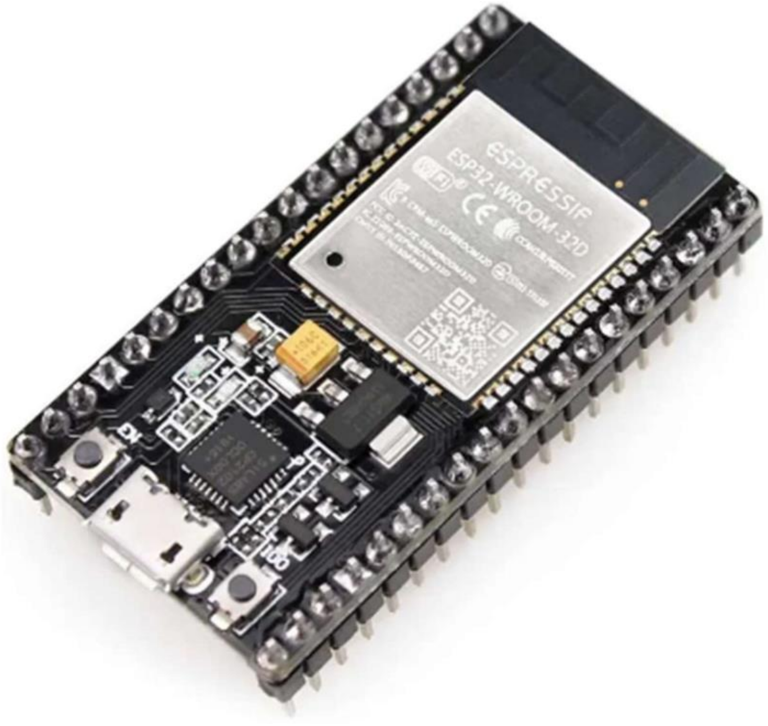


Figure 2: ESP32

4.1.3 DC Gear Motors

The mobile base of the robot is driven by four JGB37-520 12 V 60 RPM 37 mm DC gear motors with encoders and 6 mm output shafts. Each motor integrates a 37 mm metal gearbox (520 series) that reduces the speed to approximately 60 RPM at 12 V, providing enough torque to move the robot smoothly while carrying the battery, electronics, and the vertical axis hardware. The 6 mm shaft allows secure coupling with the chosen 58 mm omni wheels and mechanical hubs.

Each JGB37-520 motor includes a rear-mounted encoder that outputs quadrature signals. These encoder signals are read by the Arduino Mega to estimate wheel rotation and compute encoder-based odometry for the robot. This odometry is used during room perimeter mapping and while navigating from the start corner to the scan center. In practice, the four motors were experimentally calibrated and individual PWM trim factors were applied to compensate for manufacturing differences between motors, resulting in straighter paths and more accurate mapping of the room boundaries.

Figure 3: DC Gear Motor

4.1.4 Omni Wheels

Each DC gear motor is coupled to a 58 mm omni wheel. Unlike standard wheels, an omni wheel has multiple small rollers mounted around its circumference, oriented perpendicular to the main rolling direction. These rollers allow the wheel to roll freely in its primary direction while permitting lateral sliding.

The four omni wheels are arranged in an X-drive configuration, where each wheel is mounted at approximately 45° relative to the robot's main axes. By controlling the speed and direction of each wheel independently, the robot can move forward, backward, sideways, and diagonally, as well as rotate around its center. This high maneuverability is particularly important for wall-coating tasks, because it allows the robot to align itself parallel to walls, adjust its lateral offset, and navigate near corners without complex steering mechanisms. The omni-wheel configuration therefore plays a key role in enabling flexible motion along room boundaries and around paintable surfaces.

Figure 4: Omni Wheel

4.1.5 L298N Motor Drivers

The L298N is a dual H-bridge motor driver capable of driving two DC motors with separate direction and PWM control. It can handle relatively high currents at voltages suitable for typical DC gear motors used in hobby and educational robotics.

In this project, two L298N modules are used to drive the four DC gear motors of the X-drive base (two motors per driver). The Arduino Mega provides direction and PWM signals to the L298N boards, which in turn modulate the motor voltage. This enables speed control, directional movement, and the application of per-motor PWM trims that were experimentally calibrated to reduce drift and ensure straighter motion during perimeter mapping.

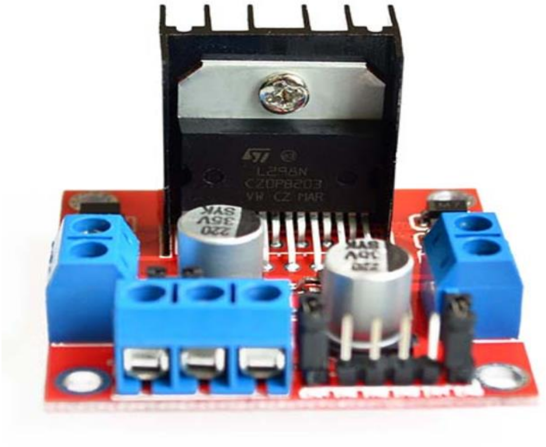


Figure 5: L298N Motor Driver

4.1.6 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor is a low-cost distance sensor that measures the time of flight of an ultrasonic pulse reflected from nearby obstacles. It typically operates in the range of a few centimeters up to several meters, with sufficient accuracy for basic wall and obstacle detection in indoor environments.

The robot is equipped with four HC-SR04 sensors mounted on its front, left, back, and right sides. These sensors are used primarily for wall detection during room perimeter mapping. In particular, the front sensor detects the distance to the wall, enabling the robot to determine when it has reached the room boundary and to maintain a safe offset (approximately 17 cm) that protects the planned vertical axis from colliding with the wall. The sensor readings are also transmitted to the PC for live visualization of wall proximity along the robot's path.

