RMIT Classification: Trusted

RMIT UNIVERSITY

EEET2610 | Engineering Design 3

School of Science, Engineering & Technology

# EEET2610 – Engineering Design 3

# Project Proposal

**Team A**

**Pham Tri Dung – s3981129**

**Vu Thien Minh Hao – s3938011**

**Phung Nam Khanh – s4033609**

**Tran Thanh Sang – s4028016**

**Huynh Nhat Anh – s3924763**

**Teacher: Dr. Son Vu**

**Submission Due Date: 23:59 PM 28th November 2025**

"We declare that in submitting all work for this assessment, I have read, understood, and agree to the content and expectations of the Assessment declaration."

School of Science, Engineering, and Technology

What's next...

# Table of Contents

# LIST OF FIGURES

# LIST OF TABLES

# Abstract

This project proposal outlines the design, development, and integration of an autonomous Mecanum-wheel robot capable of omnidirectional mobility and intelligent navigation within indoor environments during a 12-week time frame of the course EEET2610 Engineering Design 3. This project emphasizes system-level synthesis across mechanical, electrical, and software engineering disciplines. Using ESP32 microcontrollers, DC motors with encoders, and an IBT-2 motor driver, the objective is to construct a four-wheel Mecanum platform with precise motion control via PID regulation and IMU-assisted odometry. The software architecture integrates camera sensing, SLAM-based navigation, LiDAR mapping, and ROS2-enabled autonomous behavior with MATLAB-based kinematic modeling for trajectory planning. From basic modeling to final demonstration, the project is guided by five structured work packages: simulation, mechanical prototype, embedded control, autonomous implementation, system integration, and pitch presentation. The report details the project's context, methodology, timeline, task and resource allocation, and risk management, ensuring feasibility within a 12-week development cycle of the course. Upon completing this project, our team's goal was to present a fully operational Mecanum-wheel robot with easy remote control and autonomous characteristics.

# 1. Introduction

## 1.1. Context and Background Information

As developments in mechanical design, embedded computing, and artificial intelligence enable the deployment of robots in more complex and dynamic situations, the field of automated mobile robotics is expanding rapidly [1], [2]. Modern platforms combine robust mechatronic structures with dedicated control hardware, onboard sensing, and network connectivity to operate reliably in environments such as warehouses, laboratories, hospitals, and smart manufacturing lines.

However, many existing robots still rely on non-holonomic drive configurations, such as differential or Ackermann steering, which constrain motion to forward–backward movement with relatively large turning radii. This often leads to inefficient back-and-forth maneuvers and makes precise positioning in confined indoor spaces challenging. Nowadays, omnidirectional mobile robots have attracted significant attention because they decouple translation and rotation, can maneuver in tight spaces, and change direction without complex steering mechanisms [3]. Mecanum-wheeled robots achieve holonomic motion using wheels equipped with angled rollers, allowing the platform to move in any direction and rotate in place without changing its footprint. By independently controlling the speed and direction of each wheel [4], [5], [10], such robots are particularly suitable for tasks that require accurate navigation and positioning in narrow or cluttered environments.

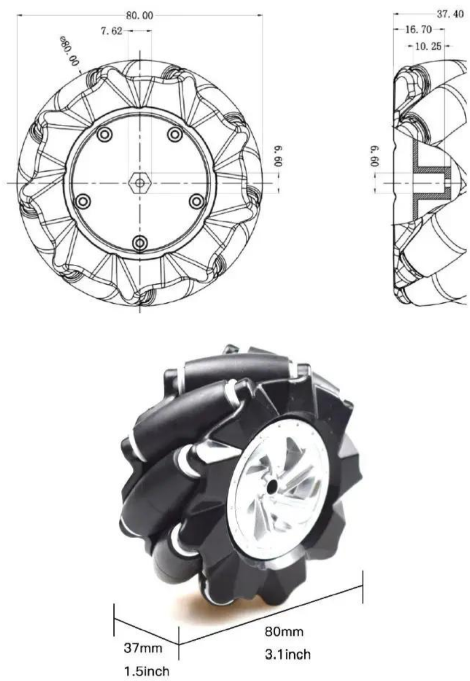


Figure 1- Mecanum-wheel structure
[https://device.report/manual/8787826]

Nonetheless, mechanical design alone does not determine the practical usefulness of a mobile robot. Its ability to perceive the environment and to present information in a usable way to human operators is equally critical. Recent surveys on vision-based navigation [6], [7] show that computer vision has become a key enabling technology for localisation, obstacle avoidance, and scene understanding in unstructured indoor environments. Cameras provide richer contextual information, including textures, labels, markers, and objects of interest. At the same time, lightweight embedded platforms have made it feasible to process this visual data on board. The ESP32-CAM module is a low-cost vision solution that combines an ESP32 microcontroller, a camera, and integrated Wi-Fi/Bluetooth connectivity in a compact form factor [11], [12]. This makes it a practical choice for adding simple vision capabilities—such as visual feedback, basic object identification, or first-person video—to a small mobile robot without requiring a full single-board computer. ROS2 (Robot Operating System 2) is a popular modular middleware framework that facilitates the autonomous actions of the robot. ROS2 provides prolific tools for sensor integration, mapping, localisation, and navigation, and is used to handle higher-level autonomous tasks such as LiDAR, camera, SLAM, and path planning.

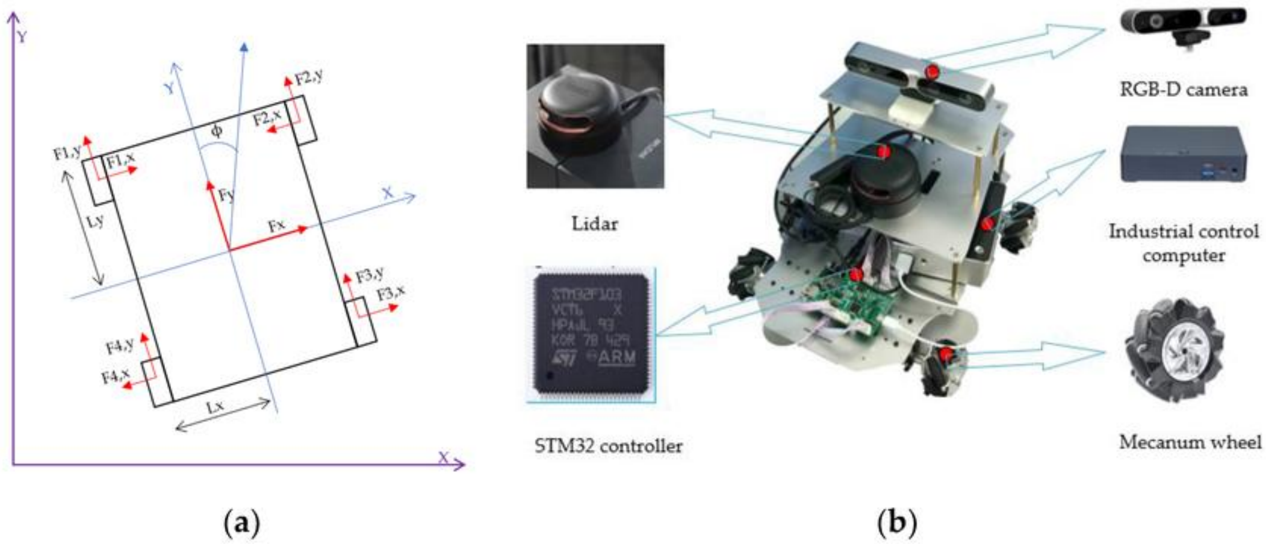


Figure 2 - Four-wheel-drive Mecanum mobile robot and power measurement

[https://www.researchgate.net/figure/Four-wheel-drive-Mecanum-mobile-robot-and-power-measurement-a-Four-wheel-Mecanum_fig1_337081740]

Concurrently, the human-robot interface also plays a critical role in a successful deployment. Previous studies on teleoperation and supervisory control have shown that graphical user interfaces (GUIs) can significantly impact operator workload, accuracy, and situational awareness [8], [9]. Overly complex or poorly designed interfaces can hinder performance even when the underlying robot hardware is capable. For non-expert users in labs or educational settings, a web interface can run through a browser and communicate with the robot via Wi-Fi, and tools like MATLAB App Designer offer an approachable way to combine controls (like buttons, sliders, and virtual joysticks). Both approaches support interactive control parameter adjustment, live ESP32-CAM video feed display, and real-time sensor data monitoring.

Motivated by these advancements, our project in this EEET260110 Engineering Design 3 course aims to focus on designing and developing a Mecanum-wheeled mobile robot that integrates a low-cost computer vision module (ESP32-CAM) capable of autonomous behaviours with an intuitive control interface. By combining omnidirectional mobility, real-time visual feedback, and user-centered interface design, we aim to provide a versatile platform for exploring complex control strategies and interaction concepts in mobile robotics and to lay the groundwork for future extensions towards higher levels of autonomy.

## 1.2. Literature review

**Mecanum-Wheeled Robot Control**

Mecanum wheels are ideal for tight spaces because they provide holonomic motion, which allows omnidirectional movement without steering [12], [13]. Because they are straightforward and simple to use, basic PID controllers are frequently used to control wheel velocities [12]. The Mecanum wheel is an omnidirectional wheel that allows the vehicle to move with three degrees of freedom, including forward, strafe, and rotation in any desired direction due to its unique roller characteristic and 4WD drivetrain. Each wheel is controlled separately with 4 motors in total attached to it, and the assigned angular velocity of each wheel depends on the direction and velocity of the main body's trajectory.

*Figure 3 Model of the Mecanum wheel [https://ecam-eurobot.github.io/Tutorials/mechanical/mecanum.html]*

The rollers are angled at a 45 relative to the rotation axis of the wheel; this orientation allows the translation of wheel rotation into omnidirectional movement without the need for the wheels to physically turn. The desired driving directions and velocity are achieved by selecting the correct combination of forces generated by the four wheels; any unwanted directional components are cancelled out, leaving only the chosen directional force. In addition, a key benefit of this design is the ability to almost instantaneously change the driving direction, which introduces an outstanding mobility surpassing most traditional wheel setups.

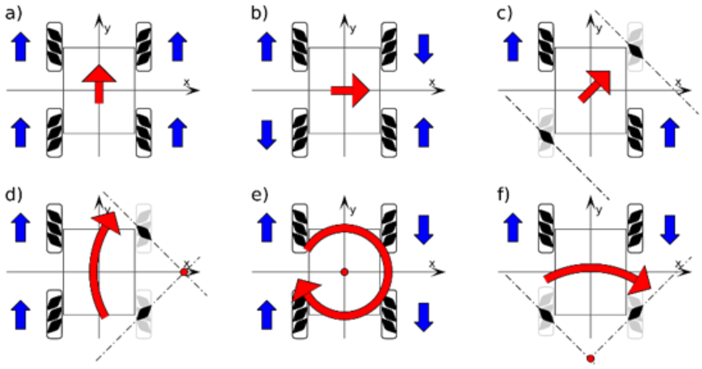


*Figure 4 Example movements of the Mecanum-wheeled robot.*

*[https://en.wikipedia.org/wiki/Mecanum_wheel]*

- Forward/Backward: All wheels are rotating at the same speed and direction.

- Sideways: The pair of wheels that are diagonal to each other have the same rotational speed and direction, with the other pair having the same rotational speed but opposite in direction.

- Diagonal: The pair of wheels that are diagonal to each other have the same rotational speed and direction while the other wheels are turned off.

- Rotational along the main body's centre: Wheels on the same side of the vehicle have the same rotational speed and direction, with the other wheels having the same rotational speed but opposite in direction.

- Rotational along an arc with the centre located on the x-axis of the main body's coordinate: Wheels on the same side of the vehicle have the same rotational speed and direction, while the others are turned off.

- Rotating along an arc with the centre located on the y-axis of the main body's coordinate: The pair of wheels on the front or rear of the vehicle has the same rotational speed and opposite direction, while the other pair is turned off.

But under real-world circumstances, more advanced methods like Model Predictive Control (MPC), Sliding Mode Control (SMC), and adaptive or robust controllers have proven to perform better, especially when slip, payload changes, and model uncertainties are present [13], [14] These techniques use kinematic and dynamic models of the Mecanum-wheeled robot to explicitly account for dynamic factors like wheel slip, payload variation, and floor conditions. Therefore, accurate modeling is essential for fine-tuning such control systems, particularly in environments that require strong disturbance rejection and precise trajectory tracking [13], [14].

**ESP32-CAM and Embedded Vision**

For low-cost robotics applications that need visual feedback, the ESP32-CAM is a popular selection because it is a small microcontroller module with an integrated camera and Wi-Fi connectivity. It can carry out simple image processing operations, including QR code scanning, basic object detection, and visual feedback to a remote operator [15]–[17]. It also supports real-time image streaming over a wireless network. Recent research has shown that ESP32-CAM-based mobile or surveillance robots can perform lightweight vision tasks and deliver live video via a web interface while maintaining low hardware costs and power consumption [15], [16]. The ESP32-CAM offers an efficient method to incorporate visual sensing into mobile robots without requiring a fully embedded PC, despite having less processing power than single-board computers. ESP32-CAM modules can be used in hospital logistics to give a remote supervisor a first-person view or to visually identify stations (for example, using QR/AprilTag markers).

