

Design and Implementation of an Indoor Autonomous Mobile Robot with LiDAR-Based Mapping

¹ Lukáš LIZÁK, ² Tibor VINCE

^{1,2} Department of Theoretical and Industrial electrotechnics, FEI, TUKE, Slovakia

¹lukas.lizak@student.tuke.sk, ²tibor.vince@tuke.sk

Abstract — This paper presents the design and implementation of an indoor autonomous mobile robot equipped with a LiDAR sensor for mapping and obstacle avoidance. The system uses two microcontrollers—Arduino MEGA 2560 for motor and sensor control, and ESP-32 for data processing and communication via TCP/IP. Infrared proximity sensors and rotary encoders complement the LiDAR to improve obstacle detection. A MATLAB application enables real-time control, speed adjustment, and visualization of the 2D environment map, as well as calculation of room perimeter and area. The proposed solution demonstrates reliable navigation in both manual and autonomous modes, offering potential applications in research, education, and prototyping of autonomous robotic systems.

Keywords — Arduino, Autonomous navigation, ESP-32, Fusion 360, LiDAR, MATLAB, Sensors

I. INTRODUCTION

Autonomous mobile robots are increasingly being used in various applications, from industrial automation to indoor navigation and mapping. Their ability to operate without human intervention, sense their environment, and make real-time decisions makes them ideal for dynamic and complex tasks. A key challenge in autonomous robotics is the integration of reliable hardware and efficient software to ensure smooth operation, accurate environment sensing, and effective mobility.

This work focuses on the design and implementation of an indoor autonomous ground robot equipped with a LiDAR sensor for mapping and obstacle detection. The system utilizes two microcontrollers—Arduino MEGA 2560 for motor and sensor control, and ESP-32 for communication and data processing. The integration of infrared sensors, rotary encoders, and a RPLiDAR A1 sensor provides comprehensive environmental awareness.

A MATLAB application serves as the central interface, enabling real-time visualization of the robot's surroundings, as well as control over speed and movement modes. The robot is capable of operating in both manual and autonomous modes and can measure the area and perimeter of the mapped environment.

II. ANALYSIS OF REQUIREMENTS

An autonomous ground robot is designed to navigate indoor environments, avoid obstacles, and process LiDAR data.

A. Hardware part

Power Supply – Selecting appropriate power sources is essential to ensure the mobile robot operates for at least 30 minutes on a single charge.

Movement – A suitable mobility system must be implemented for smooth movement on flat indoor surfaces.

Sensors – Sensors are required to enable autonomous navigation and obstacle avoidance.

Structure – The design and material of the robot's body should support stable movement while maintaining structural integrity comparable to plexiglass.

LiDAR Sensor – A key objective is the integration of a ground-based LiDAR sensor and processing its data. The selected sensor must be suitable for terrestrial use.

B. Software part

Autonomy – The system is designed to operate independently without human intervention, using integrated sensors such as LiDAR and infrared to navigate and avoid obstacles in real time.

Manual control – A user interface allows switching between autonomous and manual modes, including control over the robot's movement speed.

Room perimeter and area detection – The system can calculate the perimeter and area of the room based on collected data.

Environment mapping – A spatial map can be generated using data from the LiDAR sensor.

III. DESIGN OF HARDWARE PART

This section focuses on the hardware design of the physical components of an autonomous mobile robot. It is based on the requirements analysis outlined in Chapter II, which must be met to ensure the robot's full functionality.

A. Power supply

The power source for the proposed mobile robot can consist of five 3.7 V L18650 batteries or a power bank with a minimum capacity of 20,000 mAh. As stated in requirement II.A, the source must power the robot for at least 30 minutes, which is achievable with either option. In a previous bachelor thesis, over 30 minutes of operation was reached using two batteries for the motors and two for the electronics, though the movement was not smooth. To improve this, H-bridges were replaced to support higher voltage, allowing for more batteries for motor power.

For this project, batteries were chosen. Three batteries power the H-bridge and motors, while the remaining two supply the microcontrollers, LiDAR sensor, IR sensors, and encoders, housed in two separate holders.