Figure 6: Ultrasonic Sensor

4.1.7 NEMA17 Stepper Motor

The NEMA17 stepper motor is a widely used stepper size in 3D printers and small CNC machines. It offers discrete angular steps (typically 1.8° per full step) and good holding torque, which allows precise control of linear or angular motion when combined with an appropriate transmission mechanism.

In this project, a NEMA17 stepper motor is used to drive the vertical linear axis (Z-axis) of the robot. Through a pulley and belt or similar mechanism, the stepper motor moves a carriage up and down along a linear rail. This carriage is intended to carry the painting tool or an attached device such as a phone mount. The use of a stepper motor allows controlled, repeatable adjustment of the coating height on the wall, which is critical for future automatic spraying or rolling operations.

Figure 7: NEMA17 Stepper Motor

4.1.8 DRV8825 Stepper Driver

The DRV8825 is a stepper motor driver from Texas Instruments capable of driving bipolar stepper motors with microstepping support. It accepts step and direction signals from a microcontroller and regulates the current through the motor coils using adjustable current limiting.

In the wall-coating robot, the DRV8825 is used to control the NEMA17 stepper motor on the Z-axis. The Arduino Mega generates step and direction pulses to move the carriage up or down by a specified number of steps. The current limit (Vref) on the DRV8825 was carefully adjusted to provide enough torque for lifting the expected load while protecting the driver and motor from overheating. This driver enables fine control over vertical positioning, forming the basis for precise, repeatable coating patterns across different wall heights.

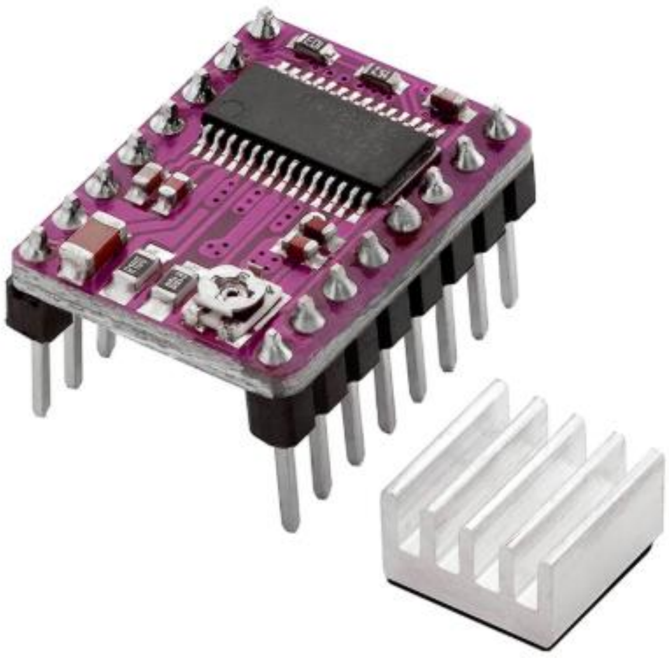


Figure 8: DRV8825 Stepper Driver

4.1.9 40 kg Servo Motor

The project employs a high-torque digital servo motor with a rated torque of approximately 40 kg·cm. This type of servo is significantly stronger than standard hobby servos and is designed to handle heavier loads, such as tools or devices mounted at the end of an arm. The servo is controlled using a standard PWM signal that specifies the desired angular position.

In this system, the 40 kg servo is used as an articulated arm joint to carry and orient a future painting tool or a smartphone used for scanning. It is connected directly to the Arduino Mega, which generates carefully calibrated pulse widths to move the servo between safe positions. During mapping, the servo holds the arm in an upper position to avoid collisions, and during scanning or coating stages it can be moved to intermediate angles that provide suitable viewing or painting geometry.

Figure 9: 40 kg Servo Motor

4.1.10 Linear Rail and Pulley Mechanism

The linear rail provides a guided track for smooth vertical motion of the carriage along the wall. It is typically constructed from metal with a matching slider or carriage that minimizes play and friction. A belt-and-pulley or similar transmission mechanism connects the carriage to the stepper motor, converting rotational motion into linear displacement.

For the wall-coating robot, the linear rail and pulley mechanism form the Z-axis structure. The NEMA17 stepper motor drives the pulley, which moves the carriage up and down the rail. This arrangement allows the painting tool or scanner mount to reach different heights on the wall while keeping the motion stable and repeatable. The mechanical design of the rail is crucial to ensure that the vertical load is well supported and that the motion remains aligned with the wall surface.

Figure 10: Linear Rail

Figure 11: Pully Mechanism

4.1.11 Battery Pack and Power Converters (LM2596 / XL4015)

The robot is powered by a rechargeable battery pack composed of three cells, providing a nominal voltage of approximately 12 V. This battery supplies energy to DC gear motors, stepper motors, and electronic components. Since different subsystems require different voltages, dedicated DC-DC buck converters are used to generate stable lower voltages from the main battery rail.

Two types of buck converters are employed in this project: LM2596-based and XL4015-based modules. These converters step down the 12 V battery voltage to approximately 5–6 V for powering logic circuits and high-torque servos. Careful consideration was given to current capacity, voltage regulation, and decoupling capacitors to avoid voltage drops or damage during high-load conditions, such as motor start-up or servo movement. Proper power distribution is essential for the reliable operation of the robot, especially during simultaneous motor and communication activity.