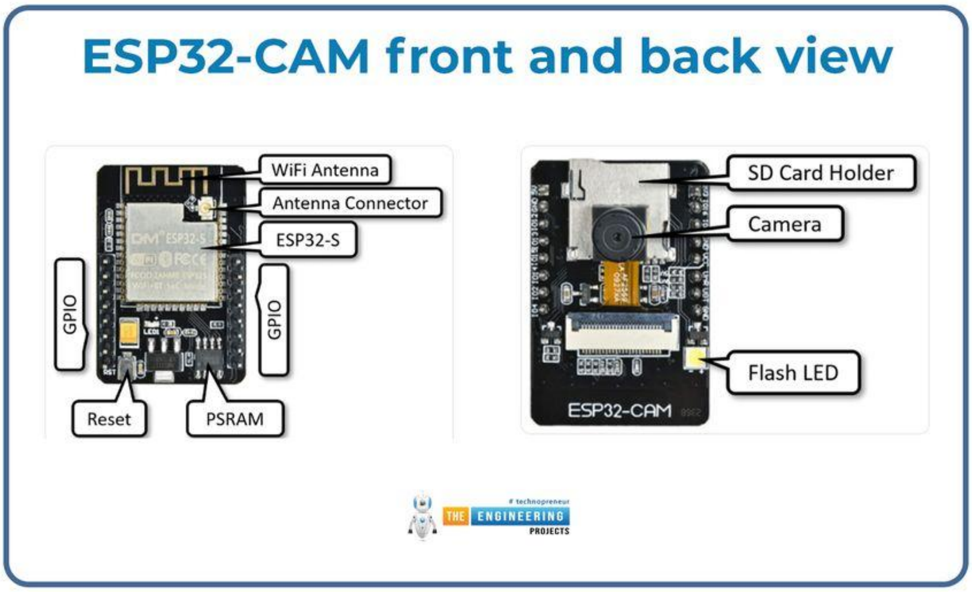


*Figure 5 - ESP32-CAM module illustration*

[https://www.theengineeringprojects.com/2025/03/how-to-program-the-esp32-cam.html]

**Autonomous Navigation with ROS2**

For autonomous robots, ROS 2 is a modern, open-source middleware framework that facilitates the modular integration of sensing, planning, and actuation. It was created to overcome the original ROS's shortcomings in areas like multi-robot scalability, real-time performance, and reliability [18]. Through packages like Navigation2 (Nav2), SLAM Toolbox, and robot_localization, ROS 2 makes it possible to use SLAM (Simultaneous Localization and Mapping), sensor fusion, and path planning [18], [19]. LiDAR, IMU, camera, and wheel encoder data are typically combined to estimate the robot's position, create or update a 2D map, and produce safe trajectories in real time. Nav2 is now widely used as a reference navigation system for ROS 2-based mobile devices after being proven in long-distance, continuous navigation experiments (also known as "Marathon 2") [20]. Because of its widespread usage, ROS 2 is now a great option for developing mobile robots, particularly when scalable autonomy and flexible sensor integration are required. An ESP32 microcontroller is responsible for low-level motor control in our proposed robot, while ROS 2 manages high-level behavior like route planning, mapping, and environment awareness.

**Mobile Robots in Healthcare Logistics**

In healthcare environments, autonomous mobile robots are becoming more popular for carrying out routine delivery and logistics tasks like moving prescription drugs, lab samples, and medical waste. These robots increase overall operational efficiency, lessen the workload for employees, and reduce the possibility of human error [11], [12]. Research on hospital transportation systems in use shows that staff walking time is significantly reduced, freeing up nurses to focus more on providing direct patient care instead of performing repetitive transport tasks [11], [12]. Mecanum-wheeled robots are especially useful in situations where conventional differential-drive robots might find it difficult to maneuver through tight spaces, crowded rooms, and clinical stations. They can navigate around obstacles and precisely align with drop-off locations thanks to their ability to move in any direction without turning, which is essential in dynamic clinical environments [13].

## 1.3. Problem statement

In most hospitals, medicines, meals, specimens, and general supplies are still transported manually by nurses and support staff. While this traditional approach is well established, it presents several underlying issues. First, repetitive delivery tasks contribute to staff fatigue and burnout, increasing the risk of human errors such as delayed deliveries or, in critical cases, medication reaching the wrong patient. Second, valuable human resources are consumed by low-complexity transport activities (e.g., moving waste, delivering consumables) instead of being allocated to tasks that require clinical judgement and direct patient care. Finally, manual transport can elevate the risk of infection transmission: staff frequently moving between wards or in and out of isolation rooms to pick up and deliver items may unintentionally contribute to the spread of pathogens and potential outbreaks.

Although some hospitals have begun to use automated carts and guided vehicles, these systems are often expensive, dependent on infrastructure, and have limited mobility in confined spaces or dynamic conditions. This project aims to investigate a more portable, mobile, and omnidirectional Mecanum-wheeled robotic vehicle as a flexible, cost-effective internal logistics assistant.

## 1.4. Contribution

This project will contribute to hospital logistics and robotics in healthcare. By downloading a map of the hospital floors, the robot can create optimal delivery paths, while the LIDAR sensor detects obstacles that need to be avoided, leveraging the unique mobility of the Mecanum wheels. This ensures that deliveries are timely and accurate, with minimal errors. Additionally, these robots can be used for waste disposal, reducing the workload of medical staff and saving significant time. This not only enhances operational efficiency but also helps prevent errors caused by overwork.

Moreover, the use of autonomous robots allows healthcare professionals, such as nurses and doctors, to limit direct interactions with others, significantly lowering the risk of infection—an essential consideration, especially during outbreaks of highly infectious diseases. These benefits contribute directly to the achievement of the Sustainable Development Goals (SDGs), fulfilling a critical project requirement.

Furthermore, this system has the potential to serve as the foundation for a wide range of applications. Beyond healthcare, it can be adapted for use in industries such as warehousing, infrastructure, and even entertainment. For example, autonomous robots could assist passengers at airports and train stations, guide disabled individuals, or support military and police personnel in high-risk scenarios, such as counter-terrorism operations.

## 1.5. Structure of the report

This project proposal is divided into 9 main sections, each containing key information and essential knowledge about the design, development, and modification processes of the project. The detailed breakdown of each section is presented below:

- **Introduction:** This section presents the project context, background, and literature review, followed by the problem statement, project objectives, and expected contribution of our team. It explains why a Mecanum-wheeled robot with ESP32-CAM (and ROS2 implementation) is beneficial for hospital logistics and clarifies the scope and limitations of the work.
- **Project Task Description:** This section breaks the project into work packages and deliverables. It describes the main technical activities, such as kinematic modelling and simulation, embedded software and interface development, mechanical and electronic design, integration, and the ROS2 autonomous implementation. Responsibilities and focus areas for each work package are outlined clearly.
- **Time Management:** A project timeline is presented using Gantt charts that cover the initiation, planning, execution, and closure phases during a 12-week timeline of this project. This section explains the scheduling of major tasks, dependencies between activities, and key milestones to ensure the prototype can be completed within the course timeframe.
- **Resource Management:** This section lists the required hardware and software resources, including a Bill of Materials (BOM) for motors, sensors, controllers, mechanical parts, and tools. It discusses budget considerations, component sourcing (course-provided vs team-purchased), and strategies for efficient use of resources.

- **Organization and Partners Presentation:** Here, potential stakeholders and partners are introduced, such as hospitals, healthcare providers, robotics companies, and academic supervisors. The section explains how these stakeholders may benefit from our proposed robot and clarifies that real deployment and commercial partnerships are outside the current project scope.
- **Risk Analysis:** This section identifies technical, logistical, and organizational risks (for example, delays in component delivery, integration issues, safety concerns, software bugs) and provides an assessment of probability and impact. Mitigation strategies and contingency plans are summarized in a risk table.
- **Team Introduction and Team Contract:** This section presents the team members, their programs, strengths, and assigned roles (robotics, electronics, software, and project management). It also includes the team contract, which defines communication rules, meeting cadence, conflict resolution, workload expectations, and consequences for not meeting agreed commitments.
- **Conclusion:** The conclusion summarizes the proposal, restates the main goals and expected outcomes of the project, and briefly outlines potential future developments, such as full ROS2-based autonomy, integration with hospital information systems, or additional sensing capabilities.
- **References and Appendix:** The final part of the report lists all cited sources and provides any supplementary material (detailed calculations, extended diagrams, datasheets, or extra planning documents) that support the main proposal.

# 2. Project Task Description

This project is organized into five Work Packages (WPs), each marking an important milestone of the project, and includes clearly defined deliverables for each stage. The following subsections provide a detailed description of each work package and its associated deliverables.

## 2.1. Work Package 1: Simulation with MATLAB

The main purpose of this work package is to simulate the motion of the Mecanum-wheeled robot with a planned trajectory using MATLAB. The four DC motors driving each individual wheels of the robot are integrated with encoders to measure the angular velocity of the wheels as a requirement for the drive-wheel odometry to estimate the exact position of the robot on the map during the trajectory. Furthermore, important parameters such as the wheel's radius, distance from the wheels to the centre of the robot's main body in x and y axis are essential to model the movement of the robot.

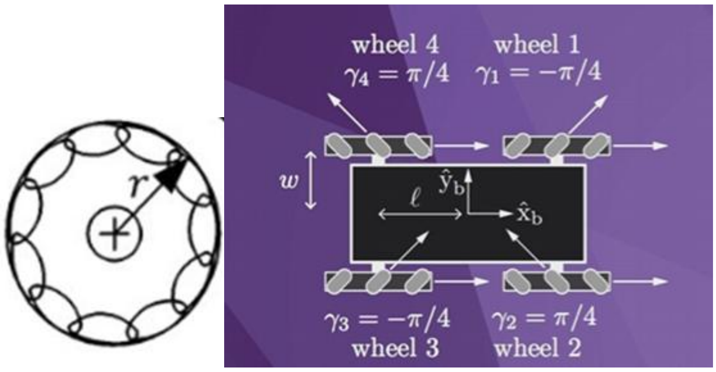


Figure 6 Parameters of the wheel and geometric parameters of the main body

[https://www.researchgate.net/figure/Parameters-of-the-Mecanum-wheel_fig1_345023500
https://www.youtube.com/watch?v=NcOT9hOsceE&list=WL&index=8&t=175s]

### 2.1.1. D1.1 Kinematics of the Mecanum robot

This combines inverse kinematics and forward kinematics of the Mecanum-wheeled robot to give insights of how to control the robot to drive in a given direction at a certain speed and when combined with the drive-wheel odometry give the ability to track its position in a simulation environment in MATLAB.

#### A. Forward kinematics

This matrix is used to calculate the chassis velocity of the robot which are the linear velocities in x and y direction with angular velocity in the z axis using the angular velocity of each individual wheels.

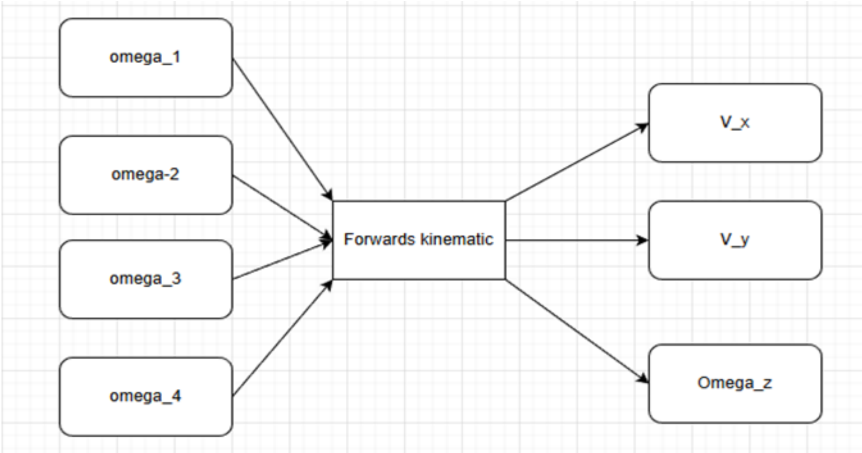


*Figure 7 Demonstration of how forward kinematic works.*

V_x: Velocity of the main body in the x axis.
V_y: Velocity of the main body in the y axis.
Omega_z: angular velocity in the z axis.
omega_1 to 4: Angular velocity of each individual wheel from wheel number 1 to number 4.

#### B. Inverse Kinematics

This matrix allows the user to calculate the required angular velocity of each wheel to achieve the desired chassis velocity of the robot.

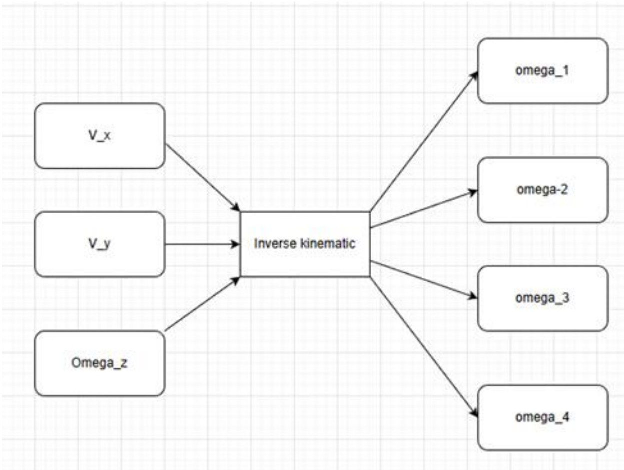


*Figure 8 Demonstration of how inverse kinematic works.*

### 2.1.2. D1.2 Trajectory planning

Trajectory planning in MATLAB facilitates the computation of a smooth and physically possible path for the robot in a defined time frame while integrating kinematic and dynamic constraint of the robot's hardware (geometric parameters of the main body, radius of the wheel). Combining this with the ability to search for path that avoids the obstacle allows the Mecanum-wheeled robot to navigate efficiently through different environmental layouts. The suggested simulation path includes a 1 m by 1 m square and a 1 m radius circle as these are the basic trajectories that can assist the user to validate the accuracy of the inverse and forward kinematic model before being implemented further.

Additionally, the two shape was chosen not only due to their simplicity but they also require most if not all the kind of movement of a Mecanum-wheeled robot including forward, strafing and rotating.

