Faculty of Electrical Technology and Engineering

DEVELOPMENT OF AUTONOMOUS ROBOT FOR ROOM MAPPING USING LIDAR

MUHAMAD RUSYAIDI HAZIM BIN SOFIAN

Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours

2024

DEVELOPMENT OF AUTONOMOUS ROBOT FOR ROOM MAPPING USING LIDAR

MUHAMAD RUSYAIDI HAZIM BIN SOFIAN

**A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics)
with Honours**

Faculty of Electrical Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I declare that this project report entitled "Development of Autonomous Robot for Room Mapping Using Lidar" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Student Name	:	MUHAMAD RUSYAIDI HAZIM BIN SOFIAN
Date	:	09/12/2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology (Industrial Automation & Robotics) with Honours.

Signature : ....

Supervisor Name : TS. DR. MOHD HANIF BIN CHE HASAN

....

Date : 09/12/2024

....

DEDICATION

I am profoundly grateful for this opportunity to embark on the development of The Autonomous Robot for Room Mapping Using Lidar. This project has allowed me to delve into the realms of robotics, a field I have always been passionate about. Through countless hours of research, testing, and refining, I have learned invaluable lessons about precision, innovation, and perseverance. This project represents more than just a technological endeavour; it signifies my dedication to pushing the boundaries of what is possible.

As I navigated the complexities of lidar technology, I faced numerous challenges, each one a stepping stone that strengthened my resolve. From calibrating sensors to ensuring accurate room mapping, every obstacle was a learning experience. I have gained not only technical skills but also a deeper understanding of the importance of teamwork, collaboration, and the relentless pursuit of excellence.

This journey has been a testament to my commitment to continuous learning and improvement. Every breakthrough, no matter how small, has been a victory that fuels my passion for robotics. The knowledge and skills I have acquired during this project will undoubtedly serve as a foundation for future endeavours in the field.

In reflecting on this journey, I recognize the invaluable support of my mentors, peers, and loved ones. Their encouragement and guidance have been instrumental in my progress. This project is not just an achievement of mine, but a collective effort that highlights the power of collaboration and shared vision.

Ultimately, this project has reinforced my belief in my capabilities and potential. I look forward to the future with optimism and excitement, ready to take on new challenges and make meaningful contributions to the world of robotics.

ABSTRACT

The project report titled "Development of Autonomous Robot for Room Mapping Using Lidar" by Muhamad Rusyaidi Hazim Bin Sofian explores the creation of an autonomous robot designed to enhance indoor mapping efficiency through the integration of lidar technology. The primary aim of the research is to improve spatial awareness and navigation capabilities for accurate real-time mapping of indoor environments, thereby addressing practical challenges in various applications such as cleaning, security, and delivery. Aims: The research aims to develop a robust mapping system that leverages lidar sensors to facilitate autonomous robot navigation, aligning with Sustainable Development Goal 9 by fostering innovation in infrastructure development through advanced robotics. Methods: The methodology involves integrating lidar sensors into the robot and developing algorithms to process lidar data for spatial mapping. The system's performance is evaluated through a series of indoor experiments to ensure accuracy and reliability. Results: The findings indicate that the autonomous robot equipped with lidar technology can effectively map indoor environments with a high degree of precision. The robot's enhanced navigation capabilities demonstrate significant potential for practical applications, contributing to the development of sustainable infrastructure. Conclusion: The study concludes that lidar-based mapping systems are instrumental in advancing autonomous robotics for indoor use. The project's outcomes highlight the impact of integrating advanced sensor technologies in robots, thereby promoting innovation and addressing societal and environmental challenges. Future research should focus on refining the algorithms for improved performance and exploring additional applications of the technology.

ABSTRAK

Laporan projek bertajuk "Pembangunan Robot Autonomi untuk Pemetaan Ruang Menggunakan Lidar" oleh Muhamad Rusyaidi Hazim Bin Sofian meneroka penciptaan robot autonomi yang direka untuk meningkatkan kecekapan pemetaan dalaman melalui integrasi teknologi lidar. Tujuan utama penyelidikan ini adalah untuk meningkatkan kesedaran ruang dan keupayaan navigasi untuk pemetaan masa nyata yang tepat dalam persekitaran dalaman, sekali gus menangani cabaran praktikal dalam pelbagai aplikasi seperti pembersihan, keselamatan dan penghantaran. Tujuan: Penyelidikan ini bertujuan untuk membangunkan sistem pemetaan yang kukuh dengan memanfaatkan sensor lidar untuk memudahkan navigasi robot autonomi, sejajar dengan Matlamat Pembangunan Lestari 9 dengan memajukan inovasi dalam pembangunan infrastruktur melalui robotik canggih. Kaedah: Metodologi melibatkan integrasi sensor lidar ke dalam robot dan pembangunan algoritma untuk memproses data lidar untuk pemetaan ruang. Prestasi sistem dinilai melalui beberapa eksperimen dalaman untuk memastikan ketepatan dan kebolehpercayaan. Hasil: Penemuan menunjukkan bahawa robot autonomi yang dilengkapi dengan teknologi lidar boleh memetakan persekitaran dalaman dengan tahap ketepatan yang tinggi. Keupayaan navigasi yang dipertingkatkan robot menunjukkan potensi yang ketara untuk aplikasi praktikal, menyumbang kepada pembangunan infrastruktur yang mampan. Kesimpulan: Kajian ini menyimpulkan bahawa sistem pemetaan berasaskan lidar adalah instrumental dalam memajukan robotik autonomi untuk kegunaan dalaman. Hasil projek ini menekankan kesan mengintegrasikan teknologi sensor canggih dalam robot, sekali gus mempromosikan inovasi dan menangani cabaran masyarakat dan alam sekitar. Penyelidikan masa hadapan perlu memberi tumpuan kepada penambahbaikan algoritma untuk prestasi yang lebih baik dan meneroka aplikasi tambahan teknologi ini.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts. Dr Mohd Hanif Bin Che Hasan, for their precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and Nur Hasriha Binti Mohamad for the financial support through claim UTeM which enables me to accomplish the project. Not forgetting my fellow colleague, Muhamad Aiman Bin Azhar for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents, parents' in-law, and family members for their love and prayer during the period of my study. An honourable mention also goes to Nurul Syahirah Binti Shahrul Anuar for all the motivation and understanding. And to Gan Hong Jun and Aiman Bin Azhar, thanks for your unwavering support and guidance throughout this project. Your insights and encouragement have been invaluable. This achievement is as much yours as it is mine. I would like to thank all the staffs at the Faculty of Electrical Technology and Engineering, FTKE, fellow colleagues and classmates, the faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

Last but not least, I want to thank me for believing in me, I want to thank me for doing all this hard work. I want to thank me for having no days off. I want to thank me for never quitting. I want to thank me for always being a giver and trying to give more than I receive. I want to thank me for trying to do more right than wrong. I want to thank me for being me at all times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

V	-	Voltage
TX pin	-	Transmit Pin
RX pin	-	Receive Pin
VCC	-	Voltage at the Common Collector
GND	-	Ground
Hz	-	Hertz
UART	-	Universal Asynchronous Receiver-Transmitter
DC	-	Direct Current
USB	-	Universal Serial Bus
PWM	-	Pulse Width Modulation
I/O	-	Input/Output
CM	-	Centimeter

LIST OF ABBREVIATIONS

LiDAR	-	Light Detection and Ranging
ROS	-	Robot Operating System
SLAM	-	Simultaneous Localization and Mapping
AUV	-	Autonomous Underwater Vehicles
SDG	-	Sustainable Development Goals
MIVIS	-	Multispectral Infrared and Visible Imaging Spectrometer
IFAC	-	International Federation of Automatic Control
LSMCL	-	Long-term Static Mapping and Cloning Localization
RGB-B	-	Red-Green-Blue Depth
IoT	-	Internet Of Things
PIC	-	Peripheral Interface Controller
3D BIM	-	Three-Dimensional Building Information Modeling
CDES	-	Corridor Detector For Environment Sensing
SOPO	-	Specific Observations For Pose Optimization
MCL	-	Monte Carlo localization
GPPs	-	Global Principal Planes
GR-LOAM	-	Generalized Robust Lidar Odometry and Mapping
BIM	-	Building Information Model
CL	-	Cloning Localization
LSM	-	Long-term Static Mapping
GAF	-	Grid association filter
DoF	-	Degrees of Freedom
Dtof	-	Technology Direct Time Of Flight
IDE	-	Integrated Development Environment
DIY	-	Do It Yourself
CAD	-	Computer-Aided Design
PLA	-	Polylactic Acid
ABS	-	Acrylonitrile Butadiene Styrene
PETG	-	Polyethylene Terephthalate Glycol
TPU	-	Thermoplastic Polyurethane
NodeMCU	-	Node MicroController
RVis	-	ROS Visualization
GIS	-	Geographic Information Systems

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Autonomous robots have become integral to various applications, ranging from industrial automation to service robots in domestic environments. One crucial capability for these robots is the ability to navigate and map their surroundings autonomously. Room mapping, an essential task in indoor navigation, involves creating a detailed representation of the environment which the robot can use to perform tasks such as cleaning, delivery, and surveillance.

The Robot Operating System (ROS) is a versatile and widely-used framework that facilitates the development of complex robotic systems by providing a standardized platform for communication, control, and integration of various sensors and actuators. Originally developed by Willow Garage in 2007, ROS has since evolved into a robust, open-source framework supported by a vibrant community of developers and researchers.

ROS's architecture is designed around a decentralized network of nodes that communicate with each other via messages. This modular approach allows for the easy integration of different functionalities, such as sensor data processing, localization, path planning, and control, which are crucial for autonomous navigation and mapping.

Key tools and libraries in ROS that support autonomous mapping include:

1. SLAM (Simultaneous Localization and Mapping): SLAM is a critical algorithm for autonomous mapping, enabling a robot to create a map of an unknown environment while simultaneously determining its position within that map. ROS provides several SLAM packages, such as Gmapping, Hector SLAM, and Cartographer, which are widely used in research and practical applications.
2. Sensor Integration: ROS supports a wide range of sensors commonly used in autonomous robots, including LIDAR, cameras, and depth sensors. These sensors are crucial for perceiving the environment and generating accurate maps.
3. Navigation Stack: The ROS Navigation Stack offers a comprehensive suite of software for robot navigation, including path planning, obstacle avoidance, and autonomous movement. This stack is highly configurable and can be tailored to the specific requirements of a given robot and environment.
4. Simulation Tools: ROS integrates seamlessly with simulation environments like Gazebo, allowing developers to test and refine their algorithms in a virtual setting before deploying them on physical robots. This is particularly useful for debugging and performance tuning.

In the context of an autonomous robot for room mapping, ROS provides a powerful and flexible framework that simplifies the integration of various components required for the task. By leveraging ROS, developers can focus on algorithm development and system integration without having to reinvent the underlying infrastructure. This accelerates the development process and enhances the reliability and functionality of the final systemIn the context of an autonomous robot for room mapping, ROS provides a powerful and flexible framework that simplifies the integration of various components required for the task. By leveraging ROS, developers can focus on algorithm development and system integration without having to reinvent the underlying infrastructure. This accelerates the development process and enhances the reliability and functionality of the final system.

1.2 Addressing Societal and Global Issues with ROS

Global The Robot Operating System (ROS) is not only a powerful tool for developing individual robotic applications but also plays a significant role in addressing broader societal and global challenges. By providing a standardized and open-source framework, ROS enables the rapid development and deployment of robotic solutions that can have far-reaching impacts on various critical issues. Here are some ways ROS is being leveraged to address these challenges.

1.2.1 Disaster Response and Recovery

Natural disasters such as earthquakes, hurricanes, and floods often result in environments that are hazardous and difficult for humans to navigate. Robots equipped with ROS can be deployed in such scenarios to perform search and rescue operations, deliver supplies, and assess structural damage. ROS-based robots can quickly adapt to changing conditions, providing real-time data to first responders and helping to save lives.

Figure 1.1: Revolutionizing emergency responce with Autonomous Robot

1.2.2 Environmental Monitoring and Conservation

Environmental degradation and climate change are pressing global issues. ROS-powered robots can be used for monitoring ecosystems, collecting data on air and water quality, tracking wildlife, and detecting illegal activities like poaching or deforestation. For example, autonomous underwater vehicles (AUVs) using ROS can map coral reefs and monitor ocean health, while terrestrial robots can gather soil and atmospheric data in remote or hazardous locations.

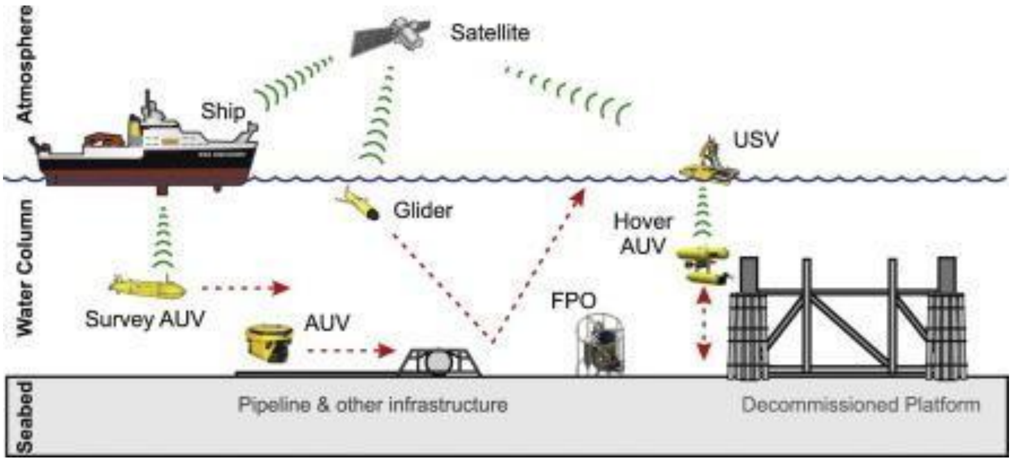


Figure 1.2: Observatories Marine Autonomous Systems

1.2.3 Agricultural Automation

The agricultural sector faces the challenge of feeding a growing global population amidst changing climate conditions and labor shortages. ROS-based robots can enhance agricultural productivity through precision farming techniques. These robots can perform tasks such as planting, weeding, and harvesting with high efficiency and accuracy. By reducing the reliance on chemical inputs and optimizing resource usage, ROS-based agricultural robots contribute to sustainable farming practices.

Figure 1.3: Robotic Integration in Indoor Agriculture

1.2.4 Healthcare and Assisted Living

An aging global population requires innovative solutions to provide adequate healthcare and support. ROS enables the development of assistive robots that can help elderly or disabled individuals with daily tasks, monitor their health, and provide companionship. In healthcare settings, ROS-powered robots can assist with logistics, such as delivering medications and supplies within hospitals, thus allowing medical staff to focus more on patient care.