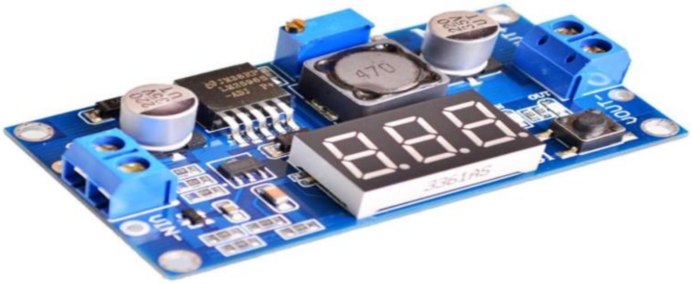


Figure 12: LM2596

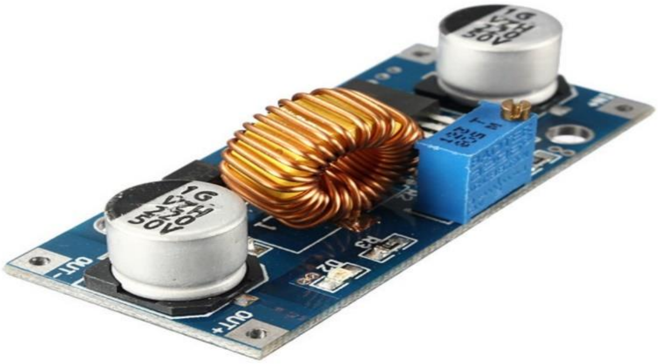


Figure 13: XL4015

4.1.12 Linear Actuator

The project also includes a compact electric linear actuator that converts rotary motion into linear displacement. The actuator is designed to provide a controlled push–pull stroke and is intended for future stages of the wall-coating system, where it can be used to adjust the distance of the coating tool from the wall or to press a spray trigger or roller mechanism in a repeatable way.

The linear actuator is mounted on the vertical structure using custom 3D-printed brackets and is mechanically linked to the end-effector through a dedicated linkage piece. Although the current prototype focuses mainly on room mapping and positioning, integrating the linear actuator at this stage prepares the hardware for more advanced coating actions and enables future experiments on automatic paint application without redesigning the mechanical assembly.

Figure 14: Linear Actuator

4.1.13 Arm Structure

This 3D-printed U-shaped part acts as a carrier for the arm that holds the gripper. It is connected to the linear actuator so that when the actuator extends or retracts, it pushes or pulls this U-shaped carrier and moves the entire arm–gripper assembly. The geometry of the part provides a guided interface between the actuator and the arm, ensuring smooth and controlled linear motion.

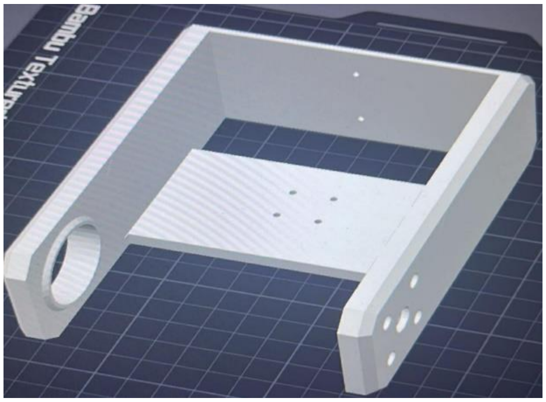


Figure 15: 3D-Printed

This 3D-printed linkage connects the moving rod of the linear actuator to the robot arm or end-effector. The tapered shape and eyelets at both ends allow it to transfer the actuator's linear motion while keeping the structure lightweight and mechanically stable.

Figure 16: 3D-Printed Linear Actuator–to–Arm Linkage

4.1.14 Gripper Mechanism

At the end of the arm, the robot uses a gripper mechanism actuated by a dedicated servo motor (such as the DS558HV). The gripper consists of a pair of jaws or a clamp that can open and close under servo control, allowing it to hold different attachments securely. Depending on the experiment, the gripper can be used to clamp a smartphone mount for 3D scanning, or to hold and actuate a future painting tool or spray trigger.

The gripper servo is controlled by the Arduino Mega using standard PWM signals, with predefined angles corresponding to “open” and “closed” states. This provides a simple and flexible way to adapt the end-effector to different tools without redesigning the entire arm. By combining the vertical motion of the Z-axis, the rotation of the 40 kg arm servo, and the gripping action at the end, the robot gains the ability to position and hold the tool in front of the wall in a controlled and repeatable manner.

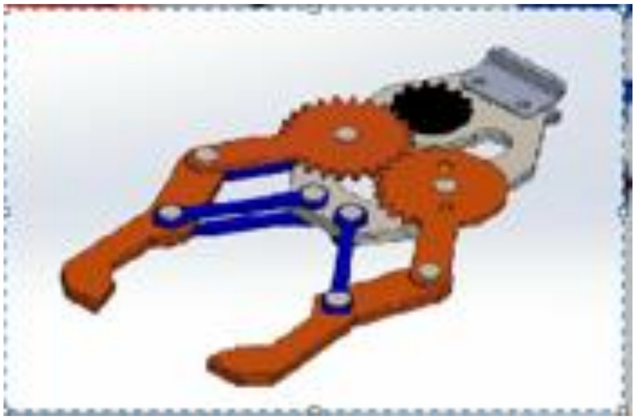


Figure 17: Gripper Mechanism

4.1.15 Gripper Servo Motor (DS558HV)

The end-effector of the robot uses a dedicated servo motor (DS558HV class) to actuate the gripper. This digital servo provides higher torque and better position holding than standard micro servos, which is important to securely clamp tools such as a smartphone holder for scanning or, in future work, a small painting tool or trigger mechanism.

The gripper servo is controlled directly by the Arduino Mega using standard PWM servo pulses. Predefined angles correspond to “open” and “closed” states of the gripper, allowing the system to reliably release or hold the attached tool during operation. In the current prototype, the servo starts from a safe neutral angle and then moves to specific angles for opening and closing, as determined experimentally. By separating the gripper actuation into its own servo, the robot gains flexibility to adapt different end-effectors in future stages without modifying the main arm or vertical axis design.

Figure 18: DS558HV

4.1.16 Wires:

We used different type of wires to adapt to the type of connectivity among components.

Figure 19: male to male wires

Figure 20: male to female wires

Figure 21: female to female wires

4.2 Mechanical Part:

The mechanical structure of the wall-coating robot is designed to provide stable motion along indoor floors and controlled positioning in front of vertical wall surfaces. The system is organized into three main mechanical subsystems: the omni-wheel mobile base, the vertical linear axis (Z-axis), and the arm–gripper assembly driven by a high-torque servo and a linear actuator.