B. Movement

Requirement II.A concerns the movement of the mobile robot. Since it is a ground robot operating indoors on flat surfaces, it will be equipped with four motors and four wheels, allowing movement across various surfaces. As the robot won't traverse rough terrain, wheels are preferred over tracks. The design uses four DC gear motors with permanent magnets and built-in encoders, with a recommended operating voltage of up to 6 V and a range between 3 V and 12 V.

C. Sensors

Another requirement in II.A focuses on connecting sensors to the mobile robot for processing information from its surroundings. The main sensor used is the RPLiDAR A1 by SLAMTEC, which detects nearby objects and measures their distance, allowing the robot to map the environment.

To perceive obstacles, the robot is also equipped with four infrared sensors (TCRT5000), placed at the front and rear. These help detect obstacles not visible to the LiDAR due to its top-mounted position, enabling the robot to react and choose a safe direction of movement.

D. Construction

As stated in requirement in II.A, the robot's chassis must be sturdy enough to support the LiDAR, sensors, and microcontroller boards, yet light enough not to overburden the motors. Initially, plexiglass was used in a previous bachelor project, but the updated design was modelled in Fusion 360.

Fig. 1 shows the bottom plate with spacer pillars supporting the top plate. The bottom plate holds the PCB with all essential electronic components, including H-bridges for motor control. The top plate (Fig. 2) consists of the LiDAR, infrared sensors, battery holders, and switches—strategically placed for optimal performance and easy access.

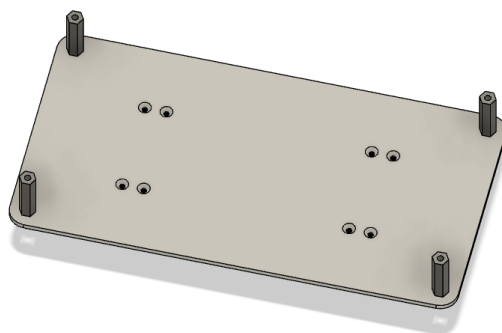


Fig. 1 Bottom plate of the robot chassis with spacer pillars and electronic component mounts

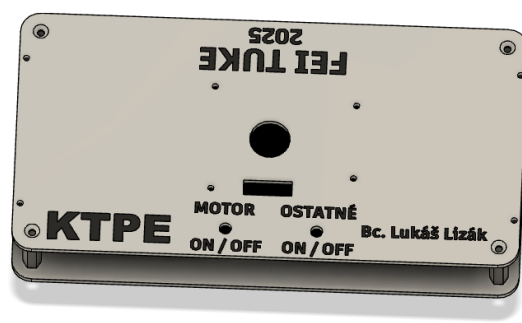


Fig. 2 Top plate layout showing LiDAR, infrared sensors, and battery holders

Fig.3 illustrates the method of mounting the motors onto the robot's chassis. This mounting approach ensures stability and proper alignment of the motors, allowing the robot to move efficiently across various surfaces.



Fig. 3 Motor mounting assembly for the chassis

Fig. 4 shows the component used to mount the infrared sensor. It is designed to securely hold the sensor in the correct position, ensuring optimal alignment for obstacle detection and reliable operation within the robot's system. There are four of these components—two at the front and two at the back—since the robot is equipped with four TCRT5000 infrared sensors. This setup enables effective obstacle detection and avoidance in both directions.

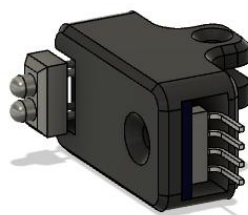


Fig. 4 Infrared sensor mounting bracket

The final design in Fusion 360 (Fig. 5) presents the complete model of the mobile robot, including

the placement of key components such as infrared sensors, LiDAR, and batteries. The layout ensures optimal positioning for efficient operation—sensors are placed for effective obstacle detection, the LiDAR is centrally mounted for environmental mapping, and the batteries are properly integrated to power all systems reliably.

