

Design and Development of a Low-Cost Obstacle-Avoiding Robot Car Using Arduino

Intelligent Systems with Robots Kyrylo Stepanets

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:	Kyrylo Stepanets
Personal number:	A22500
Study programme:	B0714A150007 Applied Informatics in Industrial Automation
Specialization:	Intelligent Systems with Robots
Type of Study:	Full-time
Work topic:	Návrh a vývoj levného robotického auta s vyhýbáním se překážkám pomocí Arduino
Work topic in English:	Design and Development of a Low-Cost Obstacle-Avoiding Robot Car Using Arduino

Theses guidelines

1. Research the topics on obstacle-avoiding robots, and Arduino projects
2. Design the robot car and sensor system
3. Build designed the robot, and create software
4. Test the robot's performance
5. Write the conclusion and summarize the thesis

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

Recommended resources:

1. BLUM, Jeremy Exploring Arduino: Tools and Techniques for Engineering Wizardry. Indianapolis: Wiley, 2013. ISBN 978-1-118-54936-0.
2. MCMANUS, John Arduino Robotics. New York: Apress, 2011. ISBN 978-1-4302-3184-8.
3. FAROUGH, Mathew Mastering Arduino: A Project-Based Approach. Birmingham: Packt Publishing, 2020. ISBN 978-1839214817.
4. SICILIANO, Bruno Robotics: Modelling, Planning, and Control. London: Springer, 2010. ISBN 978-1-84628-641-4.
5. HASSAN, Syed Omar Faruk Towaha Arduino Made Simple. Birmingham: Packt Publishing, 2020. ISBN 978-1838826035.

Supervisors of bachelor thesis: **Ing. Václav Mach, 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 bakalářská práce se zaměřuje na návrh a vývoj nízkonákladového robotického vozidla, které je schopné autonomně se vyhýbat překážkám. Systém je řízen mikrokontrolérem Arduino a využívá ultrazvukové senzory k detekci okolních objektů a provádění rozhodnutí o navigaci v reálném čase. Cílem je vytvořit funkční, spolehlivý a cenově dostupný robot, který se dokáže pohybovat ve svém prostředí bez lidského zásahu, což jej činí vhodným pro různé praktické aplikace.

Klíčová slova: Autonomní robot, Vyhýbání se překážkám, Arduino, Ultrazvukové senzory, Navigace v reálném čase, Nízkonákladové robotické vozidlo, Systém založený na mikrokontroléru, Autonomní navigace.

ABSTRACT

This thesis focuses on the design and development of a low-cost robotic vehicle capable of autonomously avoiding obstacles. The system is controlled by an Arduino microcontroller and utilizes ultrasonic sensors to detect nearby objects and make real-time navigation decisions. The aim is to create a functional, reliable, and affordable robot that can navigate its environment without human intervention, making it suitable for various practical applications.

Keywords: Autonomous robot, Obstacle avoidance, Arduino, Ultrasonic sensors, Real-time navigation, Low-cost robotic vehicle, Microcontroller-based system, Autonomous navigation.

ACKNOWLEDGEMENTS

Acknowledgements, motto and a declaration of honour saying that the print version of the Bachelor's/Master's thesis and the electronic version of the thesis deposited in the IS/STAG system are identical, worded as follows:

I hereby declare that the print version of my Bachelor's/Master's thesis and the electronic version of my thesis deposited in the IS/STAG system are identical.

CONTENTS

INTRODUCTION	10
I THEORETICAL PART	12
1. FUNDAMENTALS OF AUTONOMOUS ROBOTICS	13
1.1 OVERVIEW OF OBSTACLE-AVOIDING ROBOTS	13
1.2 LOW-COST ROBOTIC SYSTEMS.....	14
1.3 ARDUINO-BASED ROBOTICS PROJECTS	14
1.3.1 Benefits of Arduino in Robotics.....	15
1.3.2 Common Challenges in Arduino Projects	15
1.3.3 Comparison with Other Microcontrollers	15
2. SENSOR TECHNOLOGIES AND NAVIGATION METHODS	16
2.1 ULTRASONIC SENSORS	16
2.2 INFRARED SENSORS.....	17
2.3 NAVIGATION ALGORITHMS AND TECHNIQUES	17
3. MOTORS AND MOTOR DRIVERS IN ROBOTICS.....	18
3.1 TYPES OF MOTORS USED IN ROBOTICS.....	18
3.2 MOTOR DRIVERS AND THEIR FUNCTIONALITY	19
3.3 INTEGRATION OF MOTORS AND DRIVERS IN THE ROBOT	19
3.4 CHALLENGES IN MOTOR CONTROL	20
3.5 IMPORTANCE OF MOTOR AND DRIVER SELECTION IN LOW-COST ROBOTICS	21
4. THE ARDUINO PLATFORM.....	22
4.1 KEY FEATURES OF ARDUINO.....	22
4.2 APPLICATION OF ARDUINO IN ROBOTICS.....	22
4.3 LIMITATIONS OF ARDUINO IN ROBOTICS	23
4.4 FUTURE OF ARDUINO IN ROBOTICS	23
II. PRACTICAL PART	24
5. SYSTEM DESIGN	25
5.1 ROBOT CAR ARCHITECTURE	25
5.1.1 Mechanical Design	25
5.1.2 Electrical Design	25
5.2 COMPONENT SELECTION	26
5.2.1 Motors and Motor Drivers.....	26
5.2.2 Ultrasonic Sensors and Tower Pro 9G Micro Servo Motor	28
5.2.3 Arduino Microcontroller	30
5.2.4 Battery and Battery Holder.....	31
5.3 CIRCUIT DESIGN.....	32

6.	IMPLEMENTATION.....	34
6.1	ASSEMBLY OF THE ROBOT	34
6.2	ARDUINO PROGRAMMING	36
6.2.1	Control Algorithm.....	36
6.2.2	Integration of Sensor Data	37
6.2.3	Error handling and debugging.....	38
6.3	INTEGRATION OF HARDWARE AND SOFTWARE.....	39
7.	TESTING AND EVALUATION	41
7.1	PERFORMANCE TESTING METHODOLOGY	41
7.1.1	Setup of Testing Environment	41
7.1.2	Criteria for Performance Evaluation	42
7.2	OBSTACLE AVOIDANCE ACCURACY.....	43
7.2.1	Sensor Detection Accuracy	43
7.2.2	Navigation Success Rate	44
7.3	LIMITATIONS AND IMPROVEMENTS	46
7.3.1	Identified Limitations.....	46
7.3.2	Proposed Improvements.....	47
	CONCLUSION	49
	BIBLIOGRAPHY.....	51
	LIST OF FIGURES	52
	LIST OF GRAPHS	53
	LIST OF ABBREVIATIONS	54
	APPENDICES.....	55

INTRODUCTION

The field of robotics has seen a lot of development, offering solutions in many places such as industrious automation, medical care, and humans' service. Among them, autonomous mobile robots have been a field of high concern. Autonomous mobile robots can move and act in an environment with no intervention of humans and have a lot of opportunity in a variety of applications. In an endeavor to produce a working and economically feasible robotic car with an ability to move in a moving motion with no intervention at all, and navigate over obstacles, a thesis titled "Design and Development of a Low-Cost Obstacle-Avoiding Robot Car Using Arduino" seeks to build a robotic car at a low cost and with no intervention at all, with an Arduino board and an ultrasonic sensor, a simple yet efficient answer for a scenario in cases when navigation and obstruction evasion is paramount.

The motivation for such work comes in view of a growing demand for simple and low-cost robotic platforms in educational, residential, and commercial settings. High-performance robotic platforms become too costly in price and out of everyone's budget for students, instructors, and enthusiasts, and a demand is generated for an alternative with similar performance and dependability with a lesser price tag.