Figure 1.4: Teleoperated robot-assisted surgical system

1.2.5 Urban Mobility and Smart Cities

As urban populations grow, cities face challenges related to congestion, pollution, and inefficient transportation systems. ROS is integral to the development of autonomous vehicles and smart transportation systems that can reduce traffic accidents, lower emissions, and improve mobility. ROS also supports the creation of smart infrastructure, such as intelligent traffic management and automated waste collection systems, contributing to the development of sustainable and efficient urban environments.

Figure 1.5: Modern Urban Mobility with Autonomous Vehicles

1.2.6 Education and Research

By making advanced robotics accessible, ROS democratizes technology and fosters innovation. Educational institutions use ROS to teach robotics, enabling students and researchers to experiment with and contribute to cutting-edge technologies. This widespread availability of robotic tools and knowledge can spur technological advancements that address various global challenges.

Figure 1.6: Collaborative Research and Development in Automation

1.3 Problem Statement

In Indoor mapping presents a multitude of challenges that can severely impact military, rescue, and exploration efforts. One of the primary issues is the inaccuracy of coordinates within confined spaces, often exacerbated by outdated technology. This can lead to incorrect data, resulting in failed missions and putting personnel at risk. Measurement errors in narrow spaces and a lack of knowledge about the internal layout of buildings further compound the problem, creating inefficiencies and hazards. Inconsistent or outdated data can also prove detrimental, leading to more complications in the indoor mapping process. Additionally, errors in data processing and a lack of understanding in using Geographic Information Systems (GIS) and other indoor mapping tools can impede effective mapping efforts.

Moreover, inadequate risk management and resource management knowledge can pose significant obstacles in indoor mapping. When these aspects are not properly handled, they can lead to operational difficulties and misallocation of resources. This underscores the importance of precise, up-to-date indoor mapping and comprehensive training in the relevant technologies and methodologies. In critical situations such as military operations, rescue missions, and indoor exploration endeavours, the accuracy and reliability of indoor maps are paramount for success and safety. Therefore, ensuring that maps are precise and up-to-date, and that personnel are well-trained in using modern indoor mapping tools and techniques, is essential for achieving mission success and ensuring the safety of all involved.

The development of ROS (Robot Operating System) for indoor area mapping aims to address these limitations. By leveraging advanced algorithms, ROS enhances the accuracy and robustness of mapping systems. This improvement facilitates smoother navigation and interaction within complex indoor settings, ensuring that the mapping data remains reliable even amidst changes in the environment. Furthermore, ROS provides a modular and open-source platform, enabling developers to customize and extend the system to meet specific needs.

Ultimately, enhancing indoor mapping with ROS leads to more functional and user-friendly autonomous systems. This development not only improves the efficiency of navigation but also provides a scalable framework that can be adapted for various applications. The increased accuracy and reliability of ROS-based mapping contribute to safer and more efficient operations in industries such as healthcare, logistics, and retail, making it an invaluable tool in the realm of robotics and automation.

1.4 Project Objective

The primary objective of this project is to develop an autonomous robot capable of efficiently and accurately mapping indoor environments using the Robot Operating System (ROS).

This will involve several specific goals:

a) To develop and integrate a robust Simultaneous Localization and Mapping (SLAM) algorithm that allows the robot to simultaneously map its environment and determine its position within the map in real-time.

b) To equip the robot with advanced sensors and algorithms to detect and avoid obstacles, both static and dynamic, ensuring safe and efficient navigation within various indoor environments.

c) To evaluate the design: test functionality, measure performance, ensure reliability and scalability, assess usability and maintainability, and analyze cost-effectiveness. Gather user feedback and conduct stress tests.

By achieving these objectives, the project aims to create a versatile and reliable autonomous room mapping system that can be used in a variety of applications, from household automation to commercial and industrial settings, leveraging the power of ROS to streamline and enhance indoor navigation and mapping tasks.

1.5 Scope of Project

The scope of this project are as follows:

a) Plan the integration of ROS packages and nodes necessary for navigation, mapping, and sensor data processing, utilizing the Nav2 ROS package running on ROS 2 Humble.

b) Assemble the robot with the chosen hardware components, ensuring all sensors, including the DTOF (Direct Time of Flight) LiDAR sensor, are properly mounted and connected.

c) Test the robot in various indoor environments with different layouts and obstacles to ensure robustness and adaptability.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The development of autonomous robots for room mapping, leveraging lidar technology, aligns with Sustainable Development Goal 9 (SDG 9) of promoting sustainable industrialization, innovation, and infrastructure. Lidar-equipped robots emit sound waves, enabling them to construct spatial maps and navigate environments efficiently. Integrating lidar with advanced algorithms enhances accuracy, contributing to resilient infrastructure development.

Early research focused on basic tasks, but modern approaches integrate lidar with other sensors for enhanced capabilities. This review explores historical development, technological evolution, and challenges, emphasizing the relevance of lidar-based mapping systems to SDG 9's goal of fostering innovation and building sustainable infrastructure for global development.

2.2 Sustainable Development Goals (SDG)

The rapid advancement of robotics and automation is transforming sectors like manufacturing, healthcare, and logistics. One vital application is autonomous room mapping, crucial for tasks like cleaning and surveillance. This project focuses on developing an autonomous robot for room mapping using the Robot Operating System (ROS).

ROS offers a flexible framework for integrating sensors and algorithms, enabling efficient navigation and mapping. Beyond technical innovation, this project aligns with Sustainable Development Goals (SDGs): SDG 9 for industry and innovation by enhancing productivity; SDG 11 for sustainable cities through better building management; SDG 12 for responsible consumption via optimized cleaning routes; SDG 13 for climate action by reducing carbon footprints; SDG 8 for economic growth by fostering innovation; and SDG 3 for health and well-being through cleaner environments. By addressing these SDGs, the project aims to advance robotics while contributing to sustainable development. The autonomous room mapping robot represents progress towards smarter, more efficient, and sustainable indoor environments.

Figure 2.1: Sustainable Development Goals (SDG)

2.3 Key Components and Technologies in Autonomous Robotics

The Autonomous robotics relies on advanced sensors like LiDAR for precise environmental mapping, powerful microcontrollers for data processing and control, and robust software frameworks like ROS for seamless integration and communication between components. These technologies enable robots to navigate complex environments, perform tasks efficiently, and adapt to changing conditions, driving innovation in fields such as healthcare, logistics, and agriculture.

2.3.1 LiDAR Sensor

A LiDAR (Light Detection and Ranging) sensor uses pulsed laser light to measure distances by emitting rapid laser pulses, which reflect off objects and return to the sensor, allowing it to calculate distances based on the time of flight. This technology is highly accurate and provides detailed, three-dimensional spatial data, making it essential for applications such as autonomous vehicles, geospatial mapping, robotics, and archaeology. Although LiDAR sensors can be expensive and are sensitive to weather conditions, their ability to create high-resolution maps and penetrate vegetation offers significant advantages for numerous industries.

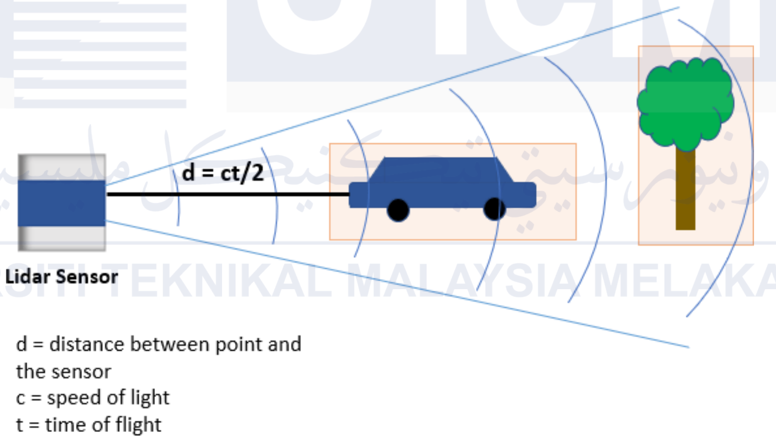


Figure 2.2: Theoretical Aspects of LiDAR Sensors

The research article [1] by Yan, Weber, and Wermter presents a novel neural model for autonomous indoor robot navigation. The study addresses the challenges of navigating in unconstrained real-world indoor environments by integrating visual and sensorimotor map information. The model not only focuses on mapping the environment but also incorporates mechanisms for robot path planning, navigation control, reflex-like obstacle avoidance, and adaptation of map connections based on feedback.

The research article [2] traditional LiDAR-based methods typically rely on point cloud reference maps, which require significant time and resources to create. In contrast, this proposed method leverages imperfect architectural skeleton information, such as walls and columns, extracted from architectural data and real scans.

The research article [3] by Giovanni Forzieri and colleagues on mapping natural and urban environments using airborne multi-sensor synergies. The research delves into the integration of data from hyperspectral MIVIS, color-infrared ADS40, and LiDAR sensors to achieve fine-scale mapping of heterogeneous landscapes. By exploring data fusion strategies, the study aims to capitalize on the high spatial resolution imagery provided by modern airborne Earth observation systems.

The research article [4] by Janis Kaltenthaler et al. published in IFAC Papers Online provides valuable insights into the integration of sensors for accurate pose estimation and mapping in autonomous systems. The study addresses the limitations of using inertial sensors alone, which are prone to drift over time, leading to errors in trajectory calculations crucial for mapping applications.

The research article [5] the implementation of indoor location for home service robots, spatial matching relationships between health tourism destinations and population aging, and temporal variations in heat exposure and hospitalizations for renal diseases. The journal showcases cutting-edge studies such as the utilization of SLAM algorithms with lidar sensors for indoor positioning of service robots, emphasizing the importance of accurate navigation in indoor environments.

The research article [6], Traditional approaches have primarily focused on 3D detection and tracking using multimodal data, but recent efforts have shifted towards large-scale semantic segmentation through data fusion rather than mere knowledge transfer or distillation .Challenges arise in effectively utilizing multimodal data due to issues such as spatiotemporal misalignment of sensors and uneven data distribution.

The research article [7] study by Lee introduces the Long-term Static Mapping and Cloning Localization (LSMCL) method for autonomous robot navigation in dynamic environments using a 3D LiDAR sensor. LSMCL aims to estimate real-time accurate locations by leveraging natural landmarks, even in environments where surrounding objects frequently change their positions.

The research article [8] authors extended a previous algorithm by incorporating two-wheel encoders and live RGB-D data for increased pose tracking robustness, improved moving object detection using optical flow and object detection, and coupled the SLAM algorithm with a path planner for simultaneous mapping and navigation.

2.3.2 Microcontroller

A microcontroller is a compact integrated circuit that includes a processor, memory, and input/output peripherals, making it capable of performing specific tasks and controlling devices within an embedded system. It features low power consumption, real-time operation, and versatility, which makes it ideal for applications in consumer electronics, automotive systems, industrial automation, medical devices, and IoT devices. Popular examples include Arduino, PIC, AVR, and ARM Cortex-M microcontrollers, which are widely used due to their efficiency, reliability, and advanced capabilities.

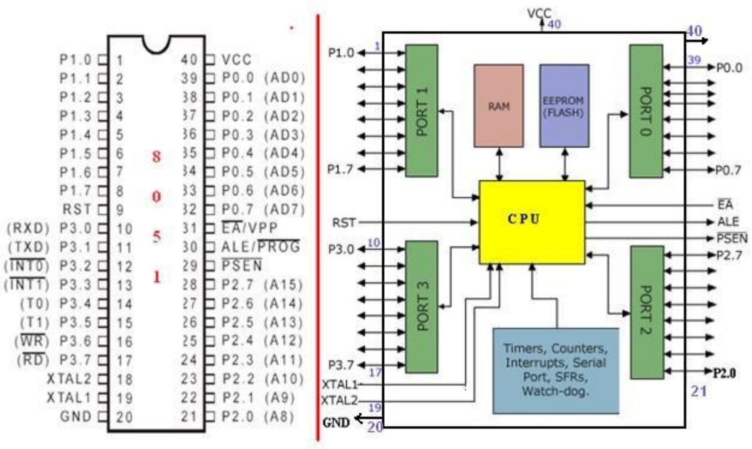


Figure 2.3: Structure of a Microcontroller

The research article [9] provides valuable insights into the use of lidar technology for indoor mapping applications. While the focus of the study is on assessing the accuracy of the iPad Pro's lidar sensor for creating 3D BIM models, it also sheds light on the importance of microcontroller technology in such applications.

The research article [10] by Zhang Zijiang et al. addresses the challenges faced by mobile robots equipped with single-line LiDAR sensors in indoor environments with long corridors. By proposing a Cartographer-SLAM based method, the study introduces innovative approaches such as the corridor detector for environment sensing (CDES) and specific observations for pose optimization (SOPO) to enhance mapping accuracy. The integration of microcontrollers, specifically the Core M4I7-D mini-host, equipped with ROS robot system under Ubuntu 16.04, demonstrates the importance of real-time data processing and control in achieving high-precision mapping results.

The research article [11] the method leverages semantic objects represented by points and structural layouts approximated by quadrilaterals, enhancing the understanding of the environment. By incorporating rooms as logical partitions in the SLAM process, RoomSLAM effectively reduces the search space in data association and aids in identifying potential loop closures.

The research article [12] by Ge et al. presents a novel approach to autonomous mobile robot localization in similar environments. The Text-MCL method combines text-level semantic information with laser scan data to achieve accurate and efficient localization. By employing a coarse-to-fine localization paradigm, the method first utilizes text-level semantic information to provide coarse localization and then refines the localization using the Monte Carlo localization (MCL) method based on laser data.

The research article [13] introduces a novel LiDAR-based mobile mapping framework that leverages Global Principal Planes (GPPs) to enhance mapping accuracy and robustness in indoor and outdoor environments. Traditional LiDAR-based mapping methods often face challenges in narrow spaces and environments lacking geometric features.

The research article [14] system combines a panoramic camera and a LiDAR sensor without the need for strict synchronization or additional auxiliary sensors, demonstrating advancements over existing methods in various indoor and outdoor scenes . The device used for data acquisition includes an Insta 360 ONE X2 panoramic camera, a Velodyne VLP-16 Lite LiDAR, a Raspberry Pi 4B, and a battery, with a delay of 100 to 200 ms between collected data due to different operating frequencies.

2.3.3 Robot Operating System, ROS

The Robot Operating System (ROS) is an open-source framework that provides tools, libraries, and conventions for developing complex and robust robotic software, featuring a modular architecture, robust messaging system, hardware abstraction, and integration with simulation and visualization tools. It is widely used for autonomous navigation, robotic manipulation, research and development, industrial automation, and education, benefiting from a large and active community that contributes various packages and tools to accelerate development.