The mobile base is built around a rigid rectangular chassis that carries the four JGB37-520 DC gear motors, the battery pack, the electronics, and the vertical structure. The four 58 mm omni wheels are mounted in an X-drive configuration, giving the robot holonomic motion: it can move forward, backward, sideways, diagonally, and rotate around its center. This layout provides high maneuverability near walls and corners, which is essential for tracing the room perimeter and aligning the robot parallel to the wall during mapping and future coating operations.

Mounted on one side of the chassis is the vertical linear axis. This Z-axis consists of a linear rail and carriage driven by a NEMA17 stepper motor through a pulley mechanism. The rail guides the carriage up and down in a straight line, allowing the end-effector height to be adjusted relative to the floor. The stepper motor provides discrete, repeatable vertical positioning, forming the basis for controlling the coating height on the wall in later stages of the project.

Attached to the moving carriage is the arm and gripper assembly. A high-torque 40 kg servo motor acts as the main joint of the arm, rotating a rigid link that extends in front of the robot. At the end of this link, a gripper mechanism driven by a separate servo holds a smartphone mount for 3D scanning or, in future work, a painting tool or spray trigger. A linear actuator, connected through custom 3D-printed U-shaped carriers and linkages, provides an additional controlled linear motion for the arm–gripper assembly, enabling fine adjustment of the distance between the tool and the wall.

The ultrasonic sensors are mounted around the perimeter of the chassis (front, left, back, and right) to measure distances to nearby walls and obstacles. Their placement is chosen so that the robot can detect the room boundary while maintaining a safe offset that protects the vertical axis and arm from collision. All mechanical elements—chassis, wheels, rail, arm, gripper, mounts, and brackets—are arranged to keep the center of mass low and to ensure sufficient stiffness for reliable motion.

This mechanical design supports the core functions of the project: autonomous room perimeter mapping, navigation to the scan center, and preparation for automated wall coating, while remaining compact and modular enough for experimentation and future upgrades.

4.3 Mechanical Design:

The mechanical design of the prototype focuses on building a simple but functional structure that can move vertically along a wall and carry a coating tool at the end of an arm. The design evolved in several stages, starting from a basic wooden frame and ending with a fully assembled platform that combines the vertical axis, arm, gripper, and electronics.

4.3.1 Wooden Base and Vertical Support

The first stage of the design consisted of a square wooden base and a vertical wooden column fixed at its center. This structure provides stable support for the linear axis and the arm while keeping the construction inexpensive and easy to modify. A metal bracket is mounted at the top of the column to prepare a rigid mounting point for the linear rail and the stepper motor assembly.

Figure 22: Wooden base and vertical support.

4.3.2 Vertical Linear Axis Assembly

In the next stage, a linear rail was attached along the front face of the wooden column, and a NEMA17 stepper motor was mounted at the top. A belt-and-pulley transmission connects the motor to a carriage that slides along the rail. The controller electronics, power supply, and test components were placed on the wooden base. This configuration allows the carriage to move up and down under precise stepper control, defining the Z-axis that will later position the coating tool at different heights on the wall.

Figure 23: Assembled vertical linear axis with linear rail, NEMA17 stepper motor.

4.3.3 Wooden Base and Vertical Support

The first stage of the design consisted of a square wooden base and a vertical wooden column fixed at its center. This structure provides stable support for the linear axis and the arm while keeping the construction inexpensive and easy to modify. A metal bracket is mounted at the top of the column to prepare a rigid mounting point for the linear rail and the stepper motor assembly.

Figure 24: Close-up of the 3D-printed geared gripper mechanism with linear actuator.

4.3.4 Arm and Spray-Can Holder

To extend the tool away from the vertical axis and orient it correctly relative to the wall, a 3D-printed arm was designed and attached to the carriage. At the end of this arm, the gripper holds a spray can used for coating tests. The arm geometry was chosen to keep the can approximately parallel to the wall while maintaining enough clearance from the wooden column. Servos in the arm and gripper allow the system to position the can and actuate the spray, demonstrating the basic coating motion along the vertical axis.

Figure 25: 3D-printed arm and gripper holding a spray can attached to the vertical axis.

4.3.5 Final Integrated Prototype

In the final prototype, the vertical axis, arm, and gripper assembly were combined with a mobile base carrying four wheels and the full set of electronics. The base supports the battery, microcontrollers, drivers, and wiring required to control the stepper, servos, and sensors. The complete system forms a compact platform where the coating tool can be raised and lowered along the rail, while future work will integrate horizontal motion and full room mapping. The wiring layout and component placement were adjusted so that the robot remains stable and the moving parts have sufficient clearance during operation.

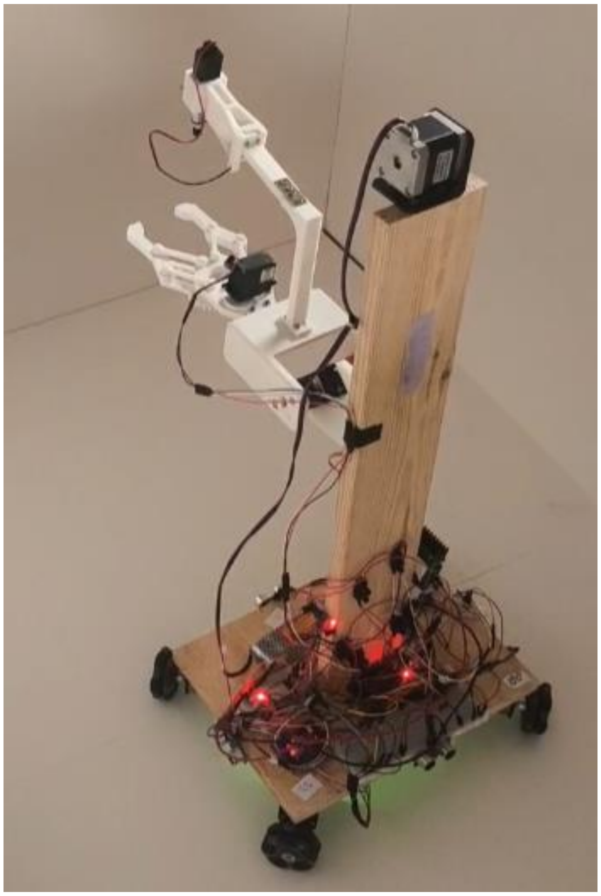


Figure 26: Final integrated prototype of the wall-coating robot.