Autonomous robots have been a backbone for cutting-edge robotic technology, combining sensor technology, algorithm controls, and real-time decision processes. Obstacle-avoiding robots, in particular, have emerged out of pioneer experiments with reaction-dependent software in mimics of human behavior in reaction to environment stimuli. With microcontrollers such as Arduino, free and accessible, becoming increasingly widespread, robotic technology is in reach for professionals and enthusiasts alike. Arduino, as described in [1], is a general-purpose platform for bringing theoretical theory to reality. Similarly, the author in [2] identifies Arduino's efficiency in robotic guidance and use of sensors, positioning Arduino ideally for use in this thesis work. With an integration of ultrasonic sensors, robotic capabilities of the car allow for it to interpret its environment with accuracy and dependability.

Technology in terms of robots continues developing at an exponential pace, yet cheap, accessible robotic technology is in short demand, most prominently for use in schools and for small companies. There is a high demand in providing cheap robotic technology accessible for students, professionals, and industries in general. Most Arduino-related work uploaded in cyberspace is not in usability and refinement for use in practice, and alternative low-cost examples have poor accuracy in sensors and inefficient integration, with poor guidance

and decision processes, therefore, for obstacles in its path. In a move towards challenging such, this thesis aims at developing a robotic car, cheap yet capable of accurately and dependably distinguishing and avoiding obstacles.

The primary objectives of the thesis include researching and comparing state-of-the-art obstacle-avoiding robotic systems, developing and building a working and cheap robot car, and developing a working model for obstacle detection and avoidance with ultrasonic sensors. Performance of the system will be gauged in terms of accuracy in navigation, dependability, and usability overall. In conclusion, a full blueprint for developing low-cost robotic systems possessing real usability, availability, and manufacturability will be developed.

To carry out these objectives, the project will entail putting together relatively accessible materials for the hardware, including an Arduino Uno microcontroller, Motor Driver Shield L293D, TT Gear Motors with wheels, Tower Pro 9G Micro Servo Motor, Tower Sensor Holder, HC-SR04 ultrasonic sensors, and 18650 Li-ion batteries. Software development will entail developing algorithms through Arduino IDE for processing values in sensors and controlling motion in motors in real-time. Testing will be localized indoors with bounded and plane obstacles, with an opportunity for testing in closely monitored environments with an assurance that the robot can move under its guidance under determined environments. Simultaneous localization and mapping (SLAM), decision-making via machine learning, and communications via wireless, respectively, represent capabilities not within the objectives of this thesis.

By putting together these parts and methodologies, will present that this project will present for low-cost robotic systems a level of viability in developing reliable robotic systems and make a contribution in an increasingly growing field of robotics through a workable and accessible solution for educational and small-packet use, instilling an impetus for continued development and innovation for low-cost technology in autonomous use.

I THEORETICAL PART

1. FUNDAMENTALS OF AUTONOMOUS ROBOTICS

The development of a robotic device capable of overcoming obstacles involves a fusion of sound robotic theory, low-cost engineering, and high accuracy in terms of programming. Creation of such a device involves a blend of fundamental robotic principles, cutting-edge technology tools, and problem-solving methodologies. Quick technological advancement in microcontrollers, together with availability and low-cost platforms such as Arduino, have helped researchers and enthusiasts develop and use efficient autonomous systems at an affordable price in comparison with commercially produced options. This section creates a theoretical background for thesis development, describing key terms and methodologies for creating an effective, low-cost robot.

The creation of robots sees a challenge in a union between efficient sensor capabilities and algorithms with real-time decision capabilities. Interaction between an assortment of gear such as ultrasonic and IR sensors and software algorithms such as path following and PID controllers defines efficient barrier-evading robots. According to the author in [4], success with autonomous robot's hinges on a union between hardware and algorithms, making them efficient in real-life settings.

1.1 Overview of Obstacle-Avoiding Robots

Obstacle-avoiding robots form a general family of autonomous machines moving in an unpopulated environment. Obstacle-aware robots make use of actuation, sensors, and algorithm decision-making for direction switch in the presence of obstacles. In *Robotics: Modelling, Planning, and Control*, the author in [4] identifies an important aspect of reactive navigation, in which real-time feedback from sensors is utilized for decision-making in robots.

The purpose of such a robot is to maximize efficiency in performance and safety, and its use can extend to industries such as logistics and medical care. Initial experiments in reactive navigation have provided a basis for current systems, and with increased microcontroller capabilities and integration with sensors, such a robot is easier to implement.

1.2 Low-Cost Robotic Systems

The affordability of robotic platforms in its use is an important consideration, particularly in educational and research environments. Most current options available are too costly or not functional at a budget price level. As described in [2], democratization of robotics via platforms such as Arduino has opened a platform for developing useful cheap alternatives at no compromise of dependability.

The low-cost robotic platforms value simplicity and frugality at no expense of such criticals such as autonomous navigation, sensor accuracy, and integration of hardware for function. The platforms suit hackers, students, and small companies, and enable them to work and use robotics at no budget restriction of high-end platforms.

1.3 Arduino-Based Robotics Projects

The arrival of Arduino changed character of robots in terms of providing an open-source and friendly board for developing embedded platforms. Affordability, versatility, and simplicity make it a best bet for instructors, enthusiasts, and engineers. As fittingly captured in [1], Arduino can work effectively in balancing theoretical practice with its practice, particularly with regard to robotics. Inbuilt modularity in the platform, inbuilt in its platform, simplifies integration of numerous modules such as servos, shields, and sensors, therefore, it enables developers to develop sophisticated structures with ease.

One Arduino platform potentials is a guaranteed backing community. There is a huge environment of Arduino platform with a plethora of supporting tools such as examples, libraries, and tutorials, and with numerous documents, and enables rapid prototyping of robotic structures. According to [3], Arduino is high in adaptability, and it can refer to a trustful platform for both novice and expert structures, for examples, in cases such as obstacle-avoiding robots. Arduino enables real-time processing of sensors, correct controlling of motors, and perfect integration of both software and firmware, and, therefore, a useful tool for low-cost development of platforms for automation.

Arduino operations, nevertheless, have not been problem less. Despite ease in most development processes facilitated through its platform, sensor calibration, controlling powers, and integration of both firmware and software can present significant impediments. As described in [5], can, nevertheless, be overcome through proper planning and iterative

testing for ultimate performance. Regardless of such impediments, Arduino ease of use and trustfulness is at the epicenter of today's robots.

1.3.1 Benefits of Arduino in Robotics

Arduino offers many beneficial aspects in robotic use:

- **Cost Effectiveness:** Arduino boards are not costly and have a low price, an ideal consideration in budget-conscious projects.
- **Ease of Use:** Arduino's ease of use and wealth of documentation make prototyping and debugging quick.
- **Modularity:** Arduino's compatibility with a range of sensors, motors, and shields make Arduino an adaptable form and function platform for its designs.

Following on, according to [3], such an advantage makes Arduino a dependable tool for the development of obstacle-avoiding robots and general autonomous machines alike.

1.3.2 Common Challenges in Arduino Projects

Despite its functionality, Arduino projects have a few weaknesses, such as:

- **Sensor Calibration:** Acquiring correct sensor readings can become a problem in environments that change.
- **Power Handling:** Having enough power for motors, sensors, and microcontroller can become a critical problem.
- **Integration Problems:** Inadequate integration between software and hardware can cause loss of efficiency and system failure.