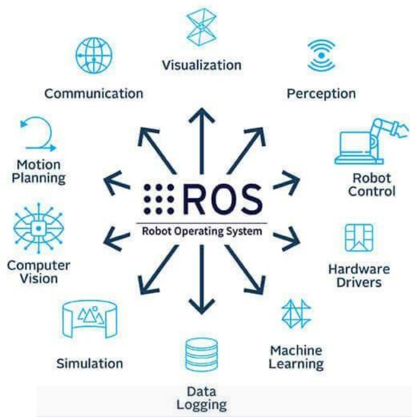


Figure 2.4: Application in ROS

The research article [15] by Siswoyo et al. presents an innovative approach to utilizing autonomous mobile robot technology in hospital settings. While the paper focuses on the design and development of the Viguro Robot, it also highlights the importance of incorporating advanced technologies such as the Robot Operating System (ROS) for efficient control and navigation.

The research article [16] In the realm of Robotics Operating System (ROS) applications within underground metal mine mapping, the study Efficient and accurate mapping method of underground metal mines using mobile mining equipment and solid-state lidar presents a significant contribution. While the research primarily focuses on the development of a novel mapping method for underground metal mines, it also leverages ROS for data processing and system integration.

The research article [17] presents a comprehensive study on simultaneous localization and mapping (SLAM) for ground robots. The proposed GR-LOAM method integrates LiDAR, inertial measurement unit (IMU), and encoder measurements in a tightly coupled scheme to enhance robot ego-motion estimation. This research contributes significantly to the field of robotics by addressing the challenges of navigating complex terrains.

The research article [18] presents a novel approach to mapping and motion estimation for mobile robots by integrating data from both cameras and Lidar sensors. In the context of Robot Operating System (ROS), this work aligns with the core principles of ROS, which is an open-source robotics middleware widely used in research and industry for robot development. ROS provides a flexible and modular framework for integrating various sensors and processing data streams, making it an ideal platform for implementing the sensor fusion techniques proposed in this study.

The research article [19] shows that recent studies have highlighted the advantages of using MEMS based 3D LIDAR sensors over traditional technologies for plant detection and mapping in agricultural robotics . These sensors provide reliable ranging information, are unaffected by changing lighting conditions, and can operate effectively in various weather conditions such as light fog and dust.

The research article [20] by H. Yin et al. presents a novel approach to semantic localization on Building Information Model (BIM)-generated maps using a 3D LiDAR sensor. The authors propose a pipeline to convert BIM to semantic point cloud maps, leveraging semantic information to filter laser points and improve data associations for localization . The study aims to bridge the gap between digital representations and localization-oriented maps without the need for manually labeled data.

2.4 Table Comparison: Key Features and Applications Based on Component

2.4.1 Research on LiDAR Sensors

Table 2.1: LiDAR Sensor article comparison

Reference	Study Focus	Key Features
1	Developing neural model for indoor robot navigation in real environments.	1. Neural-inspired model for robot navigation. 2. Learning spatial knowledge from visual and sensorimotor inputs. 3. Addressing noise, dynamics, and complexity in real-world environments.
2	Developing LiDAR method for precise indoor robot localization.	1. Utilizes imperfect architectural skeleton information for localization accuracy. 2. Extracts feature patterns from LiDAR scans for precise positioning. 3. Avoids the need for GNSS, radiation sources, and point cloud maps. 4. Suitable for large-scale indoor environments with complex structures.
3	Optimizing classification performance using multi-sensor data for landscape mapping.	1. Utilization of Maximum Likelihood algorithm for classification accuracy. 2. Integration of ADS40, LiDAR, and MIVIS data for mapping. 3. Identification of optimal data fusion configurations for high accuracy. 4. Focus on capturing spatial variability of natural and human-dominated environments. 5. Exploration of synergies among multi-sensor airborne data for mapping.
4	Enhancing pose estimation accuracy using IMU and LiDAR technology.	1. Integration of Inertial Measurement Unit (IMU) and Light Detection and Ranging (LiDAR) sensors. 2. Reduction of sensor drift for improved trajectory calculation precision. 3. Detection of standstill phases using LiDAR data. 4. Enhancement of IMU position estimation accuracy. 5. Application in autonomous motion, mapping, and navigation systems.
5	Indoor location study enhances home service robot navigation technology.	1. Utilization of SLAM algorithm with lidar sensor technology. 2. Implementation of indoor positioning, mapping, and navigation functionalities. 3. Integration of environmental data analysis and voice broadcasting capabilities. 4. Autonomous navigation and control of abnormal environmental conditions. 5. Multi-layered robot design with specific components for functionality.

6	Focusing on multimodal fusion for improved semantic segmentation accuracy.	1. Exploration of soft joint mechanism in LiDAR-camera fusion. 2. Proposal of JoSS for semantic segmentation using dual-stream transformer. 3. Unimodal data augmentation for enhanced multimodal 3D segmentation accuracy.
7	Long-term Static Mapping and Cloning Localization method enables accurate robot localization in dynamic environments.	1. Long-term Static Mapping (LSM) for creating static feature and grid maps. 2. Cloning Localization (CL) for real-time localization in dynamic environments. 3. Use of 3D LiDAR sensor for accurate mapping and localization. 4. Decoupling mapping and localization processes for enhanced performance. 5. Utilization of natural landmarks for autonomous robot navigation. 6. 3D SLAM technology for estimating global location in dynamic spaces. 7. Grid association filter (GAF) for generating feature maps of static objects. 8. Particle filter for initial 2D global position estimation. 9. Tracking position trajectory in 3D space with 6 DoF. 10. Novel methodology for stable mapping and localization in dynamic environments.
8	Development of efficient RGB-D SLAM for mobile robot navigation.	1. Lightweight and efficient for onboard processing 2. Accurate pose tracking in dynamic environments 3. Robustness to moving objects 4. Room-scale consistency in mapping 5. Integration with a navigation strategy for on-the-fly mapping.

2.4.2 Research on Microcontroller

Table 2.2: Microcontroller article comparison

Reference	Study Focus	Key Features
1	Evaluate iPad Pro lidar accuracy for 3D indoor mapping.	1. Utilization of dTOF technology with a laser emitting near-infrared spectrum. 2. Capture 576 points per frame using flash illumination. 3. Focal length of 26 mm and maximum scanning distance of 5m. 4. Comparison with reference data from Leica Disto D810, GeoSLAM ZEB Horizon, and TLS Leica RTC360.

		5. Assessment of accuracy for indoor mapping and scan-to-BIM applications.
2	Improving indoor mapping with single LiDAR sensor technology.	1. Lightweight. 2. Robust. 3. Accurate. 4. Low-cost. 5. Single LiDAR sensor. 6. Online obstacle monitoring. 7. Pose feature extraction. 8. Corridor environment detection. 9. Cumulative error reduction. 10. Compact SLAM system.
3	Developing SLAM method for mobile robots in indoor environments.	1. Utilization of semantic entities (objects and walls) for mapping. 2. Representation of indoor layout structures using quadrilaterals. 3. Efficient data association and parameter estimation within rooms. 4. Detection of potential loop closures using room information. 5. Higher level of semantic information for autonomous robot navigation.
4	Developing Text-MCL system for accurate robot localization in similar environments.	1. Coarse-to-fine localization paradigm. 2. Text-level semantic information utilization. 3. Integration of MCL method and laser sensor data. 4. Camera orientation adjustment for optimal image capture. 5. Moving strategy to avoid random walking.
5	LiDAR-based method using GPPs for accurate indoor mapping.	1. Global Principal Planes (GPPs). 2. Feature Extraction Methods. 3. Degeneracy Handling. 4. IMU Integration. 5. Mapping Accuracy. 6. Indoor and Outdoor Environments.
6	PanoVLM study focuses on cost-effective, accurate 3D mapping techniques.	1. Low-cost mapping solution. 2. Panoramic camera utilization. 3. Elimination of strict sensor synchronization. 4. Joint camera-LiDAR pose estimation. 5. Improved mapping accuracy and robustness.

2.4.3 Research on Robot Operation System, ROS

Table 2.3: ROS article comparison

Reference	Study Focus	Key Features
1	Designing autonomous robot for hospital visitor guidance and navigation.	1. Autonomous navigation using smart sensors and controllers. 2. Interactive human-robot interface for visitor interaction. 3. Localization and mapping capabilities for route planning. 4. Obstacle avoidance technology for safe navigation. 5. Ability to guide visitors to their intended destinations efficiently.
2	Efficient, accurate mapping method for underground metal mines using technology.	1. Voxel grid-based point cloud processing method. 2. Improved point-to-plane matching method. 3. Ground point cloud constraints for vertical accuracy. 4. Remote control mobile mining equipment and solid-state lidar. 5. Enhanced efficiency and accuracy in underground mine mapping.
3	LiDAR-based GR-LOAM enhances robot navigation in complex environments.	1. Fusion of LiDAR, IMU, and encoder measurements. 2. Tightly coupled sensor data integration for accurate localization. 3. Handling complex terrain and dynamic environments effectively. 4. Real-time performance suitable for ground robot applications.
4	Hybrid strategy fuses Lidar and camera for robust mapping.	1. Heuristic switching strategy for fusing Lidar and camera measurements. 2. Multi-stage loop closure detection strategy for accurate loop closure. 3. Novel hybrid mapping method for constructing accurate metric-feature maps. 4. Multi-layer optimization strategy for obtaining a global consistent map.
5	Application of 3D LIDAR sensors for plant detection in agriculture.	1. Reliable ranging data. 2. Robustness against changing lighting conditions. 3. Ability to operate in various weather conditions. 4. Accurate plant detection and mapping. 5. Real-time detection of individual plants. 6. Support for autonomous navigation in agricultural environments.
6	Enhancing sensor localization accuracy using	1. Utilizes Building Information Models (BIM) for semantic localization. 2. Semantic point cloud maps improve sensor localization accuracy.

	BIM and semantic maps.	3. Coarse-to-fine localization approach with iterative closest point registration. 4. Mapping-free localization framework. 5. Single 3D LiDAR sensor deployment. 6. Experimental validation in real-world construction environments. 7. Semantic information filtering for laser points. 8. Reduced localization errors on BIM-generated maps. 9. No deep learning is required for feature extraction or pose regression. 10. Interpretable and efficient localization framework.

2.5 Summary

The development of an autonomous robot for room mapping using lidar involves creating a system that can navigate and generate detailed maps of indoor environments autonomously. This project leverages lidar sensors, which use sound waves to detect obstacles and measure distances, allowing the robot to construct spatial maps. The Robot Operating System (ROS) provides the framework for integrating sensors, actuators, and mapping algorithms, enabling efficient real-time navigation and mapping. This project not only advances robotic technology but also supports Sustainable Development Goals (SDGs) by promoting innovation (SDG 9), optimizing urban infrastructure (SDG 11), reducing energy consumption (SDG 12), supporting climate action (SDG 13), fostering economic growth (SDG 8), and enhancing health and well-being (SDG 3).

CHAPTER 3

METHODOLOGY

3.1 Introduction

In general, the development of an autonomous robot for room mapping requires a well-structured integration of hardware components and software systems. The block diagram provided illustrates the key components and their interactions within the robot, highlighting the flow of data from sensors to processing nodes, and from processing nodes to control nodes. This diagram is essential for understanding the overall architecture and functionality of the robot, ensuring that each subsystem works cohesively to achieve efficient and accurate room mapping.

3.2 Selecting and Evaluating Tools for Sustainable Development

For my sustainable development project focused on automated tank room mapping, I have chosen to employ a Raspberry Pi 4b as the core processing unit, with an Arduino Nano for motor control, to orchestrate the entire system. The primary aim is to create an efficient, reliable, and cost-effective solution for mapping and navigating tank rooms, which can be a challenging environment due to limited accessibility and potential hazards.

To detect obstacles and navigate paths within the tank room, I am utilizing an RP Lidar C1 in combination with a servo motor. The RP Lidar C1 is a laser-based sensor adept at measuring distances by emitting laser pulses and calculating the time it takes for the reflections to return after hitting an object. This information is processed by the Raspberry Pi 4b running Ubuntu 22.04 with ROS 2 Humble distribution and the Nav2 ROS package to determine the presence and distance of obstacles in real time. The servo motor allows for precise control over the direction of the RP Lidar C1, enabling a comprehensive scan of the surroundings by adjusting the sensor’s angle. This dynamic scanning capability is crucial for identifying and avoiding obstacles, thereby ensuring safe and effective navigation within the tank room.

For mapping the room area, I have integrated the RP Lidar C1 sensor into the system. The RP Lidar C1 is a 360-degree laser scanner capable of producing high-resolution, real-time maps of the environment. It works by emitting laser pulses and measuring the time taken for the reflected signals to return from the surfaces they encounter. This data is then compiled by the Raspberry Pi 4b to create a detailed and accurate representation of the room's layout. The choice of RP Lidar C1 is driven by its high accuracy, affordability, and robustness, making it an ideal candidate for environmental mapping in confined and potentially harsh conditions.

The Raspberry Pi 4b was selected due to its versatility, extensive I/O capabilities, and sufficient processing power to handle multiple sensors and perform real-time data analysis. Its ability to run Ubuntu 22.04 and support various ROS 2 packages also simplifies the integration of the RP Lidar C1 and ensures smooth operation and coordination among the components.

By combining these tools, the automated tank room mapping system aims to enhance operational efficiency and safety. The RP Lidar C1 and servo motor provide a reliable means to detect and navigate around obstacles, while the Raspberry Pi 4b ensures precise mapping of the environment. The comprehensive data collected can be used for various applications, such as monitoring tank conditions, planning maintenance activities, and improving safety protocols. This approach not only advances the automation and accuracy of tank room management but also contributes to sustainable practices by optimizing resource usage and reducing the need for manual intervention in potentially hazardous environments.

3.3 Methodology

This thesis is very important that to make sure the progress of this project is followed step-by-step and complete the tasks that arranged in schedule. Therefore, a flowchart will be drew. A workflow or process can be graphically represented using something called a flowchart. It involves carrying out a series of distinct steps in a predetermined order. In most cases, a flowchart will display the steps as a series of boxes of varying types. Steps are linked so that anyone can examine the flowchart and follow its directions in the appropriate order from the very beginning to the very end. This is accomplished by connecting lines and directed arrows. It is also useful for a variety of other objectives, including as documenting, studying, planning, refining, and describing complicated processes in a style that is easy to understand and straightforward.