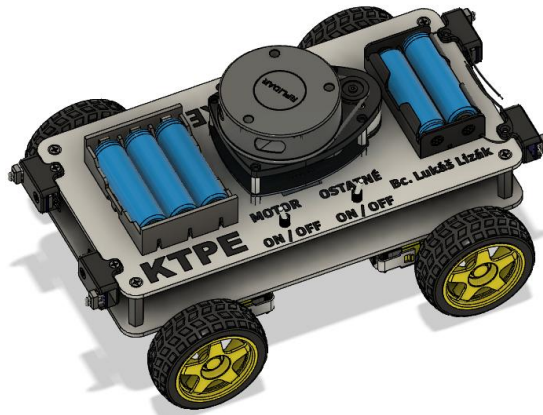


Fig. 5 Final robot design modeled in Fusion 360

E. Printed circuit board

Although sensors and microcontrollers could be connected using flexible wires, a wiring diagram was created to ensure stable and permanent connections (Fig. 6). This diagram served as the basis for designing the printed circuit board (PCB). The design also includes space for connecting an external H-bridge, essential for motor control and enabling efficient robot movement. Based on this schematic, a PCB was developed to provide all necessary electrical connections between components.

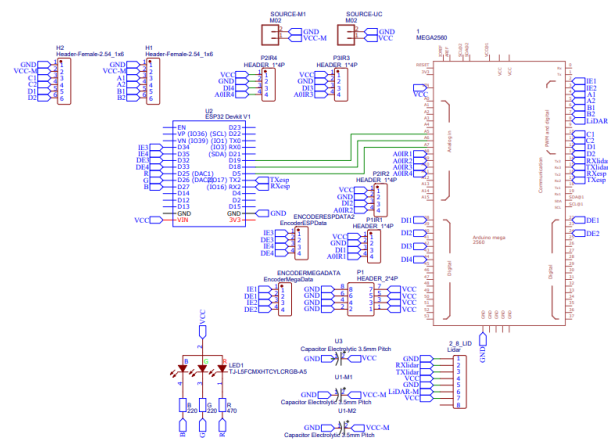


Fig. 6 Wiring schematic of the printed circuit board (PCB)

Fig. 7 shows the PCB design based on the schematic in Figure 6. It illustrates the connections between sensor terminals and the development boards. Standard trace width is 0.4 mm, with a single 1 mm trace used for power lines leading to the H-bridge connections. This thicker trace was chosen to handle higher current during motor operation, helping prevent overheating or damage to the circuit.

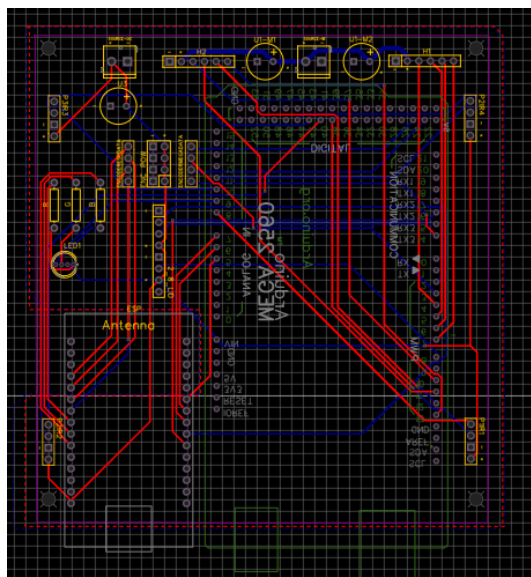


Fig. 7 PCB layout showing power and signal traces

IV. DESIGN OF SOFTWARE PART

This autonomous mobile robot requires not only hardware but also firmware to enable its movement and to process the acquired data.

A. Code for Arduino

The code uploaded to the Arduino MEGA microcontroller should include a program for acquiring data from the LiDAR sensor, enabling object detection and analysis around the robot. It should also control the robot's movement and implement obstacle detection and avoidance algorithms. Additionally, the code must handle UART communication between the Arduino MEGA and the ESP-32, ensuring efficient data exchange between the two devices.

B. Code for ESP-32

The ESP-32 code should handle receiving data from the Arduino MEGA and forwarding it to the program responsible for processing LiDAR sensor data. It will also receive commands from MATLAB regarding switching between manual and autonomous operation modes, which it then relays to the Arduino. In MANUAL mode, the ESP-32 will also receive movement commands indicating the direction in which the robot should move.