Adopting careful planning and testing is important in order to function appropriately

1.3.3 Comparison with Other Microcontrollers

Arduino stands out for its ease and affordability over options such as Raspberry Pi and ESP32. Despite the fact that Raspberry Pi supports additional computation and multitasking, Arduino is best for single-task, real-time operations such as obstacle avoidance. As described in [5], Arduino's capabilities in real-time make it an ideal platform for such operations with high accuracy and reduced latency.

2. SENSOR TECHNOLOGIES AND NAVIGATION METHODS

Sensor technologies contribute a lot towards obstruction-avoiding robots, and through them, such robots can actually perceive and respond to obstacles in an environment. One of the most prevalent sensors in use is through use of an ultrasonic sensor, such as an HC-SR04. Ultrasound sensors function through emittance of sound waves and taking a reading in terms of elapsed times for waves reflection off obstacles, with accuracy in calculation of distance. In Arduino Robotics, the author in [2] documents that ultrasonic sensors present an efficient and cost-effective alternative for use in low-cost robots for obstruction evasion.

Infrared (IR) sensors present an alternative for use in near range through the emission and testing of emitted and reflected infrared radiation. IR sensors, due to their affordability, have a drawback when exposed to surrounding lights and reflective materials, making them, in most cases, less reliable compared to ultrasonic sensors [1]. Navigation algorithms interpret feedback through sensors and calculate the motion of a robotic system. According to [4], reactive navigation utilizes real-time adaptations to environmental stimuli and programmed actions through rule-based systems, driven by sensor feedback. PID (Proportional-Integral-Derivative) control provides smooth and precise motion, improving overall accuracy in terms of orientation, position, and distance.

The synergy between efficient technology in sensors and optimized algorithms in a robotic system sees a reliable robotic performance even under changing environments. All these factors harmonize with one another, and such harmony between software and hardware is paramount in attaining reliable obstruction evasion in a robotic system.

2.1 Ultrasonic Sensors

Ultrasonic sensors, including the HC-SR04, are commonly used in robots for distance measurement. These sensors function by emitting ultrasonic waves and calculating the time taken for the waves to collide with an object and echo back. As documented in [2], ultrasonic sensors are effective and affordable, making them ideal for cost-effective robots.

2.2 Infrared Sensors

Infrared (IR) sensors present a competing alternative for near field obstacles to ultrasonic sensors. IR sensors function through a mechanism in which infrared radiation is emitted and subsequently examined through echo reflection. IR sensors, despite costing less, suffer a larger vulnerability to interfering factors in an environment, such as reflective surfaces and ambient lighting, and therefore a less effective performance in many cases compared to ultrasonic sensors.

2.3 Navigation Algorithms and Techniques

Navigation algorithms interpret feedback from sensors and specify motion for a robot. Techniques include:

- Reactive Navigation: Real-time response to obstacles, with no preplanned path.
- Rule-Based Systems: Simple algorithms specifying actions under programmed stimuli in sensors (e.g., when an obstruction is in your path to your right, turn).
- PID Control: Proportional-Integral-Derivative control, smoothing and accuracy through feedback and modulating motor velocity.

Dependence on efficiency in navigation is partially a function of combining these algorithms with the robot's hardware for seamless and effective use, as described in [4].

3. MOTORS AND MOTOR DRIVERS IN ROBOTICS

Within the field of autonomous robots, motors and motor drivers form critical parts that enable both locomotion and direction, specifically in robots with capabilities for obstacle avoidance. Motors' principal role is to enable mechanical actuation for motion, and for that purpose, motor drivers act as go-between entities between microcontrollers and motors, providing electrical current in proper magnitudes and directions. Together, these parts allow for efficient velocity, direction, and torque management, all three being important for successful realization of obstacle avoidance in real-life settings.

3.1 Types of Motors Used in Robotics

The choice of motors for robotic use is determined by several factors, such as torque, velocity, efficiency, and specific motion requirements specific to a scenario. In this work, TT gear motors have been utilized in the obstacle-avoiding robotic platform for its simple mechanism, efficiency, and suitability for use with low-powered systems.

- TT Gear Motors and Wheel Set:

The motors feature a configuration combining low-voltage direct current (DC) motors with gearboxes, with an improvement in both torque and velocity in a regulated form, and therefore apt for use in small robotic platforms for mobilization. Smooth and uniform motion, even in rough environments and during sharp turns, is guaranteed with gear reduction technology. As discussed in [3], gear motors with a configuration for use with DC motors are common in use in robotic platforms for compatibility and ease of integration with driving motors.

- Tower Pro 9G Micro Servo Motor:

The servo motor is critical in controlling mounted ultrasonic sensors, designed to map and detect obstacles in an environment. Unlike direct current (DC) motors, servo motors have a specific configuration that values positional accuracy, a feature critical in efficient mapping and surveying of obstacles in an environment. Tower Pro 9G Micro Servo Motor plays a critical role in such an arrangement, allowing real-time scan of an environment through its use in rotating an ultrasonic sensor. By changing its orientation, its field of view is dynamically increased, and obstacles can then be detected in many angles and not a single direction. This feature maximizes efficiency in predicting and acting in anticipation of obstacles, enhancing real-time performance and accuracy in guiding. Positioning of a sensor is

achieved through pulse-width modulation (PWM) signals specifically designed for use in servo motors.

3.2 Motor Drivers and Their Functionality

Motors utilized in robotic platforms cannot be driven by microcontrollers, including Arduino Uno, directly, because of a high voltage and current output that a microcontroller cannot deliver. In such cases, use of a motor driver, in this case, an L293D motor driver shield, is needed.

- L293D Motor Driver Shield:

The L293D motor driver shield is a dual H-bridge driver, allowing two or more motors to run at the same time. It complements Arduino capabilities with direction and speed controls of motors through converting low current drive signals into high current output. With a dual H-bridge configuration, motors can run in reverse and forwards through changing polarity in a power source. As seen in [3], motor drivers in robotic applications serve to protect microcontrollers from high current loads and have flexible driving capabilities for motors.

- Operational Mechanism of the Motor Driver:

Arduino sends PWM signals to the motor driver shield, which modulates voltage output to the motors, effectively controlling velocity. Directional control is achieved through digital inputs that specify polarity of current flowing through the motors. For example, when Motor 1 and Motor 2 receive different PWM signals, the robot can implement differential steering, allowing for smooth turns and obstacle evasion.

3.3 Integration of Motors and Drivers in the Robot

The connection between drivers and motors is critical in providing autonomous capabilities for the robotic device. TT gear motors are wired with L293D, a motor driver, and mounted onto Arduino Uno. L293D takes commands from Arduino, received through sensors, and controls voltage delivered to motors. In addition, servo motor controlling the ultrasonic sensor is operated through Arduino with PWM signals.

Key Features of Motor System Integration:

- Differential Steering:

The robot utilizes differential steering via velocity modulation of both the left and right motors. For example, when Motor 1 and Motor 2 run at a high velocity compared to Motor

3 and Motor 4, then the robot turns to the right, and under its reverse, it turns to the left. Differential steering enables efficient and precise turns, a critical function in avoiding obstacles.

- Real-Time Control:

The Arduino controls distance values received through the use of an ultrasonic sensor and dynamically adjusts the velocity of motors to ensure efficient maneuverability for the robot in case of obstacles. In addition, servo motor, mounted onto a sensor holder, rotates an ultrasonic sensor in a manner that expands the perceptual field of a robot.

3.4 Challenges in Motor Control

While motors and respective drivers enable robotic locomotion, they have specific requirements that require careful consideration during both development and integration phases:

- Power Handling:

The motors require high power, and inefficient distribution can cause reduced performance, even system failure. Integration with 18650 Li-ion batteries and with the KUQOIY battery holder yields a reliable source of power, thus enhancing effective use of motors and respective parts.