3.3.1 Block Diagram

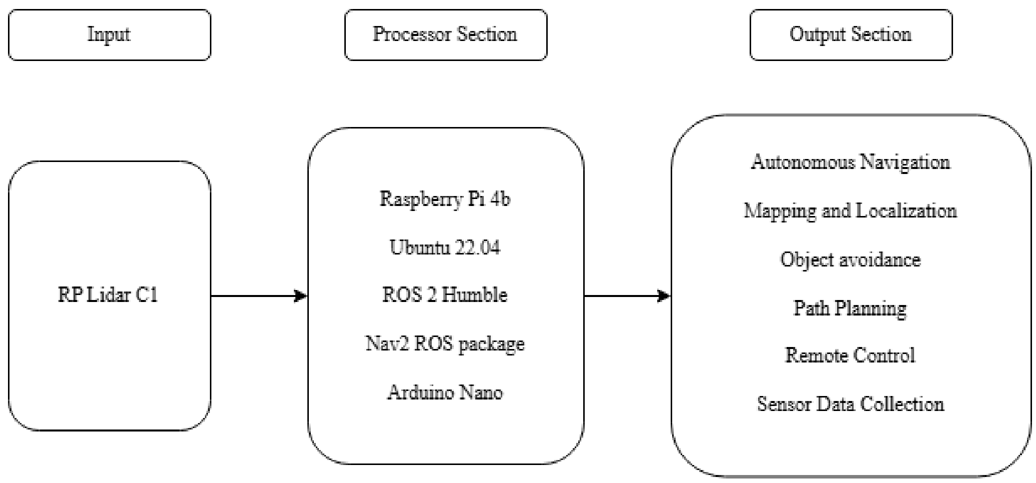


Figure 3.1: General Process Flow Estimation

3.3.1.1 Description of Block Diagram

The image is a block diagram illustrating the components and flow of a robotic system. It shows the input section (RP Lidar C1), the processor section (Raspberry Pi 4b with Ubuntu 22.04, ROS 2 Humble, Nav2 ROS package, and Arduino Nano), and the output section (Autonomous Navigation, Mapping and Localization, Path Planning, Remote Control, and Sensor Data Collection). This diagram is relevant as it outlines the architecture of a robotic system used for autonomous navigation and data collection.

Input Section:

- RP Lidar C1: Laser-based sensor for detecting 360° and measuring distance.

Processor Section:

- Raspberry Pi 4b: Central computing unit.
    1. Ubuntu 22.04: Operating system providing a robust and stable enviroment for runing application.
    2. ROS 2 Humble: Middleware framework specifically designed for robotics applications.
- Nav 2 ROS package: Used for all type navigation.
- Arduino Nano: Controls motor with encoders, providing precise control over motor functions.

Output Section:

- Autonomous Navigation: The robot can navigate autonomously in its environment using the ROS 2 Navigation Stack (Nav2). The LiDAR sensor will help create a map and detect obstacles, allowing the robot to move from one point to another without human intervention.

- Mapping and Localization: Using SLAM (Simultaneous Localization and Mapping), this robot can explore an unknown environment and build a map while keeping track of its own location within that map.
- Object Avoidance: The robot can detect and avoid obstacles in real-time. This is useful in dynamic environments where objects might move or appear unexpectedly.
- Path Planning: With the Nav2 package, this robot can plan optimal paths to reach a destination. This includes navigating around obstacles and selecting the most efficient route.
- Remote Control: The robot can remotely via a wireless connection, issuing commands from a computer or control device.
- Sensor Data Collection: The Lidar and encoder can collect data about the environment and the robot's movements, which can be used for analysis or further development.

3.3.2 Flowchart

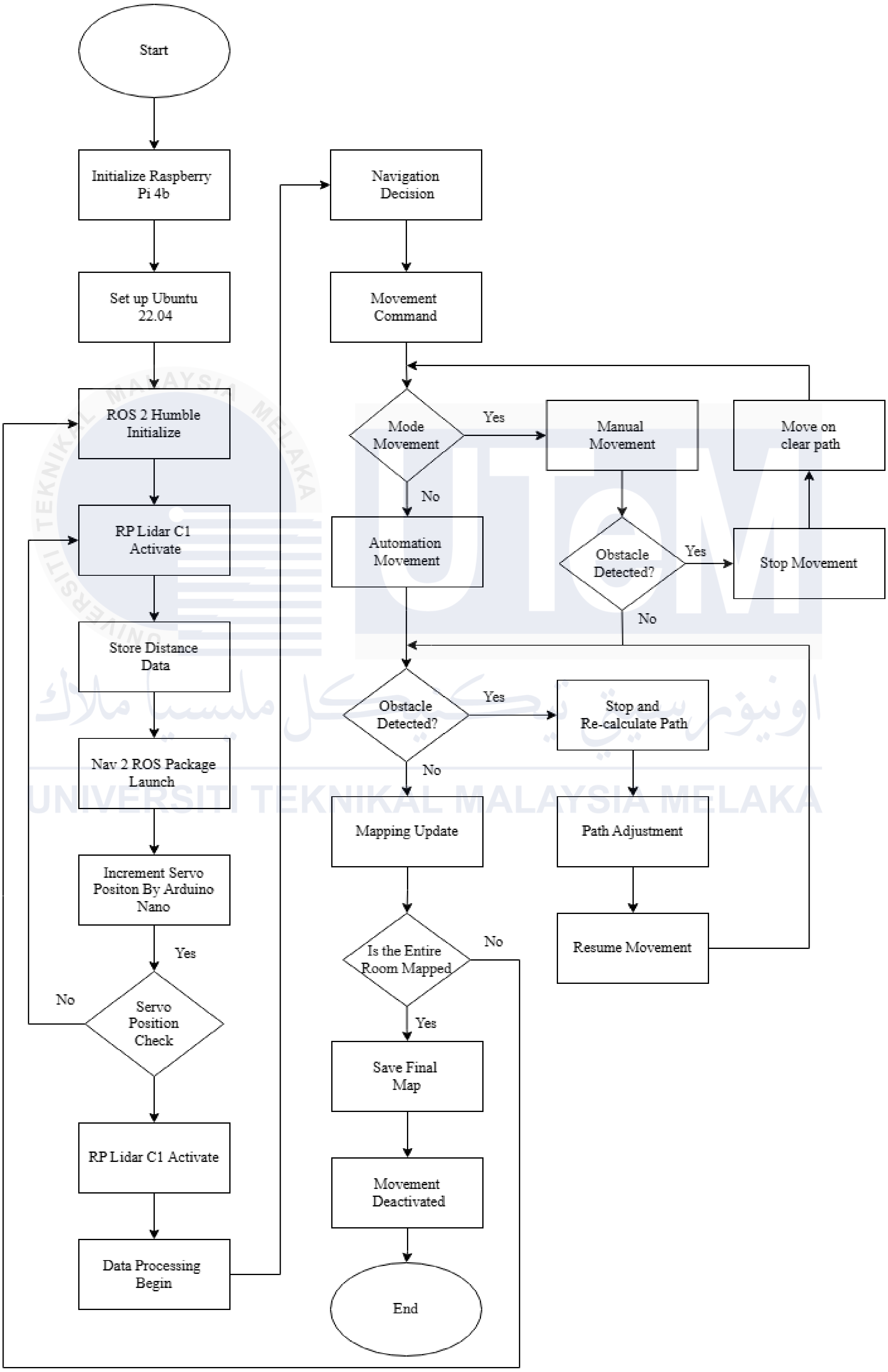


Figure 3.2: Comprehensive Process Flow Estimation

3.3.2.1 Flowchart Description

1. **Initialize System**
   - Set up the Raspberry Pi 4b with Ubuntu 22.04.
   - Launch ROS 2 Humble.
2. **Activate Sensors**
   - Turn on the RP Lidar C1 to start collecting distance measurements.
3. **Data Processing**
   - Process the sensor data using the Nav 2 ROS Package.
4. **Control System**
   - Send command signlas to the Arduino Nano, which controls the motors with encoders.
5. **Navigation and Movement**
   - Execute autonomous navigation commands, mapping and localization, path planning, remote control operations, and sensor data collection.
6. **Decision-Making**
   - Check for obstacles
   - Make navigation decisions based on processed data.
7. **Movement**
   - Manual Movement
     - Check for obstacles.
     - Move along the clear path.
   - Automated Movement
     - Continuously check for obstacles.
     - Update mapping and verify if the mapping is complete.
     - Save the final map and stop movement if mapping is complete.
     - Adjust servo position if mapping is not complete

3.3.3 Experimental Setup for Mapping

SLAMTEC provides a Lidars plugin in RoboStudio for users in test and evaluation. From this figure 3.3, the scan result directly in the UI and save the scan result to files for further processing. To facilitate the usage of RPLIDAR C1 in product development and speed up the development for users, SLAMTEC has provided the Framegrabber plugin in RoboStudio for testing and debugging as well as the SDK available under x86 Windows, x86 Linux, MacOS, and Arm Linux.

Figure 3.3: The Framegrabber plugin in RoboStudio

3.3.4 Robot System Integration

3.3.4.1 Wiring Diagram Setup

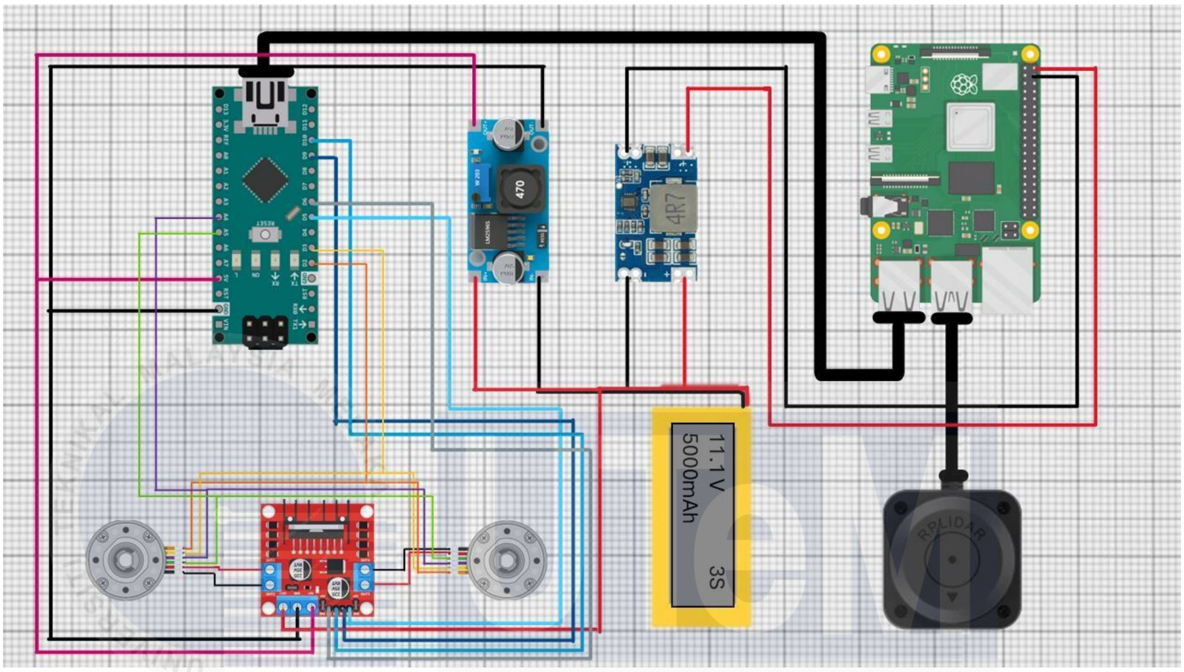


Figure 3.4: Circuit Layout Configuration

To set up the connections for this robot, start by connecting the Arduino Nano and RPLiDAR C1 to the Raspberry Pi 4b. Then, connect the 11.1V LiPo battery to the step-down LM2596 for the motor driver and to the mini 560 DC 5V to supply to the Raspberry Pi 4b, with the motor driver also connected to the LiPo battery.

For the motor encoders, connect the motor power terminal positive to OUT1 and OUT3, and the motor power terminal negative to OUT2 and OUT4. For both encoders, connect pin Left A to Arduino pin D2, pin Left B to Arduino pin D3, pin Right A to Arduino pin A4, and pin Right B to Arduino pin A5. Finally, for the L298N motor driver, connect pin IN2 (L Fwd) to Arduino pin D10, pin IN1 (L Rev) to Arduino pin D6, pin IN3 (R Fwd) to Arduino pin D9, and pin IN4 (R Rev) to Arduino pin D5. This setup will ensure all components are correctly connected for this robotic project.

3.3.4.2 Power Supply Usage

Choosing a LiPo battery with 11.1V (3S) and 5000mAh for this autonomous robot project is a wise decision for several reasons. First, the 11.1V (3S) configuration provides sufficient voltage to power the Raspberry Pi 4b, motor encoder, motor driver, Arduino Nano, and RPLidar C1, ensuring that all components receive adequate power. Second, the 5000mAh capacity offers a good balance between battery life and size, allowing the robot to operate for extended periods without frequent recharging. This is particularly important for autonomous robots engaged in tasks such as room mapping, where continuous operation is required. Additionally, LiPo batteries are known for their high energy density and ability to deliver consistent power output, making them ideal for robotics applications that demand reliability and performance.

Opting for a 5000mAh battery for this autonomous robot project provides numerous benefits:

- **Extended Operating Time**: A 5000mAh capacity ensures that this robot can run for longer periods without the need for frequent recharging. This is crucial for room mapping tasks, which require continuous operation to thoroughly scan and map the environment.
- **Balanced Power and Weight**: 5000mAh provides a good balance between power capacity and weight. It supplies ample energy to this components (Raspberry Pi 4b, motor encoder, motor driver, Arduino Nano, and RPLidar C1) while keeping the robot light enough for efficient movement and operation.
- **Reliable Power Supply**: A higher capacity battery like 5000mAh offers consistent and reliable power output, essential for maintaining stable

performance of the robot's systems, ensuring there are no power dips that could disrupt operations.

3.3.4.3 Robot Chasis

Overall Design

For the current robotic system setup, the integration of various components ensures efficiency and robust performance. This well-thought-out design leverages advanced technologies to achieve optimal functionality in autonomous navigation and other applications.

The setup includes a LiDAR sensor on top, while the Raspberry Pi 4B, Arduino Nano, LiPo battery, step-down converter, and motor controller are neatly housed inside. This compact and efficient design ensures all components are protected and organized, facilitating ease of maintenance and system upgrades.

For movement, the design employs differential wheels, allowing for precise control and manoeuvrability. Using AutoCAD to plan and visualize this setup helps ensure that all elements are optimally placed, enhancing the system's overall performance and robustness in various robotic applications.

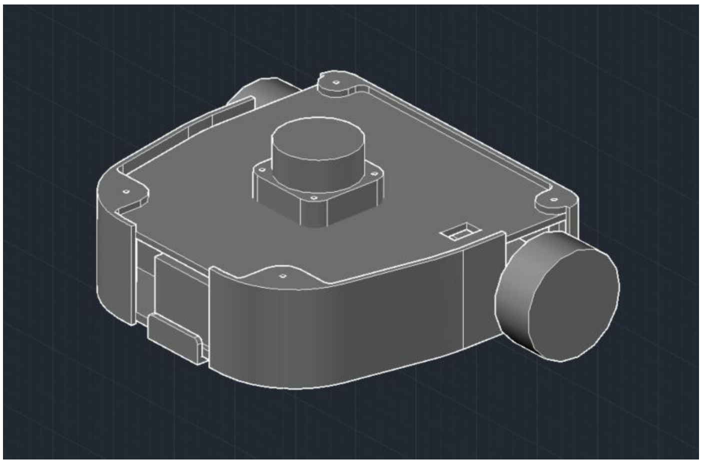


Figure 3.5: Efficient navigation, customizable design and reliable performance.