C. Code for MATLAB

MATLAB will receive data from the ESP-32 for processing and visualizing the environmental map in which the autonomous mobile robot operates. The code will process LiDAR data to generate a 2D map of the surroundings. MATLAB will also allow switching between autonomous and manual robot modes. Additionally, it will provide functions for measuring the perimeter and area of the room and setting the robot's movement speed.

V. IMPLEMENTATION OF HARDWARE PART

This subsection focuses in more detail on the implementation of the hardware part of the autonomous mobile robot. It follows Chapter III, which presents the hardware design proposals.

A. Power supply

In the previous work [4] the robot's motors were powered by two batteries, with two additional batteries for the electronics. However, the voltage provided was insufficient for the motors. Therefore, in this project, five 3.7 V L18650 batteries were used (Fig 8). Three batteries power the motors, and two are dedicated to the electronics. This power distribution was chosen to prevent voltage fluctuations caused by the motors from affecting the electronics' performance.



Fig. 8 Power supply configuration with five 3.7 V L18650 batteries

B. Movement

The movement of the robot is driven by four DC gear motors coupled with wheels, chosen for their reliability, ease of integration, and cost efficiency. In this implementation, DC motors equipped with rotary encoders (Fig. 9) were used to enable precise feedback on motor rotation. This allows for accurate control of speed and direction, which is critical for autonomous navigation and obstacle avoidance.

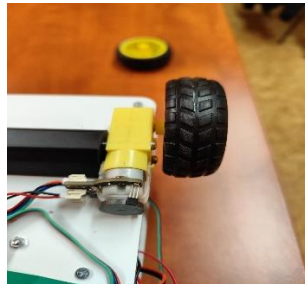


Fig. 9 DC gear motors with integrated encoders

The L298N Dual H-Bridge DC motor driver (Fig. 10) is used to control four DC motors, utilizing two modules. It supports an operating voltage range from 5 V to 35 V and delivers up to 2 A per channel. Its integrated heat sink ensures thermal stability, enabling efficient and reliable motor control for autonomous robotic movement.



Fig. 10 L298N dual H-bridge motor driver module

C. Sensors

A key component of this work is the use of the RPLiDAR sensor from SLAMTEC, mounted on the mobile robot (Fig. 11). Its primary function is to collect environmental data and generate a map of the robot's surroundings. In autonomous mode, the RPLiDAR also helps prevent collisions by maintaining 500 mm from walls and enabling the robot to navigate alongside them.



Fig. 11 RPLiDAR A1 sensor mounted on the chassis

Four infrared sensors with LM393 chips are mounted on the robot—two at the front and two at the rear (Fig. 12). These sensors have a sensing range of 2 cm to 30 cm and assist in detecting low obstacles that the RPLiDAR may not capture. They provide additional data for object detection near the robot, enhancing its ability to avoid obstacles and navigate safely in complex environments.



Fig. 12 Infrared proximity sensors (front and rear)

D. Printed circuit board

Fig. 13 shows the completed printed circuit board, which includes dedicated connections for the H-bridges, Arduino Mega 2560, NODE MCU ESP-32 DEVKIT V1, RPLiDAR sensor, LEDs, capacitors, encoder connections, and infrared sensors.

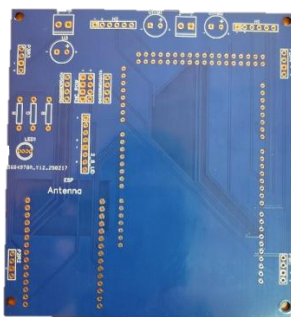


Fig. 13 Fabricated PCB before component assembly

E. Construction

Fig. 14 shows the lower part of the robot's structure, where the printed circuit board and H-bridges are mounted. This section also includes spacer pillars used to support and secure the upper part of the robot.

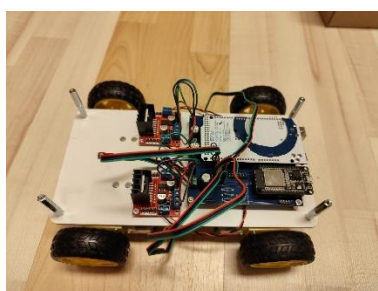


Fig. 14 Lower chassis section with PCB and motor drivers installed

Fig. 15 shows the final construction with an upper part of the robot, where power switches for the electronics and motors are located. The RPLiDAR sensor is centrally positioned to minimize its relative motion during robot rotation. Battery holders are also placed here, along with infrared sensors mounted at the front and rear for obstacle detection.