- Motor Synchronization:

Careful maintenance of a harmonized and uniform state through proper coordination of all motors is critical for ensuring sustained stability and equilibrium in the system. To maintain efficient performance over long runs, proper testing and calibration processes must effectively modulate PWM signals.

- Heat and Overcurrent Protection:

Longer runs of motors can cause excessive temperature build-up, and in case of failure, cause irreparable damage to the motor driver. Overcurrent protection forms a part of protective capabilities of L293D, discussed in [3].

3.5 Importance of Motor and Driver Selection in Low-Cost Robotics

The selection of high-performance driving parts and motors is critical in driving the development of economically efficient robotic platforms with operational efficiency. With the use of TT gear motors, L293D driving shields, and Arduino Uno, a proper balancing between function, economy, and reproduction in robotic platforms can be achieved.

According to Farough [3], a key first step in developing reliable and economically effective robotic platforms is through the use of modular parts, whose ease in substitution and updating can make them efficient for long-term use. With a general-purpose device in form of L293D and a modular platform in form of Arduino, such a combination is an ideal selection for educational and experimental work.

4. THE ARDUINO PLATFORM

Arduino is an open-source electronics platform that is also referred to as friendly hardware and software, which aims to assist in the making of interactive projects in an effective manner. Its simplicity, inexpensiveness, and flexibility have made it highly popular in the field of robotics. The platform specializes in microcontroller boards like the Arduino Uno, Mega, and Nano, which offer an economical means of introducing sensors, actuators, and communication modules into robotic platforms.

4.1 Key Features of Arduino

- **Ease of Programming:** Arduino uses a simplified version of C++ in the Arduino IDE that is simple for novices to learn yet is robust enough to be used in complicated applications.
- **Open-Source Nature:** Both hardware and software are open-source, allowing developers to extend and modify Arduino functionality for specific use.
- **Comprehensive Community Support:** Large online community offers tutorials, code repositories, and technical support, making it easier for new users to learn.
- **Modularity and Compatibility:** Arduino boards are compatible with a large number of sensors, shields, and communication modules, making it easy to integrate into various robotics projects.

4.2 Application of Arduino in Robotics

Arduino becomes significant in robotics because it supports real-time processing, sensor compatibility, and functions related to motor control. Most robots that move through obstacles, like the robot in this thesis, use Arduino for one reason or another:

- **Sensor Integration:** Ultrasonic, infrared, and other sensors can be directly interfaced with Arduino for environmental perception.
- **Motor Control:** Thanks to motor driver shields like L293D, Arduino can control DC motors, stepper motors, and servos easily for robot autonomous motion.
- **Data Processing:** Arduino interprets sensor data and produces real-time navigation instructions based on algorithms pre-programmed.

4.3 Limitations of Arduino in Robotics

Despite its advantages, Arduino has some limitations:

- **Limited Processing Power:** The Arduino has less computational power compared to microcontrollers such as the Raspberry Pi or ESP32, and hence is less appropriate for more complex AI applications.
- **Limited Memory Capacity:** The RAM and memory available on Arduino boards are quite limited, which may be restrictive for long programs.
- **Lack of Multitasking:** Unlike microcomputers, Arduino cannot handle multiple processes simultaneously, which may require external modules for enhanced performance.

4.4 Future of Arduino in Robotics

As discussed in [3], Arduino remains a core tool for robotics education and prototyping. With the introduction of Wi-Fi, Bluetooth, and IoT capabilities in newer models, the possibilities of Arduino application in autonomous robots are also expanding. Advances in sensor integration, edge computing, and real-time control will also increase the application of Arduino in robots.

This chapter highlights the significance of Arduino as a valuable tool in the construction of obstacle-avoiding robots, stressing its capacity to aid development while being functional and inexpensive. As embedded systems evolve, it is envisioned that Arduino will continue to be a vital component in the advancement of intelligent robot systems.

II. PRACTICAL PART

5. SYSTEM DESIGN

The design of the robotic car with an obstacle-detection capability is a complex integration of mechanical and electrical processes harmonized with one another. It is constructed with cheap, readily accessible parts such as Arduino Uno, shields for motors, and sensors, with a perfect harmony between price and performance. In the following sections, a critical analysis of the robotic structure, selection of parts, and creation of circuit is included.

5.1 Robot Car Architecture

The architecture of the robot is such that it is optimized for efficient path planning and obstacle-avoidance through modular subsystems. The most important subsystems of the robot are mechanical support, obstacle detection through sensors, drive through motors, and real-time processing through Arduino.

5.1.1 Mechanical Design

The mechanical build is towards a strong yet lightweight platform with an intention to secure all parts in a safe position. Arduino, drive for a motor, motors, ultrasonic sensor, servo motor, and source of power are stored in a base platform. TT Gear Motors and wheel assembly are mounted over a base, offering motion in a direction of motion and rotation. Tower Pro 9G Servo Motor accommodates an ultrasonic sensor, offering a scan of obstacles in a path of a robot in a dynamic manner.

For offering a physical steadiness to a robot, parts are balanced over a platform, and jumper wires for flexible connectivity are utilized. Smooth motion with integrity in a structure is facilitated through such a build.

5.1.2 Electrical Design

The electrical section involves integration of microcontroller, actuators, sensors, and source of power. Arduino Uno is a processing unit, and motion of a robot can be controlled in terms of feedback via an ultrasonic sensor. Motor Driver Shield L293D is a bridge between Arduino and motors, and direct direction and velocity can be regulated via it.

Both Arduino and motors are powered via 18650 Li-ion batteries via a KUQOIY Battery Holder with a switch, and simple switch off and switch on of a robot can be facilitated via it. In circuits, male and female jumper wires make a bridge between Arduino and a motor driver and sensors.

5.2 Component Selection

The component selection for the robot car for obstacles aims at cost-effectiveness, dependability, and integration ease. All parts have a significant role in contributing to overall performance in the system. In this section, motors, sensors, microcontroller, and a power source for powering the robot's autonomous capabilities are discussed in detail.

5.2.1 Motors and Motor Drivers

- **TT Gear Motors and Wheels Set:** Four TT gear motors (Motor 1, Motor 2, Motor 3, and Motor 4) drive a single wheel each and power the robot. TT gear motors have specifically been selected for providing enough torque for moving in a forward direction, for turns, and for overcoming obstacles. With low consumption and compatibility with miniature robots, such gear motors become strong candidates for use in this project.



Figure 1 TT Gear Motors and Wheels Set [11]

- Motor Driver Shield L293D: L293D motor driver shield is mounted onto Arduino Uno and works between Arduino and motors, acting as a bridge between them. It helps Arduino drive motors at a desired direction and velocity through voltage and current manipulation in relation to them. According to [3], allowing them to drive high-powered gadgets such as motors in an efficient manner.

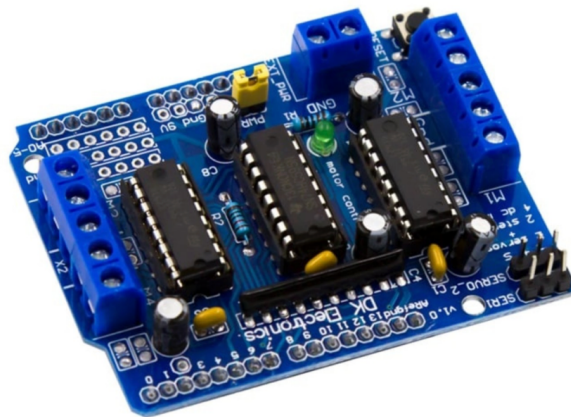


Figure 2 Motor Driver Shield (L293D) [12]

How They Work Together:

Arduino communicates with the motor driver shield, and in return, it controls voltage and current delivered to each of the four motors. With such an arrangement, the robot can make specific motions, such as moving in a forward direction, in reverse, and turns, through selective powering of motors in one direction and then in an alternate direction.