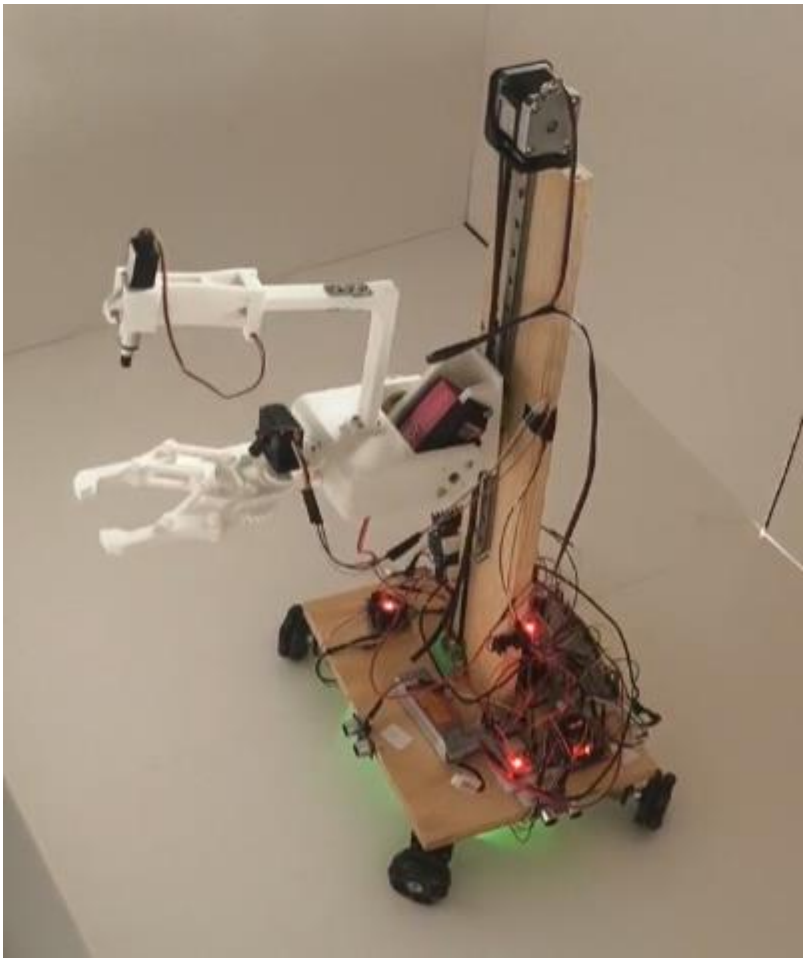


Figure 27: Side view of the complete prototype highlighting.

4.4 Features

This section describes the main functional features of the wall-coating prototype. The implemented features focus on room perimeter mapping, communication with a PC for live visualization, and basic scan-center navigation. In addition, workflow is defined for integrating smartphone-based 3D wall scans and for specifying no-paint regions that will guide future coating operations.

4.4.1 Room Perimeter Mapping and Odometry

The robot performs autonomous room perimeter mapping using a combination of encoder-based odometry and ultrasonic distance measurements. The control software on the Arduino Mega is organized as a finite state machine with four main states: **WAIT_START**, **SEEK_FIRST_WALL**, **RUNNING**, and **FINISHED**.

After power-up, the robot remains in the WAIT_START state until it receives a start command from the PC. It then enters the SEEK_FIRST_WALL state, where it drives forward while monitoring the front ultrasonic sensor. When the distance to the wall falls below a predefined threshold (approximately 17 cm), the robot considers this position as the origin of the room coordinate frame and switches to the RUNNING state.

In the RUNNING state, the robot follows a rectangular path around the room using its omni-wheel base. The motion is divided into four straight segments corresponding to the nominal +X, +Y, −X, and −Y directions. For each segment, the robot commands appropriate wheel speeds and uses the encoder counts from the four JGB37-520 motors to estimate its position and orientation (odometry). At the corners, a “corner creep” behavior is used, in which the robot slightly overshoots and then corrects its heading to approximate a right-angle corner instead of cutting the corner diagonally.

During mapping, the four ultrasonic sensors (front, left, back, and right) continuously measure distances to nearby walls. The front sensor is used primarily to detect the first wall and to maintain a safe offset from the boundary, while the other sensors provide additional information about side walls and potential obstacles. When the robot has completed at least one full loop and its current position is close to the starting point within a small tolerance, it switches to the FINISHED state and stops, signaling that the perimeter has been mapped.

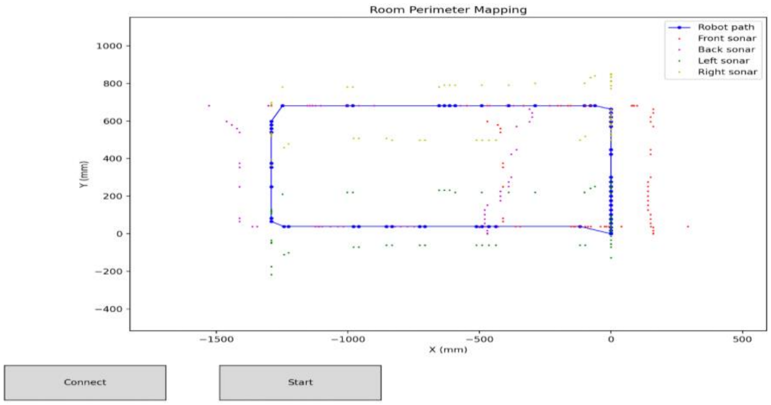


Figure 28: Roome Mapping.

