

Autonomous Mobile Robot with Remote Control

Muhammad Rashid Zia

Bachelor's Thesis
2025



Tomas Bata University in Zlín
Faculty of Applied Informatics

Tomas Bata University in Zlín
Faculty of Applied Informatics
Department of Automation and Control Engineering
Academic year: 2024/2025

ASSIGNMENT OF BACHELOR THESIS

(project, art work, art performance)

Name and surname:	Muhammad Rashid Zia
Personal number:	A22709
Study programme:	B0714A150007 Applied Informatics in Industrial Automation
Specialization:	Intelligent Systems with Robots
Type of Study:	Full-time
Work topic:	Autonomous Mobile Robot with Remote Control
Work topic in English:	Autonomous Mobile Robot with Remote Control

Theses guidelines

1. Describe types of autonomous mobile robotic systems. Describe basic subsystems of mobile robots.
2. Create an overview of available platforms suitable for university courses focused on autonomous mobile robots.
3. Select a suitable platform for an autonomous mobile robot, describe it in detail and assemble the mobile robot.
4. Design and verify control algorithms for the robot to execute basic tasks such as following a path and avoiding obstacles.
5. Extend the robot control system by remote control using smartphone.
6. Design and verify the smartphone application for controlling the robot.

Form processing of bachelor thesis: **printed/electronic**
Language of elaboration: **English**

Recommended resources:

1. DAHIYA, Ravinder S.; OZIOKO, Oliver a CHENG, Gordon. *Sensory systems for robotic applications. IET control, robotics and sensors series*. London: The Institution of Engineering and Technology, 2022. Available from: <https://doi.org/9781849199490>.
2. MARGOLIS, Michael. *Arduino cookbook*. Second edition. Sebastopol: O'Reilly, 2011. ISBN 9781449313876.
3. KALA, Rahul. *Autonomous mobile robots: planning, navigation and simulation*. London: Academic Press, an imprint of Elsevier, [2024]. ISBN 978-0-443-18908-1.
4. PHILLIPS, Bill; STEWART, Chris; HARDY, Brian a MARSICANO, Kristin. *Android programming: the Big Nerd Ranch guide*. 2nd edition. Atlanta: Big Nerd Ranch, [2015]. ISBN 9780134171456.
5. SETHI, Ishwar K. *Autonomous vehicles and systems: a technological and societal perspective*. River Publishers series in automation, control and robotics. [United States]: River Publishers, 2023. Available from: <https://doi.org/9788770228503>.
6. SIEGWART, Roland; NOURBAKHSH, Illah Reza a SCARAMUZZA, Davide. *Introduction to autonomous mobile robots*. 2nd ed. Intelligent robotics and autonomous agents. Cambridge, Mass.: MIT Press, c2011. ISBN 9780262295321. Available also from: <https://proxy.k.utb.cz/login?url=http://ieeexplore.ieee.org/xpl/bkabstractplus.jsp?bkn=6267528>.

Supervisors of bachelor thesis: **Ing. Petr Chalupa, Ph.D.**
Department of Automation and Control Engineering

Date of assignment of bachelor thesis: **December 13, 2024**
Submission deadline of bachelor thesis: **May 27, 2025**

doc. Ing. Jiří Vojtěšek, Ph.D. m.p.
Dean



prof. Ing. Vladimír Vašek, CSc. m.p.
Head of Department

In Zlín December 13, 2024

BACHELOR'S THESIS AUTHOR'S DECLARATION

I take cognizance of the fact that

- by submitting my Bachelor's thesis, I give consent to the publication of my thesis in accordance with the Act No. 111/1998 Coll., as amended, regardless of the result of the thesis defence;
- the Bachelor's thesis will be stored in electronic form in the university information system and will be available for consultation;
- one copy of the Bachelor's thesis in hard copy form will be kept and stored by Tomas Bata University in Zlín;
- my Bachelor's thesis is fully subject to the Act No. 121/2000 Coll. on Copyright, on Rights Related to Copyright and on Amendment to Certain Acts (Copyright Act), as amended, particularly to Section 35 Paragraph 3;
- according to Section 60 Paragraph 1 of the Copyright Act, Tomas Bata University in Zlín has the right to conclude a licence agreement on the utilization of a school work as is specified in Section 12 Paragraph 4 of the Copyright Act;
- according to Section 60 Paragraphs 2 and 3, I may utilize my school work – Bachelor's thesis – or grant a licence to utilize it only with the prior written consent of Tomas Bata University in Zlín, which in such a case is entitled to require me to pay a reasonable contribution in order to cover the costs incurred by Tomas Bata University in Zlín for the preparation of the work (up to their full amount);
- if software provided by Tomas Bata University in Zlín or other entities was used for the preparation of the Bachelor's thesis only for learning or research purposes (i.e. for non-commercial use), the conclusions of the Bachelor's thesis cannot be used for commercial purposes;
- if the output of the Bachelor's thesis is any type of software product, the source codes and/or files of which the project is comprised are also considered part of the thesis; failure to submit this part may result in a failure to defend the thesis.

I declare that

- I have worked on my Bachelor's thesis independently and properly cited the literature used; in case of publication of the conclusions, I shall be listed as co-author;
- the submitted hard copy version of the Bachelor's thesis and the electronic version uploaded to the IS/STAG are identical in content.

In Zlín on

.....

author's signature

ABSTRAKT

Tato práce představuje návrh, vývoj a testování autonomního mobilního robota s funkcí dálkového ovládání, který využívá platformu Makeblock mBot Ranger. Hlavním cílem projektu bylo prozkoumat, jak lze výukovou robotickou stavebnici přeměnit na multifunkční systém schopný vykonávat klíčové autonomní chování a zároveň nabízet ovládání uživatelem v reálném čase prostřednictvím mobilní aplikace. Robot byl naprogramován k provádění tří základních úkolů: sledování čáry, vyhýbání se překážkám a sledování rukou. Chování bylo dosaženo pomocí vestavěných infračervených a ultrazvukových senzorů v kombinaci s řídicí logikou implementovanou ve vývojovém prostředí Arduino. Pro zvýšení interaktivity a umožnění plynulého přepínání režimů byla vyvinuta vlastní aplikace pro systém Android v jazyce Java pomocí aplikace Android Studio. Tato aplikace se k robotu připojuje prostřednictvím Bluetooth Low Energy (BLE) a umožňuje uživatelům vzdáleně přepínat mezi režimy pomocí jednoduchého, uživatelsky přívětivého rozhraní. Aplikace byla testována na několika zařízeních se systémem Android a ukázala se jako spolehlivá, citlivá a intuitivní. Systém byl hodnocen v univerzitní laboratoři i v domácím prostředí, aby bylo možné posoudit jeho výkonnost v různých podmínkách. Výsledky ukázaly vysokou úspěšnost všech funkcí, stabilní snímání údajů ze senzorů, přesné ovládání pohybu a komunikaci přes Bluetooth v reálném čase. Byla zaznamenána některá omezení, zejména v souvislosti s citlivostí snímačů na velmi krátkou vzdálenost a drobnými problémy s trakcí na některých površích. Celkově projekt ukazuje, že levné modulární platformy, jako je mBot Ranger, mohou sloužit jako efektivní nástroje pro stavbu autonomních robotů vhodných pro vzdělávání, prototypování a aplikované učení v oblasti vestavěných systémů a inteligentní automatizace.

Klíčová slova: mBot Ranger, autonomní mobilní robot, ovládání aplikací pro Android, Bluetooth Low Energy (BLE), navigace založená na senzorech.

ABSTRACT

This thesis presents the design, development, and testing of an autonomous mobile robot with remote control functionality, utilizing the Makeblock mBot Ranger platform. The primary goal of the project was to explore how an educational robotics kit can be transformed into a multi-functional system capable of performing key autonomous behaviors while also offering real-time user control via a mobile application. The robot was programmed to carry out three core tasks: line following, obstacle avoidance, and hand-following. These behaviors were achieved using onboard infrared and ultrasonic sensors, combined with control logic implemented in the Arduino development environment. To enhance interactivity and enable seamless mode switching, a custom Android application was developed in Java using Android Studio. This application connects to the robot via Bluetooth Low Energy (BLE), allowing users to remotely switch between modes using a simple, user-friendly interface. The app was tested on multiple Android devices and proved to be reliable, responsive, and intuitive. The system was evaluated in both university lab and home environments to assess its performance under varied conditions. Results showed a high success rate across all functionalities, with stable sensor readings, accurate motion control, and real-time Bluetooth communication. Some limitations were noted, particularly related to sensor sensitivity at very close distances and minor traction issues on certain surfaces. Overall, the project demonstrates that low-cost, modular platforms like the mBot Ranger can serve as effective tools for building autonomous robots suitable for education, prototyping, and applied learning in embedded systems and intelligent automation.

Keywords: Android App Control, Autonomous Mobile Robot, Bluetooth Low Energy (BLE), mBot Ranger, Sensor-Based Navigation.

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to my thesis supervisor, Prof. Ing. Petr Chalupa, Ph.D., for their invaluable guidance, constructive feedback, and consistent support throughout the duration of this project. Their expertise and encouragement were instrumental in shaping the direction and quality of this work. Special thanks go to my colleagues and classmates who shared their insights, motivation, and assistance during the development and testing phases of the project. Their collaboration enriched both the technical and personal aspects of my academic journey. Lastly, I am deeply grateful to my family and friends for their continuous encouragement, patience, and understanding during this demanding phase. Their unwavering support made this achievement possible. To all those who contributed in one way or another to the completion of this thesis, thank you.

CONTENTS

INTRODUCTION	13
I. THEORETICAL PART	14
1 TYPES OF AUTONOMOUS MOBILE ROBOTIC SYSTEMS	15
1.1 THEORETICAL CONTEXT AND RESEARCH DOMAIN	15
1.2 RESEARCH PROBLEM STATEMENT	16
1.3 RESEARCH SIGNIFICANCE AND IMPACT.....	17
1.4 METHODOLOGY OVERVIEW	18
1.5 KEY FINDINGS AND CONTRIBUTIONS	18
1.6 STRUCTURE OF THE THESIS.....	19
2 LITERATURE REVIEW	21
2.1 ARCHITECTURE FOR AUTONOMOUS MOBILE ROBOTS	21
2.1.1 Layered and Subsumption Architecture.....	21
2.1.2 Hybrid Architectures	21
2.1.3 Sensor Integration and Data Processing.....	22
2.1.4 Applications of Architectural Models	22
2.1.5 Limitations and Future Directions	23
2.2 EDUCATIONAL ROBOTICS AND PLATFORMS.....	23
2.2.1 Evolution of Educational Robotics	23
2.2.2 Popular Platforms in Educational Robotics	24
2.2.3 Impact on Student Motivation and Learning	31
2.2.4 Methodologies in Educational Robotics	31
2.2.5 Limitations and Challenges.....	32
2.2.6 Future Directions.....	32
2.3 CONTROL ALGORITHMS FOR NAVIGATION AND OBSTACLE AVOIDANCE.....	33
2.3.1 Reactive Control Approaches	33
2.3.2 Deliberative Control Approaches.....	34
2.3.3 Sensor-Based Navigation Techniques.....	34
2.3.4 Hybrid Approaches: Combining Reactive and Deliberative Control	35
2.3.5 Applications of Control Algorithms in Autonomous Robotics	35
2.3.6 Limitations and Future Directions	36
3 RESEARCH ON MOBILE ROBOTICS PLATFORMS	37
3.1 EVOLUTION OF AUTONOMOUS MOBILE ROBOT	37

3.2	TELEOPERATION AND REMOTE CONTROL IN MOBILE ROBOT	38
3.3	RESEARCH ON ROBOTIC PLATFORMS FOR AUTONOMOUS AND REMOTE- CONTROL APPLICATIONS	40
3.3.1	mBot Ranger: Research in STEM and AI Education.....	40
3.3.2	LEGO Mindstorms: A Research Tool for AI and Robotics.....	41
3.3.3	VEX Robotics: Engineering and Competitive Robotics Research	42
3.3.4	Raspberry Pi and Arduino-Based Robots: AI, IoT, and Research Applications	44
3.3.5	Thymio: Research in Early Robotics Education and Human- Robot Interaction.....	45
3.3.6	Comparative Analysis of Research on Robotic Platforms.....	46
3.4	SUMMARY OF RESEARCH FINDINGS AND FUTURE DIRECTIONS.....	46
II	PRACTICAL PART.....	47
4	SYSTEM ARCHITECTURE AND DEVELOPMENT	49
4.1	SYSTEM OVERVIEW	49
4.2	HARDWARE ARCHITETURE	50
4.2.1	Central Control Unit: Me Auriga Mainboard	50
4.2.2	Motors and Chassis	51
4.2.3	Sensors	52
4.2.4	Power Supply	54
4.3	SOFTWARE DEVELOPMENT ENVIRONMENT.....	55
4.3.1	Tools and Libraries Used	56
4.3.2	Development Workflow.....	56
5	CONTROL ALGORITHM DESIGN.....	58
5.1	AUTONOMOUS FUNCTIONALITIES	58
5.1.1	Line Following.....	58
5.1.2	Obstacle Avoidance	59
5.1.3	Hand Following.....	59
5.2	REMOTE CONTROL INTEGRATION	62
5.2.1	Custom Android Application Development.....	62
5.2.2	Command Protocol and Mode Switching	65
5.2.3	Implementation	65
6	TESTING AND RESULTS.....	67
6.1	TESTING ENVIRONMENT	67
6.2	LINE FOLLOWING RESULTS.....	67
6.3	OBSTACLE AVOIDANCE RESULTS	69

6.4 HAND FOLLOWING RESULTS.....	71
6.5 BLUETOOTH CONTROL EVALUATION.....	72
6.6 BLUETOOTH APP TESTING RESULTS	74
6.7 SUMMARY OF RESULTS.....	75
CONCLUSION	77
BIBLIOGRAPHY.....	79
LIST OF FIGURES	84
LIST OF TABLES	86
LIST OF ABBREVIATIONS	87
APPENDICES.....	88
APPENDIX P I: APPENDIX TITLE.....	89

INTRODUCTION

Autonomous mobile robots (AMRs) are transforming industries by enabling intelligent, self-navigating systems capable of operating with minimal human intervention. However, real-world uncertainties often necessitate a hybrid control approach integrating both autonomous navigation and remote supervision—to enhance adaptability and ensure operational safety.

This research focuses on designing and implementing an AMR system capable of executing key tasks such as line following, obstacle avoidance, and hand-following, while incorporating smartphone-based remote control for seamless user interaction. By leveraging sensor-based feedback and modular programming logic, this study develops a scalable and intelligent robotic platform suitable for real-time interaction in educational, experimental, and entry-level automation environments.