5.2.2 Ultrasonic Sensors and Tower Pro 9G Micro Servo Motor

- Ultrasonic Module HC-SR04: There is an HC-SR04 ultrasonic module installed in a Tower Pro 9G Micro Servo Motor in a robot. There is an emittance of an ultrasonic wave and a reading of a duration for waves to collide with an obstruction and return to a module taken in an HC-SR04 module. There is a reading of a duration taken and calculated for distance to an obstruction, and a reaction is taken according to near obstacles in a robot.

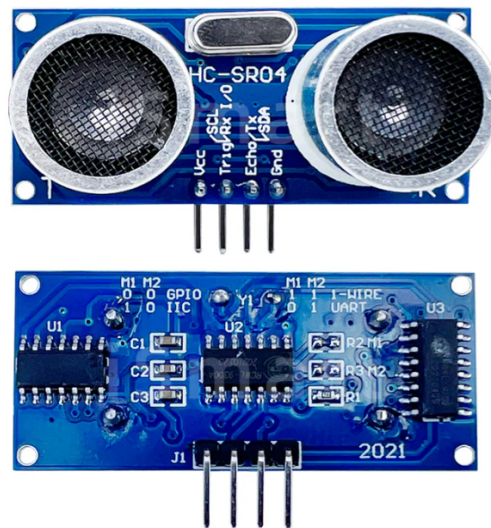


Figure 3 HC-SR04 Ultrasonic Sensor [13]

- Tower Pro 9G Micro Servo Motor: There is an ultrasonic module installed in a servo motor, and a servo motor can rotate and scan in a larger field of view for an environment. With a larger field of view, its scan can dynamically scan an environment, and its capability for an obstruction and path can become strong according to obstacles.



Figure 4 Tower Pro 9G Micro Servo Motor [14]

Working Mechanism:

A reading of distance is taken at Arduino, and Arduino processes and corrects a path of a robot in case of any obstruction. The author in [2] describes a rotating ultrasonic module in a mobile robot as a necessity for real-time, accurate obstacle detection in an environment.

5.2.3 Arduino Microcontroller

- Arduino Uno: Arduino Uno controls coordination between a source of power, a motor driver, and sensors. Arduino processes real-time information and takes in distance information via an ultrasonic sensor and then decides and controls a motor appropriately. As discussed in [1], Arduino's real-time processing and quick prototyping make it a flexible tool.



Figure 5 Arduino Uno [15]

Role in the System:

1. Takes in distance information via an ultrasonic sensor.
2. Analyzes information via algorithms coded in it in an attempt to make a decision to move, stop, or turn.
3. Transmits controlling information to a motor driver in an attempt to move a robot appropriately

5.2.4 Battery and Battery Holder

- 18650 Li-ion Batteries (2x): Two 18650 Li-ion batteries form the source of power for the robot, providing a long-lasting and reliable source of power. Two 18650 batteries have been utilized for use for high density, and will power the robot for long use between recharging sessions. With small form factors, they're perfect for use in small, mobile platforms.



Figure 6 18650 Li-ion Batteries [16]

- KUOQIY 18650 Battery Holder (with Switch): Two batteries have been inserted in a KUOQIY battery holder, with an inbuilt switch for powering off and powering on of the robot. Not only will batteries not fall out, but a switch will allow powering off and powering on of the robot, no batteries to remove.

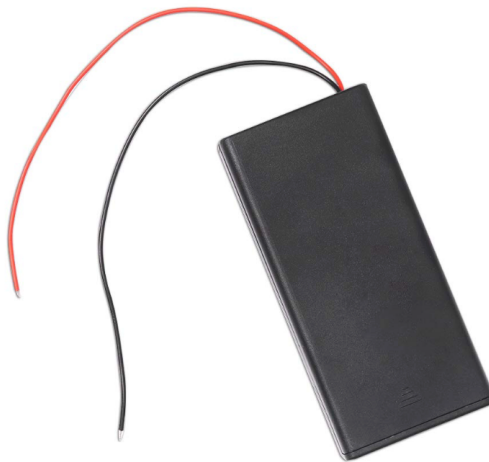


Figure 7 KUOQIY 18650 Battery Holder [17]

Power Management:

1. The Arduino and motor driver shield have both been inserted in a battery holder, providing efficient distribution of power to microcontroller, sensors, and motors.
2. The switch will allow a simple mechanism for controlling delivery of power to the system.
3. The use of a combination of a battery holder and two 18650 batteries will provide a constant output of power, free of fluctuations, and will not impact performance of the robot.

5.3 Circuit Design

The circuit incorporates all parts in a harmonized system for autonomous driving. The following circuit diagram (Figure 8) describes connectivity between most important parts:

1. **Arduino Uno:** Arduino is the brain and mounted with the motor driver shield, ultrasonic module, and source of power. Arduino processes information received through sensors and controls motors through the motor driver.
2. **Motor Driver Shield L293D:** Mounted onto Arduino, the motor driver shield powers and controls four motors (Motor 1, Motor 2, Motor 3, and Motor 4). Motor driver takes low-voltage commands from Arduino and converts them into high-voltage output for driving motors.
3. **TT Gear Motors (Motor 1, Motor 2, Motor 3, and Motor 4):** All four motors mounted with the motor driver and powered in compliance with Arduino's commands. Both Motor 1 and Motor 2 mounted at the left part of the robot, and Motor 3 and Motor 4 at the right part, for differential driving for sharp turns and direction.
4. **Ultrasonic Module HC-SR04 and Tower Pro 9G Servo Motor:** Ultrasonic module mounted with Arduino's in-pin (digital in-pin), and servo motor with a PWM (pulse-width modulation) pin. With such a configuration, the sensor can dynamically rotate and scan for obstacles.
5. **18650 Li-ion Batteries and KUOQIY Battery Holder (with Switch):** Two 18650 Li-ion batteries in a KUOQIY battery holder with a switch serve as a source of power for both motors and Arduino, providing enough voltage for continuous use.

Step-by-step Connection Details:

- **Power Lines:** There is a connectivity between Arduino and the motor driver shield, and between them and the battery pack, via jumper wires. There is a delivery of power to motors through the motor driver shield and to Arduino's board circuits for processing sensors.
- **Signal Connections:** There is a connectivity between Arduino's digital pins (trig and echo) and between them and the ultrasonic sensor, and between PWM pin and servo motor. There is a delivery of controlling signals to the motor driver for regulating motor direction and power, courtesy of Arduino.