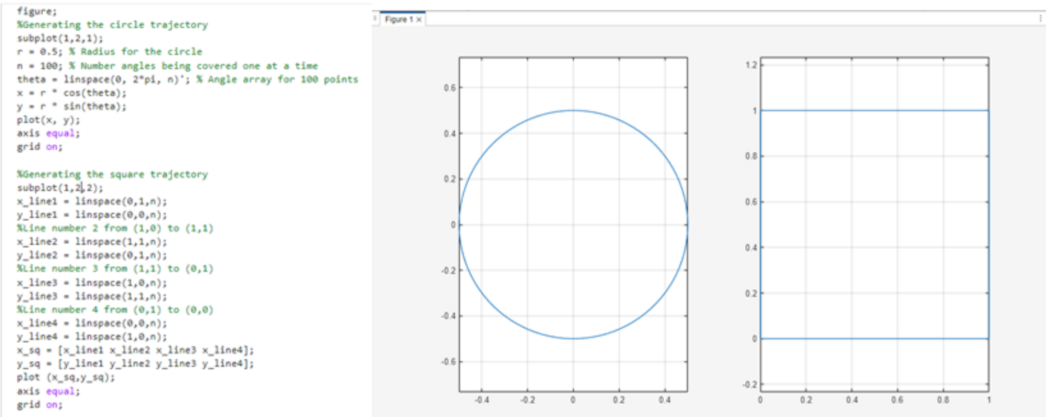


*Figure 9 Circular and Square trajectories.*

After the accuracy of the kinematic models are verified to be precise, the simulation process can move on to the next step by testing the Mecanum-wheeled robot with more complex environmental layouts filled with objects/obstacles which mimics a fraction of the difficulty the robot might face navigating in the real world.

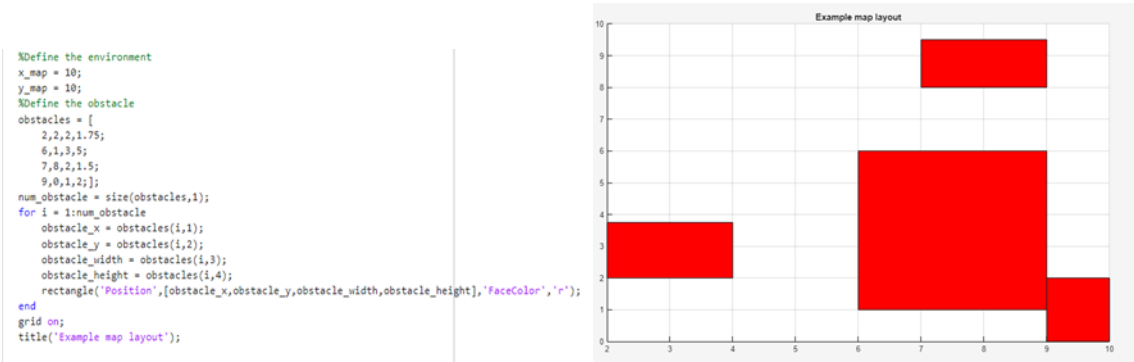


*Figure 10 Example map layout.*

## 2.2. Work Package 2: Prototyping of the Mecanum Robot

### 2.2.1. D2.1 CAD Design

To model our Mecanum robot, our team decided to use Fusion 360 instead of SolidWorks because Fusion makes collaboration much easier. By sharing the same Fusion hub, we can all work on the project at the same time and keep our files in one place. Each team member is assigned a specific part of the robot to draw, so everyone is involved in the modelling process and the final design can reflect all of our expectations. The workload is divided based on what each person is good at. The leader of this work package will contribute the most, in the beginning, the leader and another teammate took measurements of all the components we were given and sketch out basic blueprints. After that, each member starts modelling a section of the robot. The leader and the members who already have experience in 3D modelling will handle the more complicated components while the others focus on simpler parts so they can concentrate their energy on the remaining work packages. We also found that Autodesk Fusion runs more smoothly while maintaining higher graphics quality. Although SolidWorks could performs better with very complex parts, we consider this unnecessary because most of our robot components are not that complicated.

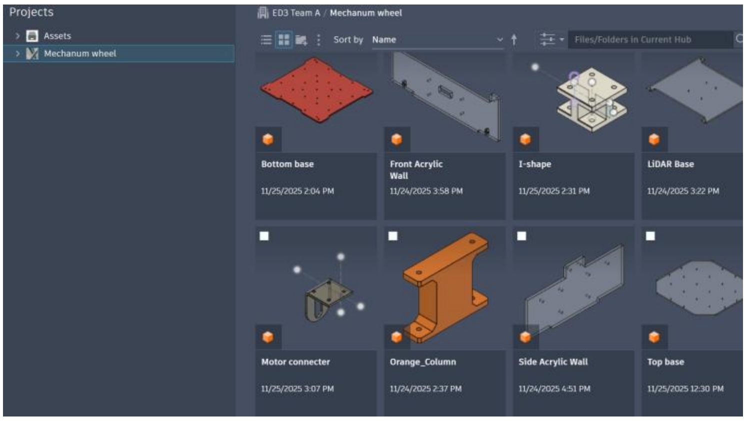


*Figure 11 Autodesk Fusion shared hub collaboration space interface*

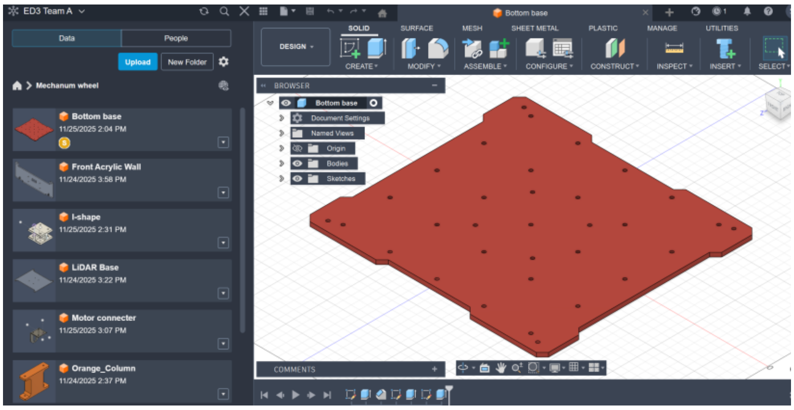


*Figure 12 Autodesk Fusion designing interface*

Once we finish recreating the robot based on the components provided, we will start discussing additional design ideas that could help us meet the project requirements and possible utilities. Since this is only the proposal stage, these ideas are still very early concepts, but we want to show our plan for how the design may evolve. For instance, we are considering 3D-printing extra parts for delivery purposes inside hospital, such as a stable platform that can hold objects securely while the robot moves. Furthermore, we should model specific holders for some medical equipment that are often used such as several perfect fit spots for first aid kid, or syringes and vials. Since all medical equipment are very fragile as they're made up of glass or electric gears that require high accuracy, these holders will ensure the deliveries move safely along with the robot.

Moreover, we also expect to design certain internal pieces to properly protect and organize the electronic components inside the robot. After finalizing and optimizing our shematics, we will likely 3D-print mounts or supports that keep the electronics stable especially with expensive devices like the raspberry pi, or vulnerable parts like the motors' drivers with their encoder requires precision feedback to control the speed. This modelling is to secure all the components within the robots when it's moving, preventing unwanted damage that could lead to lower efficiency of the robot's application.

*Figure 13 3D-printed syringes holder on autonomous robot*

*[https://3druck.com/en/research/implementation-of-a-new-type-of-syringe-holder-using-3d-printing-5483552/]*

Lastly, we'll try implement our final robot's model into the motion simulation and mapping on ROS2 ensuring that the pre-testing simulation process of our robots will be as accurate as the real-life model.

### 2.2.2. D2.2 PCB Design

To design the PCB for our robot, our team decided to use EasyEDA instead of Cadence, mainly because EasyEDA feels smoother to work with and has a cleaner, simpler interface. Although Cadence is very powerful and capable of handling more advanced electronic designs including detailed discrete components, its interface feel overwhelming and overkill for a simpler PCB that we need. Since most of our electronic components are pre-designed modules, we need a PCB mainly to connect everything together in a stable and organised way. This includes managing the power supply, ensuring all components can handle the input/output signals together with the embedded code of feedback systems and fine tuning from the controller.

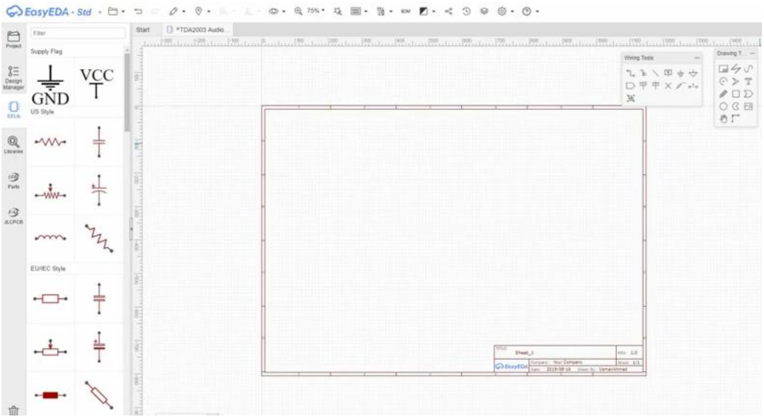

*Figure 14 EasyEDA simple user interface*

Our process starts by reading through the datasheets of all the modules to understand their voltage requirements, pin functions, current limits, and any special conditions such as logic-level thresholds or communication protocols. After gathering this information, we begin sketching the schematics that define how each module will be wired into the system. Then, we try arranging the components inside the robot's 3D model so we can see how everything fits physically and connect the modules on a breadboard to test whether the entire system behaves as expected and to confirm that the wiring plan is correct. Finally, we design and manufacture the PCB according to the test breadboard and solder everything onto it. Further testing should be done to ensure the soldering process is flawless and the PCB is working.

## 2.3. Work Package 3: Embedded and control system

### 2.3.1. D3.1 DC motor control

#### 2.3.1.1. Power Supply

DC motors require a stable power supply to function effectively. In this project, which involves controlling a mobile robot, a **4S LiPo battery** (14.8V nominal) is selected to power the motors. The robot's powertrain consists of four **12V, 330RPM DC motors**.

Using a 4S battery ensures that the voltage closely matches the motors' rated voltage, optimizing both energy efficiency and motor protection. Operating at a voltage too low can result in underperformance, while using a voltage higher than the motor's rated value may cause overvoltage conditions, potentially damaging the motor and other electrical components.

With an expected **30-minute runtime**, a battery capacity of **2200mAh** is selected, considering a **4C** discharge rate to ensure sufficient current handling, given that the working current per motor is approximately 1A. This battery provides a balance between runtime and weight, making it ideal for this application.

#### 2.3.1.2. Direction and Voltage Control Using Motor Driver

The **IBT-2 motor driver** plays a critical role in regulating the voltage supplied to DC motors. The **IBT-2** is capable of controlling the direction and speed of the motor using **PWM** signals, allowing for precise control of the robot's movement. This motor driver is designed for high-power applications, handling motor currents up to **43A peak**, making it well-suited for the 12V, 330RPM motors in your mobile robot. The **IBT-2** utilizes **MOSFET H-Bridge** technology, enabling bi-directional control of the DC motors, which is essential for forward and reverse motion.

The motor driver's power is supplied by the same **4S LiPo battery** that powers the motors. The motor driver receives PWM signals from the **ESP32 microcontroller** to regulate the motor's speed by adjusting the duty cycle, which in turn modifies the average voltage supplied to the motor. This allows the microcontroller to control both the **speed** and **torque** of the motors. With this setup, the robot can execute smooth, controlled movements while maintaining energy efficiency. The **IBT-2** also features built-in **overcurrent protection** and **thermal shutdown**, ensuring the motor driver and other components are safeguarded from electrical faults.

#### 2.3.1.3. Set Up the ESP32 Microcontroller

The ESP32 microcontroller is at the heart of this control system, receiving input from sensors, processing control algorithms, and sending commands to the motor driver. The ESP32 processes data from sensors such as motor encoders, which provide feedback on the motor’s position and speed, enabling closed-loop control. This allows for precise speed and position adjustments. The ESP32 operates on **3.3V**, and to power it from the same battery as the motors, a buck converter is used to step down the voltage from 14.8V (the 4S battery) to the required 3.3V-5V range for the ESP32. This setup ensures that the microcontroller receives a stable and regulated power supply, crucial for reliable operation. The ESP32 can be programmed to handle various tasks, including PID control, sensor readings, and wireless communication.

#### 2.3.1.4. Set Up Serial Communication Between the ESP32 and the Computer

To upload the control code and monitor the robot remotely, the ESP32 must communicate with a computer. While a direct USB connection could be used, this would limit the mobility of the robot. Instead, the ESP32 utilizes its built-in **Wi-Fi** capabilities to communicate wirelessly. The ESP32 can connect to a Wi-Fi router or a cloud server, enabling remote control and real-time data exchange. This setup allows for the robot to be controlled from anywhere within the range of the Wi-Fi network, making it highly mobile. The communication setup can also facilitate real-time updates from the robot, including encoder data and performance metrics. Additionally, the onboard Wi-Fi can be used for firmware updates, ensuring the robot remains adaptable and upgradable.

#### 2.3.1.5. Control Algorithms for Motor Speed, Encoder Feedback, and PID Control

To maintain precise control over the robot's movement, a closed-loop control system is implemented. The motor encoders provide feedback on the motor's speed and position, which is processed by the ESP32. The PID control algorithm (Proportional-Integral-Derivative) is used to adjust the motor's behavior based on the error between the desired and actual speed/position. Each term of the PID controller plays a specific role:

- **Proportional**: Corrects the error based on the current difference.
- **Integral:** Accounts for past errors to eliminate residual steady-state error.
- **Derivative:** Predicts future errors and reduces overshooting.

This combination of feedback and control ensures that the robot maintains consistent speed, regardless of changes in load or terrain. The algorithm continuously adjusts motor commands in real-time to minimize error and optimize performance. Tuning the PID values ensures smooth, stable motion, preventing oscillations or sluggishness in the robot's movement.