4.4.2 Communication and Live Mapping Software

To visualize the mapping process in real time and to control high-level actions, the robot communicates wirelessly with a PC application. The ESP32 module acts as a Wi-Fi bridge: it opens a TCP server and forwards data between the Arduino Mega (over UART) and the PC.

The Arduino periodically sends telemetry lines of the form `POS,x,y,theta` and `SONARS,d1,d2,d3,d4`, where $(x, y)$is the estimated position in millimetres and $\theta$is the robot heading, while $d_1 \ldots d_4$are the ultrasonic readings from the four sensors. The PC application, implemented in Python, connects to the ESP32, parses these messages, and plots the robot path on a 2D graph together with the sonar points. Simple filtering, such as median filtering and outlier rejection, is applied to the distance data to reduce noise.

When the mapping loop is complete and the robot sends a `DONE` message, the PC application processes the recorded trajectory to close the rectangular path. A helper function trims the path and snaps the final point to the starting point, removing diagonal artefacts that may result from odometry drift. The resulting path provides an approximate 2D representation of the room perimeter and is used to compute the scan center in the next feature.

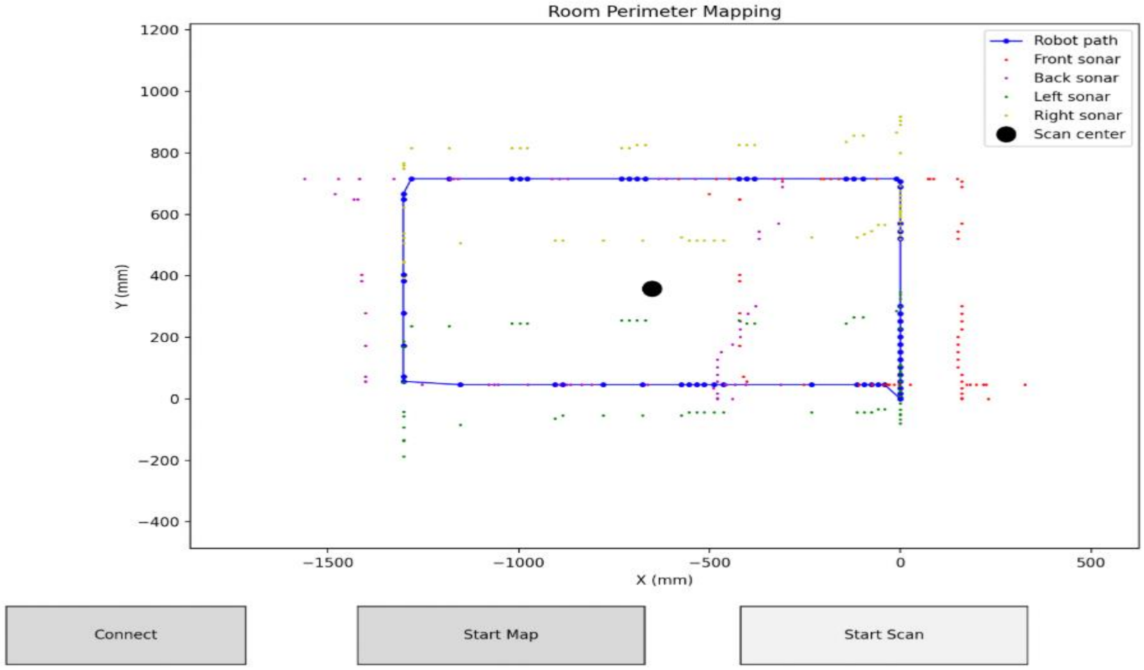
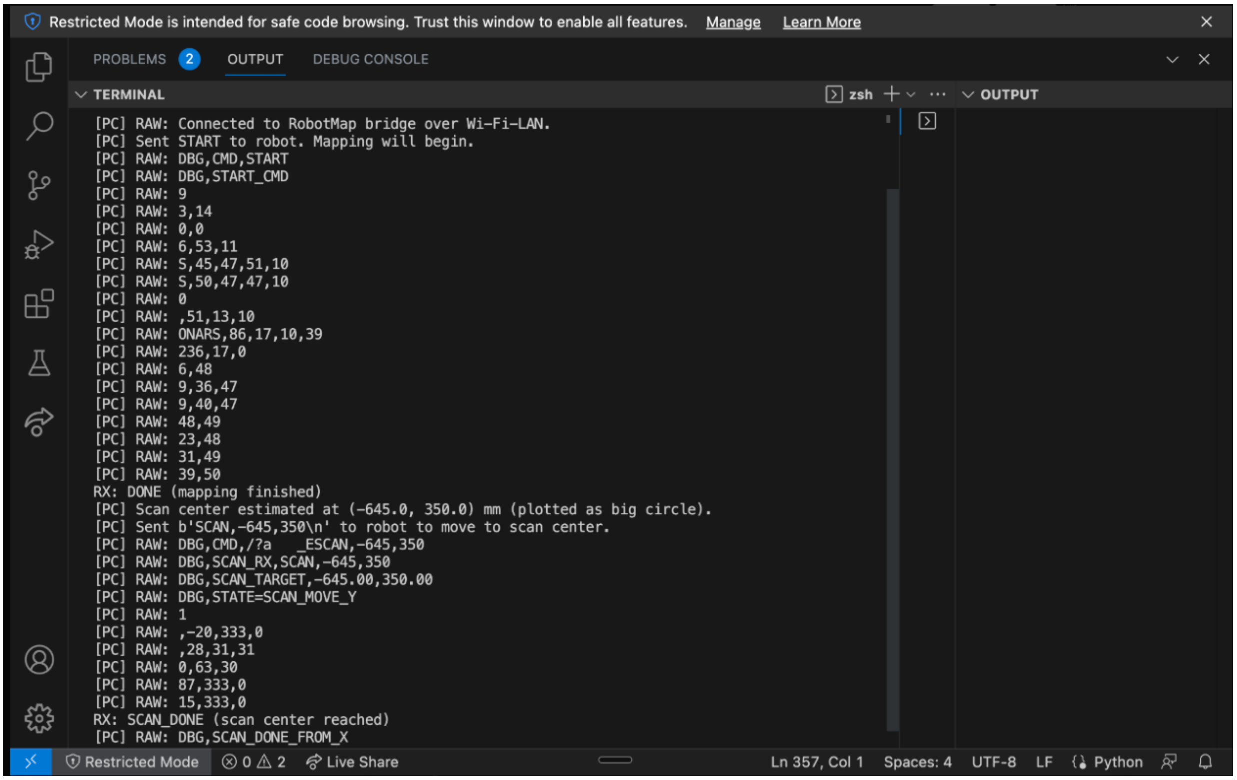