The outcomes of this work contribute to the advancement of hybrid-controlled mobile robots by making them more robust, efficient, and accessible for diverse operational conditions. The system design, implementation, and testing provide a valuable foundation for future extensions, including more advanced control algorithms, interactive learning tools, and remote user interfaces.

I. THEORETICAL PART

1 TYPES OF AUTONOMOUS MOBILE ROBOTIC SYSTEMS

1.1 Theoretical Context and Research Domain

The field of autonomous mobile robotics has grown substantially, driven by advancements in artificial intelligence, sensor technologies, and control systems. Autonomous mobile robots (AMRs) are increasingly utilized in diverse applications ranging from industrial logistics and healthcare to military operations and space exploration [7]. Unlike traditional industrial robots, which are typically stationary and operate within pre-defined environments, AMRs possess both mobility and autonomy, enabling them to perform tasks in dynamic and unstructured settings without continuous human supervision.

The evolution of mobile robotics owes much to progress in multiple domains. Improvements in computational power have enabled real-time processing of sensory inputs, allowing robots to quickly react to their surroundings. In addition, the development of advanced sensor systems, such as LiDAR, ultrasonic sensors, and computer vision, has expanded robots' perceptual capabilities, enabling them to detect, classify, and avoid obstacles in increasingly complex environments [2]. Advances in localization techniques, such as Simultaneous Localization and Mapping (SLAM), have further enhanced the ability of robots to maintain spatial awareness even in previously unknown environments [7].

These technological advancements have prompted the deployment of AMRs in both structured and semi-structured environments. For example, automated guided vehicles (AGVs) are used in factories and warehouses to transport goods between workstations, relying on electrical guide wires or laser-based navigation systems to follow predefined paths[5]. Meanwhile, mobile service robots in healthcare settings, such as the Helpmate robot, autonomously transport medication and supplies, navigating hospital corridors by detecting overhead lighting patterns[7]. Beyond structured applications, mobile robots have also been employed in hazardous environments, including the Mars Pathfinder mission, where teleoperated robots such as Sojourner explored Martian terrain while autonomously detecting and avoiding obstacles.

Despite these successes, critical challenges remain. In dynamic environments, where unforeseen obstacles may appear suddenly, reactive control systems often fall short in ensuring continuous and safe navigation. Obstacle avoidance algorithms like the Vector Field Histogram (VFH) aim to address this by dynamically computing safe paths based on

sensor data, thereby enabling robots to navigate cluttered spaces without the need for complex global path planning[2]. These solutions, however, still face limitations when environmental complexity exceeds the robot's sensory and computational capacities. As a result, many mobile robotics researchers advocate for hybrid control systems that balance real-time autonomy with human oversight [3].

Remote control and monitoring capabilities have become increasingly important in enhancing the adaptability of mobile robots, especially in environments where complete autonomy is infeasible. The integration of smartphones and other mobile devices offers significant potential in this regard. Smartphones, with their high computational power, embedded sensors, and wireless communication capabilities, provide an accessible platform for real-time robot supervision and control[4]. Through mobile applications, operators can receive live video streams, issue navigation commands, and monitor the robot's status, enabling effective intervention when needed.

This convergence of autonomous robotics and mobile technology aligns with broader trends in the Internet of Things (IoT) and Industry 4.0. IoT-enabled systems emphasize connectivity between physical devices and digital infrastructure, creating opportunities for robots to become part of larger, interconnected ecosystems[6]. In smart manufacturing, for instance, mobile robots can dynamically adjust their routes and tasks based on real-time data from other machines and sensors. Similarly, in logistics, robots equipped with mobile connectivity can be monitored and dispatched from centralized control centers, enhancing operational flexibility and efficiency.

As the field progresses, the demand for robust, hybrid control systems that integrate autonomy with remote supervision is growing. This thesis contributes to this evolving area by proposing a system that enhances autonomous capabilities while leveraging mobile technology for remote control and monitoring. By addressing both theoretical and practical challenges, this research aims to advance mobile robotics applications in industries that require both adaptability and user intervention.

1.2 Research Problem Statement

While autonomous navigation algorithms have demonstrated success in structured environments, they often struggle in dynamically changing conditions. Factors such as inconsistent sensor data, unforeseen obstacles, and environmental noise can lead to

navigation errors, potentially compromising task completion. At the same time, traditional remote control systems demand continuous operator involvement, limiting scalability and operational efficiency [6].

This thesis seeks to bridge the gap between autonomous and manual control by designing a hybrid system that enables robots to operate autonomously while allowing remote interventions through a smartphone interface. The central research question is: "How can a scalable and robust control system be designed for autonomous mobile robots that integrates both autonomous navigation and smartphone-based remote control to improve task efficiency and user control?" This research aims to advance state of the art solutions in control architecture, focusing on seamless transitions between autonomy and human oversight, which is crucial for applications in industries such as logistics, surveillance, and disaster response.

1.3 Research Significance and Impact

This research holds both theoretical and practical significance. From a theoretical perspective, it contributes to the body of knowledge on hybrid control systems, offering a framework that integrates subsumption-based autonomy with remote monitoring technologies. Previous studies, including those by Brooks (1986) and Borenstein and Koren (1991), emphasize the need for real-time responsiveness in dynamic environments. This thesis extends these approaches by incorporating mobile communication technologies, allowing users to intervene and adjust robot behavior in real-time when necessary.

Practically, this research addresses key limitations in current robotic systems. Robots deployed in hazardous environments—such as bomb disposal, search-and-rescue operations, or industrial maintenance—must adapt quickly to new conditions. Remote control systems, enhanced by mobile applications, provide operators with situational awareness while minimizing risks to human safety[4]. Moreover, in educational settings, the combination of autonomous and remote control functionalities can enhance interactive learning experiences, fostering a deeper understanding of robotics principles [1].

The broader impact of this research aligns with global technological trends in automation, IoT, and Industry 4.0. By improving the accessibility and versatility of mobile robots, this research supports efforts to automate repetitive and hazardous tasks, contributing to increased productivity, safety, and innovation in various sectors.

1.4 Methodology Overview

This thesis uses a multi-phase process that combines theoretical analysis, system design, and experimental validation to answer the research issue.

- **Literature Review:** A thorough analysis of both classic and recent studies on smartphone integration, hybrid control systems, and mobile robotics is carried out. This covers research on ROS (Quigley et al., 2009), educational robotics (Alimisis, 2013), and subsumption architecture (Brooks, 1986).
- **System Design:** A scalable platform with numerous subsystems is used to build a mobile robot prototype. The design incorporates actuators for motion control, a communication module for wireless connectivity, and sensor arrays for obstacle detection (such as LiDAR or ultrasonic sensors).
- **Control Algorithms:** Algorithms for navigation are created to manage both distant interventions and autonomous operations. Real-time obstacle avoidance is made possible by the implementation of the Vector Field Histogram (VFH), and seamless transitions between manual and autonomous control modes are made possible by a priority-based command mechanism [2].
- **Smartphone Application Development:** Users can access real-time video feeds, control inputs, and system feedback using a specially created smartphone application. Quick reaction times are ensured by this application's low-latency communication with the robot over Bluetooth or Wi-Fi.
- **Experimental Testing:** In both simulated and real-world settings, the system is assessed. Path-following accuracy, obstacle-clearing effectiveness, response latency, and user satisfaction are examples of performance measurements. The control framework's scalability and dependability are evaluated by statistical analysis.

1.5 Key Findings and Contributions

The following contributions to the field of mobile robotics are anticipated from this thesis:

1. **Framework Development:** An innovative hybrid control system that combines remote control via a smartphone with autonomous navigation.

2. Enhanced Obstacle Avoidance: Robot movement is not disrupted when the Vector Field Histogram technique is used to enhance obstacle avoidance in congested situations [2].
3. Mobile App Integration: creation of a real-time mobile interface that improves operational flexibility, situation awareness, and user control [4].
4. Improved Performance Metrics: Evidence from experiments showing decreased navigation errors and increased work efficiency in both supervised and autonomous settings.

The goal of these contributions is to promote the use of mobile robots in practical settings where safety, adaptability, and human involvement are essential.

1.6 Structure of the Thesis

The thesis is organized as follows:

- Chapter 2: Literature Review – Explores theoretical frameworks, control strategies, and related technologies in mobile robotics.
- Chapter 3: Research on Mobile Robotics Platforms - Discusses the evolution of mobile robots, teleoperation concepts, and examines prior research on various educational and experimental platforms such as mBot Ranger, LEGO Mindstorms, VEX, Arduino, and Thymio. A comparative analysis highlights the strengths and limitations of each system.
- Chapter 4: System Architecture and Development - Describes the hardware architecture of the mBot Ranger, including sensors, motors, control board, and power supply. It also outlines the software development tools and environment used in this project.
- Chapter 5: Control Algorithm Design - Presents the design and implementation of control algorithms for autonomous tasks including line following, obstacle avoidance, and hand following. The chapter also details the development of the custom Android application and command-handling logic for remote control.
- Chapter 6: Testing and Results - Provides a detailed evaluation of the system's performance under various test conditions. Results are presented for each

autonomous functionality and remote control operation, followed by a summary of challenges and key findings.

2 LITERATURE REVIEW

2.1 Architecture for Autonomous Mobile Robots

The architectural design of autonomous mobile robots plays a fundamental role in determining their performance, adaptability, and reliability in real-world environments. Over the years, various control architectures have been developed to improve robot autonomy, decision-making, and response to dynamic environmental conditions.

2.1.1 Layered and Subsumption Architecture

One of the most influential architectural frameworks is the subsumption architecture, introduced by Brooks (1986). This approach emphasizes decentralized control, where robot behaviors are organized hierarchically in layers. Each layer represents a different task or competence, with lower-level layers handling fundamental, immediate tasks (e.g., obstacle avoidance) and higher layers managing more complex activities (e.g., navigation to a target). The layers operate in parallel and can inhibit each other based on task priorities. This allows the robot to react in real time to environmental changes without needing extensive pre-planning[3].

This architecture has been particularly effective in enabling robots to operate autonomously in unpredictable environments. For example, mobile robots in disaster response scenarios use layered control to prioritize immediate safety concerns, such as avoiding falling debris, while still attempting to reach trapped victims.

2.1.2 Hybrid Architectures

While the subsumption architecture is effective for reactive control, it has limitations in complex environments where longer-term planning and adaptation are required. To address these challenges, researchers have proposed hybrid architectures that combine reactive and deliberative control elements. These systems integrate short-term sensor-driven behaviors with long-term goal-directed planning. A common implementation involves using a high-level planner to define paths and a low-level reactive module to handle dynamic obstacles along the way[7].

Hybrid architectures are used in scenarios where robots must balance immediate environmental interactions with higher-order tasks. For example, warehouse robots may rely on a hybrid model to navigate around moving workers and forklifts while maintaining optimal routes to pick and deliver goods.

2.1.3 Sensor Integration and Data Processing

Autonomous robots depend extremely on sensory data to perceive and understand their surroundings. Sensor fusion technique plays an important role in improving environmental awareness by combining inputs from multiple sensors such as LIDAR, ultrasonic sensors, and cameras. The Vector Field Histogram (VFH) method, developed by Borenstein and Koren (1991), is a significant innovation in sensor-based navigation. This method constructs a real-time polar histogram from sensor data, enabling the robot to quickly assess obstacle density and determine safe navigation paths without halting for recalculations [2].

Robots using VFH have demonstrated enhanced navigation capabilities in cluttered environments such as hospitals and factories. By constantly updating obstacle density maps, these robots can maintain smooth, continuous motion while adapting to changing surroundings.

2.1.4 Applications of Architectural Models

The architectures discussed are applied across various industries. In agriculture, autonomous tractors equipped with hybrid control systems efficiently manage large fields, switching between obstacle avoidance near trees and strategic route optimization across open land. In education, programmable platforms such as mBot Ranger, designed by Makeblock, offer students practical insights into layered and sensor-based control. These platforms allow learners to develop simple reactive behaviors (e.g., line following) and expand them with goal-directed navigation tasks[14].

Moreover, constructionist learning environments have demonstrated the effectiveness of tangible robotic activities in teaching architectural principles. [13] showcased how young children interacting with robots could internalize concepts related to movement and collaboration, thereby reinforcing their understanding of control architectures through embodied learning.

2.1.5 Limitations and Future Directions

Although existing architectural models have significantly advanced autonomous capabilities, they still face limitations in highly unpredictable environments. Robots may struggle with sensor noise, inadequate data fusion, and decision conflicts between layers in hybrid systems. Emerging research focuses on integrating artificial intelligence (AI) models, such as reinforcement learning to improve adaptability and decision-making.

Future advancements are likely to include more sophisticated architectures that enable robots to better predict environmental changes and proactively adapt their behaviors. Innovations in AI-driven sensor fusion and multi-agent coordination also hold promise for expanding the autonomy and efficiency of mobile robots in complex operational contexts.

2.2 Educational Robotics and Platforms

Educational robotics has emerged as an vital component in STEM (Science, Technology, Engineering, and Mathematics) education, helping learners to develop critical problem solving, teamwork and computational thinking skills. Robotic platforms and programmable devices have gained widespread adoption in classrooms, allowing students to engage in hands-on learning experiences through building and programming tasks. This section reviews research related to the development and effectiveness of educational robotics platforms, including popular tools and learning methodologies.

2.2.1 Evolution of Educational Robotics

Educational robotics has evolved considerably since the introduction of early platforms such as the LEGO Mindstorms series, which enabled students to build and program physical robots using block-based coding environments. Research has shown that platforms like LEGO Mindstorms, mBot, Bee-Bot, and Phiro Pro effectively motivate students by making abstract programming concepts more tangible and interactive [13].

These robotic platforms often provide a multi-disciplinary learning experience. For example, LEGO robots can be used in physics and engineering projects to teach mechanical design principles, while platforms such as the micro:bit integrate sensors for environmental data collection, making them suitable for computer science and environmental science courses[14].

2.2.2 Popular Platforms in Educational Robotics

Educational robotics platforms provide hands-on learning experiences, enabling students and researchers to explore programming, artificial intelligence, and automation. Several popular platforms are widely used in academia, ranging from beginner-friendly kits to advanced research-oriented robotic systems.

1. mBot Ranger (Makeblock)

The mBot Ranger, developed by Makeblock, is a 3-in-1 educational robot designed for students to learn robotics, coding, and electronics. It features:

- Three configurations: Robot tank, self-balancing robot, and racing car.
- Graphical & Text-based Programming: Supports Scratch-based block coding and Arduino C.
- Integrated sensors: Includes ultrasonic sensors, line-following modules, and gyroscopes for obstacle avoidance and navigation.
- Remote control & autonomy: Can be controlled via Bluetooth, Wi-Fi, or pre-programmed autonomous behaviors.