### 2.3.2. D3.2: Integration of an IMU ICM_20948

The ICM-20948 sensor ensures accurate motion tracking by providing 9-axis data (accelerometer, gyroscope, and magnetometer), crucial for precise orientation and position estimation in autonomous systems like a Mecanum-wheel robot. It enhances stability, control, and position estimation, allowing the robot to track its movements and adjust accordingly. We use I2C or SPI as communication protocol between the sensor and both esp32 and Rasberry PI, then both device can process it in real time and use it for odometry calculations or control algorithms to ensure precise autonomous navigation

## 2.4. Work Package 4: Autonomous implementation with ROS2

This work package is one of the most important tasks and focuses on implementing the autonomous behaviors of our robot. Upon the completion of this stage, our robot will be able to perceive its surroundings, build or use a map, and safely navigate to target positions while detecting and avoiding obstacles along the road. ROS2 will run on a companion computer (Raspberry Pi will be used in this project) and communicate with the robot's ESP32-based controller over a wireless network (a hotspot) created by our device. To achieve this, we will have to integrate the LiDAR sensor and configure a camera pipeline, then combine LiDAR, odometry, and vision data for mapping and navigation. Due to the complexity of ROS2, this task is quite challenging and requires considerable effort to complete.

### 2.4.1. Deliverable 4.1 Lidar implementation

In this deliverable, our main goal is to integrate the LiDAR sensor into the ROS2 ecosystem to provide range data of the surrounding environment to the robot. The LiDAR will be mechanically mounted on the robot and connected to the Raspberry Pi to process and send the data. The ROS2 rplidar_ros package will be utilized and will be installed and configured to publish laser scan data on a dedicated topic at an appropriate update rate.

The first step is to verify that the LiDAR sensor is correctly detected by the operating system as well as connected to other components, and that ROS2 can receive data. We will visualize the live scan using RViz2 to confirm that obstacles, walls, frontiers, and open spaces are represented accurately relative to the robot frame. Basic parameters, such as minimum and maximum range, scan angle, and frame IDs, will be tuned so that the LiDAR data aligns with our robot's coordinate system.
Once the raw data is validated, the LiDAR will be used to generate a simple 2D occupancy grid map of an indoor area (in this case, it will be our lab room). This can be achieved either by recording a

*Figure 15 A commonly used LiDAR sensor in robotics project - A1M8*

*[https://www.amazon.com/Slamtec-RPLIDAR-Scanning-Avoidance-Navigation/dp/B07TJW5SXF]*

rosbag and using an offline mapping tool or by running a SLAM node in real time. The expected outcome of this deliverable is a reliable LiDAR pipeline in ROS2 that produces stable scan data and a basic environmental map suitable for navigation and obstacle avoidance.

### 2.4.2. Deliverable 4.2 Camera implementation

In order to give our robot visual feedback, this deliverable focuses on incorporating a camera module into the ROS2 environment. Either an ESP32-CAM stream or a Raspberry Pi Camera Module will be utilized, depending on the final hardware selection. The camera module will be directly connected to a Raspberry Pi if one is available, and the camera_ros package will expose it to ROS2. Otherwise, a custom node or pre-existing tools that subscribe to the HTTP/MJPEG stream and publish images as ROS2 messages will be used to bridge the ESP32-CAM's video stream into ROS2

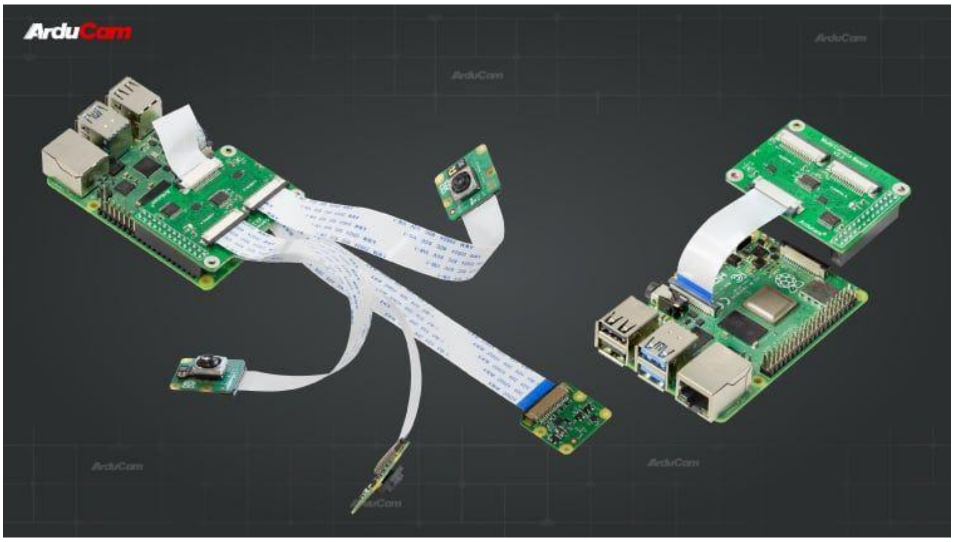


*Figure 16 RPI Module V3 attached to a Raspberry Pi*

*[https://blog.arducam.com/top-accessories-camera-module-3/ ]*

Our main tasks include configuring the camera node to publish images at a reasonable resolution and frame rate, assigning consistent frame IDs, and verifying the feed in RViz2 or another ROS2-compatible viewer. Once the basic image pipeline is operational, simple experiments will be conducted by the team to investigate higher-level applications, such as identifying QR codes or AprilTags to mark stations or waypoints in the environment. An AprilTag detection node can be added to publish tag poses that can subsequently support docking or localization behaviors if we have sufficient time and resources.

A stable camera topic in ROS2 that offers a real-time first-person view of the robot's surroundings is the expected outcome of this deliverable; more sophisticated implementation may allow us to see visual landmarks. Additionally, the software team's user interface and remote teleoperation tools can make use of this camera feed. The successful integration of this Deliverable will facilitate us in developing more advanced features for our robot.

### 2.4.3. Deliverable 4.3 Navigation and Mapping

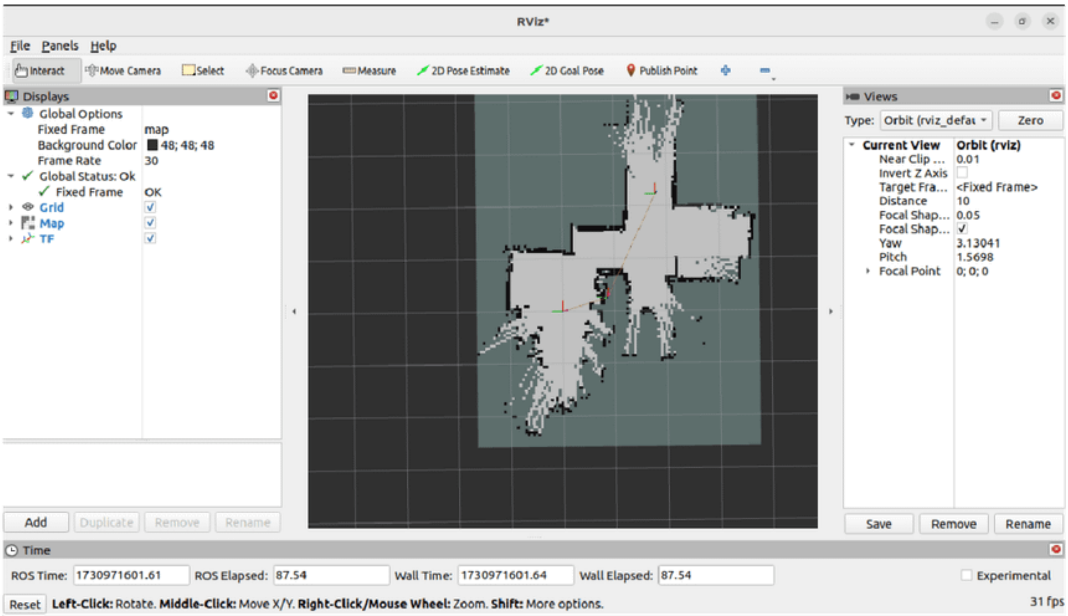


*Figure 17 Figure 17 - SLAM for environment mapping*

*[https://inomacreate.com/ros2-slam_toolbox/]*

In this deliverable, we will combine the individual sensing and control components to enable autonomous navigation for our robot. The ultimate goal is for the robot to move from a start position to a target position in an indoor environment while avoiding obstacles and preventing crashing into the wall. ROS2 navigation tools such as nav2, robot_localization, and a SLAM or map server will be used to achieve this task.

The robot's TF (transform) tree, which links frames like base_link, laser, camera_link, and odom, must first be defined by our team. In order to estimate the robot's pose over time (odometry), wheel encoder data and IMU measurements from the ESP32 will be sent to ROS2 and combined using the robot_localization package. The robot will be localized on a pre-recorded map, or a map will be constructed using the LiDAR data from Deliverable 4.1 (online SLAM). The Navigation2 stack will then receive this map and process it for the robot's pose estimation.The navigation stack will then be set up with the proper obstacle inflation parameters, costmaps, and local and global planners. Our team will use RViz2 to set navigation objectives and observe how the robot plans and follows routes while reacting to LiDAR-detected obstacles. Moving along a "corridor," changing into a "room," and docking close to a designated station are examples of test scenarios that represent some realistic tasks for future adoption.

The goal of D4.3 is to accomplish reliable autonomous navigation in a small, controlled area rather than a complex, real-life environment due to the course's time constraints. In order to demonstrate the potential of our system for future deployment in healthcare logistics, we anticipate that after completing this task, our robot will automatically generate paths and be instructed (via the ESP32 controller) to reach specific waypoints while avoiding collisions. Furthermore, a control user interface is also anticipated to be developed in order to provide an interactive and intuitive collaboration with the robot. A web interface can run in any popular browser and communicate with the robot via Wi-Fi using HTTP, WebSocket, or other protocols, or an App Designer that facilitates rapid desktop application development that integrates directly with MATLAB scripts and plots can be the desired outcomes. An interactive control and cooperation combine with the autonomous behavior of the robot, shaping a complete system integration in the future and marking an important milestone in our project.

## 2.5. Work Package 5: Integration and pitch presentation

This work package is the final milestone of the project, which focuses on combining the outcomes of all previous work packages into a functioning prototype and is ready for real-world demonstrations. As previous packages have addressed simulation, mechanical and electronic design, embedded control, and ROS2-based autonomy of the robot, during this Work Package, we need to ensure that these elements are properly integrated, tested, and combined. The expected outcome of this deliverable includes the final assembly and wiring of the robot, software integration between the ESP32 controller, sensors, ROS2, and user interfaces. Furthermore, we will also prepare a concise and professional video presentation about our project.

### 2.5.1. Deliverable 5.1 Full integration of the Robot System

At this stage of the project, all components, including the mechanical structure, electronics, embedded software, and ROS2, should be combined into a single working system to achieve a stable prototype capable of reliable motion and demonstrate the desired tasks. The chassis, Mecanum wheels, motors, drivers, ESP32, camera module, IMU, LiDAR, and power supply will be assembled and wired according to the CAD and PCB designs from Work Package 2. They will be organized neatly for aesthetic, troubleshooting, and safety. The final version of ESP32 firmware will be uploaded and verified, ensuring correct motor control, encoder, and IMU reading, as well as its communication with the Raspberry Pi. While ROS2 nodes for LiDAR, camera, odometry, and navigation will be configured and tested when used in a real environment. Moreover, a collection of tests will be created to check manual or semi-autonomous control (via a GUI or web interface) of the robot. Basic Mecanum movements such as forward, sideways, and rotating, sensor feedback, and autonomous behaviors like traveling to set waypoints will be taken into consideration. Therefore, any integration issues, such as unstable motion, communication disconnection, and sensor noise, will be identified and mitigated through troubleshooting and fine-tuning of control parameters. The expected outcome of this Deliverable is a fully assembled and tested prototype that can demonstrate the main features of the project in a real-life environment with obstacles, while ensuring camera vision capabilities can produce stable and clear image data. The robot should be able to move in the Cartesian direction, with keyboard, joystick control, or a WebApp developed by our team. The MATLAB simulation shows the potential movement of the robot, and the mapping of the room with the Lidar is needed to showcase our successful implementation.  Additionally, navigation inside the room with nav2 and a scenario solving one of the SDGs chosen by our group should be showcased and clearly described.

### 2.5.2. Deliverable 5.2 PowerPoint Presentation

Ultimately, after completing all previous responsibilities, at this last point of the project, the team will deliver a succinct, realistic demonstration of the final product and the development process on the demonstration day. A brief PowerPoint presentation will be prepared to present the work done throughout this project, including key design decisions and challenges, test results, limitations, and future improvement directions. Additionally, a detailed video presentation will also be recorded and submitted, as the demo time is limited to 10 – 15 minutes per group, which is not enough to explain the details of the project. During this video presentation, each team member will be assigned speaking roles aligned with their contributions so that both technical and non-technical aspects are presented carefully, and they will dive deeper into different aspects of our project. To deliver the best experience, the presentation will ensure clarity and explain the project's sophistication with clear and professional communication skills. The expected outcome of this Deliverable is a concise and convincing pitch, supported by a successful demonstration of the prototype, that clearly expresses the value of our robot and its potential for future extension and real-world deployment.

# 3. Time Management

The project is divided into six major work packages (WP0-WP5), each of which addresses a distinct stage of development, and is scheduled over a 12-week period. Tasks are divided into manageable units, with team members assigned in a clear manner and deadlines specified. To maximize available time, tasks that are independent of one another are often scheduled in parallel. In order to account for learning curves and potential delays, more complex task such as ROS2 development and embedded system integration, are given longer durations and buffers. The timeline guarantees that all deliverables are finished and prepared for submission by Week 12. This structure is reflected in the Gantt chart, which shows minimal idle days for the team, overlapping workflows, and highlighted milestone coordination.