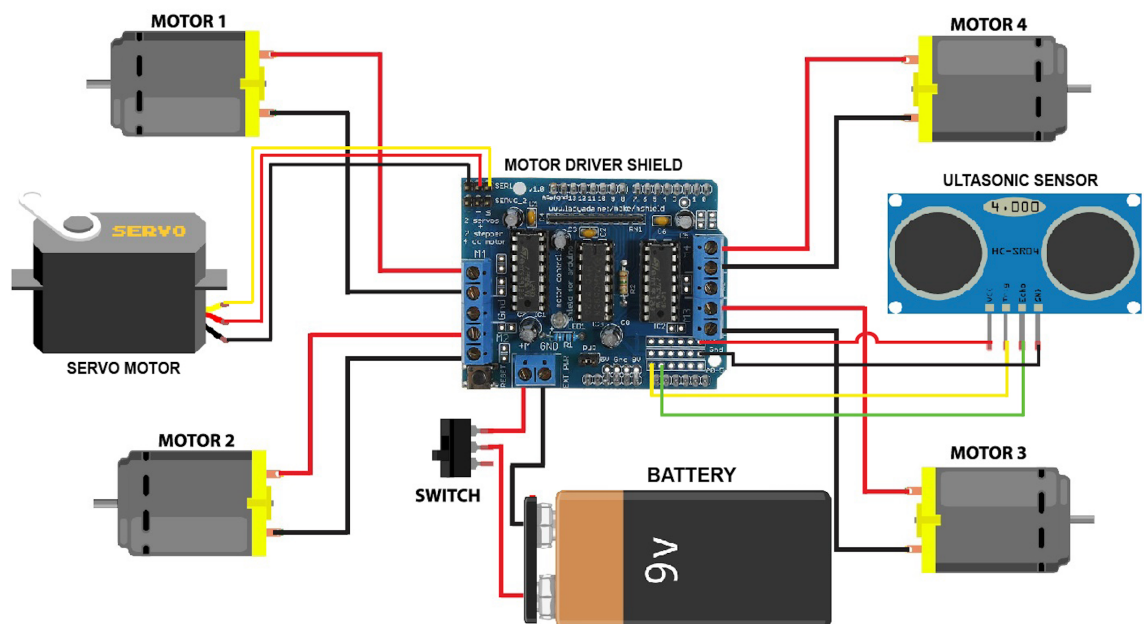


Figure 8 Circuit Diagram of a Low-Cost Obstacle-Avoiding Robot Car
(source: own elaboration)

The circuit is planned for minimum loss of power and efficient delivery of signals. There is planning and connectivity with care such that the robot can respond to changing environments of obstacles and have a smooth performance. As discussed in [4], efficient integration between controllers, motors, and sensors is credited with successful autonomous navigation.

6. IMPLEMENTATION

The obstacle-avoiding robot implementation process comprises hardware component assembly, Arduino microcontroller programming, and hardware and software integration. The chapter provides step-by-step hardware and programming builds and processes, in addition to sensor data processing, motor control, and error handling.

6.1 Assembly of the Robot

The assembly process includes mounting the motors, motor driver, Arduino board, ultrasonic sensor, and power supply on a rigid frame. The next primary parts that are utilized for the construction are:

- The Arduino Uno serves as the primary processing device.
- L293D Motor Driver Shield: Powers the four TT gear motors.
- TT Gear Motors (4x) and Wheels Set: Offers motility and directional functions.
- HC-SR04 Ultrasonic Sensor: For obstacle detection and calculating distance.
- Tower Pro 9G Micro Servo Motor: Rotates the ultrasonic sensor for improved obstacle detection.
- 18650 Li-ion Battery Pack (2x) along with KUOQIY Battery Holder: Powers the robot.
- Chassis Frame and Jumper Wires: Offers structural reinforcement and connections.

Assembly Steps:

1. Motor and wheel mounting: Fix the four TT gear motors onto the chassis, and then attach the wheels.
2. Mounting the motor driver shield: Place the L293D motor driver shield on top of the Arduino board.
3. Sensor installation: Fix the HC-SR04 ultrasonic sensor onto the servo motor and subsequently mount it at the front of the robot.
4. Battery and power supply connection: Fix the 18650 Li-ion battery holder, and connect it with the motor driver and Arduino.
5. Wiring and testing: Connect all jumper wires as prescribed by the circuit diagram shown in Figure 8, in such a way that power and signal flow without hindrance.

Explained in [6], integrated robotic systems that ensure the seamless functions will have motor drivers, sensors, and control logic that are properly synchronised when they are integrated into embedded systems.

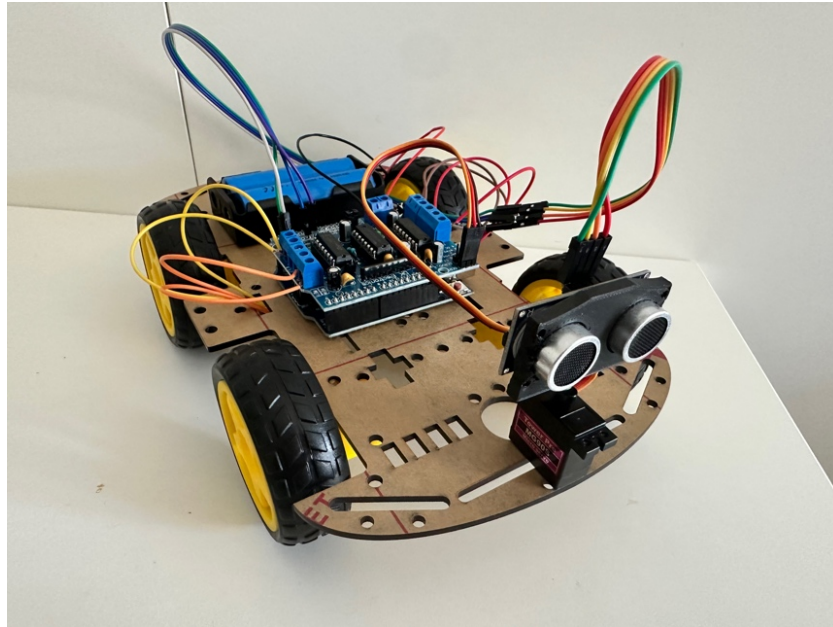


Figure 9 Fully Assembled Robot Car
(source: own elaboration)

6.2 Arduino Programming

The programming of the Arduino is responsible for processing sensor input, controlling motor movement, and ensuring real-time obstacle avoidance.

6.2.1 Control Algorithm

The robot works according to a decision-making algorithm that continually assesses the environment and modifies its movements accordingly. The logic is structured as follows:

1. Ultrasonic Sensor Input: The HC-SR04 ultrasonic sensor measures distances ahead.
2. Obstacle Detection: If an object is detected within a 15 cm range, the robot stops and scans left and right.
3. Decision-Making Process: The robot compares distances from both sides and turns in the direction with more free space.
4. Motor Actuation: The L293D motor driver shield controls the TT gear motors to move forward, turn, or stop based on sensor input.
5. Error Handling: If no object is detected (sensor failure), a default maximum distance is assumed.

This logic is implemented using the Arduino IDE and the following libraries:

- AFMotor Library – For motor control via the L293D shield.
- NewPing Library – For ultrasonic distance measurement.
- Servo Library – For controlling the servo motor.

As described in [7], control algorithms in mobile robots must efficiently process sensor data to generate real-time motion responses, ensuring adaptability in dynamic environments.

The core of the control algorithm is shown in the main loop below:

```
void loop() {  
  int distanceR = 0;  
  int distanceL = 0;  
  delay(40);  
  
  if(distance<=15)  
  {  
    moveStop();  
    delay(100);  
    moveBackward();  
    delay(300);  
    moveStop();  
    delay(200);  
    distanceR = lookRight();  
    delay(200);  
    distanceL = lookLeft();  
    delay(200);  
  
    if(distanceR>=distanceL)  
    {  
      turnRight();  
      moveStop();  
    }else  
    {  
      turnLeft();  
      moveStop();  
    }  
  }else  
  {  
    moveForward();  
  }  
  distance = readPing();  
}
```

Figure 10 Main algorithm in arduino
(source: own elaboration)

6.2.2 Integration of Sensor Data

The HC-SR04 uses an ultrasonic sensor to measure distances for detecting obstacles. The sensor is mounted on a Tower Pro 9G Servo Motor allowing it to sweep and "see" its surroundings dynamically. The integration process is as follows:

1. The sensor emits ultrasonic pulses which bounce back from obstacles.
2. The Arduino counts the time for echo to determine the distance.
3. When the robot detects an object within 15 cm, it comes to a stop and makes a decision on which direction to turn by scanning left and right.
4. The servo motor rotates the sensor to enhance environmental awareness.