Figure 29: Connect with ESP32.



Restricted Mode is intended for safe code browsing. Trust this window to enable all features.

```
[PC] RAW: Connected to RobotMap bridge over Wi-Fi-LAN.
[PC] Sent START to robot. Mapping will begin.
[PC] RAW: DBG,CMD,START
[PC] RAW: DBG,START_CMD
[PC] RAW: 9
[PC] RAW: 3,14
[PC] RAW: 0,0
[PC] RAW: 6,53,11
[PC] RAW: S,45,47,51,10
[PC] RAW: S,50,47,47,10
[PC] RAW: 0
[PC] RAW: ,51,13,10
[PC] RAW: ONARS,86,17,10,39
[PC] RAW: 236,17,0
[PC] RAW: 6,48
[PC] RAW: 9,36,47
[PC] RAW: 9,40,47
[PC] RAW: 48,49
[PC] RAW: 23,48
[PC] RAW: 31,49
[PC] RAW: 39,50
RX: DONE (mapping finished)
[PC] Scan center estimated at (-645.0, 350.0) mm (plotted as big circle).
[PC] Sent b'SCAN,-645,350\n' to robot to move to scan center.
[PC] RAW: DBG,CMD,/?a   _ESCAN,-645,350
[PC] RAW: DBG,SCAN_RX,SCAN,-645,350
[PC] RAW: DBG,SCAN_TARGET,-645.00,350.00
[PC] RAW: DBG,STATE=SCAN_MOVE_Y
[PC] RAW: 1
[PC] RAW: ,-20,333,0
[PC] RAW: ,28,31,31
[PC] RAW: 0,63,30
[PC] RAW: 87,333,0
[PC] RAW: 15,333,0
RX: SCAN_DONE (scan center reached)
[PC] RAW: DBG,SCAN_DONE_FROM_X
```

Figure 30: *Communication with ESP.*

4.4.3 Scan Center Estimation and Navigation

After a successful mapping run, the PC application computes an approximate scan center for the room. In the current prototype, this center is obtained by analyzing the recorded perimeter path and calculating a point near the middle of the room, for example, by averaging the minimum and maximum x- and y-coordinates of the trajectory. The computed center is displayed on the live map as a marker.

When the user presses the “Start Scan” button, the PC sends a command of the form `SCAN, cx, cy` to the robot, where (c_x, c_y) are the target coordinates of the scan center in the same map frame used during mapping. The Arduino Mega then enters a scan-navigation mode in which it moves from the starting corner to the scan center using its existing straight-line movement functions.

To simplify the control, the robot first corrects its lateral position along the Y direction (moving left or right based on encoder odometry), and then corrects its position along the X direction (moving forward or backward). Movement continues until the distance to the target in each axis falls below a small tolerance. Once the target is reached, the robot stops at the scan center, where the vertical axis and arm can be used to position a smartphone for 3D scanning or a future coating tool.

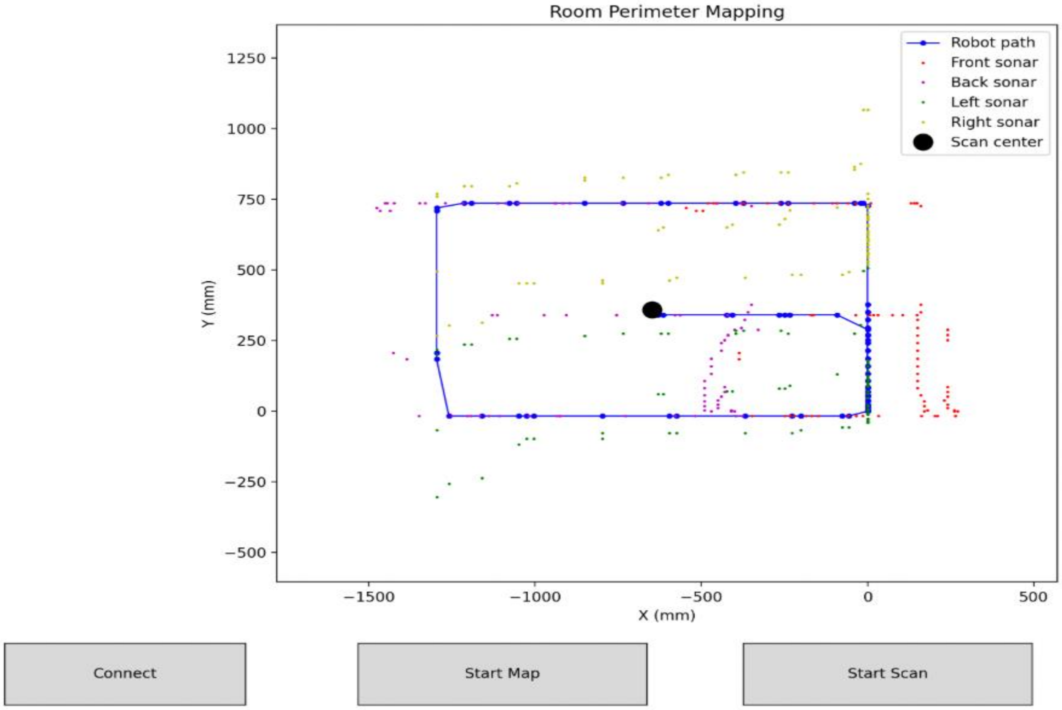


Figure 31: Scan Center.