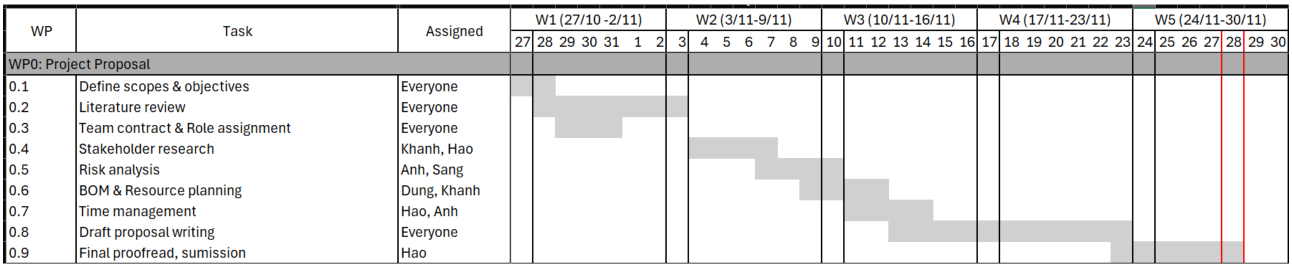


*Figure 18 WP0 Milestone Gantt Chart: Project Proposal*

For the first four weeks, the team focuses on building a strong foundation for the project. To guarantee agreement on team objectives, contract, role responsibilities, we start with cooperative ideation (T0.1), literature review (T0.2), and contract drafting (T0.3). In order to make sure that all important factors are taken into consideration before execution, these efforts work in tandem with stakeholder and application research (T0.4), risk identification (T0.5), and resource planning (T0.6). Realistic scheduling is done through management planning (T0.7) after initial risk and resource awareness has developed. As early as Week 3, draft proposal writing (T0.8) begins, incorporating contributions from every member. To guarantee clarity, consistency and adherence to submission standards, the final proofreading and submission task (T0.9) is assigned to a single member with supervision.

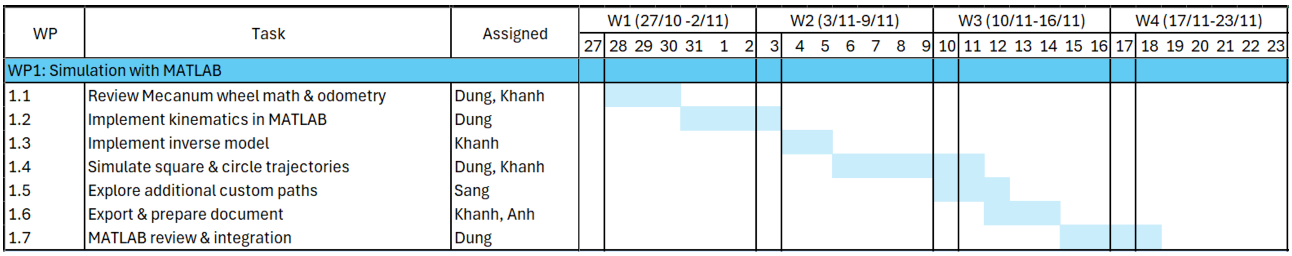


*Figure 19 WP1 Milestone Gantt Chart - Simulation with MATLAB*

The MATLAB simulation stage begins by reviewing Mecanum wheel kinematics and odometry principles (T1.1), followed by implementation of a forward and inverse model (T1.2-T1.3). This mathematical model serves as the foundation for all subsequent trajectory planning (T1.4-T1.5), which includes both standard paths and creative motion testing.

The documentation and export task (T1.6) ensures that all visual data and plots are available for later integration with the ROS2 platform. A final review and cleanup task (T1.7) ensures the simulation results are reliable, clearly presented, and aligned with the physical design constraints developed in WP2. Simulation runs in parallel with early CAD design to maximize time usage.

| WP | Task | Assigned | W2 (3/11-9/11) | | | | | | | W3 (10/11-16/11) | | | | | | | W4 (17/11-23/11) | | | | | | | W5 (24/11-30/11) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| WP2: Prototyping of the Mecanum robot | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.1 | CAD sketch basic layout | Dung | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.2 | Design mounts (LiDAR, sensors, camera) | Khanh | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.3 | Review & 3d print prepare | Sang | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.4 | Submit STL + order print | Dung | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.5 | Schematic Wiring in EasyEDA | Khanh, Sang | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.6 | Assign footprint & pinouts | Sang | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.7 | PCB routing & 3D view | Dung | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.8 | Adjustment after test print / feedback | Sang | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.9 | 3D preview + export for fabrication | Khanh | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.10 | Order PCB + parts | Dung | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2.11 | Export & prepare document | Dung, Hao | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

*Figure 20 WP2 Milestone Gantt Chart: Prototyping of the Mecanum Robot*

WP2 combines CAD and PCB development in parallel over the course of four Sensor mount design (T2.2) and chassis modeling (T2.1) are given priority because they specify the physical dimensions and limitations for subsequent electronics and navigation features. In order to expedite 3D printing, design reviews and print preparation (T2.3-T2.4) must be conducted.

Schematic wiring, footprint assignment, and routing (T2.5-T2.7) all advance concurrently in EasyEDA. Refinement of PCB and CAD models is supported by feedback from CAD prints (T2.8). The hardware design phase ends with final previews, fabrication exports, and part ordering (T2.9-T2.10). The deliverables are completed with document preparation (T2.11).

| WP | Task | Assigned | W5 (24/11-30/11) | | | | | | | W6 (1/12-7/12) | | | | | | | W7 (8/12-14/12) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| WP3: Embedded and control system | | | | | | | | | | | | | | | | | | | | | | | |
| 3.1 | Hardware setup, safety test | Hao, Anh, Khanh | | | | | | | | | | | | | | | | | | | | | |
| 3.2 | Read encoder value (ESP32-PC) | Hao, Dung | | | | | | | | | | | | | | | | | | | | | |
| 3.3 | Implement + tune basic PID (one motor) | Anh | | | | | | | | | | | | | | | | | | | | | |
| 3.4 | Multi-motor abstraction | Hao | | | | | | | | | | | | | | | | | | | | | |
| 3.5 | Serial bridge between ESP32 ↔ PC | Anh, Sang | | | | | | | | | | | | | | | | | | | | | |
| 3.6 | IMU integration + test orientation logic | Hao, Anh | | | | | | | | | | | | | | | | | | | | | |
| 3.7 | Merge motor + IMU + encoder output to ROS2 stub | Dung, Anh | | | | | | | | | | | | | | | | | | | | | |
| 3.8 | Debug + test full firmware + documenting | Hao, Sang | | | | | | | | | | | | | | | | | | | | | |

*Figure 21 WP3 Milestone Gantt Chart: Embedded and Control System*

The embedded phase builds on completed CAD/PCB outputs and begins with hardware bench testing (T3.1), confirming safe operation for smoother progress. Encoder reading and PID tuning (T3.2-T3.3) enable accurate low-level control. These are followed by multi-motor abstraction (T3.4) and serial communication bridging (T3.5), preparing data delivery to ROS2.

IMU integration and orientation logic (T3.6) are developed parallelly, followed by merging of encoder + IMU data into ROS2-compatible odometry (T3.7). Debugging, documentation, and final verification (T3.8) start right after, ensuring system stability. ROS2 bridge logic is timed so ROS2 can begin processing fused odometry as early as Week 5.

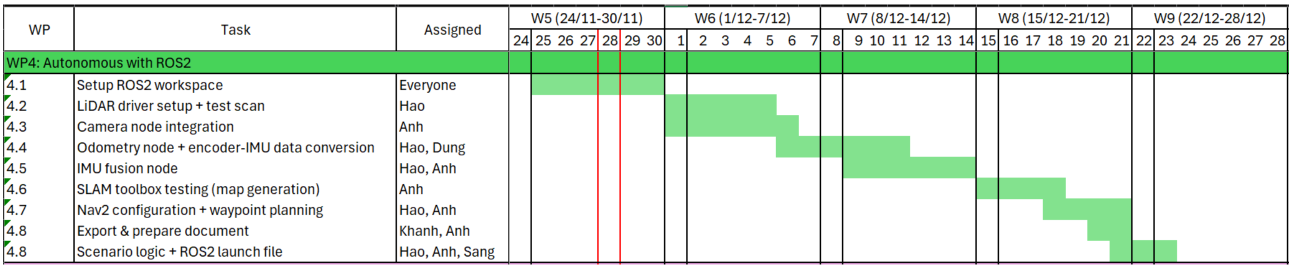


*Figure 22 WP4 Milestone Gantt Chart: Autonomous with ROS2*

As ROS2 is complex and new to the team, WP4 begins in Week 5 and runs until Week 9. The workspace is set up early for the rest of the week (T4.1), helping everyone to get familiar with the tools. Sensor nodes are developed in parallel: LiDAR (T4.2) and camera (T4.3). Odometry node development (T4.4) starts once encoder and IMU outputs are available, followed by the IMU fusion node (T4.5) using robot_localization.

After completing odometry, SLAM testing (T4.6) and Nav2 configuration (T4.7) are scheduled. Finally, the autonomous behavior setup is completed with scenario logic and ROS2 launch integration (T4.9).

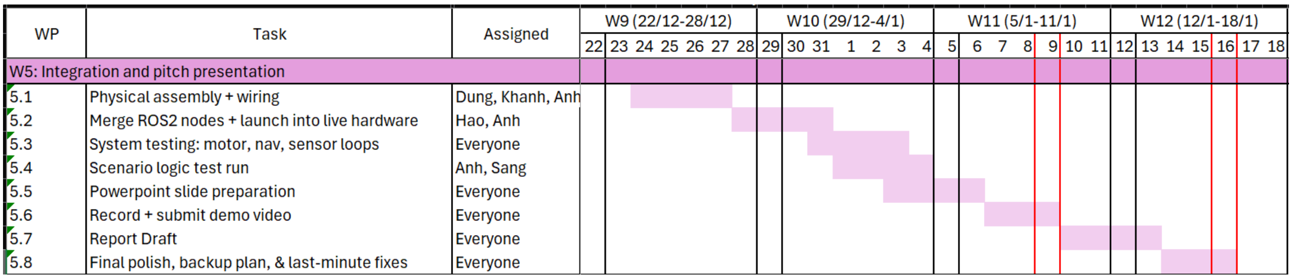


*Figure 23 WP5 Milestone Gantt Chart: System Integration and Presentation*

In Week 9, physical assembly (T5.1) and ROS2 node merging (T5.2) mark the start of system integration. Everyone in the team immediately conducts live testing of the navigation, sensors, and actuators (T5.3) to guarantee complete functionality prior to the demo.

After testing scenario-specific logic (T5.4) for alignment with SDG themes, the PowerPoint slides (T5.5) and demonstration video (T5.6) are jointly created. In parallel, the draft report (T5.7) is written with the help of documents given in the previous Work Packages. Before submission, a final buffer (T5.8) protects against unexpected hardware malfunctions or software regressions

# 4. Resources management

## 4.1. Hardware management

| BILL OF MATERIAL | | | | | | |
|---|---|---|---|---|---|---|
| **Num** | **Part Name** | **Description** | **Qty** | **Unit Cost (VND)** | **Total Cost (VND)** | **Status** |
| 1 | JGB37-520 | DC Geared Motor | 4 | 215,000 | 860,000 | Provided |
| 2 | BTS7960 43A | High-power Motor Driver | 4 | 76,000 | 304,000 | Provided |
| 3 | ESP32-DevKit V1 | 30 pins ESP32, CH340 Type-C development board | 1 | 125,000 | 125,000 | Provided |
| 4 | Terminal Adapter | 30 pins Terminal Adapter for ESP32 DevKit | 1 | 45,000 | 45,000 | Provided |
| 5 | ESP32 - CAM | Bluetooth Camera | 1 | 190,000 | 190,000 | Self-bought |
| 6 | Ovonic 2200mah 4S 120C | 2200mah 4S 120C LiPo Battery | 1 | 570,000 | 570,000 | Self-bought |
| 7 | RPLIDAR A1M8-R6 | 360° Laser Range Scanner | 1 | 2,484,000 | 2,484,000 | Provided |
| 8 | Rasberry Pi 5 (4GB) | Mini-sized computer | 1 | 2,160,000 | 2,160,000 | Provided |
| 9 | Raspberry Pi Camera Module 3 Standard | 75° IMX708 12MP sensor and autofocus camera module | 1 | 864,000 | 864,000 | Provided |
| 10 | IMU ICM-20948 | Motion tracking device | 1 | 262,500 | 262,500 | Provided |
| 11 | PCB | Custom printed IC board | 1 | 180,000 (Expected) | 180,000 (Expected) | Order |
| 12 | Mechanum Wheels | 79mm mechanum wheels | 4 | 91,000 | 366,000 | Provided |
| 13 | Top Plate | Hold SDG service device | 1 | N/A | N/A | Provided |
| 14 | Orange Column | Support top plate and leave space for LiDAR sensor | 2 | N/A | N/A | Provided |
| 15 | LiDAR Base | LiDAR sensor attachment | 1 | N/A | N/A | Provided |
| 16 | Double aluminum extrusion frame | Contribute as main chassis frame. Size 20x40x300 | 2 | 361,169 | 722,338 | Provided |

| | | | | | | |
|---|---|---|---|---|---|---|
| 17 | Single Aluminum Extrusion frame | Contribute as main chassis frame. Size 20x20x300 | 2 | 308,524 | 617,048 | Provided |
| 18 | L-Shape Corner Connector Joint Bracket | Hold 4 aluminum extrusion | 4 | 7,000 | 28,000 | Provided |
| 19 | Front Wall | Electronics container | 2 | N/A | N/A | Provided |
| 20 | Side Wall | | 2 | N/A | N/A | Provided |
| 21 | Bottom Plate | | 4 | N/A | N/A | Provided |
| 22 | Motor Mounting Braket | Secure motor to the chassis frame | 4 | 25,000 | 100,000 | Provided |
| 23 | I-Beam support structure | Connects main chassis to the Motor Mounting Braket | 1 | N/A | N/A | Provided |
| 24 | Hexagon M3 x 8mm Screws | Secure components together | 100 | 400 | 40,000 | Self-aquired |
| 25 | M3 Screw nuts | | 100 | 500 | 50,000 | Self-aquired |
| 26 | Screw drivers set for M3 screws | Drive screws | 1 | 58,000 | 58,000 | Self-aquired |
| 27 | Electrical wire | Electronics connectors | 1 | 23,000 | 23,000 | Provided |
| 28 | Breadboard | Distribution device for electronics | 2 | 16,000 | 16,000 | Self-aquired |
| 29 | Medical container | Contains medical utilities | 1 | 60,000 | 60,000 | Self-aquired |
| **Total item: 29 items** | | | **Total cost: 10,124,000 VND** | | | |

The link for the components can be found in the reference – BOM.