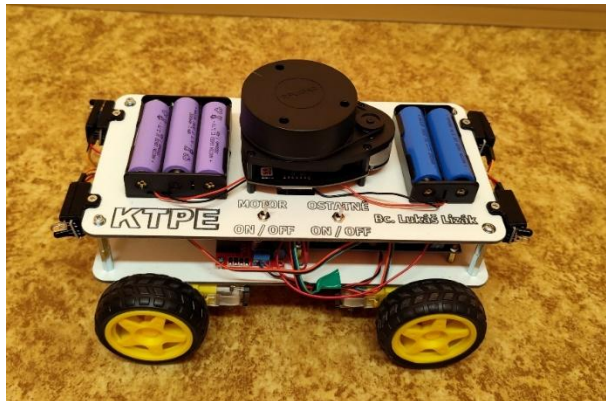


Fig. 15 Fully assembled robot with LiDAR and sensors positioned

VI. IMPLEMENTATION OF SOFTWARE PART

A. Code for Arduino

The Arduino code controls the autonomous mobile robot equipped with a RPLiDAR sensor, infrared sensors, and DC motors with encoders. Upon initialization, UART communication is established and pins for motors and sensors are configured. The robot continuously receives distance data from the RPLiDAR, which it analyses based on signal quality and the angle of detection. The measured distances are stored in an array indexed by angle and transmitted via UART to the ESP-32 module.

The robot operates in two primary modes: manual and autonomous. In autonomous mode, it avoids obstacles detected within a predefined distance and follows walls on its left side while maintaining a specific distance. In manual mode, the robot responds to user commands but still uses infrared sensors to stop when an obstacle is detected. Additionally, the robot features adjustable speed control, which can be set through a MATLAB interface.

B. Code for ESP - 32

To transfer data from the Arduino Mega to MATLAB, the NODE MCU ESP-32 microcontroller is used, leveraging its built-in Wi-Fi to communicate via the TCP/IP protocol. Acting as a bridge, the ESP-32 uses UART to receive data from the Arduino and forward it to MATLAB.

After configuring the network settings (IP address, gateway, subnet, DNS), the ESP-32 connects to Wi-Fi and waits for a client connection. Once connected to MATLAB, it regularly sends a 360-byte array containing LiDAR distance data, where each index represents a specific angle.

Besides data transmission, the ESP-32 receives commands from MATLAB such as switching between manual and autonomous modes, directional instructions in manual mode, and speed values. These are then forwarded to the Arduino for execution.

C. Code for MATLAB

After receiving data from the ESP-32 via the TCP/IP protocol, MATLAB processes the RPLiDAR sensor data to generate a map of the environment. Based on the measured distances and angles, a 2D representation of the surroundings is created.

Two visualizations illustrate the accuracy of the mapping (Fig. 17): one with the robot positioned in the centre of the room (left), and the other near a wall (right). These demonstrate that the LiDAR correctly captures the spatial layout relative to the robot's position.

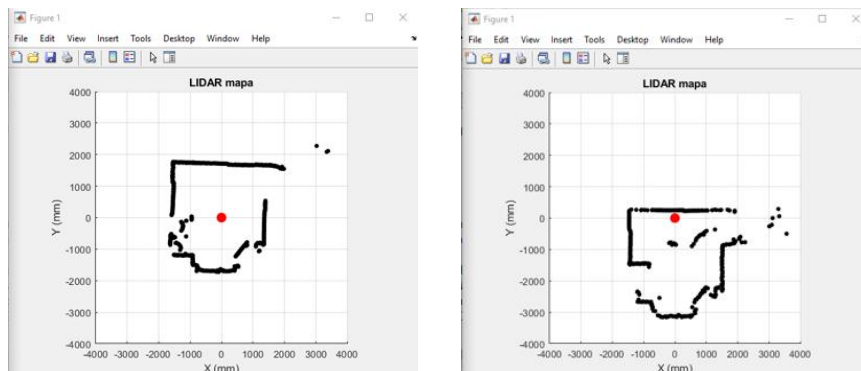


Fig. 16 LiDAR mapping results: robot positioned in the room center (left) and near the wall (right)

A custom control panel has been developed for the robot. It allows manual control by pressing the Manual buttons (Fig.17 left), after which the robot can be directed forward, backward, left, or right.