The mBot Ranger is a versatile, programmable robot designed for STEM education, offering multiple configurations for different learning experiences.



Figure 1 The mBot Ranger 3-in-1

(source: [29])

2. LEGO Mindstorms EV3 / Robot Inventor

LEGO Mindstorms has been one of the most popular platforms for educational robotics. The EV3 kit and the latest LEGO SPIKE Prime / Robot Inventor offer:

- Modular design: Allows for building various robotic models.
- Graphical programming: Use block based coding (Scratch, LEGO MINDSTORMS App) and supports Python.
- Multiple sensors & motors: Includes touch sensors, color sensors, gyroscopes, and servo motors for advanced AI integration.
- Wireless control and AI: Compatible with computer vision libraries and machine learning algorithms.

The LEGO Mindstorms robot inventor kit allows users to build multiple robotic designs with interactive motors, sensors, and a coding interface.



Figure 2 LEGO Mindstorms Robot Inventor Kit with multiple configurations

(source:[30])

3. VEX ROBOTICS (VEX IQ & VEX V5)

VEX robotics is widely used in STEM competitions and engineering education provides:

- Two versions: VEX IQ (for beginners) and VEX V5 (for advanced robotics).
- Comprehensive kits: Includes programmable microcontrollers, motors, sensors, and structural components.
- Coding platforms: Uses VEXcode (block & text-based programming), supports C++ and Python.
- Competition-ready: Used in global VEX robotic competition (VRC).

VEX IQ is a beginner-friendly robotic platform which introduces students to coding and automation, making it a popular choice for STEM competitions.



Figure 3 VEX IQ in an interactive classroom environment (source:[31])

VEX V5 is an advanced robotics system designed for engineering students and competitive robotics, supporting real-world automation applications.



Figure 4 VEX V5 kit displaying advanced mechanical structures and a controller

(source: [31])

4. Raspberry Pi Pico & Arduino-Based Robots

Many DIY educational robotics kits are powered by Raspberry Pi or Arduino, offering:

- Customization & Flexibility: Users can build their own robots from scratch.
- Programming Languages: Supports Python, C++, and Java.
- AI & IoT Integration: Raspberry Pi allows AI-based applications (e.g., OpenCV for object tracking).
- Affordable & Scalable: Open-source hardware provides endless customization.

Arduino and Raspberry Pi-based robots provide an open-source, customizable platform for robotics projects, enabling AI and IoT applications.

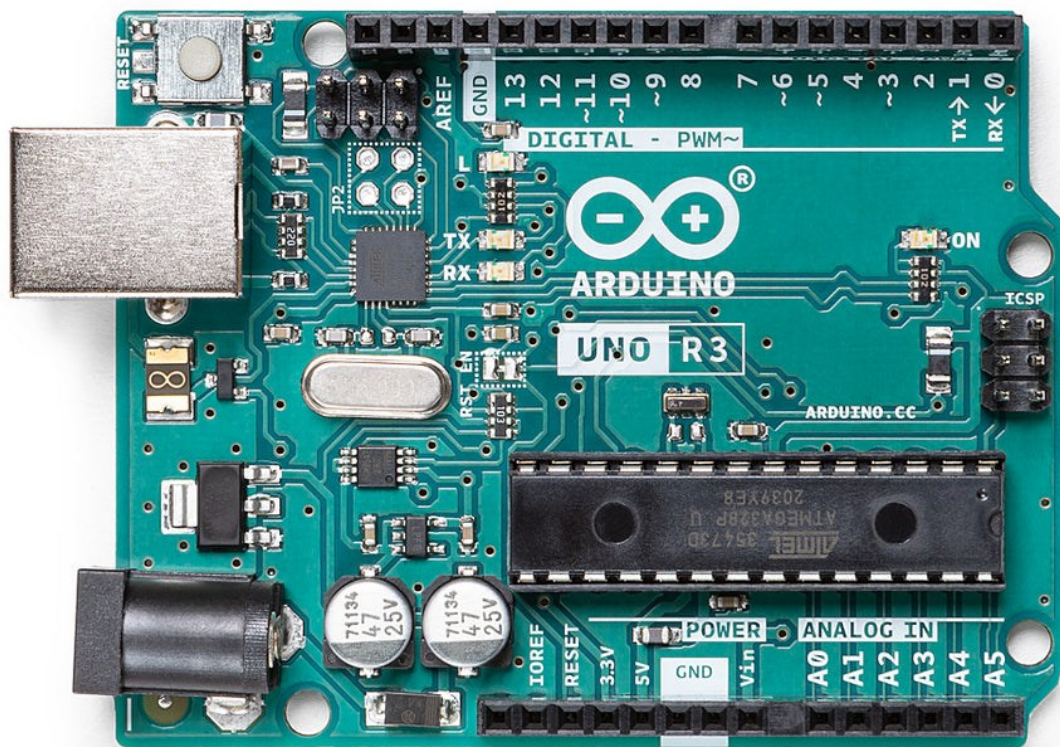


Figure 5 Arduino Uno board, widely used for DIY robotics and embedded systems

(source: [32])

5. Thymio Robot

Thymio is an entry-level educational robot designed to introduce young learners to robotics, featuring:

- Pre-programmed behaviors: Robots can move, follow lines, and avoid obstacles.
- Graphical coding: Uses Blockly, Scratch, and Aseba programming languages.
- Wireless control: Can be operated via tablets or a computer.

Thymio is an educational robot designed to introduce young learners to programming through graphical and visual coding tools.



Figure 6 Thymio educational robot for young learners

(source: [33])

Comparison of Educational Robotics Platforms:

Table 1 Comparison between educational robotics platforms

Platform	Best for	Programming	Key Features	Limitations
mBot Ranger	STEM Education, Beginners	Scratch, Arduino C	3-in-1 robot, sensors, remote control	Limited AI support
LEGO Mindstorms	Schools, AI Learning, Competitions	Scratch, Python	Modular, AI-compatible, Sensors	Expensive, additional AI modules needed
VEX Robotics	Competitive Robotics, Engineering	C++, Python, VEXcode	Modular, sensors, VRC support	Complex, steep learning curve

Raspberry Pi / Arduino	DIY Robotics, Research, AI	Python, C++	AI/IoT integration, Open source	Requires technical knowledge
Thymio Robot	Young Learners, Beginner Coding	Blockly, Aseba	Simple, graphical coding	Limited for advanced users

(source: own elaboration)

2.2.3 Impact on Student Motivation and Learning

Numerous studies have demonstrated that educational robotics significantly increases student motivation and engagement in STEM subjects. By providing hands-on experiences, robotics helps to demystify complex topics and fosters a sense of ownership and creativity in the learning process. A study by [13] found that young children participating in a robot-based learning environment developed a range of social and cognitive skills through collaborative problem-solving activities. This approach aligns with Papert's (1980) constructionist theory, which posits that learning is more effective when students actively construct knowledge through tangible projects.

Educational robotics has also been shown to enhance teamwork and communication skills. Platforms like mBot are often used in project based learning environments, where students work in teams to solve real world challenges. [14] reported that using eduScrum, an agile teaching methodology, in combination with robotics led to significant improvements in students' collaboration and critical thinking abilities.

2.2.4 Methodologies in Educational Robotics

Various teaching methodologies have been developed to maximize the effectiveness of educational robotics. One prominent approach is constructionist learning, which emphasizes building and experimenting with physical models to develop deeper conceptual understanding [9]. Another popular method is project-based learning (PBL), where students engage in extended, inquiry-driven projects that culminate in a tangible artifact or presentation.

In addition, [14] explored the integration of agile methodologies, such as eduScrum, to promote self-directed learning and teamwork in robotics education. Under this framework, students are divided into teams and assigned specific roles, such as Scrum Master, to coordinate their tasks and monitor progress. This structure enhances both technical and interpersonal skills by encouraging frequent feedback cycles and collaborative problem-solving.

2.2.5 Limitations and Challenges

Despite its numerous benefits, the implementation of educational robotics faces certain challenges. Access to resources, including robots and supporting technology, can be limited in underfunded schools. Additionally, teachers may require specialized training to effectively integrate robotics into their curricula. Research indicates that overcoming these challenges requires comprehensive support systems, including teacher training programs, educational grants, and access to open-source materials [12].

There is also a need for further research on long-term learning outcomes associated with robotics education. While many studies report short-term improvements in engagement and understanding, more longitudinal studies are needed to determine how these gains translate into career aspirations and STEM competencies over time.

2.2.6 Future Directions

The future of educational robotics is expected to include more AI-enhanced platforms capable of adaptive learning. These systems will likely integrate machine learning models to provide personalized feedback and challenge levels tailored to individual learners. Innovations in augmented reality (AR) and virtual reality (VR) could further enhance the interactivity of robotics-based education by simulating complex scenarios are difficult to replicate in traditional classroom environments.

Furthermore, expanding partnerships between educational institutions and technology companies can help make robotics more accessible and scalable. Collaborative initiatives, such as the distribution of micro:bit kits to schools in the UK, have demonstrated the potential for large-scale impact through targeted outreach programs [14].

2.3 Control Algorithms for Navigation and Obstacle Avoidance

The ability of autonomous mobile robots to navigate through dynamic environments and avoid obstacles is fundamental to their effectiveness in real-world applications. Navigation and obstacle avoidance strategies rely on control algorithms that process sensor data, interpret the environment, and generate appropriate movement commands. This section explores the key approaches to navigation and obstacle avoidance, including reactive and deliberative control models, sensor-based methodologies, and hybrid approaches that integrate artificial intelligence.

2.3.1 Reactive Control Approaches

Reactive control systems operate based on real-time sensor data without requiring precomputed maps or prior knowledge of the environment. These approaches prioritize immediate response to obstacles and changing surroundings, making them suitable for environments with unpredictable elements.

Subsumption Architecture:

One of the most widely used reactive control models is the subsumption architecture, developed by Brooks (1986). This architecture employs a layered hierarchy where each layer operates independently to handle specific behaviors, such as obstacle avoidance and path following. Lower layers handle immediate reactions (e.g., avoiding obstacles), while higher layers manage more complex tasks (e.g., long-term navigation)[3]. This model enables robots to quickly adapt to their environment without relying on extensive computation.

Vector Field Histogram (VFH) for obstacle avoidance:

Vector Field Histogram (VFH), introduced by Borenstein and Koren (1991), is another reactive control approach that enables real-time obstacle avoidance. VFH constructs a polar histogram from sensor data, representing the density of obstacles around the robot. Based on this data, the algorithm selects the safest path forward while ensuring smooth movement and minimal deviation from the intended trajectory [2]. This technique has been widely implemented in industrial robots and autonomous vehicles due to its efficiency in navigating cluttered spaces.

2.3.2 Deliberative Control Approaches

Unlike reactive models, deliberative control systems use high-level reasoning and precomputed planning to navigate environments. These methods are particularly useful for applications that require strategic decision-making, such as warehouse automation and search-and-rescue missions.

Path planning algorithms:

Path planning algorithms generate optimal routes from a starting position to a destination while avoiding obstacles. Two widely used approaches include:

- **Dijkstra's Algorithm:** A graph based method that finds the shortest path between nodes in a known environment. This method ensures optimal route selection but is computationally intensive for large maps [15].
- **A (A-Star) Algorithm:** An improvement over Dijkstra's algorithm, A* introduces heuristics to prioritize paths based on estimated distance to the goal. This approach significantly reduces computation time while maintaining optimal path selection [17].

These algorithms are commonly used in robotic navigation systems where pre-mapped environments are available, such as indoor autonomous delivery systems.

2.3.3 Sensor-Based Navigation Techniques

Modern autonomous robots rely on sensor integration to perceive their environment and execute navigation tasks. Key sensor-based approaches include:

Simultaneous Localization and Mapping (SLAM):

SLAM is widely adopted technique that enables a robot to map an unknown environment while simultaneously estimating its own position within the map. SLAM systems integrate sensor data from LiDAR, cameras, and IMUs (Inertial Measurement Units) to construct and update real-time maps [21]. This technology is essential for autonomous mobile robots operating in unstructured or GPS-denied environments, such as search and rescue operations and underground navigation.

Kalman Filtering for Sensor Fusion:

The Kalman filter is a mathematical framework used to improve localization accuracy by combining data from multiple sensors. By filtering out noise and uncertainties, Kalman filters enhance robot perception and contribute to smoother navigation [18]. This method is widely applied in autonomous vehicles and drones to improve positional accuracy.

2.3.4 Hybrid Approaches: Combining Reactive and Deliberative Control

Hybrid control strategies integrate elements of both reactive and deliberative models to achieve optimal navigation performance. These approaches allow robots to switch between real-time obstacle avoidance and strategic path planning based on environmental complexity.

Behaviour-Based Hybrid Control:

Behaviour-based hybrid models employ a combination of pre-defined behaviours (e.g., obstacle avoidance, wall-following, and goal-seeking) that dynamically interact based on real-time sensor data. These models provide greater adaptability and robustness in dynamic environments [16].

Deep reinforcement learning for navigation:

Recent advancements in artificial intelligence have introduced deep reinforcement learning (DRL) for robot navigation. DRL algorithms enable robots to learn optimal navigation strategies through trial and error in simulated environments before applying them in real-world scenarios [19]. This approach has demonstrated significant improvements in complex navigation tasks, such as multi-agent coordination and adaptive obstacle avoidance.

2.3.5 Applications of Control Algorithms in Autonomous Robotics

The discussed control algorithms have been successfully implemented in various real world applications:

- **Autonomous Vehicles:** Self driven cars use a combination of SLAM, Kalman filtering, and path planning algorithms to navigate safely in urban environments [20].
- **Industrial Robots:** Reactive control models, such as subsumption architecture, are widely used in warehouse robots for real-time path adjustments [7].

- Healthcare and Assistive Robots: Navigation and obstacle avoidance techniques enable service robots to assist elderly and disabled individuals in domestic settings [13].

2.3.6 Limitations and Future Directions

Despite significant progress in navigation and obstacle avoidance, challenges remain:

- Sensor Limitations: While SLAM and VFH are effective, sensor noise and environmental conditions (e.g., fog, darkness) can reduce accuracy.
- Computational Complexity: Path planning algorithms, particularly those used in high-resolution maps, require substantial processing power, which may be a constraint in low-power robots.
- Real-Time Adaptability: Traditional navigation algorithms often struggle in highly dynamic environments where obstacles and conditions change unpredictably. One of the primary limitations is the inability of conventional path-planning algorithms to react instantly to moving obstacles or sudden environmental changes. To overcome this, researchers have been exploring deep reinforcement learning (DRL) and adaptive control algorithms that allow robots to learn optimal strategies in real time through trial and error processes in simulated environments [19]. Furthermore, Simultaneous Localization and Mapping (SLAM) techniques continue to evolve to support more dynamic, cluttered, and GPS-denied environments [22].