## 4.2. Software management

- Simulation and trajectory planning: MATLAB and Simulink
- Frabication: CAD Design: Fusion; PCB Design: Easy EDA
- Embedded and Control System: Visual Studio Code with Platform IO, Github
- Autonomous Implementation: ROS2
- Project planning: Teams and Planner.

# 5. Organization and Partners

## 5.1. RMIT Staff and students

The development of this robot has a direct relevant with RMIT School of Science Engineering and Technology. It helps students understand about the development of autonomous robot, dealing with hardware and software, and preparing them for the workforce. For lecturers, it helps them to understand even more about the subject, understand insight of student intellect in class, and possibly even further innovation and future development for the robot itself.

## 5.2. Robotics

- **Vin Robotics:**

VinRobotics is a high-technology subsidiary of Vingroup focused on advanced robotics, industrial automation, and AI-driven manufacturing solutions. Established in 2024, the company aims to develop intelligent robotic systems for production, services, and future smart-factory applications in Vietnam. With strong financial backing and national-scale infrastructure, VinRobotics represents a strategic stakeholder for innovation projects, providing potential pathways for collaboration, technology transfer, and real-world deployment of autonomous robotic platforms.

- **ABB Robotics**

ABB Robotics is a global leader in industrial automation, robotics, and digital manufacturing solutions. The company specializes in robotic arms, factory automation systems, and AI-enabled production technologies. With a strong presence across automotive, manufacturing, and logistics sectors, ABB represents a key stakeholder for projects involving advanced robotics, system integration, and high-precision automation.

- **Ecovacs Robotics**

Ecovacs Robotics is a global consumer robotics company specializing in autonomous cleaning robots, sensing systems, and home-focused AI navigation technologies. Their experience in mass-producing reliable, low-cost autonomous mobile platforms positions them as a relevant stakeholder for projects exploring indoor mobility, sensor fusion, and autonomous path-planning in compact environments.

## 5.3. Tech

- **FPT Robotics & FPT Software**

FPT is one of Vietnam’s largest technology corporations, with active development across AI, robotics, automation, and digital transformation. Through FPT Software and FPT Smart Factory, the company provides end-to-end solutions involving industrial IoT, autonomous systems, computer vision, and AI-driven automation. Their growing focus on robotics R&D positions FPT as a strategic domestic stakeholder capable of supporting integration, software development, and large-scale deployment of intelligent robotic platforms.

- **Samsung Vietnam / Samsung R&D**

Samsung operates major manufacturing and R&D centers in Vietnam, focusing on electronics, AI, advanced manufacturing systems, and smart factory technologies. Samsung's robotics initiatives emphasize automation efficiency, high-precision assembly, and next-generation intelligent systems. Their presence as a stakeholder highlights the region's increasing adoption of robotics for manufacturing optimization, while providing potential pathways for collaboration, internships, and technology co-development related to autonomous robotics and industrial automation.

## 5.4. Medical

- **GE Healthcare**

GE Healthcare drives innovation in medical imaging, diagnostics, and hospital automation systems. As hospitals scale in complexity, autonomous robots supporting routine logistics, specimen movement, and supply-chain automation become increasingly valuable. GE Healthcare's emphasis on digitalized hospital systems makes them a fitting stakeholder for autonomous indoor mobility platforms.

- **Omron Healthcare**

Omron operates across medical devices, sensing technology, and AI-driven health systems. Their robotics initiatives extend into hospital automation and intelligent monitoring. Autonomous mobile robots can integrate with Omron's sensor ecosystem to support safe navigation, contactless delivery, and automated patient-support services in medical facilities.

- **Local Hospitals & Medical Networks (e.g., Vinmec, Hoan My, Tam Anh Hospital)**

Vietnamese hospital systems are rapidly modernizing through digital transformation and smart-hospital initiatives. Facilities such as Vinmec International Hospital already adopt automation and robotics for tasks like pharmacy management, diagnostics, and patient-care workflows. Autonomous robots capable of precise indoor navigation address growing needs in sample transport, medication delivery, and supply movement, making hospital networks essential end-user stakeholders.

## 5.5. Defences

- **Viettel High Tech**

Viettel High Tech, the R&D arm of Viettel Group, focuses on advanced electronics, communication systems, AI platforms, robotics, and defense-grade technologies. Their capabilities span sensor development, embedded systems, wireless communication, and autonomous control solutions. As a stakeholder, Viettel High Tech offers strong alignment with high-reliability robotics, secure communication protocols, and intelligent monitoring or automation technologies suited for industrial and national-scale applications.

# 6. Risk Analysis

*Table 1 Risk Impact Table*

| Level | Consequence | Description |
|---|---|---|
| 1 | Insignificant | Negligible effect: no impact on deliverables, minor adjustments |
| 2 | Minor | Small effect: slight delay, small hardware/software rework |
| 3 | Moderate | Noticeable effect: schedule delay (1-3 days) or rework of subsystem, may have minor cost increase |
| 4 | Major | Serious effect: significant delay (> 7 days), multiple subsystems require redesign, milestone impact |
| 5 | Catastrophic | Critical failure: project cannot continue, or safety hazard arises; major schedule or budget failure |

*Table 2 Risk probability table*

| Level | Likelihood | Description |
|---|---|---|
| 1 | Rare | Very low chance: may occur in exceptional circumstances |
| 2 | Unlikely | Low chance: could occur at some time, but not expected in normal operation |
| 3 | Possible | Moderate chance: might occur during the project, though not frequent |
| 4 | Likely | High chance: will occur in most circumstances without mitigation |
| 5 | Almost certain | Very high chance: expected to occur in most cases |

*Table 3 Risk Evaluation Matrix*

| Likelihook | Consequences | | | | |
|---|---|---|---|---|---|
| | Insignificant | Minor | Moderate | Major | Catastrophic |
| Rare | 1 | 2 | 3 | 4 | 5 |
| Unlikely | 2 | 4 | 6 | 8 | 10 |
| Possible | 3 | 6 | 9 | 12 | 15 |
| Likely | 4 | 8 | 12 | 16 | 20 |
| Almost Certain | 5 | 10 | 15 | 20 | 25 |

**Matrix explaination:**

>12 Extreme (E): immediate intervention required.

8-12 High (H): serious risk; must be monitored closely.

4-6: Moderate (M): manageable but should not be ignored.

1-3: Low (L): negligible; routine oversight suffices.

*Table 4 Risk Analysis*

| Number | Risk | Likelihood | Impact | Score (LxC) | Level | Migitation |
|---|---|---|---|---|---|---|
| Technical Risks | | | | | | |
| 1 | IMU malfunction causing severe odometry drift; Mecanum slip amplifies drift. Dead-wheel odometry fallback too complex. | 3 | 3 | 9 | High | Early calibration, long-run tests, filtering. Keep dead-wheel odometry only as a theoretical backup. |
| 2 | Motor driver (IBT-2) overheating or failing under load | 2 | 4 | 8 | High | Current monitoring, ramped acceleration, spare drivers |
| 3 | Incorrect encoder readings create PID instability. | 3 | 3 | 9 | High | Shielded wiring, incremental tuning, test each wheel separately |
| 4 | ROS2 navigation misconfiguration (TF tree, cost maps) | 3 | 3 | 9 | High | Validate TF in RViz2, reuse working configs |
| 5 | LiDAR misalignment causing distorted scans | 2 | 3 | 6 | Moderate | Mechanical leveling, parameter tuning, stable mounting |
| 6 | Wi-Fi instability between ESP32 and Raspberry Pi. | 2 | 3 | 6 | Moderate | Dedicated hotspot, retry logic, controlled publish rate |
| 7 | Battery voltage fluctuations causing resets or weak motor output | 3 | 2 | 6 | Moderate | Clean power rails, buck converter smoothing, voltage monitoring |
| Non-Technical Risks | | | | | | |
| 8 | Minor scheduling conflicts or casual meeting delays | 2 | 2 | 4 | Moderate | Shared calendar, following team contract, meeting minutes |
| 9 | Late PCB or small hardware items arriving close to deadline | 2 | 2 | 4 | Moderate | Order early, maintain backups, prototype on breadboard first |
| 10 | Members struggling with ROS2 learning curve | 3 | 2 | 6 | Moderate | Tutorials early, assign ROS2 leads, reuse reference configs, seeks help from lecturer |
| 11 | Internal miscommunication causing duplicated work or minor inconsistencies | 2 | 2 | 4 | Moderate | Clear task allocation, weekly updates, shared documentation |
| 12 | Minor safety issues | 1 | 2 | 2 | Low | Basic safety rules, handle LiPo carefully, test motors on stands |
| 13 | Documentation inconsistencies | 2 | 1 | 2 | Low | Final proofreading, version control, shared editing |

# 7. Team introduction and team contract

## 7.1. Team introduction

Teams have always been a crucial factor in the success of any product development cycle. In an engineering project, this is even more evident, as the outcome relies on individuals with different areas of expertise working together towards a common goal. This section highlights the skills and responsibilities of each team member, acknowledging their contributions and commitment throughout this project. The following profiles provide a brief overview of each team member, showcasing their background, strengths, and role within the development of the robot.