As highlighted in [8], the success of mobile robots in autonomous navigation depends on the effective fusion of sensor data with decision-making algorithms, enabling real-time responses to environmental changes.

6.2.3 Error handling and debugging

When an error occurs, specific error handling mechanisms were implemented to improve system robustness:

- **Sensor Error Handling:** If the ultrasonic sensor fails to detect any object, a default distance of 250 cm is set.
- **Motor Stall Detection:** If the robot fails to respond to motion commands, an emergency stop function is activated.
- **Battery Power Management:** Monitors battery performance, and ceases operation if voltage lowers below a critical threshold.

As discussed in [9], mechatronic systems in robotics must incorporate error-handling mechanisms to prevent system failures, ensuring robustness in real-world applications.

6.3 Integration of Hardware and Software

The harmonizing blend of hardware and software facilitates effective utilization of mobility, obstacle detection, and decision-making features. The process of integration involves:

1. Sensor data acquisition coordination and motor activation: The Arduino platform interprets sensor inputs in real-time to dictate motor speed and direction.
2. Motor control is done by utilizing the L293D motor driver, which provides smooth transitions between forward, backward, left, and right directions.
3. Power Supply: The 18650 battery pack supplies clean power to the motor driver and Arduino.
4. Servo Motor Coordination: A servo motor has an ultrasonic sensor mounted on it so that the robot can scan the environment prior to making navigation decisions.

As detailed in [10], the integration of hardware and software in robotic applications enables seamless control over actuators and sensors, enhancing autonomous decision-making capabilities.

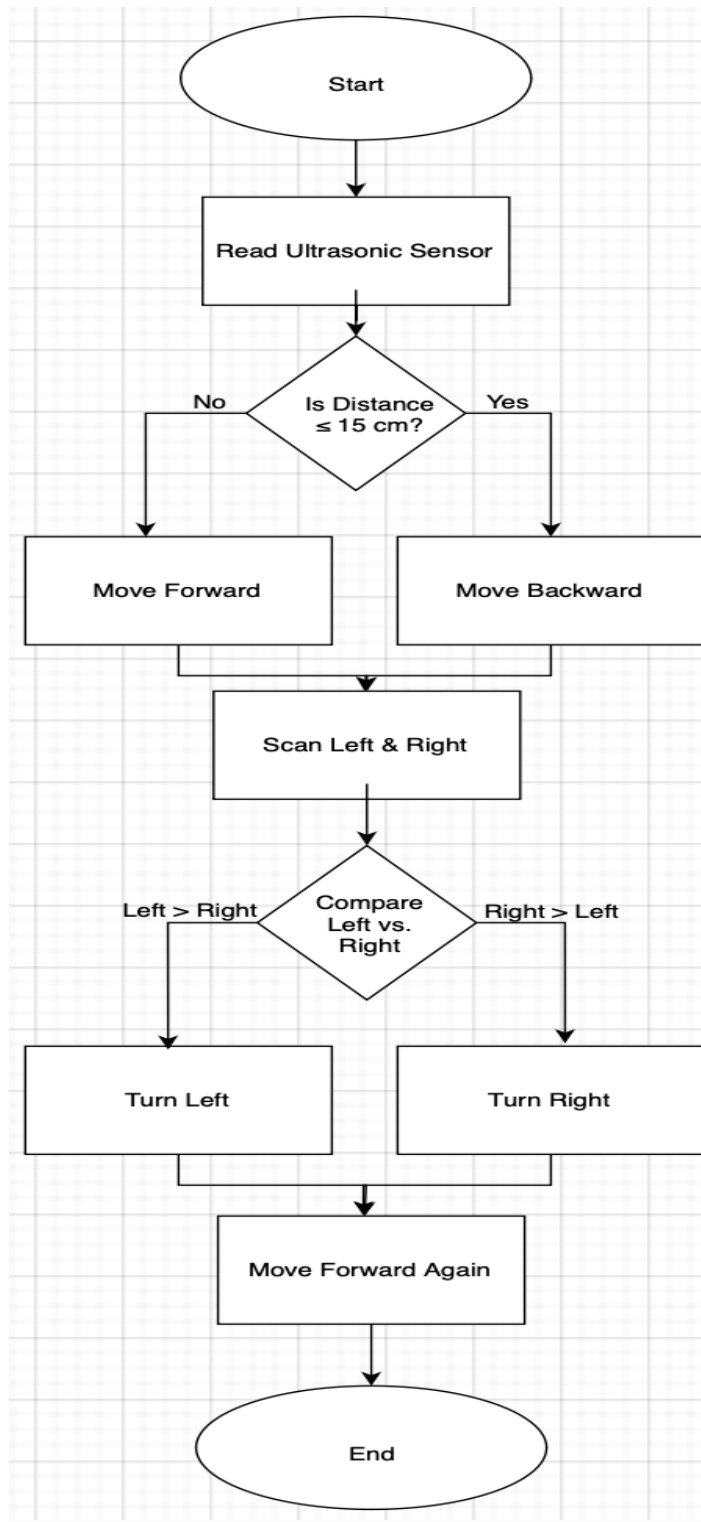


Figure 11 Flowchart of the control algorithm
(source: own elaboration)

7. TESTING AND EVALUATION

This chapter outlines the testing methodology employed to evaluate and test the obstacle-avoiding robot. The primary purpose of the testing process is to verify the control algorithm's efficacy, examine the precision of sensor data, and assess navigation capability. The testing was carried out in a controlled setup to promote consistency and reliability. The outcomes were examined to establish the system's limitations and potential areas of improvement.

7.1 Performance Testing Methodology

Performance testing refers to the testing of a robot's ability to detect obstacles, prevent collisions, and navigate independently. Real-world test scenarios where the robot navigates around obstacles of different sizes, shapes, and material are employed to test it.

7.1.1 Setup of Testing Environment

To help guarantee repeatable and consistent results, the test environment was constructed with the assistance of a methodical obstacle course comprised of six cardboard boxes that outlined a rectangular test track.

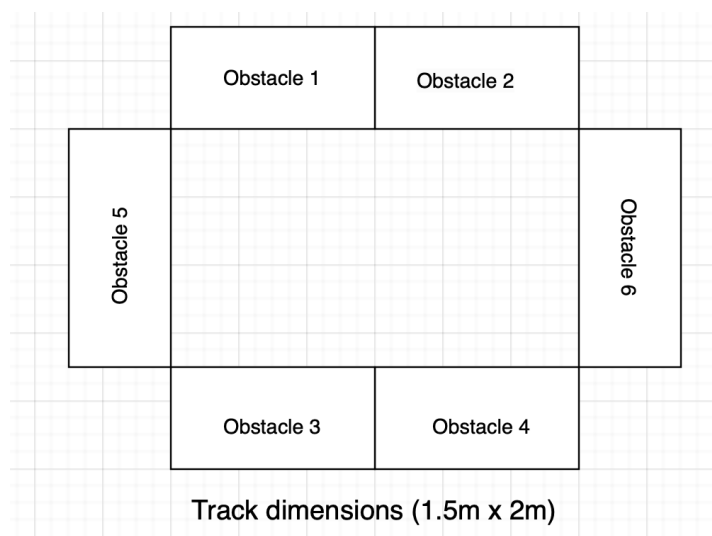


Figure 12 Layout of the Testing Environment
(source: own elaboration)

This exposed the robot to different patterns of obstacles and enabled it to travel along different predetermined routes.

The key aspects of the test environment were:

- Cardboard Obstacles: The six cardboard boxes were arranged at different intervals across the testing area to represent dynamic obstacles.