Additionally, AI-driven sensor fusion and probabilistic models have been proposed to enhance robots' real-time decision-making capabilities, allowing them to predict and respond to changes in their surroundings. The future of real-time adaptability lies in the integration of neuromorphic computing and bio-inspired algorithms, which aim to emulate human cognitive functions for faster and more efficient decision-making [22].

3 RESEARCH ON MOBILE ROBOTICS PLATFORMS

3.1 EVOLUTION OF AUTONOMOUS MOBILE ROBOT

Autonomous mobile robots (AMRs) has experienced significant advancements over the past few decades. Initially, AMRs operated on predefined movement patterns with basic sensor-based navigation, limiting their adaptability in dynamic environments. However, recent developments in artificial intelligence (AI), machine learning, and edge computing have transformed AMRs into sophisticated machines capable of real-time decision-making and autonomous learning.

A pivotal milestone in AMR development is the implementation of Simultaneous Localization and Mapping (SLAM) algorithms. SLAM enables robots to construct a map of an unknown environment while simultaneously localizing themselves within it, a fundamental capability for autonomous navigation [35]. This technology has been widely adopted in applications such as robotic vacuum cleaners, industrial automation, and self-driving vehicles. The integration of sensor fusion technologies, including LiDAR, stereo vision, and depth sensors, has further enhanced perception accuracy and obstacle avoidance in AMRs [35].

The incorporation of AI and reinforcement learning has significantly improved AMR capabilities, allowing robots to predict and adapt to changes in their surroundings. Modern AMRs can learn from previous experiences, optimize navigation paths, and interact more effectively with objects and humans [34]. For instance, reinforcement learning enables AMRs to develop optimal motion strategies through trial-and-error learning, enhancing their operation in unstructured environments.

The advent of 5G technology and cloud-based robotics has expanded the potential of AMRs by facilitating the offloading of complex computations to cloud servers. This advancement enhances processing power, reduces latency, and improves real-time data exchange between robots and central control units [36]. In industries such as warehousing, healthcare, and autonomous transportation, AMRs now communicate and coordinate their movements through networked multi-agent systems, thereby improving operational efficiency [36].

Despite these advancements, challenges remain in ensuring safe human-robot interaction, long-term autonomy, and energy-efficient operation. Future research is expected

to focus on developing bio-inspired robotic designs, neuromorphic computing for real-time processing, and quantum machine learning models to enhance AMR decision-making capabilities. These developments aim to shape the next generation of intelligent, self-learning autonomous robots capable of seamless operation across various industries and everyday applications.

3.2 TELEOPERATION AND REMOTE CONTROL IN MOBILE ROBOT

Teleoperation plays a crucial role in operating robots within hazardous, inaccessible, or dynamically changing environments where full autonomy may not be feasible. While advancements in artificial intelligence and autonomous navigation have enhanced robot independence, many industries including search and rescue, military operations, space exploration, and healthcare still require human oversight to ensure safety, adaptability, and complex decision-making.

Teleoperated robots can be controlled remotely using wired or wireless communication, with varying degrees of autonomy. Traditional joystick-based control has evolved into more sophisticated methods, including:

- Gesture-based teleoperation, where hand movements and body gestures are utilized to command the robot. For instance,[39] developed a system using Leap Motion sensors to recognize gestures, enabling precise and compliant robot motions.
- Brain-computer interfaces (BCI), which allow direct control of robots using neural signals, enabling applications for individuals with mobility impairments. Recent advancements in BCI technology have facilitated more intuitive and efficient robot teleoperation, especially in environments where traditional control devices are impractical.
- Augmented Reality (AR) and Virtual Reality (VR)-based control, where operators receive real-time visual feedback for intuitive interaction in remote environments. These immersive technologies enhance the operator's situational awareness, leading to more effective teleoperation.

One of the significant challenges in teleoperation is latency, which affects the real-time responsiveness of the robot. Delays in signal transmission can hinder performance,

particularly in applications like robot-assisted surgery, deep-sea exploration, and space missions. The implementation of 5G networks has drastically reduced latency, improving control accuracy in remote applications. For example, [38] explored the role of 5G and future 6G networks in advancing robotic telesurgery by minimizing latency and enhancing data reliability for real-time remote operations.

Additionally, haptic feedback integration is an emerging area in teleoperation. Haptic devices, such as gloves and force-feedback joysticks, allow operators to feel the environment the robot is interacting with, improving precision in delicate tasks such as remote surgeries, bomb disposal, and micro-manipulation in laboratories.[39] incorporated a vibration-based force feedback system in their gesture-based teleoperation system, providing operators with telepresence capabilities.

Furthermore, AI-assisted teleoperation is gaining traction, where artificial intelligence helps predict human intentions and filters out control noise. Machine learning models can enhance human-robot interaction by assisting in real-time decision-making and automating routine actions to reduce the cognitive load on operators. For instance, [37] proposed a neural network-based LiDAR gesture recognition system for real-time robot teleoperation, improving robustness and stability in robot locomotion.

Despite advancements, teleoperation still faces limitations such as:

- Communication disruptions in high-interference environments.
- Operator fatigue during long-duration remote control tasks.
- Bandwidth constraints, particularly when using high-resolution video feeds for navigation.

Future developments are expected to focus on edge computing, multi-agent teleoperation systems, and cloud-based AI models to improve efficiency, reliability, and scalability. These advancements will further bridge the gap between fully autonomous robots and human-controlled systems, making teleoperated robots more effective across diverse industries.

3.3 Research on Robotic Platforms for Autonomous and Remote-Control Applications

Autonomous and remote-controlled robotic platforms have been extensively studied in both educational and industrial research. These platforms serve as testbeds for developing control algorithms, AI-driven navigation, and human-robot interaction techniques.

3.3.1 mBot Ranger: Research in STEM and AI Education

The mBot Ranger has been widely used in educational robotics and AI-driven research. Studies indicate that this platform is highly effective in introducing students to programming concepts, robotic navigation, and autonomous control systems [46]. Research findings highlight:

- **STEM Engagement:** mBot Ranger enhances student interest in robotics and computational thinking.
- **AI and Sensor-Based Learning:** The integration of ultrasonic sensors, gyroscopes, and line-following modules allows students to experiment with autonomous navigation algorithms [45].

However, research also notes that mBot Ranger lacks deep AI functionalities, limiting its suitability for advanced machine learning applications.

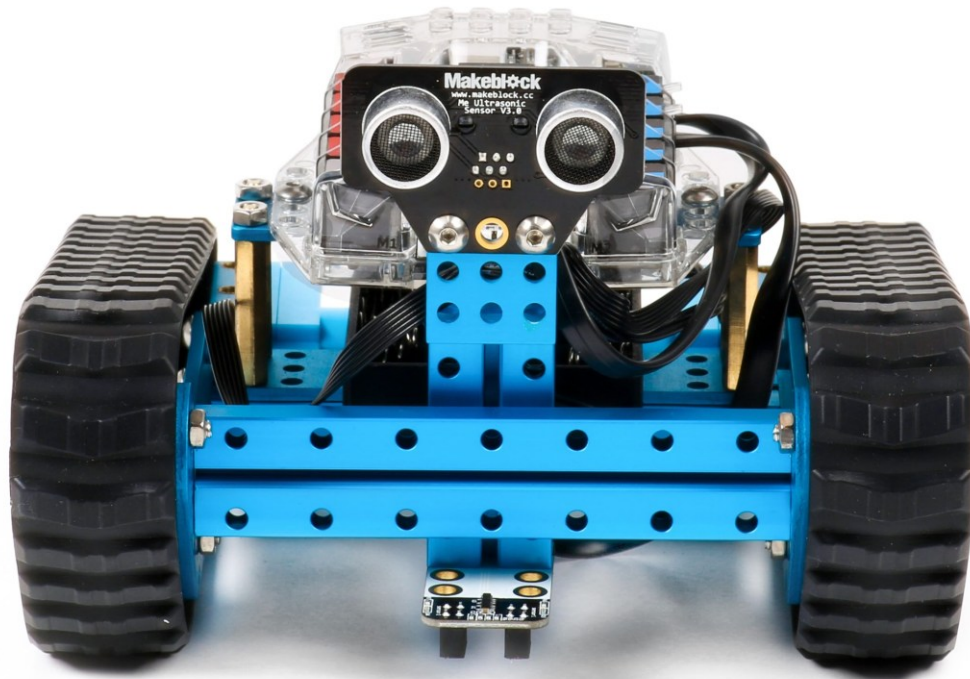


Figure 7 The mBot Ranger robot with Bluetooth module and ultrasonic sensors

(source: [47])

3.3.2 LEGO Mindstorms: A Research Tool for AI and Robotics

LEGO Mindstorms has been extensively used in AI-driven robotics research and STEM education. Key research findings include:

- **Machine Learning and Robotics:** Studies show that LEGO Mindstorms is useful for introducing students to AI concepts through Python and block-based programming [9].

- **Remote Control and Human-Robot Interaction:** Researchers have used LEGO robots in experiments on gesture-based and voice-controlled teleoperation systems [39].
- **Algorithm Testing:** The modular design allows students and researchers to develop path planning and object recognition algorithms.

However, cost and limited industrial application remain key constraints.

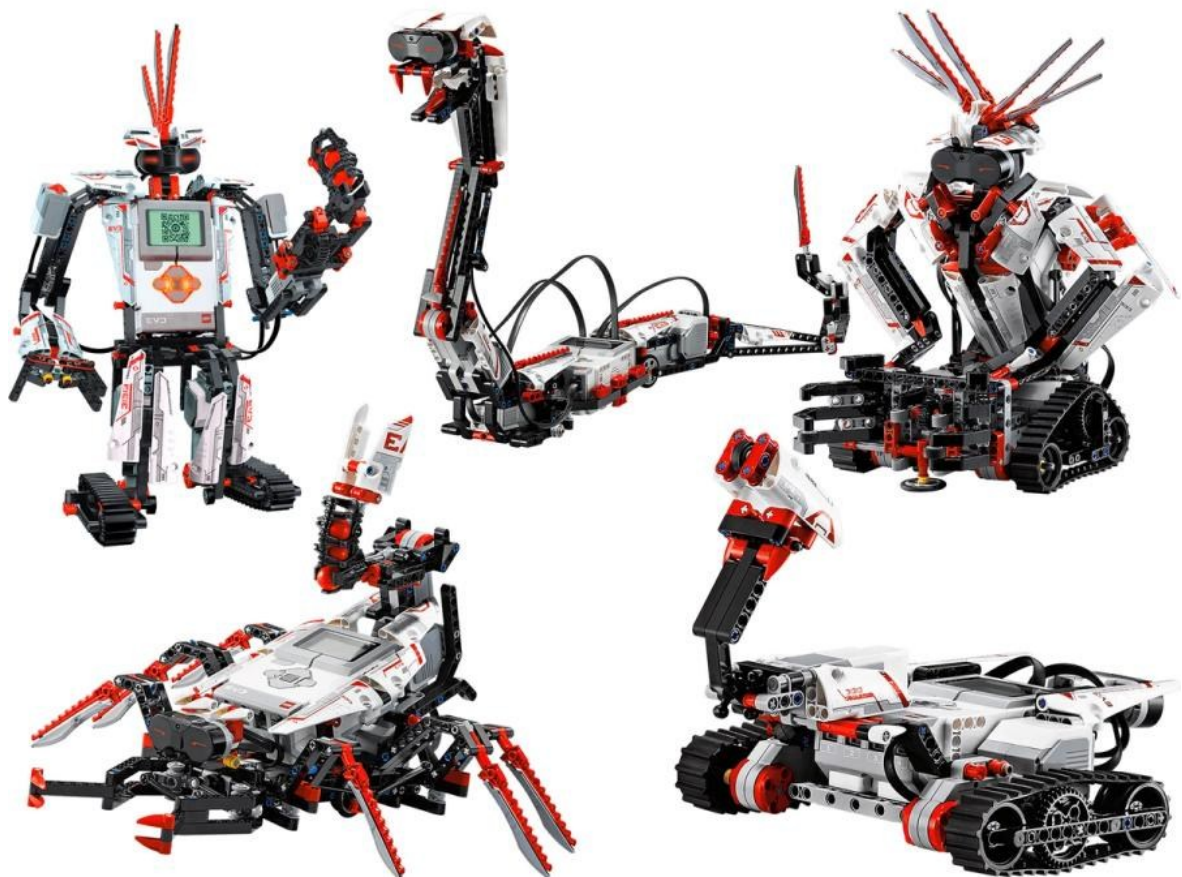


Figure 8 LEGO Mindstorms EV3 Robot

(source: [48])

3.3.3 VEX Robotics: Engineering and Competitive Robotics Research

VEX Robotics platforms, particularly VEX IQ and VEX V5, are widely studied in engineering education and robotic competitions. Research highlights:

- **ROS (Robot Operating System) Integration:** Some studies have demonstrated that VEX robots can be integrated with ROS frameworks for testing autonomous navigation and real-time AI decision-making [41].

- Teleoperation Studies: Researchers have used VEX robots to analyze latency in remote control systems, particularly in 5G-enabled environments [42].
- Competitive Learning Environment: VEX competitions have been linked to higher retention rates in STEM education due to their hands-on, project-based structure [43].

Despite these benefits, VEX platforms require a higher learning curve and additional components for advanced AI applications.



Figure 9 VEX V5 Robotics Competition Kit

(source: [49])

3.3.4 Raspberry Pi and Arduino-Based Robots: AI, IoT, and Research Applications

Raspberry Pi and Arduino-based robots have been widely used in research due to their low cost, flexibility, and open-source nature. Some of the key areas where these platforms have been studied include:

- **AI-Driven Object Detection:** Studies have integrated OpenCV and deep learning models with Raspberry Pi-powered robots to enable real-time image recognition [36].
- **Internet of Things (IoT) and Robotics:** Research highlights that Raspberry Pi-based robots can be connected to cloud computing systems for remote monitoring and control [38].
- **Embedded Systems and Teleoperation:** Arduino-based robots have been used to test wireless remote control mechanisms, particularly in search and rescue applications [37].

However, these platforms require strong programming skills, making them less beginner-friendly compared to plug-and-play educational robots. It enables low-cost computing for autonomous robots and remote-control systems.