The panel also includes an Autonomous mode (Fig. 17 right), which is activated by pressing the corresponding button, enabling the robot to navigate autonomously.

Additionally, a speed slider is integrated, allowing users to adjust the robot's speed. The current speed is displayed as a percentage, where 100% corresponds to a numeric value of 255.

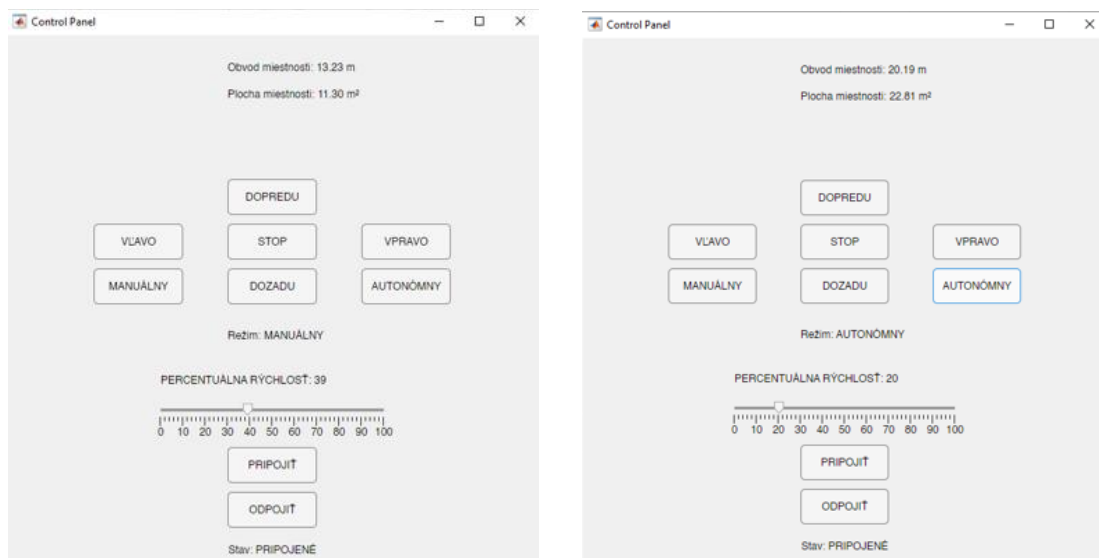


Fig. 17 MATLAB control panel in manual mode (left) and autonomous mode (right)

Based on the map generated, it is also possible to calculate the perimeter and area of the room (Fig. 18) in which the robot operates. These measurements are derived directly from the 2D environment map created using data from the RPLiDAR sensor.

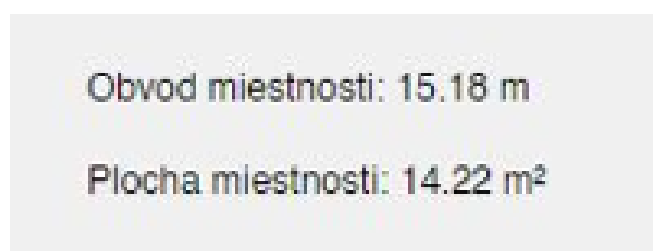


Fig. 18 Display of calculated room area and perimeter from LiDAR-generated map

VII. CONCLUSION

The developed autonomous mobile robot successfully demonstrates indoor navigation, environment mapping, and obstacle avoidance using a combination of LiDAR, infrared sensors, and rotary encoders. The integration of Arduino MEGA 2560 and ESP-32 microcontrollers enables reliable sensor data processing, motor control, and seamless communication with MATLAB via TCP/IP. The custom-designed chassis and printed circuit board ensure mechanical stability and wiring reliability, while the MATLAB interface allows intuitive control and real-time visualization. The system's modular hardware and software architecture make it suitable for further enhancements, including advanced SLAM algorithms, improved localization accuracy, and integration with cloud-based data processing for extended autonomy.