3.3.4.4 Software Setup

For the software setup of an autonomous robot project, Ubuntu 22.04 is utilized as the operating system, providing a stable and reliable environment for running robotic software. ROS 2 Humble is the chosen Robot Operating System, which offers a framework for developing functionalities such as communication between components, sensor integration, and control algorithms. Additionally, the Nav2 ROS package is employed, which is a powerful set of tools for navigation in ROS 2. This package enables the robot to autonomously navigate and map its environment, making it essential for the development of room mapping projects using lidar. Combining these software components ensures a robust and efficient software stack capable of handling complex tasks and autonomous operations.

3.3.4.5 Data Retrieval

For an autonomous robot room mapping project, software setup involves using Ubuntu 22.04 with ROS 2 Humble and the Nav2 package to enable navigation, mapping, and robust control of the robot's functions.

1. Install ROS 2 and Nav2

Open a terminal and run the following commands to install ROS 2 Humble and the Nav2 package:

```
# Add the ROS 2 apt repository
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository universe
sudo apt update
sudo apt install curl gnupg2 lsb-release
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | su
sudo sh -c 'echo "deb http://packages.ros.org/ros2/ubuntu $(lsb_release -cs

# Install ROS 2 Humble
sudo apt update
sudo apt install ros-humble-desktop

# Source ROS 2 setup script
source /opt/ros/humble/setup.bash

# Install Nav2
sudo apt install ros-humble-nav2-bringup
```

2. Setup and launch Nav2

Create a workspace, clone the Nav2 repository, and build it:

```
# Create a workspace
mkdir -p ~/nav2_ws/src
cd ~/nav2_ws/src

# Clone the Nav2 repository
git clone https://github.com/ros-planning/navigation2.git -b humble

# Build the workspace
cd ~/nav2_ws
colcon build
```

3. Launch the Nav2 stack

Create a launch file to start the Nav2 stack. Create a file named in the directory with the following content:

```
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument
from launch_ros.actions import Node
from launch.substitutions import LaunchConfiguration

def generate_launch_description():
    return LaunchDescription([
        DeclareLaunchArgument(
            'use_sim_time',
            default_value='false',
            description='Use simulation (Gazebo) clock if true'
        ),
        Node(
            package='nav2_bringup',
            executable='bringup_launch',
            output='screen',
            parameters=[{
                'use_sim_time': LaunchConfiguration('use_sim_time')
            }]
        )
    ])
```


4. Run the Nav2 stack

Launch the Nav2 stack with the following command:

```
# Source the workspace
source ~/nav2_ws/install/setup.bash

# Launch Nav2
ros2 launch nav2_bringup nav2_launch.py use_sim_time:=false
```

3.3.4.6 Parameters of LiDAR Sensor

The RPLIDAR C1 is a versatile and efficient DTOF (Direct Time of Flight) LiDAR sensor designed for a range of applications, including robotics, mapping, and obstacle detection. It features a measurement range from 0.05 meters to 12 meters, making it suitable for both close-range and mid-range scanning tasks. The sensor offers an angular resolution of 1°, which can be improved to 0.5° with interpolation, providing detailed and accurate environmental data.

Operating at a default scan rate of 5 Hz, the RPLIDAR C1 can be adjusted to scan up to 10 Hz, depending on the requirements of the application. This flexibility allows for a balance between scan speed and data density. With a measurement frequency of 5000 samples per second, the RPLIDAR C1 ensures high-resolution data capture across its full 360-degree field of view, enabling comprehensive environmental mapping.

The RPLIDAR C1 requires a 5V DC power supply and consumes less than 500 mA during operation, making it energy efficient. It communicates via a 3.3V TTL UART interface, ensuring compatibility with various microcontrollers and development platforms. Weighing approximately 190 grams, the compact design of the RPLIDAR C1 allows for easy integration into different systems and projects. Overall, the RPLIDAR C1's robust performance and flexibility make it an ideal choice for enhancing the capabilities of autonomous systems.

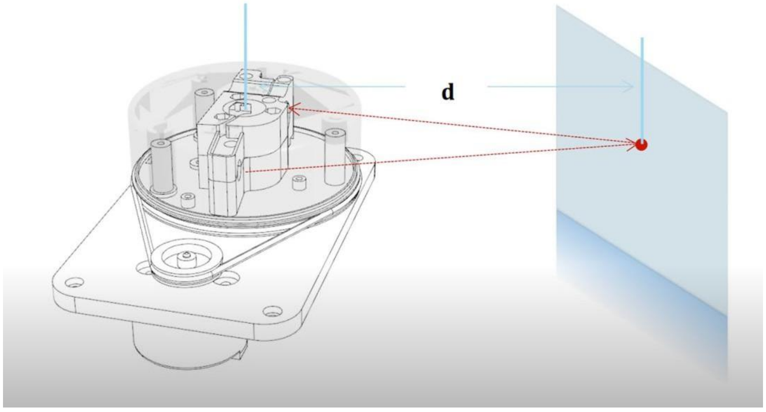


Figure 3.6: The RPLiDAR working schematic

3.3.4.7 RViz Interface

RViz, short for "ROS Visualization," is a 3D visualization tool in the Robot Operating System (ROS) ecosystem. It allows developers and researchers to visualize and interact with data from robots, sensors, and simulations in real-time. This powerful tool supports the visualization of a wide range of information, including sensor data like laser scans, point clouds, and camera feeds, as well as the state of the robot itself, such as its position, orientation, and movement. Users can create custom configurations to visualize specific data that is crucial for their projects, making RViz an essential tool for developing and debugging robot systems.

One of the key strengths of RViz is its versatility and extensibility. It provides a modular interface where various visualization plugins can be added or customized according to the user's needs. This flexibility makes RViz suitable for a broad range of applications, from simple robot demonstrations to complex autonomous navigation and manipulation tasks. Additionally, RViz supports interactive markers, allowing users to interact with the visualized data and control the robot or simulate different scenarios. Overall, RViz is a critical tool for anyone working within the ROS ecosystem, offering comprehensive visualization capabilities to enhance the development and deployment of robotic systems.

3.3.5 Project Hardware and Components

3.3.5.1 Raspberry Pi 4b

The Raspberry Pi 4 Model B is a powerful single-board computer that offers significant upgrades over its predecessors. It features a Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC running at 1.8 GHz, providing robust processing power for a variety of applications. The Raspberry Pi 4B is available with 1GB, 2GB, 4GB, or 8GB of LPDDR4-3200 SDRAM, allowing users to choose the amount of memory that best suits their needs.

One of the standout features of the Raspberry Pi 4B is its dual micro-HDMI ports, which support dual 4K monitors at 60Hz. This makes it an excellent choice for projects requiring multiple displays, such as media centers or educational setups. Additionally, the Raspberry Pi 4B includes Gigabit Ethernet, 2 USB 3.0 ports, 2 USB 2.0 ports, Bluetooth 5.0, and 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless connectivity, ensuring fast and versatile networking options.

The Raspberry Pi 4B also boasts improved energy efficiency and runs silently without a fan, making it an environmentally friendly and quiet computing solution. It supports Power over Ethernet (PoE) with a separate PoE HAT, providing flexibility in power delivery. With its compact size and extensive connectivity options, the Raspberry Pi 4B is a versatile and powerful tool for hobbyists, educators, and developers alike.

Moreover, the Raspberry Pi 4B supports various operating systems, including the official Raspberry Pi OS, Ubuntu, and other Linux distributions, offering flexibility and catering to a wide range of user needs. The active community and extensive online resources make it easy to find support, tutorials, and project ideas, making the Raspberry Pi 4B an accessible and valuable platform for both beginners and experienced users in the field of computing and electronics.

Figure 3.7: The Raspberry Pi 4b

3.3.5.2 Arduino Nano V3

The Arduino Nano is a small, complete, and flexible microcontroller board based on the ATmega328P. It was developed by Arduino.ccin Italy in 2008 and is designed to be breadboard-friendly. The Nano features 30 male I/O headers configured in a DIP30 style, making it easy to integrate into various projects.

The board includes digital pins and analog pins, along with reset pins and power pins. It is programmed using the Arduino IDE, which can be downloaded from the Arduino official site. The Nano also has a USB type- C connector for easy programming and power supply. With its compact size (45 mm x 18 mm) and lightweight design (7 grams), the Arduino Nano is perfect for projects where space is limited. It is widely used in educational settings, hobbyist projects, and prototyping due to its versatility and ease of use.

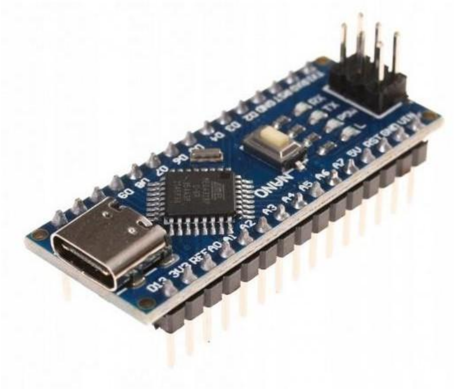


Figure 3.8: Arduino Nano V3 Atmega 328p Development Board

3.3.5.3 RPLiDAR C1

Another noteworthy aspect of the RPLIDAR C1 is its high-definition fusion technology. This advanced technology leverages the strengths of both triangulation and DTOF (Direct Time of Flight) ranging methods. By combining these techniques, the sensor can deliver highly detailed and accurate data, especially for close-range objects. This makes it particularly advantageous for tasks requiring precise environmental perception, such as robot positioning, mapping, and navigation.

The RPLIDAR C1 also features a robust filtering mechanism that significantly reduces measurement noise and enhances data quality. This ensures that even in environments with potential interferences, the sensor can maintain a high level of accuracy and reliability. This capability is essential for autonomous systems that rely on precise and consistent data for effective operation. The improved filtering mechanism helps the sensor to deliver stable performance across diverse conditions, making it a reliable choice for various applications.

Furthermore, the RPLIDAR C1 is designed with an intuitive user interface and easy integration capabilities. It supports a range of development environments and is compatible with various microcontrollers and development platforms. This ease of use allows developers to quickly integrate the sensor into their projects and start leveraging its capabilities. Whether for educational purposes, commercial robots, or low-speed unmanned vehicles, the RPLIDAR C1's combination of high-performance features and user-friendly design makes it a valuable asset in the realm of autonomous systems.

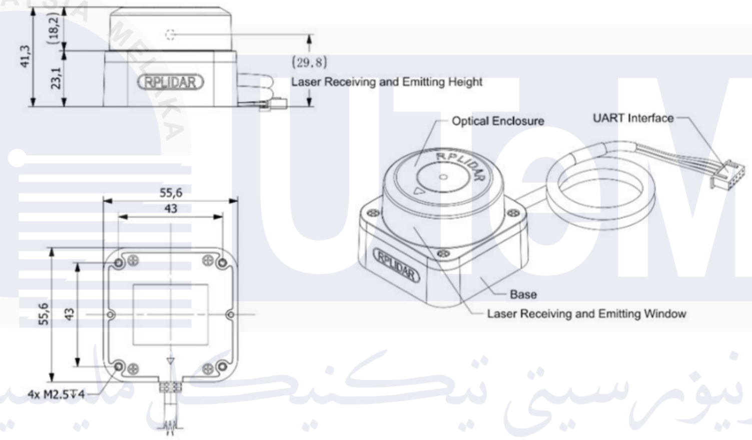


Figure 3.9: RPLidar C1 Mechanical Demension

3.3.5.4 Motor Driver

The L298N Motor Driver for Arduino is a versatile module that controls DC and stepper motors with an operating voltage of 5V to 35V and an output current of up to 2A per channel. It allows for independent control of two motors, including speed and direction, via H-Bridge configurations and PWM signals. The module features screw terminals for motor and power connections and pin headers for control signals. It is ideal for robotics, automation, and DIY projects, providing reliable motor control and ease of integration with Arduino.

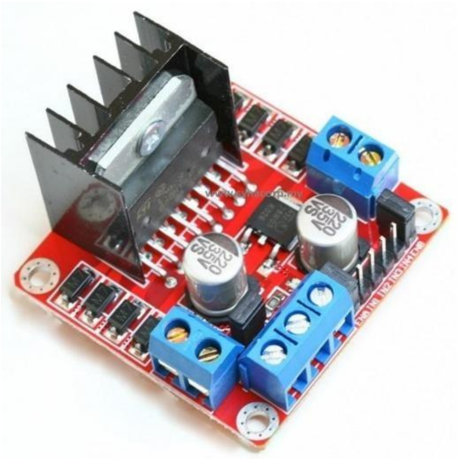


Figure 3.10: Motor Driver model L298N

3.3.5.5 Mini 560 DC 5V 5A step-down Stabilized Module

The Mini 560 DC 5V 5A step-down stabilized module is designed to convert higher input voltages (ranging from 7V to 20V) to a stable 5V output. It employs an integrally molded power inductor and a synchronous rectifier control chip, ensuring efficient and reliable power conversion. This module is particularly useful for providing a consistent 5V power supply to devices such as the Raspberry Pi 4b, ensuring stable operation without voltage fluctuations. Its compact size and high efficiency make it suitable for various applications, including robotics, mobile power supplies, and communication equipment.

Figure 3.11: Mini 560 DC 5V 5A step-down Stabilized Module

3.3.5.6 LM2596 DC-DC Adjustable Step Down Voltage Converter

The LM2596 DC-DC Adjustable Step Down Voltage Converter is a highly efficient voltage regulator, also known as a buck converter. It steps down higher input voltages (4.5V to 40V) to a stable, adjustable output voltage (1.2V to 37V). The LM2596 operates at a fixed frequency of 150 kHz, allowing for smaller-sized filter components. It requires minimal external components, making it easy to use and integrate into various applications. The module also features thermal shutdown and current-limit protection to ensure reliable performance.

Figure 3.12: LM2596 DC-DC Adjustable Step Down Voltage Converter

3.4 Limitation of proposed methodology

A limitation of the proposed methodology using the Raspberry Pi 4 Model B for running ROS and an Arduino Nano for motor control is the potential complexity in integration and communication between the two devices. While the Raspberry Pi 4 Model B offers significant computational power and memory, ensuring efficient data processing and real-time decision-making, the overall system may still face challenges in handling multiple tasks simultaneously if not optimized properly. Additionally, the learning curve and setup time for ROS can be considerable, potentially affecting the development speed and robustness of more sophisticated applications. Consequently, while this setup offers a powerful and flexible solution, it may require careful management and expertise to fully leverage its capabilities for demanding or computationally intensive tasks.

3.5 Summary

This chapter presents the limitation of using the Raspberry Pi 4 Model B for running ROS and an Arduino Nano for motor control. While the Raspberry Pi 4 Model B offers significant computational power and memory capacity, ensuring efficient data processing, real-time decision-making, and handling complex tasks, the integration and communication between the Raspberry Pi and Arduino Nano can be challenging. Additionally, the extensive software support and community resources for ROS require a steep learning curve, which may impact the development speed of more advanced applications. Nevertheless, this setup provides a powerful and flexible solution, but it requires careful management and expertise to fully leverage its high-performance and intensive processing capabilities.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

In this chapter, the results of the investigation into various key aspects of the system's performance are presented and analyzed. The discussion begins with an examination of navigation performance, assessing the system's ability to navigate accurately and efficiently. This is followed by an evaluation of mapping and localization accuracy, exploring how effectively the system can map its environment and accurately determine its location within that mapped space. The object avoidance effectiveness section discusses the system's capability to detect and avoid obstacles, ensuring smooth and safe operation.