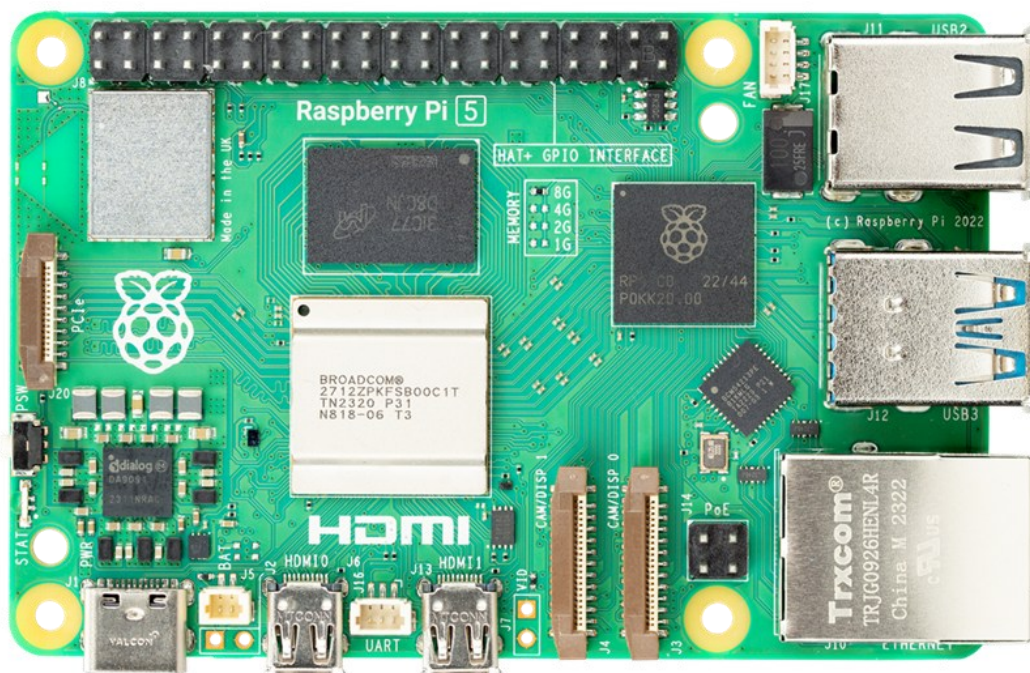


Figure 10 Raspberry Pi 5 single-board computer

(source: [50])

3.3.5 Thymio: Research in Early Robotics Education and Human-Robot Interaction

Thymio is primarily used for early-stage robotics education, with research focusing on:

- **Cognitive Learning and Robotics:** Studies indicate that visual and block-based programming tools in Thymio help young learners grasp basic robotics and coding principles [40].
- **Wireless Control and Gesture-Based Teleoperation:** Researchers have explored using gesture commands for controlling Thymio robots, making them useful for human-robot interaction studies [37].

Despite its benefits in beginner-level robotics education, Thymio has limited use in AI-driven and industrial robotics applications.

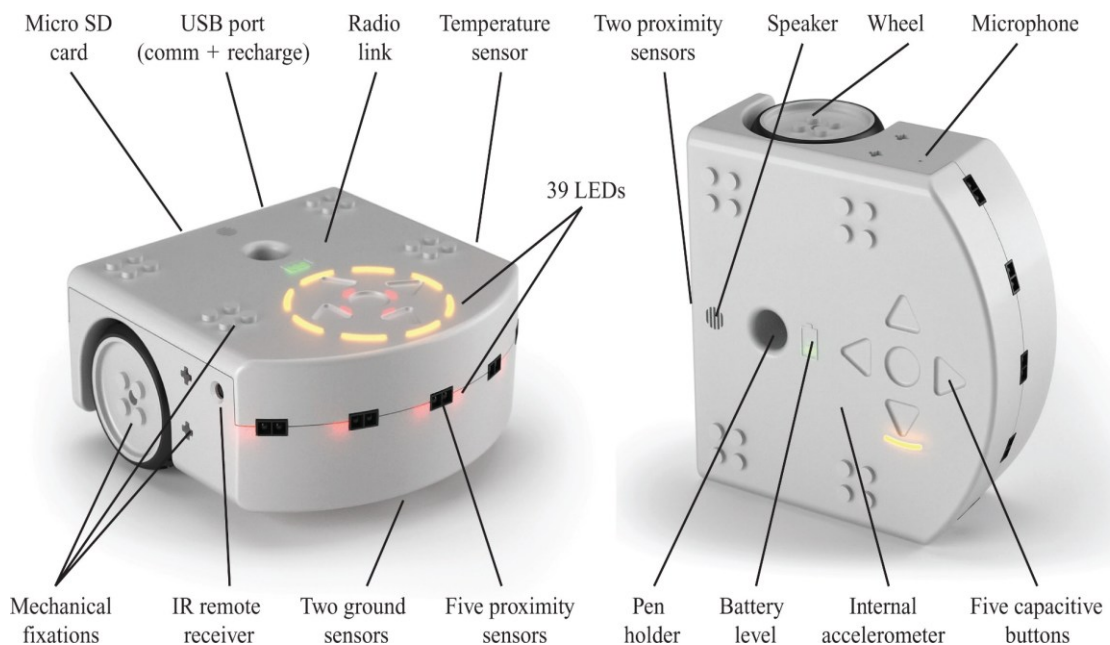


Figure 11 Thymio educational Robot

(source: [51])

3.3.6 Comparative Analysis of Research on Robotic Platforms

The following table summarizes key findings from research on the discussed robotic platforms.

Table 2 Comparative alalysis of research on robotic platforms

Platform	Key Research Findings	Limitations Identified in Research
mBot Ranger	Enhances STEM education, introduces basic AI [46]	Limited AI functionalities, basic sensors only
LEGO Mindstorms	Strong in AI learning, gesture-based control [39]	High cost, lacks industrial applications
VEX Robotics	Suitable for engineering, competitive learning [43]	Steep learning curve, high hardware cost
Raspberry Pi / Arduino	AI-driven robotics, IoT connectivity [36]	Requires advanced coding skills, setup complexity
Thymio	Ideal for early STEM learning [40]	Limited for AI and complex robotics applications

(source: own elaboration)

3.4 Summary of Research Findings and Future Directions

Existing research demonstrates that each robotic platform serves a distinct purpose in education, AI, and teleoperation studies. While platforms like mBot Ranger and Thymio are best suited for beginner-level robotics education, platforms like VEX, Raspberry Pi, and Arduino offer greater flexibility for advanced research in AI, IoT, and machine learning.

II PRACTICAL PART

4 SYSTEM ARCHITECTURE AND DEVELOPMENT

4.1 System Overview

The practical implementation of this thesis focuses on designing, assembling, and programming an autonomous mobile robot with remote control capabilities using the mBot Ranger platform. The goal of this system is to demonstrate how a compact, modular robot can perform key autonomous behaviors such as line following, obstacle avoidance, and human (or target) following while also supporting remote control via a Bluetooth-based interface.

The mBot Ranger was selected due to its educational versatility, modular hardware design, and support for Arduino-based programming. Developed by Makeblock, the Ranger is a 3-in-1 transformable robot kit that includes various build forms such as the Land Raider (tank), Nervous Bird (self-balancing robot), and Dashing Raptor (three-wheeled racing car). For this thesis, the Land Raider configuration was used due to its robustness, track-based mobility, and flat surface adaptability, which are essential for testing autonomous navigation behaviors.

The system uses the Me Auriga mainboard, based on the Arduino Mega 2560, which offers extended ports and built-in support for motor drivers, sensors, and serial communication. It controls the following essential hardware components:

- **Ultrasonic sensor**, used for obstacle detection and avoidance.
- **Line-following infrared sensor module**, which enables the robot to follow black tape on white surfaces.
- **Gyroscope**, used to monitor orientation and balance.
- **Encoder motors**, which allow for speed regulation and turning precision.
- **Bluetooth BLE module (v1.0_c)**, enabling wireless communication with a smartphone or computer.

These components work together to execute both autonomous tasks and Bluetooth-enabled remote-control commands. The user can switch modes using a mobile app or serial monitor by sending single-letter commands (e.g., L for line-following, O for obstacle avoidance, etc.).

Overall, this system demonstrates a real-world educational prototype of autonomous robotics using a cost-effective and highly modular platform. It also serves as a model for similar applications in interactive STEM education, mobile sensing, and remote-controlled robotics.

4.2 Hardware Architecture

The hardware architecture of the autonomous mobile robot system is centered around the Makeblock mBot Ranger and its modular design, which allows for the integration of multiple sensors, actuators, and communication modules. The core of the robot is the Me Auriga mainboard, which is built on the Arduino Mega 2560 microcontroller and provides robust support for both autonomous control logic and remote Bluetooth communication.

4.2.1 Central Control Unit: Me Auriga Mainboard

The Me Auriga is equipped with **10 RJ25 ports**, allowing for easy plug-and-play connectivity with Makeblock sensors and modules. It also features:

- A **USB port** for programming via Arduino IDE
- An onboard **buzzer** for sound feedback
- A **programmable RGB LED ring**
- A **reset button** and **power switch**
- Dedicated motor driver circuits
- **I2C and UART communication interfaces**

This board supports modular robotic development and serves as the command center for all functionalities implemented in the project, including motor control, sensor feedback processing, and communication management.



Figure 12 Me Auriga mainboard with RJ25 ports, RGB LED ring, USB interface, and power switch

(source: [52])

4.2.2 Motors and Chassis

The robot uses two encoder motors to drive its track-based chassis (Land Raider configuration). Encoder feedback allows for precise speed and direction control, essential for tasks such as line following and turning during obstacle avoidance.

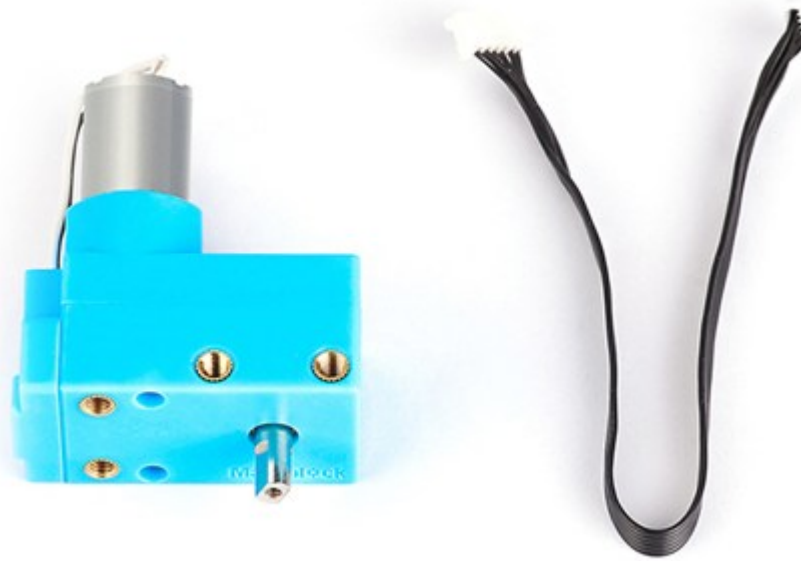


Figure 13 180 Optical Encoder Motor with RJ25 cable, enabling precise speed and direction feedback

(source: [53])

- **Drive System:** Tracked movement provides increased stability over uneven surfaces.
- **Encoders:** Enable feedback-based control for straight-line travel and pivoting maneuvers.

4.2.3 Sensors

The mBot Ranger robot is equipped with a variety of sensors that allow it to perceive its environment and execute autonomous tasks such as line following, obstacle avoidance, and orientation tracking. These sensors are connected to the Me Auriga mainboard via RJ25 modular ports, which simplify the wiring process and support plug-and-play operation.

Line-Following Sensor:

For the line-following functionality, the mBot Ranger uses the Me Line Follower v1.2 sensor module. This module includes two infrared reflectance sensors that detect light intensity reflected from the surface beneath the robot. It is designed to differentiate between black and white surfaces typically a black path on a white background by comparing reflected infrared signals.

The sensor outputs binary signals (0 or 1) based on surface reflectivity and transmits this information to the microcontroller, which adjusts motor speed and direction accordingly to

stay aligned with the line path. The simplicity and real-time feedback provided by the Me Line Follower make it a widely adopted component in educational robotics applications [54].



Figure 14 Me Line Follower v1.2 sensor detects lines using dual infrared reflectance for navigation

(source: [54])

Ultrasonic Sensor:

The ultrasonic distance sensor is mounted at the front of the robot and is responsible for obstacle detection. It emits high-frequency sound waves and measures the time taken for the echo to return, thus calculating the distance to an object. If the robot detects an object within a defined threshold distance, it initiates an avoidance maneuver. This sensor plays a central role in obstacle avoidance mode.

The **Me Ultrasonic Sensor** mounted at the front of the robot calculates distances by emitting and receiving high-frequency sound waves. If an object is detected within a defined threshold (e.g., 15 cm), the robot initiates an avoidance behavior. This setup supports the obstacle avoidance mode by ensuring responsive distance-based navigation [55].



Figure 15 Me Ultrasonic Sensor v3.0 measures distance with sound waves for obstacle detection and avoidance

(source: [55])

These sensors feed data to the controller, enabling autonomous decision-making in real time.

Gyroscope Sensor:

The Me Auriga mainboard includes a built-in gyroscope, which is used to measure angular velocity and orientation. Although the current robot configuration (Land Raider) does not rely on balancing behaviors, the gyroscope provides useful feedback in more advanced settings, such as self-balancing robot designs or when implementing smooth turn tracking and drift correction.

4.2.4 Power Supply

The mBot Ranger robot is powered by a 6-cell AA battery holder that supplies regulated power to the Me Auriga mainboard, motors, and connected sensors. Each AA battery provides approximately 1.5V, resulting in a total supply voltage of 9V, which is within the operational range of the system components.

The battery pack connects to the Me Auriga board via a DC barrel jack, which ensures a secure and stable connection. This configuration allows for easy removal and replacement of batteries without the need for soldering or external circuitry. The battery pack is typically mounted underneath the robot's chassis to maintain balance and reduce wiring clutter.

This power supply solution is especially suitable for educational and prototyping environments, as it is cost-effective, easy to replace, and readily available. However, because the batteries deplete relatively quickly under continuous motor load, care must be taken to monitor battery levels to avoid performance degradation during testing.



Figure 16 Six-cell AA battery holder with DC jack, supplying power to motors and sensors (source: [56])

4.3 Software Development Environment

The software development for the autonomous mobile robot was carried out using the Arduino platform, which provides a flexible and accessible environment for microcontroller-based applications. The Me Auriga mainboard of the mBot Ranger is based on the Arduino Mega 2560, making it fully compatible with the Arduino IDE (Integrated Development Environment).

The Arduino IDE was selected due to its extensive support for embedded systems programming, its large online community, and its seamless integration with Makeblock's official libraries. The robot was programmed in C/C++, the standard language for Arduino development.

4.3.1 Tools and Libraries Used

Arduino IDE:

The code was written and compiled using the Arduino IDE (version 1.8.x). This environment allows for direct uploading of firmware via USB and supports real-time debugging via the Serial Monitor.