4.4.4 Wall 3D Scanning and 2D Projection Workflow

To obtain a more detailed description of the walls and to define regions that should not be coated, the project uses a workflow based on smartphone 3D scanning and point-cloud processing. A wall is first scanned using a mobile application such as Scaniverse, which generates a 3D point cloud of the environment and exports it in PLY format. The PLY file is then opened in CloudCompare on a PC.

Within CloudCompare, the “Point picking” tool is used to select reference points on the wall and corner points of windows or doors. Typically, three reference points are selected to define the wall plane and a local 2D coordinate system, and four corner points are picked for each opening. These points are exported as CSV files for further processing.

A Python script reads the CSV files, constructs a local 2D wall coordinate frame from the reference points, and projects the 3D corner points onto this plane. From these projected points, the script computes axis-aligned rectangles that represent no-paint areas on the wall. The resulting rectangles are stored in a simple configuration file containing $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$values in wall coordinates. This configuration can then be sent to the robot via the existing TCP–Serial communication channel.

4.4.5 No-Paint Zones and Planned Coating Logic

The no-paint rectangles generated in the previous step are intended to guide the future coating process. On the Arduino Mega, these rectangles can be stored in memory as a list of 2D regions defined in the same wall coordinate frame used during mapping and scanning. When the robot executes a coating path, the control logic can compare the current tool position (x, y)against all stored rectangles and temporarily disable spraying whenever the position lies inside any no-paint region.

Although the current prototype focuses on mapping, scan-center navigation, and the preparation of the vertical axis and arm, this no-paint concept provides a clear path towards a complete autonomous coating system. In future work, the vertical stepper axis and arm will be used to move a spray can or other coating tool along planned vertical and horizontal trajectories while respecting the no-paint zones. In this way, the same mechanical and software foundation presented in this project can be extended to perform safe, selective coating of walls that include windows, doors, or other areas that must remain unpainted.

5 Conclusion

The development of the wall-coating prototype has demonstrated that a low-cost, modular platform can provide a solid foundation for future automated painting systems. The project combined an omni-wheel mobile base, a vertical linear axis, and an arm–gripper assembly into a single robot capable of moving close to walls and adjusting the height and orientation of a coating tool. On the software side, the system implements encoder-based room perimeter mapping, wireless communication with a PC for live visualization, and a basic scan-center navigation feature.

In addition, a workflow was designed that uses smartphone-based 3D scanning (Scaniverse), point-cloud processing in CloudCompare, and Python scripts to project wall geometry into 2D and define configurable no-paint regions such as windows and doors. Although full automatic coating of complete rooms is not yet implemented, the results show that the chosen mechanical design, sensing approach, and communication architecture are suitable for building a more advanced wall-coating system in future work.

6 Future work

In future work, the main direction will be to integrate artificial intelligence into the system so that more of the planning and decision-making becomes automatic. Instead of manually defining paths and no-paint areas, AI methods can analyze the environment and generate the required actions. Possible extensions include:

1. **1. AI-based wall analysis and path planning:**
   In future versions, the robot will include its own depth or 3D camera mounted permanently on the arm, so it can scan the walls directly without switching between a smartphone and a spray can. Computer vision and machine-learning models will analyze the live 3D data from this camera, automatically detect walls and openings, and generate optimal coating paths, including where to start, how to move, and how many passes are needed.
2. **Automatic detection of no-paint areas:**
   Apply AI to automatically recognize windows, doors, and other regions that must not be coated directly from images or 3D data, without requiring the user to manually select points.
3. **Intelligent quality checking:**
   Use vision-based AI to check coverage after coating (for example, detecting unpainted spots or uneven areas) and let the robot decide where it needs to re-coat.

By adding these AI capabilities in the future, the system can move from a semi-automatic prototype toward a more fully wall-coating robot that requires minimal human intervention.

6 References

[1] N. Barak, N. Gaba, and S. Aggarwal, “Two-Dimensional Mapping by using a Single Ultrasonic Sensor,” Int. J. Advanced Research in Computer Science, vol. 7, no. 3, pp. 254–257, May–Jun. 2016.

[2] A. H. Kassem and M. Asem, “A low-cost robotic system for simultaneous localization and mapping,” Journal of Engineering and Applied Science, vol. 71, art. 158, 2024, Doi: 10.1186/s44147-024-00486-8.

[3] M. Sukop, “Simple Ultrasonic-Based Localization System for Mobile Robots,” Applied Sciences, vol. 14, no. 9, 3625, 2024, Doi: 10.3390/app14093625.

[4] O. Catharia et al., “Smartphone LiDAR Data: A Case Study for Numeration of Indoor Buildings in Railway Stations,” Sensors, vol. 23, no. 4, 1967, 2023, Doi: 10.3390/s23041967.

[5] D. Hu, V. J. L. Gan, and C. Yin, “Robot-assisted mobile scanning for automated 3D reconstruction and point cloud semantic segmentation of building interiors,” Automation in Construction, vol. 152, 104949, 2023, Doi: 10.1016/j.autcon.2023.104949.

[6] Okibo Ltd., “Smart Robots for Construction Sites – EG7 Robot,” Okibo official website, accessed Jan. 22, 2026.

[7] Endless Robotics, “WALT the wall paint robot,” RBTX Solution Description, accessed Jan. 22, 2026.

[8] Shenzhen Legend Robot Technology Co., Ltd., “Latex Paint Spraying Robot (6.2m),” Legend Robot official product page, accessed Jan. 22, 2026.

[9] Niantic, Inc., “Scaniverse – Your invitation to explore the world in 3D,” Scaniverse official website, accessed Jan. 22, 2026.