Next, the efficiency of path planning is analyzed, examining how well the system can plan optimal paths based on the given criteria. Finally, the chapter concludes with an analysis of remote control and feedback mechanisms, assessing the responsiveness and reliability of remote control operations and the quality of feedback provided to the user. Each of these subsections provides detailed insights and observations that are critical for understanding the overall performance and robustness of the system.

4.2 Results and Analysis

4.2.1 Navigation Performance Integration

Integration and operation of this setup involve several steps. First, the RPLIDAR C1 collects laser scan data and sends it to the Raspberry Pi 4B via USB. The Raspberry Pi 4B, running ROS 2 Humble, processes this data to create a map of the environment and detect obstacles. The Nav 2 ROS package then uses this processed data to plan and execute navigation tasks, determining the best path to the target while avoiding obstacles. Motor control is managed by the Arduino Nano, which is connected to the Raspberry Pi via USB and receives commands to steer the robot based on the navigation plan.

Performance Results

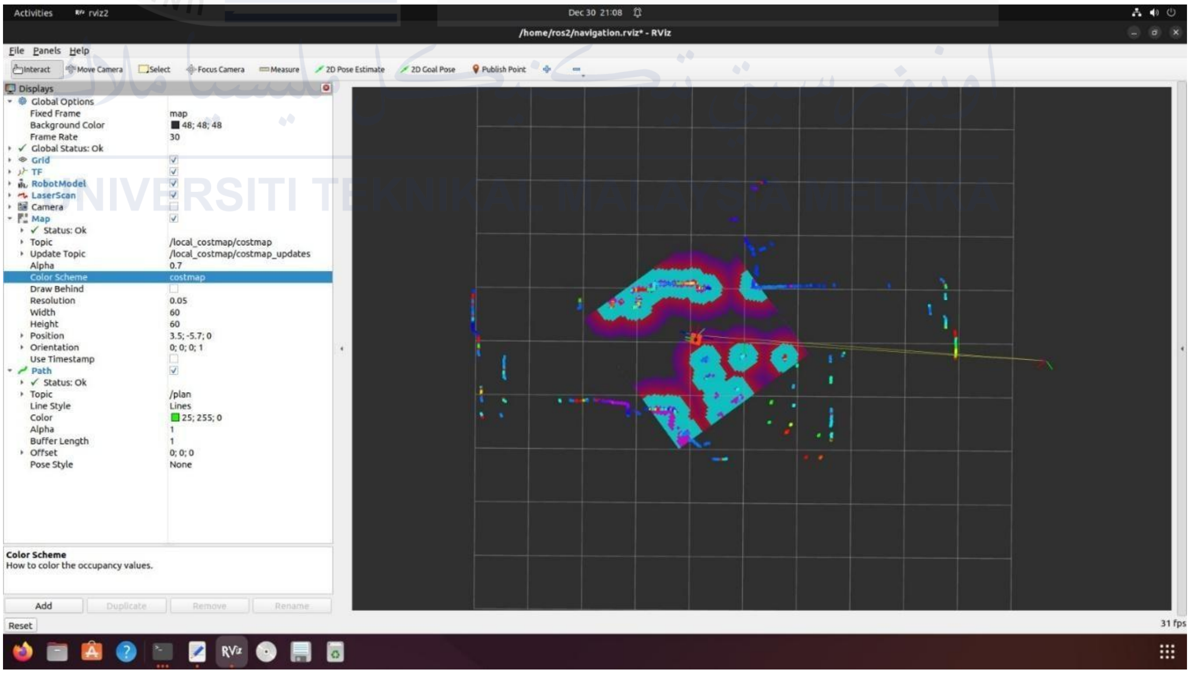


Figure 4.1: Visualization of robot's navigation costmap in RViz interface

Table 4.1: The performance of the robot to reach target

Test	To The Box (m)	Error (m)	Accuracy	Time (sec)
1	9 .0	0.02	High	37.0
2	7.2	0.02	High	29.0
3	5.4	0.01	Very High	22.0
4	3.6	0.02	High	14.0
5	1.8	0.01	Very High	7.0

Performance on different terrains.

The testing phase of this robot's development is a critical stage where its ability to navigate towards a target across various terrains is observed. This involves setting a designated target and monitoring the robot's movement as it attempts to reach this goal in different environmental conditions, such as carpet, cement, grass, and tile within the house. Figure 4.2 shows the illustration of the robot's performance testing. These tests are not just about reaching the target, but also about how effectively and efficiently the robot can do so. The observation of the robot's performance is tabulated as shown in Table 5.

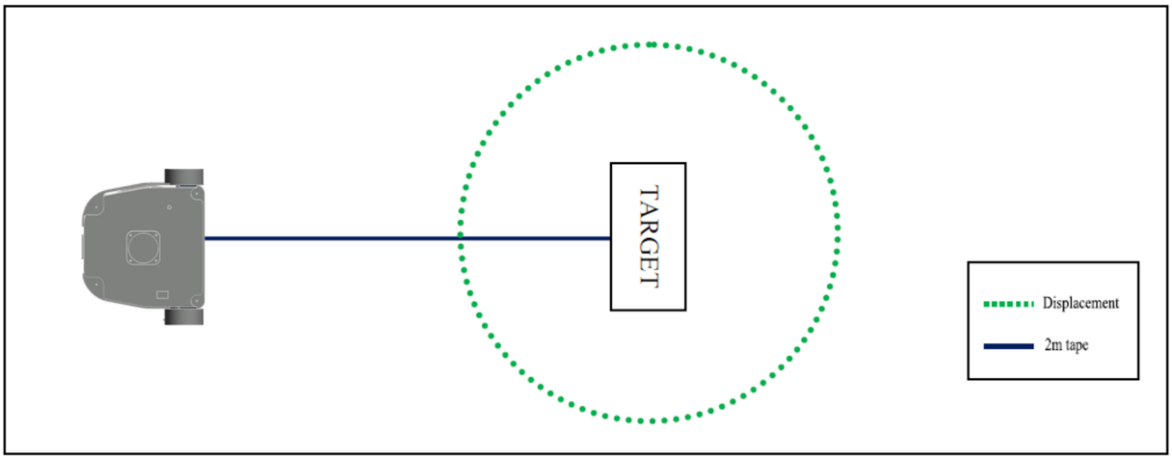


Figure 4.2: Robot performance testing procedure

Table 4.2: Terrain type performance of robot

	Terrain Type In 2m		
Test	**Carpet**	**Cement**	**Tile**
1	✗	✗	✓
2	✓	✓	✓
3	✗	✓	✓

The robot's performance was tested over three different terrain types for a 2-meter distance across three trials. In the first test, the robot failed to navigate on both the carpet and cement surfaces, but succeeded on the tile surface. In the second test, the robot successfully navigated all three surfaces: carpet, cement, and tile. In the third test, the robot failed on the carpet but managed to pass on the cement and tile surfaces. This evaluation highlights the varying degrees of success the robot experienced on different terrains, with the most consistent success observed on the tile surface.

Carpet | Cement | Tile

Figure 4.3: Type Of Terrain

4.2.2 Mapping and Localization Accuraccy

Integration Mapping and localization in this setup involve a series of coordinated steps. The RPLIDAR C1 collects laser scan data to understand the environment. This data is sent to the Raspberry Pi 4B running Ubuntu 22.04 and ROS 2 Humble, where it is processed. The processing involves creating a detailed map of the surroundings and pinpointing the robot's position within that map. The Nav 2 ROS package then utilizes this map to plan navigation tasks and ensure accurate localization. The motor control is handled by the Arduino Nano, which receives movement commands via USB from the Raspberry Pi to navigate the robot along the planned path.

Mapping and Localization Result

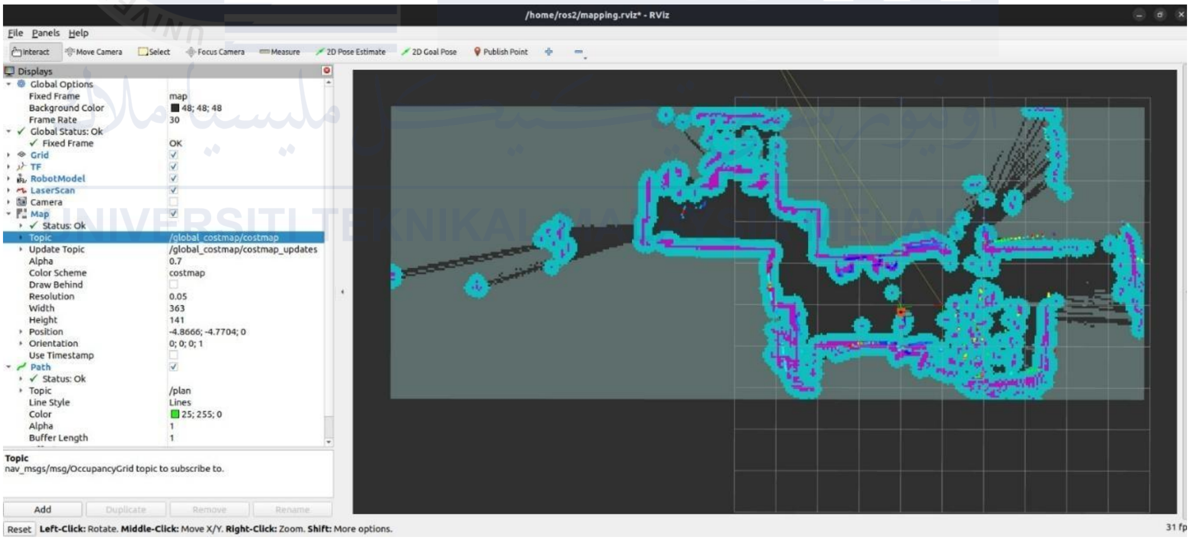


Figure 4.4: RViz visualization of robot mapping from living room to kitchen.

Table 4.3: The results of the localization after mapping have been saved

Test	Starting	End Point	Time Taken to Reach (min/sec)
1	Living Room	Kitchen	1 minutes 20 seconds
2	Living Room	Dining area	40 second
3	Porch	Kitchen	1 minutes 30 seconds

The table, titled "Table 6: The results of the localization after mapping have been saved," summarizes the robot's performance in navigating within a house. In Test 1, the robot takes 1 minute and 20 seconds to move from the Living Room to the Kitchen. Test 2 shows the robot taking 40 seconds to travel from the Living Room to the Dining Area. In Test 3, the robot takes 1 minute and 30 seconds to go from the Porch to the Kitchen. This is because of the obstacles that the robot needs to avoid to reach its destination. This table highlights the time required for the robot to reach various destinations from specific starting points within the house.

4.2.3 Object Avoidance Effectivness

Object avoidance in this setup operates through a combination of sensor data collection, processing, and motor control. The RPLIDAR C1 collects laser scan data to detect objects and obstacles in the environment. This data is sent to the Raspberry Pi 4B, which runs Ubuntu 22.04 and ROS 2 Humble, to process the information. The Nav 2 ROS package uses this processed data to determine the location of obstacles and plans the best path to avoid them while navigating. The Arduino Nano, connected via USB to the Raspberry Pi, receives commands to control the motors, steering the robot to avoid detected obstacles. This dynamic adjustment ensures the robot can navigate safely and efficiently.

Obstacles Avoidance Result

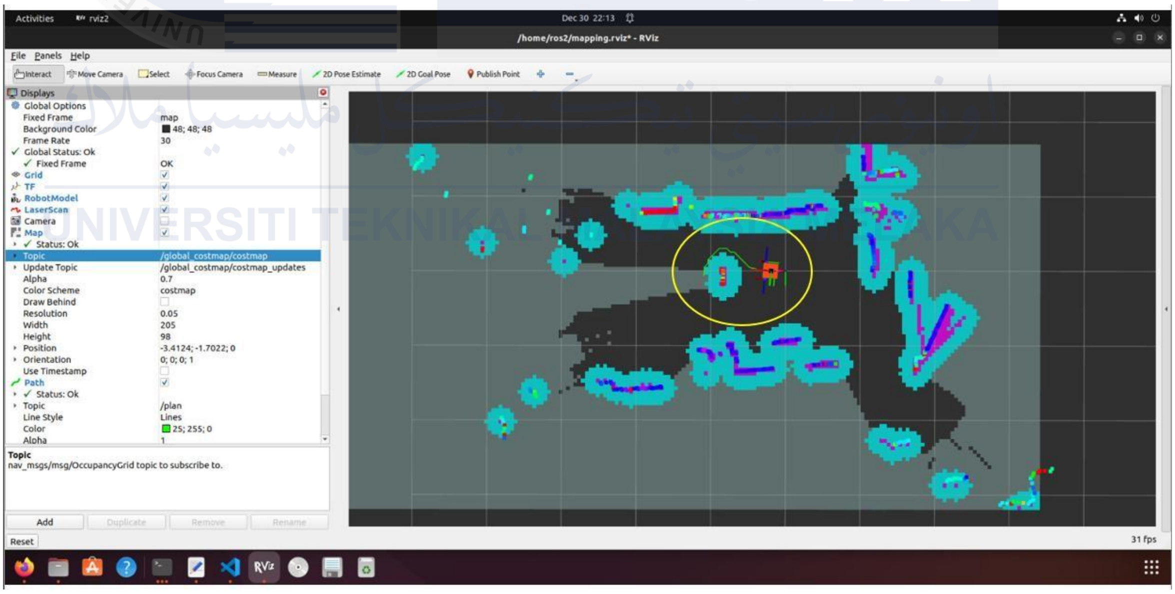


Figure 4.5: RViz visualization of the robot's obstacle avoidance capabilities.

Table 4.4: Type of obstacle to avoid

Test	Static Obstacle Avoidance (Success/Failed)	Time Taken to Pass the Obstacle (s)	Dynamic Obstacle Avoidance Situation (Success/Failed)	Time Taken to Pass the Obstacle (s)	Reach Goal (Yes/No)
1	Success	18.0	When someone walking Success	20.0	Yes
2	Success	18.0	When cat running Success	15.0	Yes

Table 4.5: The robot navigates through the space between two obstacles

Test	Distance Between Two Obstacle (m)	Time Taken to Configure Pass Obstacle(s)	Reach Goal
1	More than 1.5 meter	1.0	Yes
2	1.5 meter	5.0	Yes
3	Less than 1 meter	10.0	No

These two tables summarizing a robot's obstacle navigation performance. It details how the robot navigates between static and dynamic obstacles, showing success rates and times for each trial. The tables highlight the robot's ability to avoid collisions and its efficiency in completing the navigation tasks, providing clear metrics on the robot's overall performance in different obstacle scenarios.