**SELF-INTRO**

Vu Thien Minh Hao – s3938011

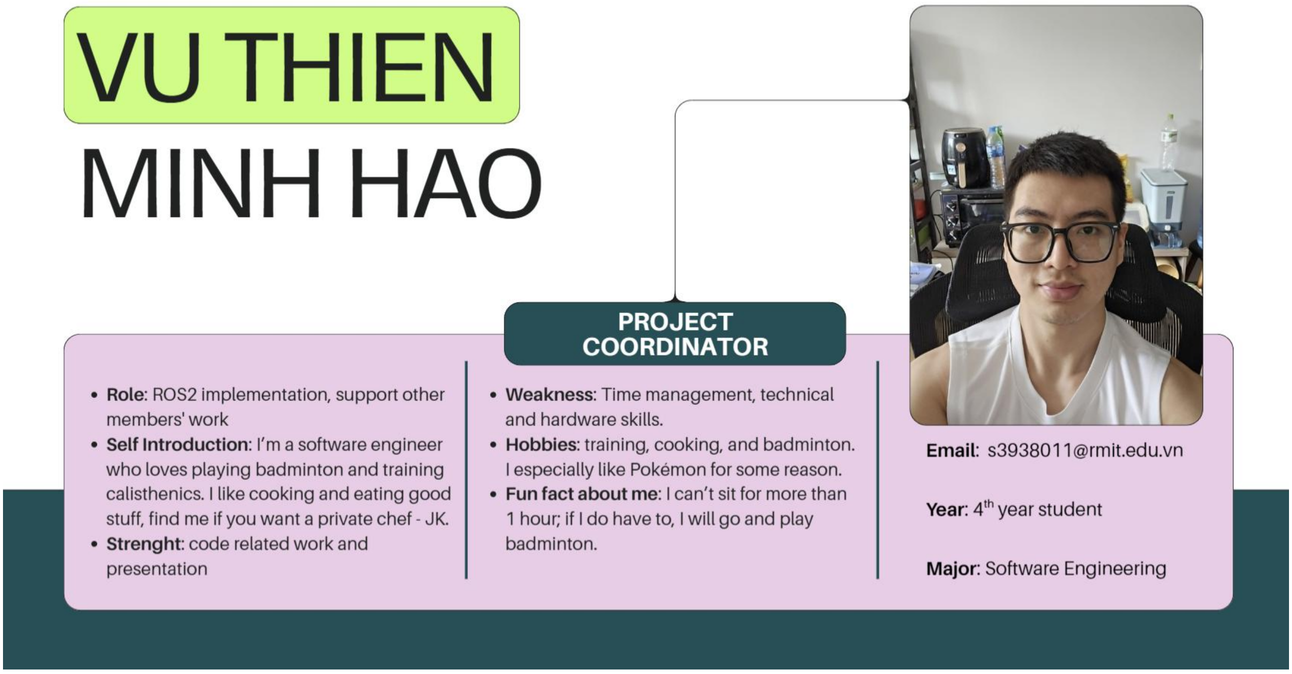


*Figure 24 Hao member's Introduction*

**SELF-INTRO**

Pham Tri Dung – s3981129

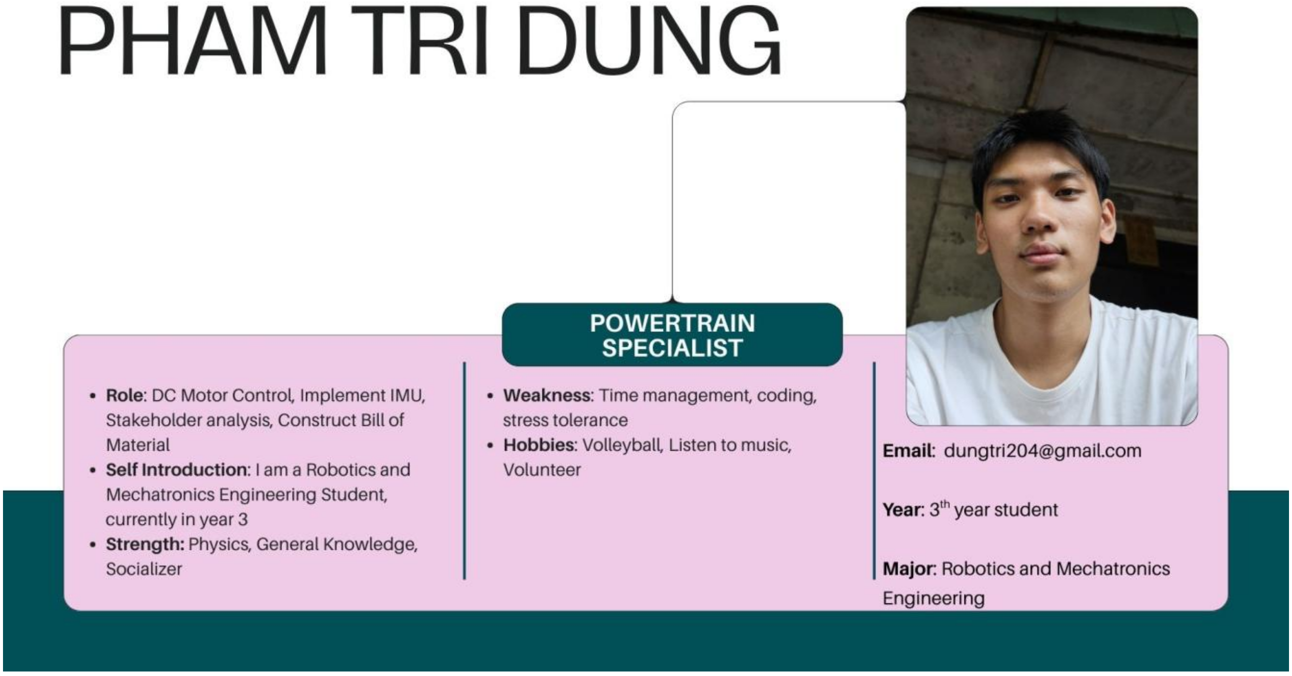


*Figure 25 - Dung's member Introduction*

**SELF-INTRO**

Phung Nam Khanh- s4033609

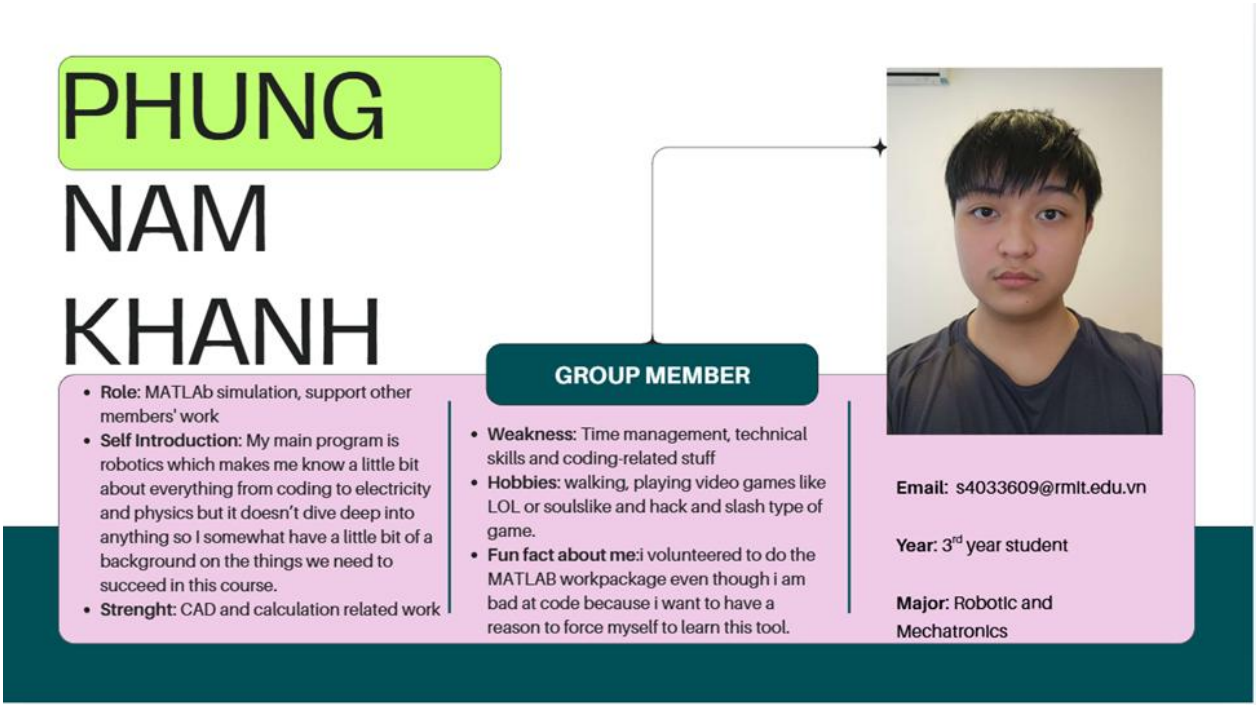


*Figure 26 Khanh's member Introduction*

**SELF-INTRO**

Tran Thanh Sang – s4028016

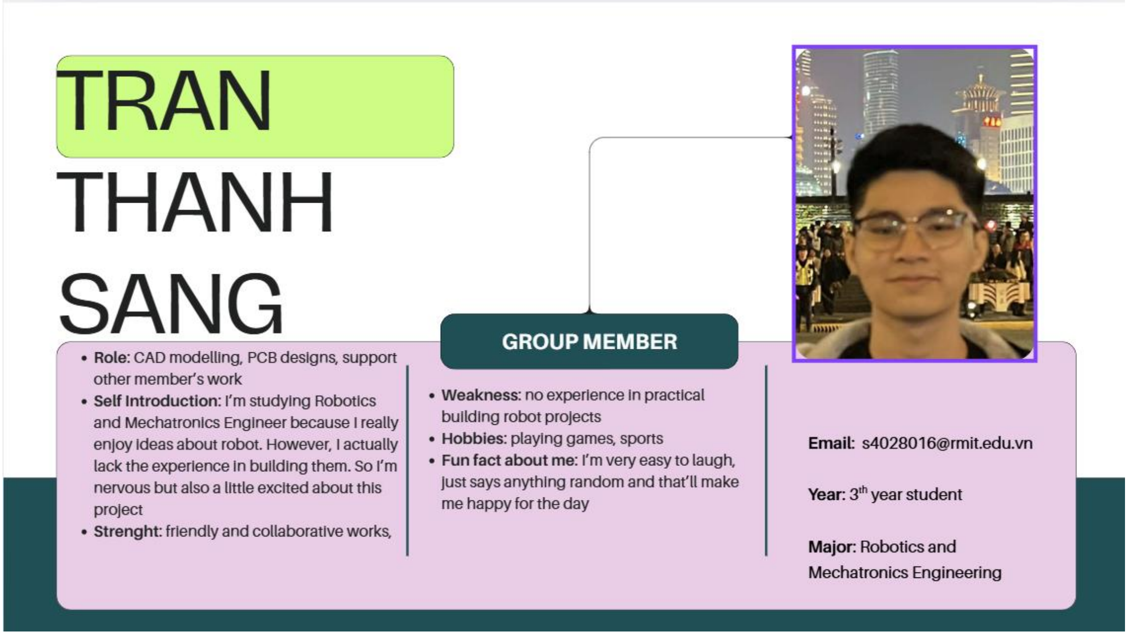


*Figure 27 Sang's member Introduction*

**SELF-INTRO**

Huynh Nhat Anh – s3924763

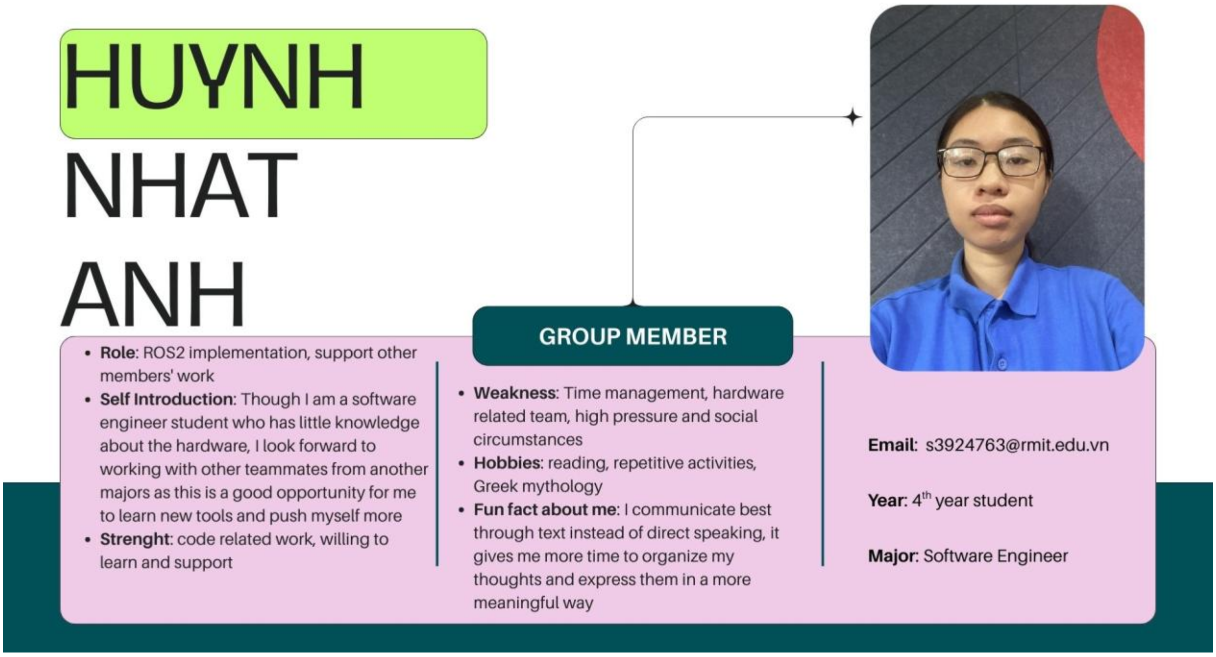


*Figure 28 Anh's member Introduction*

### 7.2. Team Contract

| Team A Contract |
|---|
| **Team members' Information, Roles, and contact Email:**<br>• Vu Thien Minh Hao – Software Engineer: RO2S implementation, website application, and remote control of the robots – s3938011@rmit.edu.vn<br>• Pham Tri Dung – Mechatronics Engineer: DC Motor control, CAD Modelling, PCB Design, Stakeholder Analysis, Construct Bill of Materials - s3981129@rmit.edu.vn<br>• Phung Khanh Nam – Mechatronics and Robotics: MATLAB simulation, literature review, DC motor. – s4033609@rmit.edu.vn<br>• Tran Sang-: PCB design, CAD, MATLAB simulation. – s4028016@rmit.edu.vn<br>• Huynh Nhat Anh - Embedded support, IMU and sensor integration, ROS2 implementation – s3924763@rmit.edu.vn |
| **Deliverables**<br>**A deliverable outlines a physical result of the project so that others may imagine what the final product should look like.**<br>• In this project, it is to produce an AGV with a front camera that can be remotely controlled and monitored with a website application. It also has to have integrated autonomous behaviors of ROS2, which allows it to automatically navigate through an environment while avoiding obstacles. |
| **Objectives**<br>**Describe the expected outcome, skills, knowledge, and experience gained from this project**<br>• Enhance team collaboration, project management, and communication skills<br>• Elevate mechanical design, electronics, and programming knowledge<br>Deliver the requirement-satisfied AGV that can be controlled by a website application |
| **Meetings and Communications**<br>**How each member should connect and interact with each others.**<br>• Must meet at least once weekly, either online or offline.<br>• Must notify the team at least 4 hours beforehand if unable to attend a meeting. If this happens more than two times in a row, a valid reason is required. If no valid reasons are provided, that team member will be fined and have to buy a cup of coffee for every other team member. Repeated absences will result in a deduction from the team's contribution score. |

- The team has the right to reconsider one's contribution in terms of idea-sharing if a member repeatedly misses meetings without a valid reason and doesn't make up the work.
- The team mainly communicates using Messenger and Outlook, holds online meetings, sets deadlines, and plans the project on Microsoft Teams. Each team member is expected to check messages regularly, communicate proactively, respond quickly, and help clarify mutually.
- A meeting minute is required for each meeting, either online or offline. This helps other members remember the main purpose of each meeting and allows work tracking to be easier.
- During the meeting, team members are expected to contribute proactively, share their thoughts, and support others.

**Task allocation**

**How tasks should be allocated and how team members should behave towards an assigned task**

- All tasks must be agreed upon by the whole team and recorded either online or via Teams allocation or online messages. If tasks are given during an offline meeting, they must still be recorded online and assigned to the desired member.
- All tasks must have a deadline. Assigning a reviewer is optional but encouraged.
- If a member misses a task's deadline without a valid reason, the team has the right to reassign it. This will affect that member's contribution score.
- All members are expected to finish the task within the allocated timeframe and are encouraged to help others if possible. Members who delay tasks frequently will be fined and have to buy every other member a cup of coffee.

**Resolving Conflict**

**How the team agrees to solve conflicts during the project**

- All conflicts must first be discussed in the group, whether online or offline, where everyone can participate. All members must express their thoughts and ideas to resolve a problem.
- Conflicts should be resolved respectfully and according to the team contract. If a conflict is not resolved after two attempts, the issue will be escalated to the lecturer for further support.

**Commitment and Code of Conduct**

**Good practices and expected commitment from each member during the development process of the project**

- Tasks should be equally distributed among members, considering their expertise.

- Every member is expected to support others regardless of expertise. Refusing to help without a valid reason is not acceptable.
- Members are required to support each other physically and mentally. If any members struggle to complete their tasks, members who have finished are welcome to help out.
- When working with hardware, safety comes first.
- Unless damaged intentionally, hardware is a shared responsibility if any unexpected accidents happen.
- Academic dishonesty is not acceptable.

Each team member is expected to contribute to the group's budget for the project. However, any arising costs should be discussed beforehand and equally distributed.