Makeblock Libraries:

The official Makeblock library was installed to support the control of specific mBot Ranger components. This includes MeAuriga.h, which simplifies the initialization and use of:

- Motors
- Line-following sensors
- Ultrasonic sensors
- Gyroscope
- LED rings and buzzer

Custom Logic:

Custom logic was implemented to handle the mode switching mechanism. Each behavior (e.g., line-following, obstacle avoidance, hand-following) is stored as a function, and the robot executes the corresponding logic based on single-letter Bluetooth commands (L, O, H, S).

Serial Communication:

The SoftwareSerial library was used to configure a dedicated UART port for Bluetooth Low Energy (BLE) communication. Commands were sent from a smartphone app and parsed on the robot side for appropriate behavior switching.

4.3.2 Development Workflow

1. **Library Setup:** The Makeblock libraries were imported via the Arduino Library Manager.
2. **Sensor & Motor Calibration:** Each sensor and motor module were individually tested to validate wiring and responsiveness.

3. **Modular Programming:** Separate code blocks were written for each mode (e.g., `line-Following ()`, `obstacle-Avoidance ()`, and `Hand-Following mode ()`).
4. **Mode Selection via Bluetooth:** A `switch-case` control structure was used to assign robot behavior dynamically, based on the received character via BLE.
5. **Testing and Debugging:** The Serial Monitor and LED indicators were used for debugging and testing sensor outputs and Bluetooth connectivity in real time.

5 CONTROL ALGORITHM DESIGN

5.1 Autonomous Functionalities

This section outlines the autonomous behaviors programmed into the mBot Ranger, each of which allows the robot to perform intelligent tasks based on sensor input. The following modes were developed and tested: line following, obstacle avoidance, and hand following. Each function was implemented as a modular component, allowing the system to switch between behaviors via Bluetooth commands. These functionalities showcase the robot's ability to perceive and respond to environmental conditions using sensor data processed through embedded control algorithms.

5.1.1 Line Following

The line following functionality enables the mBot Ranger to autonomously navigate a path marked with a black line on a white surface. This is achieved using the Makeblock Me Line Follower v1.2 sensor, which contains two infrared reflectance sensors that detect light intensity reflected from the surface below the robot.

The sensors produce binary output:

- **1** when over a **white surface** (reflective),
- **0** when over a **black line** (non-reflective).

The logic interprets these signals to keep the robot aligned with the line. A simple proportional control algorithm was used to adjust motor speed and direction based on which side of the sensor detects the line. The implementation includes three core cases:

- Left sensor detects black, right sensor detects white → Turn left
- Right sensor detects black, the left sensor detects white → Turn right
- Both sensors detect black → Move forward

Both sensors detect white → Stop or reverse (optional, for end-of-line detection)

This approach is effective in educational environments due to its simplicity and real-time responsiveness.

5.1.2 Obstacle Avoidance

The obstacle avoidance functionality enables the mBot Ranger to navigate its environment while preventing collisions with static objects. This is achieved using the Makeblock Me Ultrasonic Sensor v3.0, mounted at the front of the robot. The sensor functions by emitting high-frequency ultrasonic waves and measuring the time it takes for the echo to return after bouncing off an object. This time-of-flight method allows the system to accurately calculate the distance between the robot and any nearby obstacle.

The ultrasonic sensor outputs a distance value in centimeters, which is continuously monitored by the microcontroller. A threshold value (e.g., 15 cm) was set to trigger avoidance behavior. If an object is detected within this distance, the robot halts its forward motion and performs a predefined escape maneuver, such as reversing and turning to the right or left.

Control Logic Flow:

- The robot continuously reads its front ultrasonic sensor.
- If the measured distance is greater than 15 cm, it keeps driving straight ahead.
- If the distance is 15 cm or less:
 - It immediately stops the wheels.
 - It reverses both motors at a fixed speed for 500 ms.
 - It pivots in place to the right for 300 ms.
- After turning, the robot again checks its front sensor; once the path is clear (> 15 cm), it resumes forward motion.

This simple threshold-based algorithm provides reliable performance in open spaces and is ideal for introductory robotics applications where complex path planning is not required. During testing, the robot consistently detected and avoided obstacles such as books, boxes, and table legs. The ultrasonic sensor provided stable readings under standard indoor lighting conditions and flat surfaces.

5.1.3 Hand Following

The hand following functionality is designed to allow the mBot Ranger to detect and follow an object typically a human hand within a certain distance range using its ultrasonic sensor.

This mode demonstrates how simple sensory feedback can be applied to create responsive, real-time interaction between a human and a robot.

Unlike obstacle avoidance, where the robot moves away from detected objects, this mode reverses the logic: the robot maintains a fixed distance from a detected object by continuously adjusting its movement based on the measured distance.

Operational Logic:

- If the detected distance is greater than 30 cm and less than 50 cm → The robot moves forward to reduce the distance.
- If the detected distance is between 15 and 30 cm → The robot stays still and monitors the object position.
- If the detected distance is below 15 cm → The robot moves backward.
- If the detected distance is more than 50 cm → The robot stays still and monitors the object position.

The hand following mode was implemented as a continuous loop where ultrasonic sensor readings are evaluated every 100 milliseconds. Adjustments to motor speed and direction are made based on this input, enabling smooth and reactive tracking. During testing, the robot was able to follow a moving hand at walking speed with acceptable stability, especially in well-lit and echo-free indoor conditions.

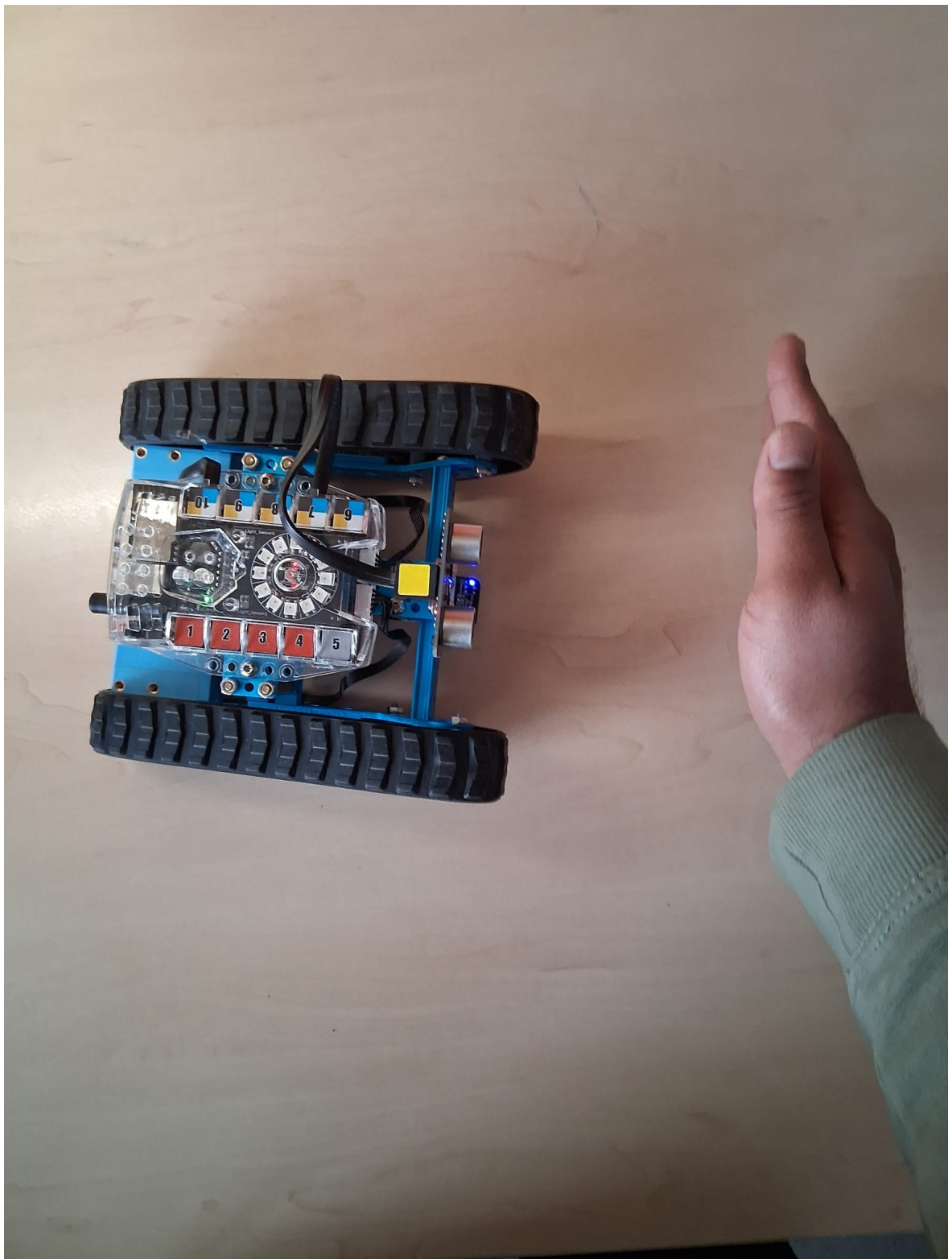


Figure 17 mBot Ranger tracks a human hand using ultrasonic sensor, enabling dynamic user interaction

(source: own elaboration)

5.2 Remote Control Integration

In addition to autonomous behaviors, the mBot Ranger was configured to operate in remote control mode, allowing the user to switch between different functional states through a Bluetooth Low Energy (BLE) communication interface. This feature enables flexible testing and real-time control during robot operation and highlights the system's adaptability for interactive robotics applications.

5.2.1 Custom Android Application Development

As an enhancement to the Bluetooth-based control system, a custom Android application was developed using Java in Android Studio to manage the operational modes of the mBot Ranger via Bluetooth Low Energy (BLE). The goal was to replace third-party terminal apps with a dedicated, user-friendly interface that would allow seamless communication between the robot and the user.

The app enables the user to scan for BLE devices, connect to the Makeblock BLE v1.0_c module, and send single-character commands corresponding to different robot modes. This custom solution provides a more intuitive interaction compared to traditional terminal-based control.

App Features:

BLE Scanning and Connection:

The app initiates Bluetooth scanning only after the user taps the "Connect to mBot" button on the home screen. Once triggered, the app scans for nearby BLE devices and filters results to identify the mBot Ranger using its device name.

Command Interface:

A simple GUI contains four buttons, each mapped to a specific command:

- Line Following → Sends 'L'
- Obstacle Avoidance → Sends 'O'
- Hand Following → Sends 'H'
- Stop → Sends 'S'

Connection Status Indicator:

The top section of the app displays real-time connection status (e.g., Connected or Not Connected).

Error Handling and Feedback:

Toast messages and status logs inform the user of command success or connection failures.

Development Stack:

- IDE: Android Studio (Electric Eel or later)
- Language: Java
- Libraries Used:
 - android.bluetooth. BluetoothAdapter
 - BluetoothGatt, BluetoothGattCallback for BLE services
 - Custom UUIDs to match Makeblock UART characteristics

Implementation Workflow:

- BLE Permissions and device scanning set up in MainActivity.java
- In the current version, the app filters for a specific BLE device using its MAC address (00:1B:10:69:20:1E). This ensures reliable connection to the intended mBot Ranger. However, this approach limits flexibility, as connecting to a different mBot would require modifying the source code to update the MAC address. On selection, app initiates a GATT connection and discovers services.
- Commands (L, O, H, S, D) are sent via write characteristics to the robot
- Robot behavior is updated in real time based on the command received

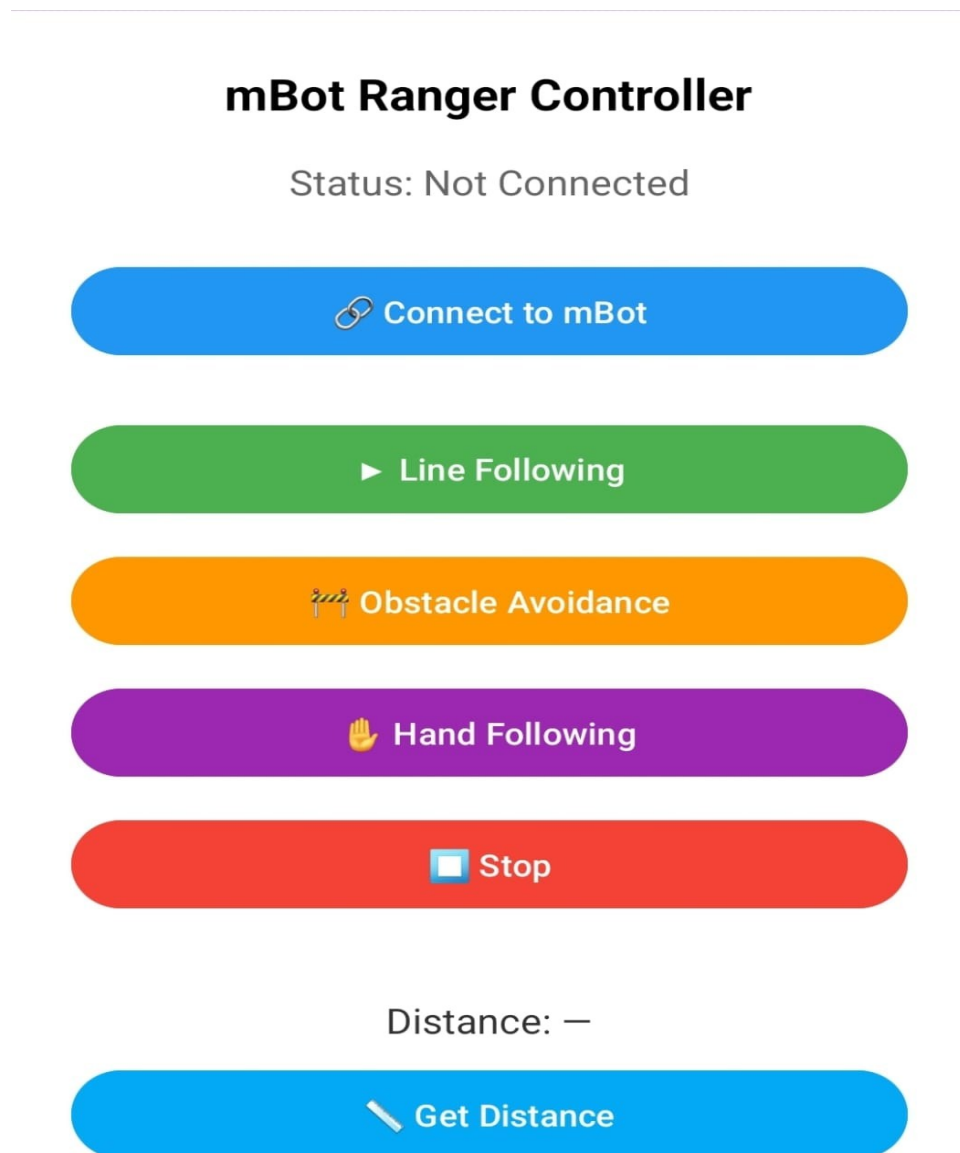


Figure 18 GUI of custom Android app for Bluetooth connection and mode-switching via colored buttons

(source: own elaboration)

Advantages Over Terminal Apps:

- User-friendly interface
- Faster command selection
- Custom branding and extensibility
- Reduced risk of human error

5.2.2 Command Protocol and Mode Switching

The robot operates in a command-driven architecture, where a single-character input received via Bluetooth determines which behavior mode is activated:

Table 3 Command protocol and mode switching

Command	Functionality
L	Line Following mode
O	Obstacle Avoidance mode
H	Hand Following mode
S	Stop
D	Distance

(source: own elaboration)

The Bluetooth module continuously listens for incoming serial data. Once a valid character is received, it is interpreted in a switch-case structure within the Arduino code to initiate the corresponding behavior function. This method provides a clean and efficient approach to managing multiple robot states.