4.2.4 Path Planing Efficiency

Path planning efficiency in this setup involves several components working together seamlessly. The RPLIDAR C1 collects laser scan data to understand the environment and detect obstacles. This data is processed by the Raspberry Pi 4B running Ubuntu 22.04 and ROS 2 Humble, which utilizes the Nav 2 ROS package to generate a map and plan efficient paths. The Nav 2 package evaluates various potential paths and selects the most optimal one, considering factors like distance and obstacle avoidance. The Arduino Nano, connected via USB to the Raspberry Pi, receives motor control commands to execute the planned path. This ensures the robot navigates smoothly and efficiently in real-time. To evaluate path planning efficiency, visualization tools like RViz in ROS 2 can be used to monitor the planned paths and actual navigation. Metrics such as the time taken to reach the target, deviations from the planned path, and the number of obstacles avoided are tracked. Adjustments to the Nav 2 parameters can further enhance path planning efficient.

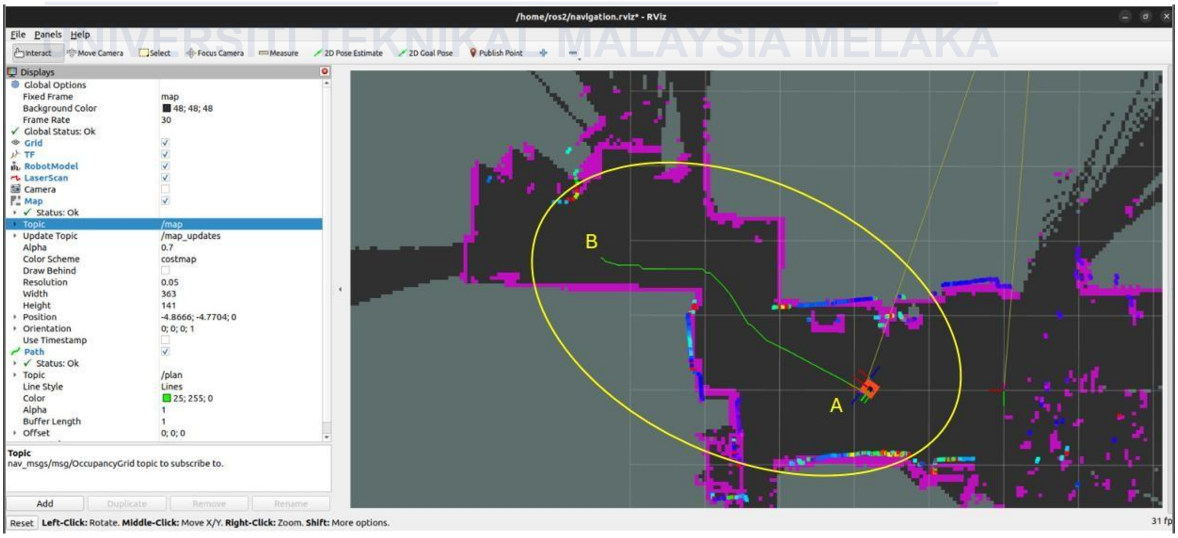


Figure 4.6: RViz visualization of the robot's of path planing. Robot will follow green line as its path to go from point A to point B

4.2.5 Remote Control and Feedback Analysis

Object Remote control and feedback analysis in this setup work through a combination of manual input and data processing. The RPLIDAR C1 collects laser scan data, which is then processed by the Raspberry Pi 4B running Ubuntu 22.04 and ROS 2 Humble. This processed data aids in navigation tasks, whether manual or autonomous. When using a keyboard for manual control or to interrupt autonomous movement, commands are sent directly to the Raspberry Pi.The Arduino Nano, connected via USB, receives these commands and controls the motors to navigate the robot accordingly. Feedback from the robot's movements and the environment is continuously collected and analyzed to ensure accurate control and navigation.

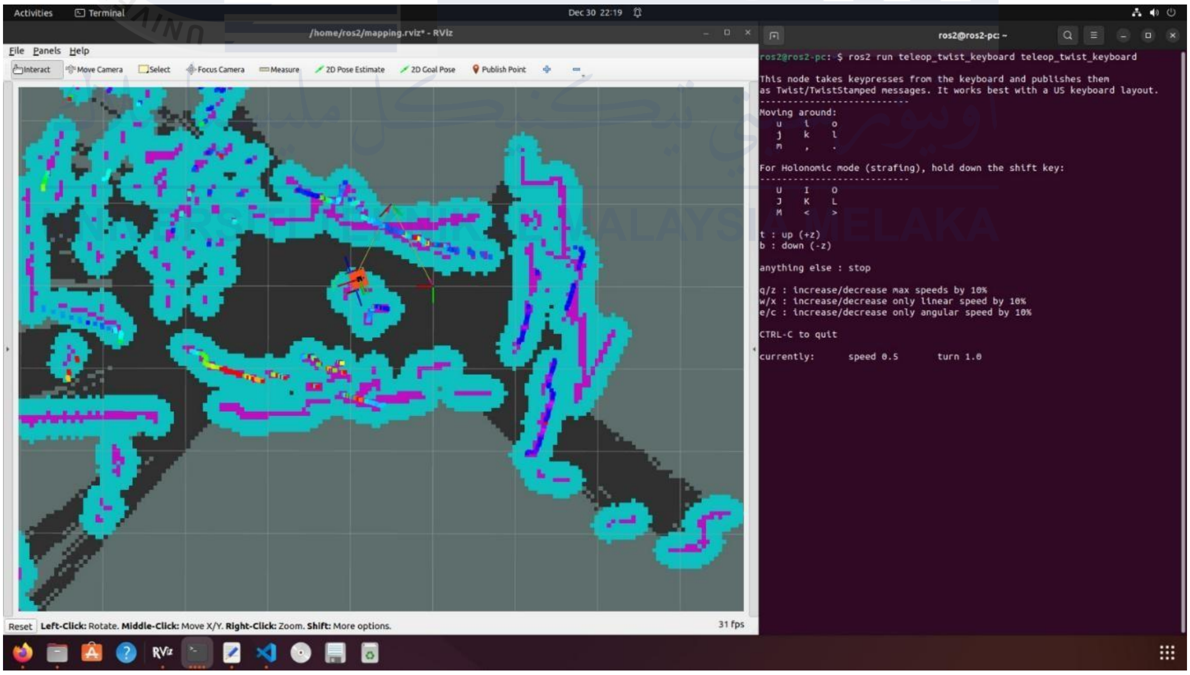


Figure 4.7: When controlling RViz manually, the visualizations indicate that the saved data contains significant noise.

4.3 Summary

This chapter delves into various performance aspects of the system, beginning with an analysis of navigation accuracy and efficiency. It then evaluates mapping and localization, focusing on how effectively the system can create accurate maps and determine its position within those maps. The discussion also covers object avoidance, highlighting the system's ability to detect and steer clear of obstacles to ensure smooth operation.

Following this, the chapter examines path planning efficiency, assessing the system's capability to plan optimal routes based on set criteria. The final section explores remote control and feedback mechanisms, evaluating the responsiveness and reliability of remote-control operations and the quality of feedback provided to users. Each section offers detailed insights, crucial for understanding the system's overall performance and robustness.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This thesis presents a project that employs the Raspberry Pi 4B running Ubuntu 22.04 with ROS 2 Humble and the Nav 2 ROS package to create an automated differential wheel room mapping system. The Raspberry Pi 4B serves as the main processor, offering robust capabilities for running complex software and handling various sensors and components. For input, the RP LiDAR C1 is used for accurate and detailed mapping of the room's area, providing real-time data for efficient navigation and obstacle avoidance.

The motor control is managed by an Arduino Nano, which is connected via USB to the Raspberry Pi 4B. This setup ensures precise motor operations and seamless integration with other components. For path and obstacle detection, an ultrasonic sensor and a servo motor are utilized, allowing the differential wheel system to navigate the room autonomously.

This system enables autonomous navigation, mapping, and localization, object avoidance, path planning, remote control, and sensor data collection. The Raspberry Pi 4B's Wi-Fi capabilities facilitate remote monitoring and control, enhancing the overall functionality of the project.

Overall, this project demonstrates the versatility and power of the Raspberry Pi 4B and Arduino Nano in advanced robotics and IoT applications, providing a reliable solution for automated room mapping and navigation.

5.2 Potential for Commercialization

The integration of RP LiDAR C1 for input, paired with a Raspberry Pi 4b running Ubuntu 22.04 and ROS 2 Humble, alongside the Arduino Nano for motor control, showcases significant potential for commercialization. This advanced setup ensures robust solutions for autonomous navigation, mapping and localization, object avoidance, and path planning. Additionally, it allows for remote control and comprehensive sensor data collection, making it a versatile platform for various applications.

The modularity and adaptability of this system enables it to be tailored to diverse use cases, from industrial automation to consumer robotics. Its flexibility makes it an appealing option for developers and businesses looking to implement advanced robotic functionalities. The integration of these components creates a comprehensive solution that can be easily scaled and customized to meet specific needs.

As industries increasingly seek innovative solutions to enhance efficiency and safety, this integrated system stands out as a highly promising commercial product. Its capacity to provide reliable and advanced robotic functionalities positions it well for broad adoption across multiple sectors. By addressing key technological challenges and offering scalable solutions, this setup is poised to become a valuable asset in the rapidly evolving field of robotics.

5.3 Recommendation Future Works

For future improvements, this section will discuss the future work needed to enhance the software functionality:

i) **Enhance Image Processing Capabilities:** Integrate advanced image processing techniques using infrared cameras and night vision. This addition will improve the system's performance in low-light or no-light environments, enhancing its ability to detect and navigate obstacles effectively.

ii) **Add Environmental Sensors:** Implement additional sensors, such as gas detection sensors, to gain more comprehensive knowledge about the surrounding environment. These sensors will enhance the system's ability to detect potential dangers, such as hazardous gases, ensuring a higher level of safety and adaptability. By gathering more detailed environmental data, the system can make better-informed decisions and respond more effectively to changing conditions.

iii) **Optimize System Performance:** Focus on optimizing the overall system performance by refining algorithms and improving hardware-software integration. This will enhance efficiency, reduce processing times, and ensure smoother operation, resulting in a more reliable and effective system.

REFERENCES

1. H. Yin, Z. Lin, and J. K. W. Yeoh, "Semantic localization on BIM-generated maps using a 3D LiDAR sensor," *Automation in Construction*, vol. 146, p. 104641, Feb. 2023, doi: 10.1016/j.autcon.2022.104641. Available: https://doi.org/10.1016/j.autcon.2022.104641
2. S. Coradeschi, A. Loutfi, and B. Wrede, "A Short Review of Symbol Grounding in Robotic and Intelligent Systems," *KI. Künstliche Intelligenz/KI - Künstliche Intelligenz*, vol. 27, no. 2, pp. 129–136, Mar. 2013, doi: 10.1007/s13218-013-0247-2. Available: https://doi.org/10.1007/s13218-013-0247-2
3. G. Forzieri, L. Tanteri, G. Moser, and F. Catani, "Mapping natural and urban environments using airborne multi-sensor ADS40–MIVIS–LiDAR synergies," *International Journal of Applied Earth Observation and Geoinformation*, vol. 23, pp. 313–323, Aug. 2013, doi: 10.1016/j.jag.2012.10.004. Available: https://doi.org/10.1016/j.jag.2012.10.004
4. J. Kaltenthaler, H. A. Lauterbach, D. Borrmann, and A. Nüchter, "Pose estimation and mapping based on IMU and LiDAR," IFAC-PapersOnLine, vol. 55, no. 8, pp. 71–76, Jan. 2022, doi: 10.1016/j.ifacol.2022.08.012. Available: https://doi.org/10.1016/j.ifacol.2022.08.012
5. H. Si, X. Hu, and L. Xu, "Research and Implementation of Indoor Location for Home Service Robot," IOP Conference Series. Materials Science and Engineering, vol. 711, no. 1, p. 012035, Jan. 2020, doi: 10.1088/1757-899x/711/1/012035. Available: https://doi.org/10.1088/1757-899x/711/1/012035
6. C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," IEEE Transactions on Robotics,

vol. 32, no. 6, pp. 1309–1332, Dec. 2016, doi: 10.1109/tro.2016.2624754. Available: https://doi.org/10.1109/tro.2016.2624754

7. Y.-C. Lee, “LSMCL: Long-term Static Mapping and Cloning Localization for autonomous robot navigation using 3D LiDAR in dynamic environments,” Expert Systems With Applications, vol. 241, p. 122688, May 2024, doi: 10.1016/j.eswa.2023.122688. Available: https://doi.org/10.1016/j.eswa.2023.122688
8. B. Canovas, A. Nègre, and M. Rombaut, “Onboard dynamic RGB-D simultaneous localization and mapping for mobile robot navigation,” *ETRI Journal*, vol. 43, no. 4, pp. 617–629, Aug. 2021, doi: 10.4218/etrij.2021-0061. Available: https://doi.org/10.4218/etrij.2021-0061
9. P. F. McManamon, LiDAR Technologies and Systems. SPIE-International Society for Optical Engineering, 2019.
10. J. A. Gonçalves and R. Henriques, “UAV photogrammetry for topographic monitoring of coastal areas,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 104, pp. 101–111, Jun. 2015, doi: 10.1016/j.isprsjprs.2015.02.009. Available: https://doi.org/10.1016/j.isprsjprs.2015.02.009
11. Z. Zijiang, W. Jingwen, C. Haoran, J. Lin, and Y. Shuo, “Mapping method of single LiDAR for indoor degraded environment,” Computers & Electrical Engineering, vol. 103, p. 108284, Oct. 2022, doi: 10.1016/j.compeleceng.2022.108284. Available: https://doi.org/10.1016/j.compeleceng.2022.108284
12. I. Rusli, B. R. Trilaksono, and W. Adiprawita, “RoomSLAM: Simultaneous Localization and Mapping With Objects and Indoor Layout Structure,” IEEE Access, vol. 8, pp. 196992–197004, Jan. 2020, doi: 10.1109/access.2020.3034537. Available: https://doi.org/10.1109/access.2020.3034537