**Member's signature**

**Tran Thanh Sang**

**Huynh Nhat Anh**

**Pham Tri Dung**

**Vu Thien Minh Hao**

**Phung Nam Khanh**

## 8. Conclusion

All in all, this project offers numerous challenges and difficulties, but it will also provide us with valuable experience and precious knowledge. The work is structured into clear work packages covering MATLAB simulation, mechanical and electronic design, embedded control on the ESP32, ROS2-based autonomy, and final system integration and presentation. Within the 12-week timeframe, the primary goal is to deliver a functional prototype capable of stable omnidirectional motion, and autonomous behaviors such as basic sensing, mapping, and navigating. In order to achieve this within such a constrained time frame, a realistic and meticulous time plan, resource management, and risk analysis will be developed to guide and instruct us along the journey. Collaboration and professional communication between members of different study backgrounds are needed to promote the accomplishments of the project.

If the project is a grand success, it will provide amazing and helpful support for hospital logistics. By helping out existing issues such as staff workload, human error in deliveries, and infection risk, the project aims to demonstrate how an omnidirectional, sensor-equipped platform can assist with internal transport tasks in an important and sensitive environment and help to address the **Good Health and Well-Being (SDG 3).** The project also contributes to **Decent Work and Economic Growth (SDG 8)** by reducing repetitive, low-value tasks for medical staff and allowing them to focus on higher-skilled care. The project further aligns with **Industry, Innovation and Infrastructure (SDG 9)** through the development of a modern, robotics-based logistics solution, and with **Sustainable Cities and Communities (SDG 11)** by exploring safer and more efficient use of shared indoor spaces in public healthcare facilities.

Future work beyond this course may extend the system towards more advanced ROS2 navigation, deeper integration with hospital workflows and industrial environments, and richer perception and interaction capabilities, building on the foundation established by this project.

# REFERENCES


[1] A. Waga and K. Chatterjee, “A survey on autonomous navigation for mobile robots,” *SN Computer Science*, vol. 6, no. 2, 2025. Accessed: Nov. 28, 2025. [Online]. Available: https://doaj.org/article/a2a7135198a1456e9a813a27a65991ed

[2] Í. R. da Costa Barros, A. R. de Almeida, and A. A. de Souza Junior, “Robotic mobile fulfillment systems: A survey on recent developments and challenges,” *Robotics and Autonomous Systems*, vol. 140, 2021. Accessed: Nov. 28, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0921889021000142

[3] H. Taheri, “Omnidirectional mobile robots, mechanisms and navigation systems: A review,” *Mechanism and Machine Theory*, vol. 150, 2020. Accessed: Nov. 28, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0094114X20301798

[4] K. Kanjanawanishkul, “Omnidirectional wheeled mobile robots: Wheel types and practical applications,” *International Journal of Advanced Mechatronic Systems*, vol. 6, no. 6, pp. 289–302, 2015. Accessed: Nov. 28, 2025. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJAMECHS.2015.074788

[5] J. Qian, H. Hu, L. Wang, Z. Wang, and M. Tomiyama, “The design and development of an omni-directional mobile robot,” *Journal of Healthcare Engineering*, vol. 2017, Article ID 5094329, 2017. Accessed: Nov. 28, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/17/9/2073

[6] G. N. DeSouza and A. C. Kak, “Vision for mobile robot navigation: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002. Accessed: Nov. 28, 2025. [Online]. Available: https://dl.acm.org/doi/10.1109/34.982903

[7] N. Gryaznov, P. Kulakov, and A. Smirnov, “Computer vision for mobile on-ground robotics,” *Procedia Manufacturing*, vol. 3, pp. 3282–3289, 2015. Accessed: Nov. 28, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978915004394

[8] Y. Wu, J. Zhang, F. Chen, and B. Liang, “A teleoperation interface for loco-manipulation control of mobile manipulators,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3574–3585, 2020. Accessed: Nov. 28, 2025. [Online]. Available: https://re.public.polimi.it/bitstream/11311/1119793/1/08764016.pdf

[9] J. Janßen, “Building a user-interface for a teleoperated mobile robot in laboratory automation,” M.Sc. thesis, TU Wien, Vienna, Austria, 2022. Accessed: Nov. 28, 2025. [Online]. Available: https://repositum.tuwien.at/handle/20.500.12708/80608

[10] T. Borbély, “Kinematic model of a four Mecanum wheeled mobile robot,” *International Journal of Computer Applications*, vol. 113, no. 3, pp. 24–29, 2015. Accessed: Nov. 28, 2025. [Online]. Available: https://www.ijcaonline.org/archives/volume113/number3/19804-1586

[11] A. Tzafestas, “Service robots in the healthcare sector,” *Robotics*, vol. 10, no. 1, p. 47, 2021. Accessed: Nov. 28, 2025. [Online]. Available: https://www.mdpi.com/2218-6581/10/1/47

[12] Q. Xu, Y. Li, H. Huang, and J. Wang, “Advancing healthcare through mobile collaboration: A survey of intelligent nursing robots research,” *BMC Medical Informatics and Decision Making*, vol. 24, no. 1, 2024. Accessed: Nov. 28, 2025. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpubh.2024.1368805

[13] N. H. Thai, N. T. Thinh, and N. D. Quang, “Trajectory tracking control for Mecanum wheel mobile robot by time-varying parameter PID controller,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 1902–1910, 2022. Accessed: Nov. 28, 2025. [Online]. Available: https://beei.org/index.php/EEI/article/view/3712

[14] D. Wang, C. Xu, and Z. Li, “Sliding mode observer-based model predictive tracking control for Mecanum-wheeled mobile robot,” *ISA Transactions*, vol. 151, pp. 51–61, 2024. Accessed: Nov. 28, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057824002672

[15] Y. Zhang, G. Chen, and X. Zhou, “Trajectory tracking control of Mecanum wheel omnidirectional mobile robot based on power reaching law,” *Journal of Shanghai University of Engineering Science*, vol. 36, no. 1, pp. 23–30, 2022. Accessed: Nov. 28, 2025. [Online]. Available: https://xuebao.sues.edu.cn/cn/article/doi/10.12299/jsues.21-0195

[16] S. T. Nowroz and M. S. Rahman, “Smart surveillance rover: Real-time monitoring with ESP32-CAM and pan-tilt servo motor integration,” *International Journal of Research in Applied Science and Engineering Technology*, vol. 11, no. 3, 2023. Accessed: Nov. 28, 2025. [Online]. Available: https://www.ijraset.com/research-paper/smart-surveillance-rover-real-time-monitoring-with-esp32-cam

[17] B. A. Kumar, “Robot vehicle with ESP32-CAM,” *International Journal of Research in Applied Science and Engineering Technology*, vol. 12, no. 4, 2024. Accessed: Nov. 28, 2025. [Online]. Available: https://www.ijraset.com/research-paper/robot-vehicle-with-esp32-cam

[18] V. V. Grankin, A. A. Dymova, and A. V. Midkhat, “Research on object recognition approaches for mobile systems with limited resources,” *Proc. Int. Workshop on Computer Vision for IoT Systems*, CEUR-WS, 2024. Accessed: Nov. 28, 2025. [Online]. Available: https://ceur-ws.org/Vol-3970/PAPER12.pdf

[19] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, eabm6074, 2022. Accessed: Nov. 28, 2025. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.abm6074

[20] S. Macenski, F. Martín, R. White, and J. G. Clavero, “The Marathon 2: A navigation system,” *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2020, pp. 2718–2725. Accessed: Nov. 28, 2025. [Online]. Available: https://dl.acm.org/doi/10.1109/IROS45743.2020.9341207

# BOM


- **[1]** "DC Geared Motor," HShop, [Online]. Available: https://hshop.vn/dong-co-dc-servo-giamtoc-ga37-v1. [Accessed: Nov. 28, 2025].

- **[2]** "High-power Motor Driver BTS7960 43A," HShop, [Online]. Available: https://hshop.vn/mach-dieu-khien-dong-co-dc-bts796043a-1-dong-co. [Accessed: Nov. 28, 2025].

- **[3]** "ESP32-DevKit V1, 30 pins ESP32, CH340 Type-C development board," LinhKienVietnam, [Online]. Available: https://linhkienvietnam.vn/module-node-mcu-32s-esp32-devkitc-dung-module-esp-wroom-32?srsltid=AfmBOop0R7GU2lBzXtln2cFOAP-HCvt_BAQ9yu8k5bfEhvcRcSA3wG8m. [Accessed: Nov. 28, 2025].

- **[4]** "Terminal Adapter for ESP32 DevKit," MShop, [Online]. Available: https://mshop.io.vn/de-ra-chan-esp32-30-chan-38-chan. [Accessed: Nov. 28, 2025].

- **[5]** "ESP32 - CAM Bluetooth Camera," TheGioiIC, [Online]. Available: https://www.thegioiic.com/module-esp32-cam-bluetooth-camera-ov2640. [Accessed: Nov. 28, 2025].

- **[6]** "Ovonic 2200mah 4S 120C LiPo Battery," MoodFPV, [Online]. Available: https://moodfpv.com/san-pham/pin-lipo-ovonic-2200mah-4s-120c/. [Accessed: Nov. 28, 2025].

- **[7]** "RPLIDAR A1M8-R6 Laser Range Scanner," HShop, [Online]. Available: https://hshop.vn/cam-bien-laser-radar-lidar-rplidar-a1. [Accessed: Nov. 28, 2025].

- **[8]** "Raspberry Pi 5 (4GB)," HShop, [Online]. Available: https://hshop.vn/may-tinh-raspberry-pi-5-made-in-uk. [Accessed: Nov. 28, 2025].

- **[9]** "Raspberry Pi Camera Module 3," HShop, [Online]. Available: https://hshop.vn/raspberry-pi-camera-module-3-imx708-12mp-sensor-and-autofocus. [Accessed: Nov. 28, 2025].

- **[10]** "ICM-20948 Motion Tracking Device," LKCG, [Online]. Available: https://lkcg.vn/cam-bien-gia-toc-9-truc-icm-20948. [Accessed: Nov. 28, 2025].

- **[11]** "Custom Printed IC Board," ThienLampPCB, [Online]. Available: https://thienlampcb.com/san-pham/dat-mach-in-tphcm-2-lop-kich-thuoc-nho-hon-1010cm/. [Accessed: Nov. 28, 2025].

- **[12]** "Mechanum Wheels 97mm," NShopVN, [Online]. Available: https://nshopvn.com/product/banh-xe-mecanum-nhua-da-huong-97mm/?srsltid=AfmBOorQAbUeWzqUpks7CPTdf4z1ef_VJMO8ZDOQM7HM6Hcpavg6SotEWcw. [Accessed: Nov. 28, 2025].

- **[13]** "Aluminum Extrusion Frame 20x40x300," Misumi, [Online]. Available: https://vn.misumi-ec.com/vona2/detail/110302685750/?srsltid=AfmBOoplT4sS5tfedZFsrlm06J4wKAYHIb7rYq-G_bMZlRLWwE8xPLOP. [Accessed: Nov. 28, 2025].

- **[14]** "Aluminum Extrusion Frame 20x20x300," Misumi, [Online]. Available: https://vn.misumi-ec.com/vona2/detail/110302685660/?srsltid=AfmBOopKvWst4c28KnsSmHLXmhrSCjCT-fLifhFTGlXYaqD_eHI1oknj. [Accessed: Nov. 28, 2025].

- **[15]** "Hexagon M3 x 8mm Screws," Shopee, [Online]. Available: https://shopee.vn/product/1618519827/44416687891?gads_t_sig=VTJGc2RHVmtYMTlxTFVSVVRrdENkZmdTUmpMeSszRElEdm91eXFrOVhzN0xQZHVqT2pBUnhGRHZyeU0vUCtZdjBWeDYxSkdtOWNOc0poZnZFOGh0Wk5GMXFQYkUwREgreWpTeGxCUU02a3R4RWpJVDhiR3orOFBJOGptcTZTZmFtUitzREQwZWZERHZ2VmYxZl. [Accessed: Nov. 28, 2025].

- **[16]** "DC Motor Coupling 37mm," HShop, [Online]. Available: https://hshop.vn/go-dong-co-duong-kenh-37mm. [Accessed: Nov. 28, 2025].

- **[17]** "M3 x 8mm Stainless Steel Hexagon Bolt," OCVitmin, [Online]. Available: https://ocvitmin.com/slgm3i8b-oc-luc-giac-dau-dep-mong-m3x8mm-inox304. [Accessed: Nov. 28, 2025].

- **[18]** "M3 x 8mm Stainless Steel Bolt," OCVitmin, [Online]. Available: https://ocvitmin.com/sum3mi-dai-oc-luc-giac-mong-m3-tan-inox304. [Accessed: Nov. 28, 2025].

- **[19]** "Test Board for Electronics," Shopee, [Online]. Available: https://shopee.vn/product/376012490/19654119450?gads_t_sig=VTJGc2RHVmtYMTlxTFVSVVRrdENkUm9yWVZMbGw0Q214S0d1TzNEZmUwNldnSElwZllMUkoxalA0a3I2czBKRGZFUnl1d1VoRkU5dXYrZXgyZjdaZjE0VzZFdkFROUkwVzlUOEd5Sm42MGNCL1VNZFFGekJDRVoxY0FGT011ZWk&gad_source=1. [Accessed: Nov. 28, 2025].

- **[20]** "Test Board for Small Components," HShop, [Online]. Available: https://hshop.vn/day-camtest-board-duc-duc-10cm40soi. [Accessed: Nov. 28, 2025].

- **[21]** "Small Test Board," HShop, [Online]. Available: https://hshop.vn/test-board-camnho-8-5-x-5-5-cm. [Accessed: Nov. 28, 2025].

- **[22]** "Medical Utility Box," HangGiaDungNhuaVietNhat, [Online]. Available: https://hanggiadungnhuavietnhat.com.vn/products/hop-dung-do-y-te-6576-1. [Accessed: Nov. 28, 2025].