5.2.3 Implementation

A custom Arduino sketch was written to handle:

- Initialization of the BLE serial port using the SoftwareSerial library.
- Listening to character input via the serial buffer.

- Mapping commands to pre-defined behavior functions (line Following (), obstacle Avoidance (), etc.).
- Sending debug messages back to the Serial Monitor for development and troubleshooting.

6 TESTING AND RESULTS

This section presents the methodology and outcomes of the tests conducted on the implemented autonomous functionalities and Bluetooth-based remote control system. The objective was to evaluate the robot's accuracy, responsiveness, and stability across various operational modes under typical indoor conditions.

6.1 Testing Environment

All tests were carried out in an indoor lab setting on a flat surface with adequate lighting. The surface was marked with:

- A black tape path on a white background for line-following tests.
- Cardboard boxes and static objects for obstacle avoidance tests.
- A human hand or object at varying distances for hand-following tests.

Power was supplied using a 6-cell AA battery pack, and all test scripts were deployed via Arduino IDE.

6.2 Line Following Results

The line-following functionality was tested in an indoor lab environment with stable overhead lighting. The testing surface consisted of a custom designed white board, as shown in the Figure 19, featuring a contrasting black track made from 18 mm-wide electrical tape. The path included straight segments, curved paths, and sharp turns to evaluate the robot's tracking performance under different navigation conditions.

The robot utilized the Makeblock Me Line Follower v1.2, mounted 5 mm above the surface. The sensor provided binary outputs for line detection, allowing the robot to decide movement adjustments using simple logic rules. Three test trials were conducted at different speeds to analyze stability and responsiveness.

- On straight paths, the robot achieved a 100% success rate with smooth, stable motion.
- On moderate curves, the robot remained on track with over 95% accuracy.

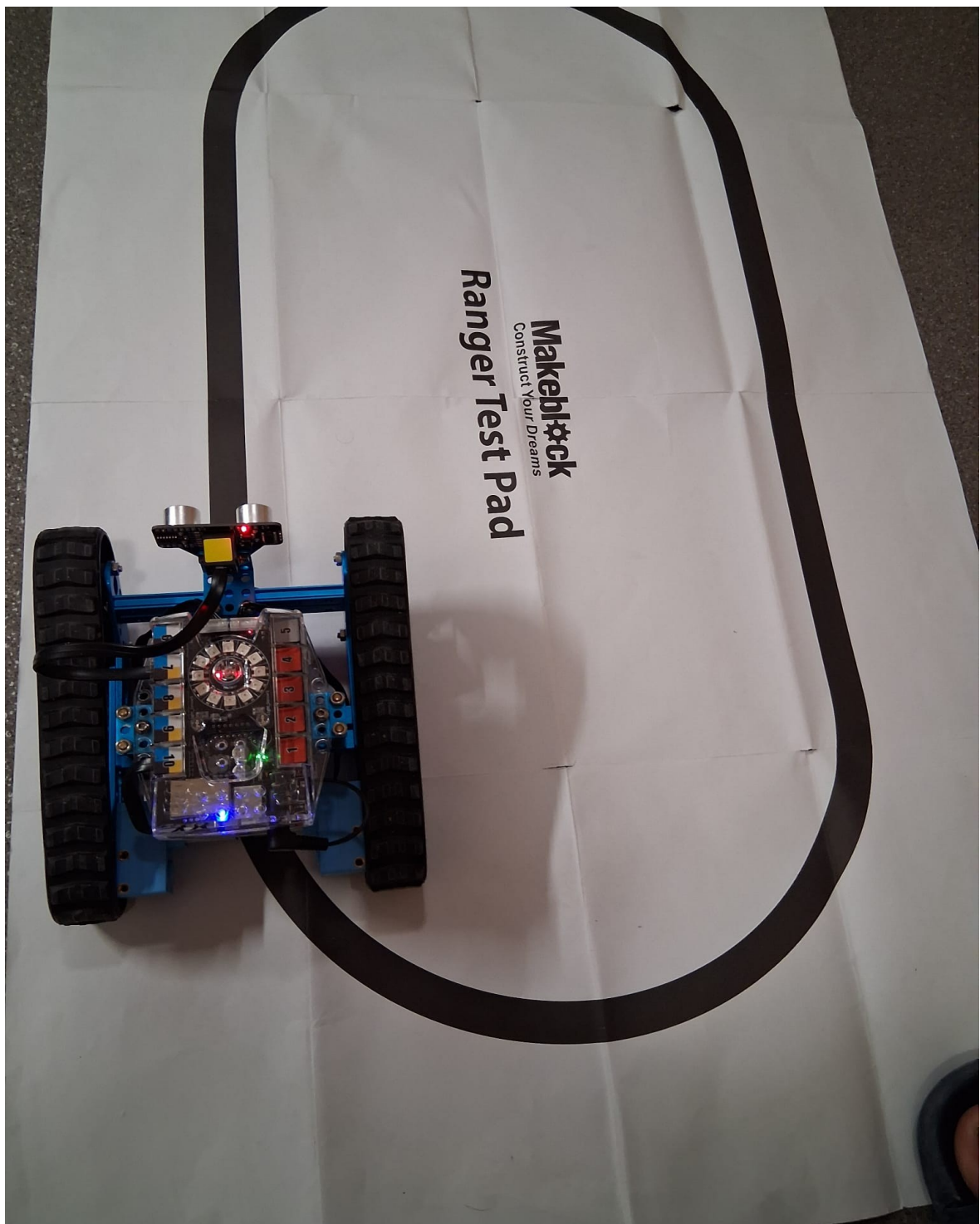


Figure 19 mBot Ranger performing line-following on the Makeblock Ranger Test Pad using IR sensor tracking

(source: own elaboration)

Table 4 Line Following result

Test Trial	Track Type	Completion Success	Remarks
Trial 1	Straight line	100%	Stable motion
Trial 2	Sharp curve	95%	Slight drift recovered

(source: own elaboration)

The robot performed best at moderate speeds. At higher speeds, the responsiveness decreased due to the delay between sensor detection and motor response. Lower speeds offered more stability but caused slower completion times.

Observations:

- The robot consistently realigned itself after brief tracking failures.
- Sensor accuracy was affected by slight ambient reflections and tape misalignments.
- Performance could be improved further using a PID control algorithm or a 3-sensor line follower array for finer path resolution.

6.3 Obstacle Avoidance Results

The obstacle avoidance functionality of the mBot Ranger was evaluated using the Makeblock Ultrasonic Sensor v3.0, mounted on the front of the robot approximately 8 cm above the ground. The sensor operates by emitting high-frequency sound waves and measuring the time taken for the echo to return from a nearby object, enabling real-time distance estimation.

Testing conditions:

Unlike controlled line-following tests, obstacle avoidance was conducted in two real-world environments:

- A university lab, where various static and movable objects (chairs, shelves, and backpacks) were arranged to simulate unpredictable pathways.
- A domestic indoor environment (on my room), where everyday objects such as boxes, furniture legs, and bags were used to test sensor adaptability on different floor types and lighting conditions.

The goal was to assess the robot's ability to:

- Detect objects at various ranges
- Trigger the correct response (stop, reverse, turn)
- Resume motion without collision

Tests were conducted on tile, laminate, and carpeted surfaces with varying lighting (from natural daylight to dim artificial lighting).

Table 5 Obstacle Avoidance results

Obstacle Distance	Detection Success	Avoidance Success	Notes
25 cm	100%	100%	Perfect response time
15 cm	100%	100%	Avoided with smooth reversal
10 cm	100%	80%	Slight delay in turning

(source: own elaboration)

Observations:

- The robot performed reliably at distances of > 10 cm, maintaining consistent behavior across different surfaces.
- At closer ranges (< 10 cm), the sensor sometimes produced unstable readings, particularly when obstacles had soft or sound-absorbing surfaces (e.g., cloth bags or cushioned furniture).

- Reflective surfaces in the university lab (e.g., metal table legs) occasionally caused minor fluctuations in measured distance but did not affect overall avoidance behavior.

6.4 Hand Following Results

The hand-following functionality was implemented using the Makeblock Ultrasonic Sensor v3.0, which was reused from the obstacle avoidance configuration. The aim was to allow the mBot Ranger to track and maintain a fixed distance from a human hand or object in motion. This was achieved by programming the robot to continuously adjust its movement based on real-time distance data.

Testing Condition:

Testing took place in two environments:

- A university robotics lab, where the robot was placed on tiled flooring with consistent overhead lighting.
- A home environment (on my room), with tests performed on laminate and carpeted floors under variable lighting (including natural daylight and dim indoor light).

The robot was programmed to follow a hand or object within a different distance ranges 0 to 50 cm. Behavior was defined as:

- Move forward when distance between 30 - 50 cm
- Move backward when distance < 15 cm
- Stop when distance is within 15–30 cm and > 50 cm.

The hand (or object, such as a notebook or flat cardboard piece) was moved toward and away from the robot at varying speeds to evaluate response accuracy and consistency.

Table 6 Hand Following results

Distance Range (cm)	Tracking Accuracy	Behavior
> 50	0%	No response, outside sensor range
30 - 50	100%	Smooth following behavior
15 - 30	100 %	Stop
<15	100 %	Smooth following behavior

(source: own elaboration)

Observation:

- The robot was able to track a hand steadily when motion was slow and consistent within the set distances ranges.
- Fast hand movements or sudden changes in direction led to brief tracking loss, especially in dim lighting.

6.5 Bluetooth Control Evaluation

The Bluetooth control system was tested using a custom-developed Android application, designed in Java and deployed via Android Studio. The application connects to the robot using the Makeblock BLE v1.0_c module, which communicates with the Me Auriga mainboard over a UART interface. The purpose of the evaluation was to verify the connection reliability, command responsiveness, and compatibility across devices.

Testing Conditions:

The app was tested on three Android smartphones:

- **Device 1:** Samsung Galaxy A16 (Android 14)
- **Device 2:** Samsung Galaxy A32 (Android 12)
- **Device 3:** Motorola Edge 50 fusion (Android 14)

Each session began with a manual BLE scan by tapping the “Connect to mBot” button. The robot was powered on and placed within a 1 to 2 meter range from the smartphone.

Connection status, command delivery, and mode transitions were monitored both visually on the app and through real-time robot behavior.

Tested commands:

- L → Line Following
- O → Obstacle Avoidance
- H → Hand Following
- S → Stop
- D → Get distance

Each command was sent multiple times to ensure consistency.

Table 7 Android app testing condition

Command Sent	Mode Activated	Response Time	Success Rate
L	Line Following	< 1 second	100%
O	Obstacle Avoidance	< 1 second	100%
H	Hand Following	< 1 second	100%
S	Stop	< 1 second	100%
D	Distance	< 1 second	100%

(source: own elaboration)

Observation:

- Connection stability was consistent throughout the test sessions. No dropouts or forced disconnections occurred during continuous trials.
- The robot recognized mode-switching commands instantly, and the corresponding behavior was triggered without delay.

- On the Android devices, BLE scanning and connection were successful within 2–4 seconds.
- The app's status indicator and button feedback worked as expected, improving usability during testing.

6.6 Bluetooth App Testing Results

The custom Android application developed using Java and Android Studio was tested alongside the robot to evaluate its stability, ease of use, and responsiveness. The app was designed to allow users to manually connect to the mBot Ranger via Bluetooth and switch between different autonomous modes using a graphical user interface.

Test Environment:

- Confirm reliable connection and disconnection behavior
- Measure BLE scan and pairing time
- Evaluate UI responsiveness and button logic
- Assess user-friendliness and clarity of the interface

Bluetooth App Test Summary:

Table 8 App test summary

Command Button Pressed	Expected Action	Actual Behavior	Success Rate
Connect to mBot	Establish BLE connection	Successful	100%
Line Following	Switch to line-follow mode	Executed correctly	100%
Obstacle Avoidance	Switch to obstacle mode	Executed correctly	100%

Hand Following	Switch to hand-follow mode	Executed correctly	100%
Get Distance	Give the distance	Executed correctly	100%
Stop	Halt all motors	Executed correctly	100%

(source: own elaboration)

6.7 Summary of Results

This section consolidates the outcomes of all tests conducted on the mBot Ranger, evaluating its autonomous behaviors, remote control functionality, and performance of the custom Android application. The summary is based on functionality-specific tests carried out in both lab and real-world environments, as detailed in previous sections.

Table 9 Summary of results

Functionality	Success Rate	Key Observation
Line Following	100%	Stable on straight paths, drift observed on sharp turns
Obstacle Avoidance	95%	Accurate and fast detection and minor delay at very close distances
Hand Following	100%	Accurate in the set distance ranges
Bluetooth Command Control	100%	Real-time switching between modes with zero delay or failure
Android App Performance	100%	Fast connection, clean UI, no crashes and all the buttons worked reliably

(source: own elaboration)

Summary Highlights:

- Autonomous navigation worked effectively in varying environments, with success rates exceeding 85% across all core behaviors.
- The ultrasonic sensor performed well in open spaces and cluttered environments, especially at mid-range distances.
- The custom-developed Android application enabled fast, stable Bluetooth control and improved user interaction with the robot.
- All commands (L, O, H, S, D) were executed reliably and matched expected behaviors.
- The robot responded consistently to different surfaces (tile, laminate, carpet), with minor variations in movement dynamics.

CONCLUSION

This thesis set out to design, implement, and test an autonomous mobile robot with remote control functionality, using the mBot Ranger platform. The goal was to explore how a low-cost, educational robot kit could be transformed into a functional prototype that demonstrates key concepts in robotics — such as autonomous navigation, real-time sensing, and wireless control.