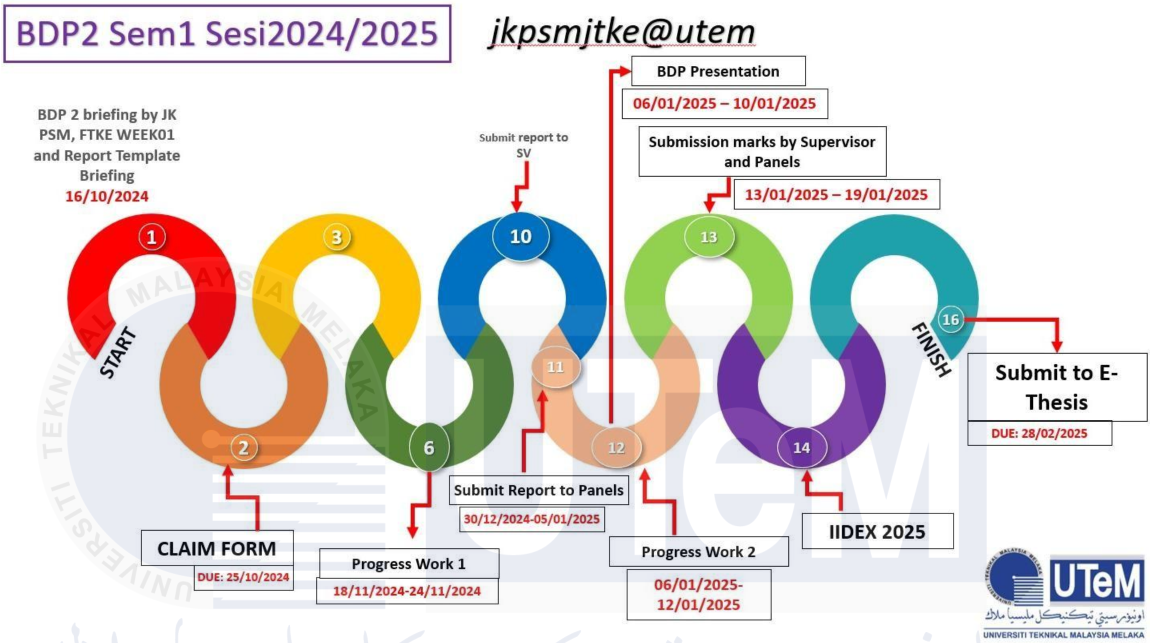
13. G. Ge, Y. Zhang, W. Wang, Q. Jiang, L. Hu, and Y. Wang, “Text-MCL: Autonomous Mobile Robot Localization in Similar Environment Using Text-Level Semantic Information,” Machines, vol. 10, no. 3, p. 169, Feb. 2022, doi: 10.3390/machines10030169. Available: https://doi.org/10.3390/machines10030169
14. D. Tu, H. Cui, and S. Shen, “PanoVLM: Low-Cost and accurate panoramic vision and LiDAR fused mapping,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 206, pp. 149–167, Dec. 2023, doi: 10.1016/j.isprsjprs.2023.11.012. Available: https://doi.org/10.1016/j.isprsjprs.2023.11.012
15. A. Siswoyo, N. P. Arbiyanti, and R. W. Indrawan, “Design and Control Development of an Autonomous Visitor Guiding Robot in a Hospital Environment,” Jurnal Teknologi/Jurnal Teknologi, vol. 13, no. 1, Jun. 2023, doi: 10.35134/jitekin.v13i1.81. Available: https://doi.org/10.35134/jitekin.v13i1.81
16. J. Wang, L. Wang, P. Peng, Y. Jiang, J. Wu, and Y. Liu, “Efficient and accurate mapping method of underground metal mines using mobile mining equipment and solid-state lidar,” Measurement, vol. 221, p. 113581, Nov. 2023, doi: 10.1016/j.measurement.2023.113581. Available: https://doi.org/10.1016/j.measurement.2023.113581
17. Y. Su, T. Wang, S. Shao, C. Yao, and Z. Wang, “GR-LOAM: LiDAR-based sensor fusion SLAM for ground robots on complex terrain,” Robotics and Autonomous Systems, vol. 140, p. 103759, Jun. 2021, doi: 10.1016/j.robot.2021.103759. Available: https://doi.org/10.1016/j.robot.2021.103759
18. C. Liu et al., “Hybrid metric-feature mapping based on camera and Lidar sensor fusion,” Measurement, vol. 207, p. 112411, Feb. 2023, doi: 10.1016/j.measurement.2022. 112411. Available: https://doi.org/10.1016/j.measurement.2022.112411

19. U. Weiss and P. Biber, “Plant detection and mapping for agricultural robots using a 3D LIDAR sensor,” *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 265–273, May 2011, doi: 10.1016/j.robot.2011.02.011. Available: https://doi.org/10.1016/j.robot.2011.02.011

20. H. Yin, Z. Lin, and J. K. W. Yeoh, “Semantic localization on BIM-generated maps using a 3D LiDAR sensor,” Automation in Construction, vol. 146, p. 104641, Feb. 2023, doi: 10.1016/j.autcon.2022.104641. Available: https://doi.org/10.1016/j.autcon.2022.104641

APPENDICES

Appendix 5.A: Infographic of BDP 2



Appendix 5.B: RPLiDAR C1 Measurement Performance

Item	Detail
Distance Range[1]	White object: 0.05-12 meters (under 70% reflection)
	Black object: 0.05-6 meters (under 10% reflection)
Sample Rate[2]	5KHz
Scanning Frequency[3]	8~12Hz, 10Hz@typical
Angular Resolution	0.72°@typical
Scan Field Flatness	0°~1.5° (can be customized)
Communication Interface	TTL UART
Communication Speed	460800
Accuracy[4]	±30mm
Resolution	15mm
Degree of Protection	IP54
Ambient Light Limit	40,000lux

Appendix 5.C: RPLiDAR C1 Optical and Interface Specfication

Item	Unit	Min	Typical	Max	Remarks
Working Wavelength	Nanometer (nm)	895	905	915	Infrared Light Band
Laser Power	Watt (W)	-	20	-	Peak power
Pulse Length	Nanosecond (ns)	-	1.4	-	-
Laser Safety Class	-	-	IEC-60825 Class 1	-	-

Optical Specfication

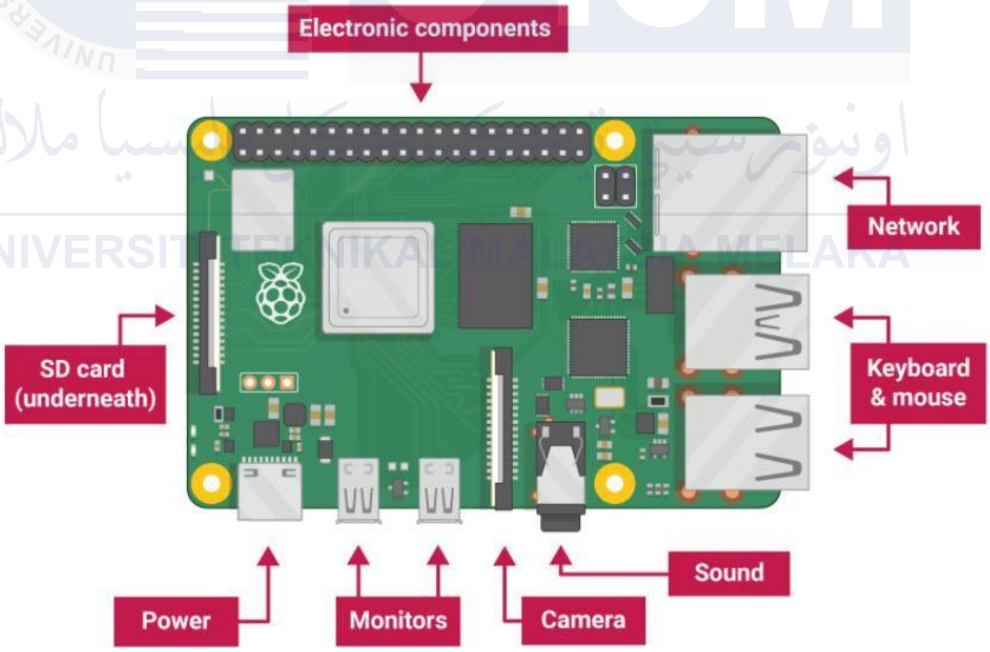
Item	Unit	Min	Typical	Max	Remarks
Baud Rate	bps	-	460800	-	-
Working Mode	-	-	8 data bits, 1 stop bit, no parity check	-	8n1
Output High Voltage	V	2.9	3.3	3.5	Output signal with high voltage
Output Low Voltage	V	-	-	0.4	Output signal with low voltage
Input High Voltage	V	2.4	3.3	3.5	Input signal with high voltage
Input Low Voltage	V	0	-	0.4	Input signal with high voltage

Interface Specfication

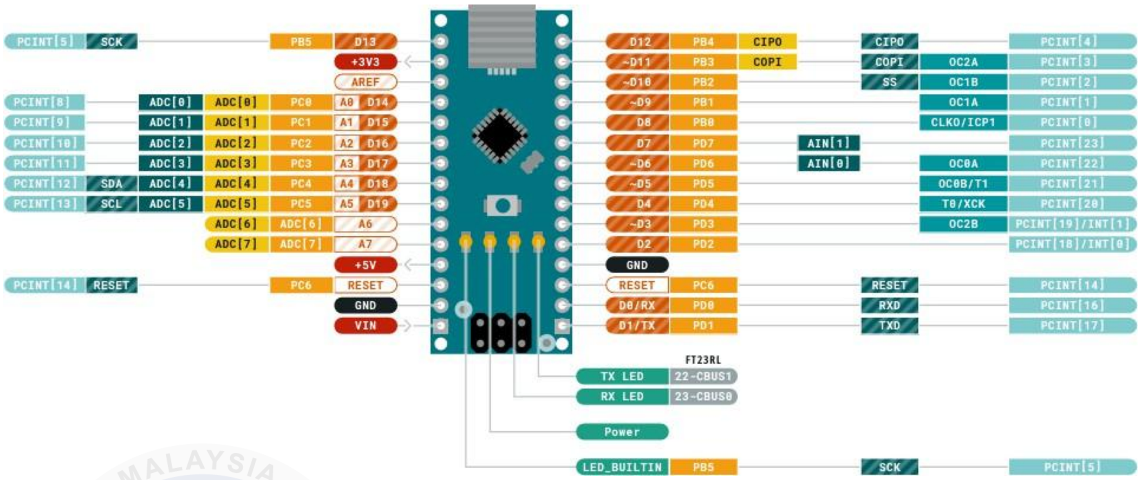
Appendix 5.D: RPLiDAR Miscellaneous Specfication

Item	Unit	Min	Typical	Max	Remarks
Weight	Gram (g)	-	110	-	
Switch-on Temperature	Degree Celsius (ºC)	0	-	-	
Working Temperature Range	Degree Celsius (ºC)	-10	25	40	
Storage Temperature Range	Degree Celsius (ºC)	-20	25	60	

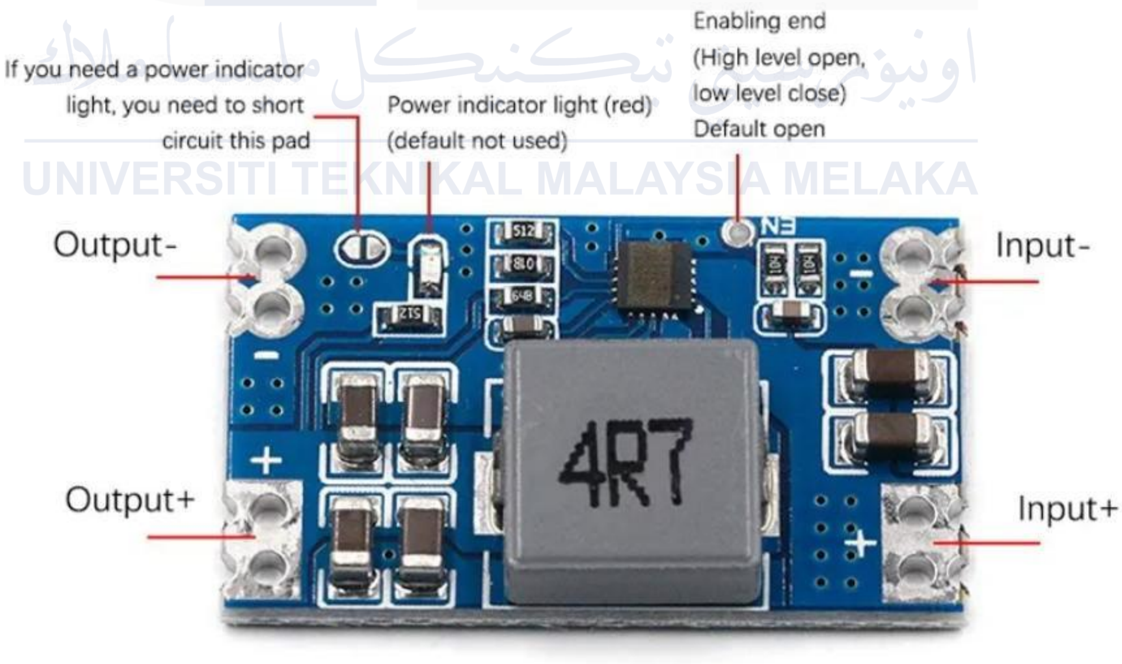
Appendix 5.E Raspberry PI 4B Compartment



Appendix 5.F: Arduino Nano V3 Pin Declaration



Appendix 5.G: Mini 560 DC 5V 5A step-down Stabilized Module Functionality



Appendix 5.H: SLAM Programming

```
import os
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument, LogInfo
from launch.conditions import UnlessCondition
from launch.substitutions import LaunchConfiguration, PythonExpression
from launch_ros.actions import Node
from ament_index_python.packages import get_package_share_directory
from nav2_common.launch import HasNodeParams
def generate_launch_description():
    use_sim_time = LaunchConfiguration('use_sim_time')
    params_file = LaunchConfiguration('params_file')
    default_params_file = os.path.join(get_package_share_directory("robot"),
                                   'config', 'mapper_params_online_async.yaml')
    declare_use_sim_time_argument = DeclareLaunchArgument(
        'use_sim_time',
        default_value='true',
        description='Use simulation/Gazebo clock')
    declare_params_file_cmd = DeclareLaunchArgument(
        'params_file',
        default_value=default_params_file,
        description='Full path to the ROS2 parameters file to use for the slam_toolbox node')
    # If the provided param file doesn't have slam_toolbox params, we must pass the
    # default_params_file instead. This could happen due to automatic propagation of
    # LaunchArguments. See:
    # https://github.com/ros-planning/navigation2/pull/2243#issuecomment-800479866
    has_node_params = HasNodeParams(source_file=params_file,
                                  node_name='slam_toolbox')
    actual_params_file = PythonExpression(['"', params_file, '" if ', has_node_params,
                                         ' else "', default_params_file, '"'])
    log_param_change = LogInfo(msg=['provided params_file ',  params_file,
                                  ' does not contain slam_toolbox parameters. Using default: ',
                                  default_params_file],
                              condition=UnlessCondition(has_node_params))
    start_async_slam_toolbox_node = Node(
        parameters=[
          actual_params_file,
          {'use_sim_time': use_sim_time}
                package='slam_toolbox',
        executable='async_slam_toolbox_node',
        name='slam_toolbox',
        output='screen')
    ld = LaunchDescription()
    ld.add_action(declare_use_sim_time_argument)
    ld.add_action(declare_params_file_cmd)
    ld.add_action(log_param_change)
    ld.add_action(start_async_slam_toolbox_node)

    return ld
```