REFERENCES

- [1] WALKER Jason. What Are Autonomous Robots? 8 Applications For Today's AMRs, [online], Available on internet: <https://locusrobotics.com/blog/what-are-autonomous-robots#:~:text=True%20autonomous%20robots%20are%20intelligent,without%20human%20control%20or%20intervention.>
- [2] INDEVA, Automation and innovation for an efficient production line, [online] Available on internet: <https://www.indevagroup.com/news/industrial-automation-automatic-guided-vehicles-production-line/>
- [3] VIROG Lee, What is an Automated Guided Vehicle (AGV)?, [online] - Available on internet: [https://www.autostoresystem.com/insights/what-is-an-automated-guided-vehicle-agv#:~:text=An%20Automated%20Guided%20Vehicle%20\(AGV\)%20system%2C%20also%20referred%20to,a%20human%20operator%20or%20driver.](https://www.autostoresystem.com/insights/what-is-an-automated-guided-vehicle-agv#:~:text=An%20Automated%20Guided%20Vehicle%20(AGV)%20system%2C%20also%20referred%20to,a%20human%20operator%20or%20driver.)
- [4] Lizák, L., Vince, T.: Mobile robot with use of RPLiDAR sensor, In: Proceeding of scientific and student's works in the field of Industrial Electrical Engineering. Volume 12., 2023, ISBN 978-80-553-4406-5
- [5] WASEEM Umar, Understanding AMR Robots: A Comprehensive Guide, [online] Available on internet: <https://www.wevolver.com/article/understanding-amr-robots-a-comprehensive-guide>
- [6] RIISGAARD Søren, RUFUS BLAS Morten, SLAM for Dummies, [online] Available on internet: https://dspace.mit.edu/bitstream/handle/1721.1/119149/16-412j-spring-2005/contents/projects/1aslambblas_repo.pdf
- [7] TEKNOLOGS Jurnal, Particle filter in simultaneous localization and mapping (Slam) using differential drive mobile robot, [online] Available on internet: [https://www.researchgate.net/publication/286763834_Particle_filter_in_simultaneous_localization_and_mapping_Slam_using_differential_drive_mobile_robot#:~:text=Simultaneous%20Localization%20and%20Mapping%20\(SLAM\)%20problem%20is%20a%20well%20D,Extended%20Kalman%20Filter%20\(EKF\).](https://www.researchgate.net/publication/286763834_Particle_filter_in_simultaneous_localization_and_mapping_Slam_using_differential_drive_mobile_robot#:~:text=Simultaneous%20Localization%20and%20Mapping%20(SLAM)%20problem%20is%20a%20well%20D,Extended%20Kalman%20Filter%20(EKF).)
- [8] A. WAN Eric, VAN DER MERWE Rudolph, The Unscented Kalman Filter for Nonlinear Estimation [online] Available on internet: <https://groups.seas.harvard.edu/courses/cs281/papers/unscented.pdf>
- [9] MONTEMERLO Michael, THRUN Sebastian FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem [online] Available on internet: <https://cdn.aaai.org/AAAI/2002/AAAI02-089.pdf>
- [10] O. HAMZAoui B. Steux, CoreSLAM : a SLAM Algorithm in less than 200 lines of C code [online] Available on internet: <https://www.semanticscholar.org/paper/CoreSLAM-%3A-a-SLAM-Algorithm-in-less-than-200-lines-Steux-Hamzaoui/1c7e7af4388b17b137badc6ec19e9724e6bd91e4>
- [11] SAAT Shahrizal HECTORSLAM 2D MAPPING FOR SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM) [online] Available on internet: [chrome-extension://efaidnbmninnbpcjpcglclefindmkaj/https://iopsience.iop.org/article/10.1088/1742-6596/1529/4/042032/pdf](https://iopsience.iop.org/article/10.1088/1742-6596/1529/4/042032/pdf)
- [12] KIM Stanley FastSLAM: Algorithm for Simultaneous Localization and Mapping [online] Available on internet: <https://medium.com/ml-purdue/fastslam-algorithm-for-simultaneous-localization-and-mapping-6fb5bd141f30>
- [13] Mikk, Simultaneous localization and mapping libraries for C#, [online] Available on internet: <https://github.com/mikkleini/slam.net>