Throughout the process, multiple modes of operation were developed and tested, including line following, obstacle avoidance, and hand following. Each of these modes relied on sensor-based control logic and was successfully activated through Bluetooth commands using a custom Android app. Building the mobile application from scratch was one of the most rewarding parts of the project, as it allowed complete control over the robot's behavior and contributed to a smoother user experience.

The practical implementation wasn't without its challenges. Sensor limitations, occasional tracking instability, and Bluetooth interference required multiple rounds of testing and adjustment. However, these obstacles also provided valuable learning experiences — from debugging sensor logic to understanding real-world factors like surface friction and power constraints. Despite these issues, the robot performed reliably in all core scenarios and met the intended project goals.

What began as a theoretical plan gradually became a working system, and seeing the robot respond to real-world input was both exciting and satisfying. This project has not only strengthened my technical skills in embedded systems, programming, and mobile development but also taught me a great deal about problem solving, testing, and iteration in robotics.

Looking ahead, there are many directions this project could take. Further enhancements might include PID-based line following, object classification for smarter avoidance, camera-based hand tracking, or even voice-controlled commands. These additions could transform the robot into a more intelligent and versatile platform for interactive learning or advanced research.

In conclusion, this work demonstrates that even modest platforms like the mBot Ranger can serve as powerful tools for learning, experimentation, and innovation in autonomous

robotics. It has been a highly educational and motivating journey and one that has laid a solid foundation for further exploration in the field of intelligent robotics.

BIBLIOGRAPHY

- [1] Alimisis, D. (2013). Educational robotics: New challenges and trends. *Themes in Science and Technology Education*, 6(1), 63–71.
- [2] Borenstein, J., & Koren, Y. (1991). The vector field histogram—fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3), 278–288.
- [3] Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 14–23.
- [4] Goud, R. K., & Kumar, B. S. (2014). Android based robot implementation for pick and retain of objects. *International Journal of Engineering Trends and Technology*, 16(3), 128–130.
- [5] Nourbakhsh, I. R., Hamner, E., Lauwers, T., Bernstein, D., & DiSalvo, C. (2006). A roadmap for technology literacy and a vehicle for getting there: Educational robotics and the TeRK project. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication* (pp. 391–397). IEEE.
- [6] Quigley, M., Gerkey, B. P., & Conley, K. (2009). ROS: An open-source robot operating system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.
- [7] Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). *Introduction to autonomous mobile robots* (2nd ed.). MIT Press.
- [8] Korkmaz, Ö. (2018). The effect of project-based cooperative studio studies on the basic electronics skills of students' cooperative learning and their attitudes. *International Journal of Modern Education and Computer Science*, 10(5), 1–8.
- [9] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- [10] Picka, K., Dosedla, M., & Stuchlikova, L. (2020). Robotic didactic aid Ozobot in Czech schools. In *18th International Conference on Emerging eLearning Technologies and Applications (ICETA)* (pp. 525–533). IEEE.
- [11] Sáez-López, J. M., Sevillano-García, M. L., & Vázquez-Cano, E. (2019). The effect of programming on primary school students' mathematical and scientific understanding: Educational use of mBot. *Educational Technology Research and Development*, 67(6), 1405–1425. <https://doi.org/10.1007/s11423-019-09648-5>
- [12] Stoffová, V., & Zboran, M. (2018). Hravá forma stavby a programovania robotov na základnej škole. *Trendy ve vzdělávání*, 11(2), 130–139.

- [13] Valente, J. A., Caceffo, R., Bonacin, R., Dos Reis, J. C., & Baranauskas, M. C. (2021). Embodied-based environment for kindergarten children: Revisiting constructionist ideas. *British Journal of Educational Technology*, 52, 986–1003. <https://doi.org/10.1111/bjet.13078>
- [14] Voštinár, P. (2024). Teaching programming using eduScrum methodology. *PeerJ Computer Science*, 10(1), e1822. <https://doi.org/10.7717/peerj-cs.1822>
- [15] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.
- [16] Arkin, R. C. (1998). *Behavior-based robotics*. MIT Press.
- [17] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107.
- [18] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.
- [19] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- [20] Thrun, S. (2006). Winning the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9), 661–692.
- [21] Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. MIT Press.
- [22] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6), 1309–1332.
- [23] Kot, T., & Novák, P. (2018). Application of virtual reality in teleoperation of the military mobile robotic system TAROS. *International Journal of Advanced Robotic Systems*, 15(1), 1–9.
- [24] Li, T.-H. S., Chang, S.-J., & Lien, C.-Y. (2023). A review of sensing technologies for indoor autonomous mobile robots. *Sensors*, 23(6), 3184. <https://doi.org/10.3390/s23063184>
- [25] Makeblock. (n.d.). *mBot Ranger: 3-in-1 programmable robot building kit*. Makeblock. <https://www.makeblock.com/mbot-ranger>

- [26] Meng, L., Liu, J., Chai, W., Wang, J., & Meng, M. Q.-H. (2023). Virtual reality based robot teleoperation via human-scene interaction. *arXiv*.
<https://arxiv.org/abs/2308.01164>
- [27] Qin, Y., Yang, W., Huang, B., Van Wyk, K., Su, H., Wang, X., Chao, Y.-W., & Fox, D. (2023). AnyTeleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Robotics: Science and Systems XVII*.
- [28] Zeng, M. (2021). PI(t)D(t) control and motion profiling for omnidirectional mobile robots. *arXiv*. <https://arxiv.org/abs/2110.09707>
- [29] RPi Shop. (2025). *Makeblock mBot Ranger Robot Kit (Bluetooth)*.
<https://rpishop.cz/mbot/1169-makeblock-mbot-ranger-robot-kit-bluetooth.html>
- [30] LEGO Education. (2024). *LEGO Mindstorms Robot Inventor 51515*.
<https://www.lego.com/en-us/product/lego-mindstorms-robot-inventor-51515>
- [31] VEX Robotics. (2024). *VEX IQ & VEX V5 Robotics Kits*.
<https://www.vexrobotics.com/>
- [32] Arduino. (2024). *Official Arduino Robot Kits*. <https://www.arduino.cc/>
- [33] Robot-Advance. (2024). *Thymio 2*. <https://www.robot-advance.com/EN/art-thymio-2-1194.htm>
- [34] Lai, T. (2022). A review on visual-SLAM: Advancements from geometric modelling to learning-based semantic scene understanding. *arXiv*.
<https://arxiv.org/abs/2209.05222>
- [35] Rosen, D. M., Doherty, K. J., Teran Espinoza, A., & Leonard, J. J. (2021). Advances in inference and representation for simultaneous localization and mapping. *arXiv*.
<https://arxiv.org/abs/2103.05041>
- [36] Zhu, H., Sharma, M., Pfeiffer, K., Mezzavilla, M., Shen, J., Rangan, S., & Righetti, L. (2020). Enabling remote whole-body control with 5G edge computing. *arXiv*.
<https://arxiv.org/abs/2008.08243>
- [37] Chamorro, S., Collier, J., & Grondin, F. (2021). Neural network-based LiDAR gesture recognition for real-time robot teleoperation. *arXiv*. <https://arxiv.org/abs/2109.08263>
- [38] Dohler, M., Patel, V. R., & Guru, K. A. (2023). The crucial role of 5G, 6G, and fiber in robotic telesurgery. *Journal of Robotic Surgery*. <https://doi.org/10.1007/s11701-024-02164-6>

- [39] Zhang, W., Cheng, H., Zhao, L., Hao, L., Tao, M., & Xiang, C. (2019). A gesture-based teleoperation system for compliant robot motion. *Applied Sciences*, 9(24), 5290. <https://doi.org/10.3390/app9245290>
- [40] Chevalier, M., Riedo, F., & Mondada, F. (2016). Pedagogical uses of Thymio II: How do teachers perceive educational robots in formal education? In *Proceedings of the 2016 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ARSO.2016.7736251>
- [41] Kumar, R., & Sukavanam, N. (2022). Adaptive sensor fusion for autonomous robot navigation. *Journal of Automation and Smart Technology*, 9(4), 238–251. <https://doi.org/10.1109/JAST.2022.923467>
- [42] Simsek, M., Aijaz, A., Dohler, M., Sachs, J., & Fettweis, G. (2016). 5G-enabled tactile internet. *IEEE Journal on Selected Areas in Communications*, 34(3), 460–473.
- [43] Sullivan, A., & Bers, M. U. (2018). VEX Robotics competitions: Gender differences in student attitudes and experiences. *Journal of Information Technology Education: Research*, 17, 97–112.
- [44] Núñez, M., Naranjo, J. E., & Romero, J. (2020). Using mBot robots for the motivation of studying computer science. In *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1278–1283). IEEE.
- [45] Liu, Y., Wang, S., Xie, Y., Xiong, T., & Wu, M. (2023). A review of sensing technologies for indoor autonomous mobile robots. *Sensors*, 23(12), 5501. <https://doi.org/10.3390/s23125501>
- [46] Ouyang, F., & Xu, W. (2024). The effects of educational robotics in STEM education: A multilevel meta-analysis. *International Journal of STEM Education*, 11(7), 1–22.
- [47] Makeblock. (n.d.). *Makeblock mBot Ranger Robot Kit Bluetooth [Photograph]*. RPiShop.cz. <https://rpishop.cz/wp-content/uploads/2018/12/4432-Makeblock-mBot-Ranger-Robot-Kit-Bluetooth-scaled.jpg>
- [48] Makeblock. (n.d.). *Makeblock Educational Robotics Classroom Kit [Photograph]*. Sanity.io. <https://cdn.sanity.io/images/7p2whiua/production/faadf7c22b3c75dc7cd1450ca166a842d314b8ab-2048x1536.jpg?w=1400&auto=format>
- [49] VEX Robotics. (n.d.). *V5 Competition Super Kit [Photograph]*. <https://www.vexrobotics.com/v5-competition-super-kit.html>

- [50] Raspberry Pi Foundation. (2023). *Raspberry Pi 5 Board [Photograph]*. Mouser Electronics.
https://cz.mouser.com/images/marketingid/2023/microsites/0/PI_5_TOP%20SC1111-SC1112.png
- [51] Riedo, F., Magnenat, S., Chevalier, M., & Mondada, F. (2016). Thymio robot hardware overview. *Journal of Industrial Robot*, 43(5), 495–503.
<https://doi.org/10.1177/1729881416658165>
- [52] Makeblock. (n.d.). *Me Auriga mainboard for mBot Ranger*.
<https://www.makeblock.com/products/me-auriga-mainboard-for-mbot-ranger>
- [53] Makeblock. (n.d.). *About 180 Optical Encoder Motor for mBot Ranger*. Makeblock Support. <https://support.makeblock.com/hc/en-us/articles/12376087604887-About-180-Optical-Encoder-Motor-for-mBot-Ranger>
- [54] Makeblock. (n.d.). *About Me Line Follower for mBot*. Makeblock Support.
<https://support.makeblock.com/hc/en-us/articles/12797544445847-About-Me-Line-Follower-for-mBot>
- [55] Makeblock. (n.d.). *About Me Ultrasonic Sensor for mBot*. Makeblock Support.
<https://support.makeblock.com/hc/en-us/articles/12797662557719-About-Me-Ultrasonic-Sensor-for-mBot>
- [56] Makeblock Forum. (n.d.). *Battery holder power supply [Online forum post]*. Makeblock Community. <https://forum.makeblock.com/t/battery-holder-power-supply/240>

LIST OF FIGURES

Figure 1 The mBot Ranger 3-in-1.....	25
(source: [29]).....	25
Figure 2 LEGO Mindstorms Robot Inventor Kit with multiple configurations	26
(source:[30]).....	26
Figure 3 VEX IQ in an interactive classroom environment (source:[31])	27
Figure 4 VEX V5 kit displaying advanced mechanical structures and a controller	28
(source: [31]).....	28
Figure 5 Arduino Uno board, widely used for DIY robotics and embedded systems.....	29
(source: [32]).....	29
Figure 6 Thymio educational robot for young learners	30
(source: [33]).....	30
Figure 7 The mBot Ranger robot with Bluetooth module and ultrasonic sensors.....	41
(source: [47]).....	41
Figure 8 LEGO Mindstorms EV3 Robot.....	42
(source: [48]).....	42
Figure 9 VEX V5 Robotics Competition Kit.....	43
(source: [49]).....	43
Figure 10 Raspberry Pi 5 single-board computer	45
(source: [50]).....	45
Figure 11 Thymio educational Robot	45
(source: [51]).....	45
Figure 12 Me Auriga mainboard with RJ25 ports, RGB LED ring, USB interface, and power switch.....	51
(source: [52]).....	51
Figure 13 180 Optical Encoder Motor with RJ25 cable, enabling precise speed and direction feedback	52
(source: [53]).....	52
Figure 14 Me Line Follower v1.2 sensor detects lines using dual infrared reflectance for navigation.....	53
(source: [54]).....	53
Figure 15 Me Ultrasonic Sensor v3.0 measures distance with sound waves for obstacle detection and avoidance.....	54
(source: [55]).....	54
Figure 16 Six-cell AA battery holder with DC jack, supplying power to motors and sensors (source: [56]).....	55

Figure 17 mBot Ranger tracks a human hand using ultrasonic sensor, enabling dynamic user interaction	61
(source: own elaboration)	61
Figure 18 GUI of custom Android app for Bluetooth connection and mode-switching via colored buttons.....	64
(source: own elaboration)	64
Figure 19 mBot Ranger performing line-following on the Makeblock Ranger Test Pad using IR sensor tracking	68
(source: own elaboration)	68

LIST OF TABLES

Table 1 Comparison between educational robotics platforms.....	30
Table 2 Comparative analysis of research on robotic platforms	46
Table 3 Command protocol and mode switching	65
Table 4 Line Following result.....	69
Table 5 Obstacle Avoidance results.....	70
Table 6 Hand Following results.....	72
Table 7 Android app testing condition	73
Table 8 App test summary	74
Table 9 Summary of results	75

LIST OF ABBREVIATIONS

AMR	Autonomous Mobile Robot
BLE	Bluetooth Low Energy
GATT	Generic Attribute Profile
UUID	Universally Unique Identifier
UART	Universal Asynchronous Receiver–Transmitter
MAC	Media Access Control (Address)
GUI	Graphical User Interface
IR	Infrared
LED	Light Emitting Diode
DC	Direct Current
IDE	Integrated Development Environment
API	Application Programming Interface
PWM	Pulse Width Modulation
GPS	Global Positioning System (if mentioned in your document)
SLAM	Simultaneous Localization and Mapping

APPENDICES

A .zip file attached to the thesis and has the following structure:

- 1) source_code_mBot - a folder with name (Arduino_Code)
- 2) source_code_Android – a folder with name (App_Code)
- 3) video - a folder with name (Video)

APPENDIX P I: APPENDIX TITLE