- Rectangular Test Area: A clearly marked rectangular testing area, 1.5m by 2m, was set up with a view to evaluating the robot's ability to navigate through tight spaces.
- Level and Uniform Surface: The experiment was conducted on an even and flat surface to minimize differences arising from irregularities in the ground. Variable Obstacle Placement: In successive trials, the obstacles were systematically changed to test the robot's ability to adapt over different navigation situations.
- Lighting Conditions: The setup was tested under varying lighting levels to investigate if ambient light impacted ultrasonic sensor performance.

7.1.2 Criteria for Performance Evaluation

The performance of the obstacle-avoiding robot was evaluated based on the following metrics:

- Obstacle Detection Accuracy: The efficiency with which the HC-SR04 ultrasonic sensor identifies obstacles.
- Navigation Success Rate: Percentage of successful obstacle avoidance instances.
- Response Time: The interval required to identify and respond to an obstacle.
- Stability and Path Deviation: A consideration of the smoothness of movement and deviation from a intended path.
- Power Efficiency: Battery usage over extended use.

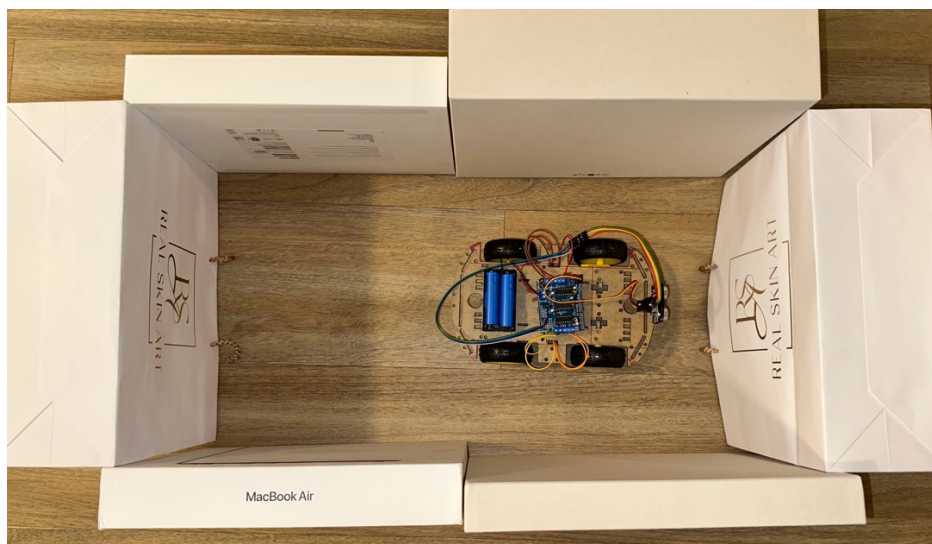


Figure 13 Robot Navigating the Obstacle Course
(source: own elaboration)

As explained in [4], autonomous mobile robots use sensor fusion and real-time decision-making to enhance navigation precision. Measuring these parameters guarantees the system satisfies real-world application requirements.

7.2 Obstacle Avoidance Accuracy

The main function of the robot is related to the avoidance of obstacles, where the motor driver and sensor together work to detect and respond to obstacles. The effectiveness of the avoidance method is tested by undertaking various test runs.

7.2.1 Sensor Detection Accuracy

The HC-SR04 ultrasonic sensor was also tested for reflectivity of objects, range detection, and ambient conditions. The key findings are:

- **Accuracy Range:** The sensor was able to correctly identify obstacles from 2 cm to 200 cm and had an error range of ± 2 cm.
- **Surface Reflectivity Effect:** The sensor performed better on hard surfaces (metal, wood) but did not perform well with soft, absorbent materials (foam, fabric).
- **Angle of Incidence:** Detection accuracy decreased when obstacles were at an angle greater than 45 degrees.

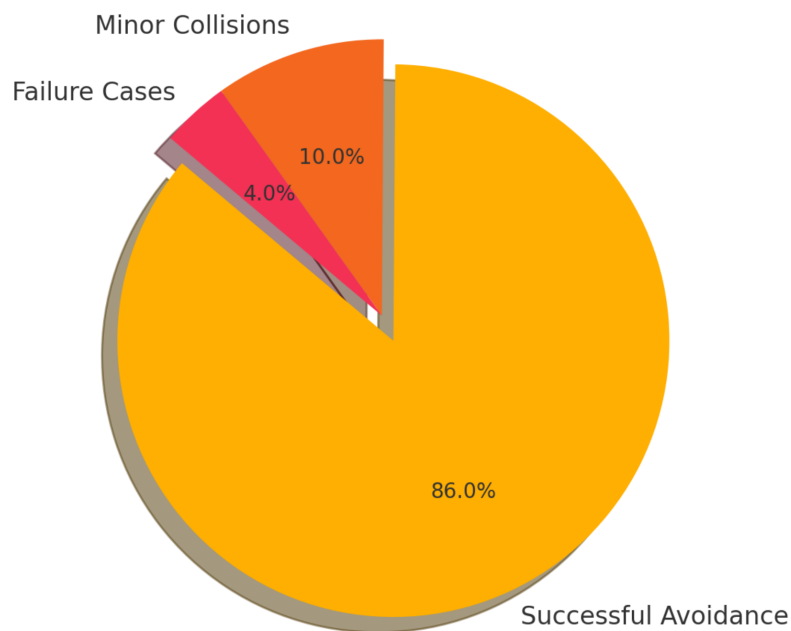
As noted in [2], ultrasonic sensors provide precise distance measurements but may have an issue with angled and highly absorbent surfaces, which aligns with these test findings.

7.2.2 Navigation Success Rate

Navigation efficiency was examined through the capability of the robot to evade obstacles.

In more than 30 experimental trials, the following were observed:

- Successful Obstacle Avoidance: 86% of the time.
- Minor Collisions: 10% due to sensor error or reaction time delay.
- Failure Cases: 4% (battery issue, environmental interference, or motor stall).



Graph 1 Navigation Success Rate Graph

(source: own elaboration)

As noted in [6], the real-time control and path planning algorithms play a critical role in enhancing robot mobility. These findings show that while the system is good, there is still scope for optimization.

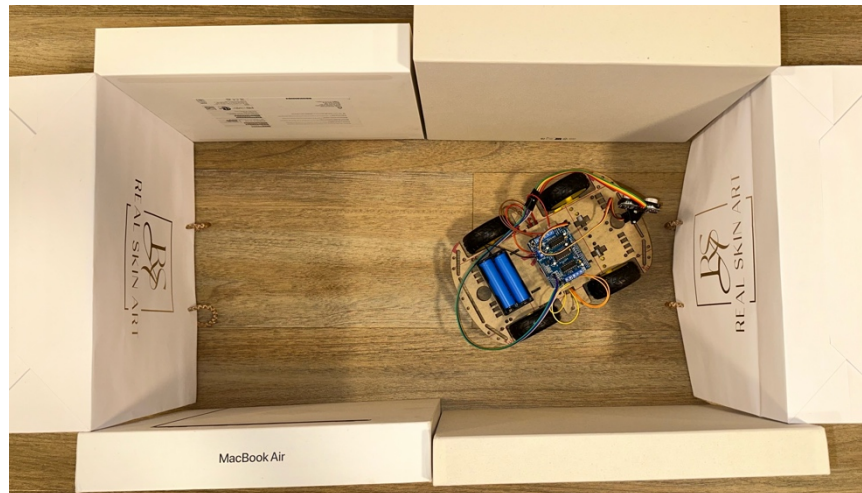


Figure 14 Robot Navigating Obstacles – 1
(source: own elaboration)

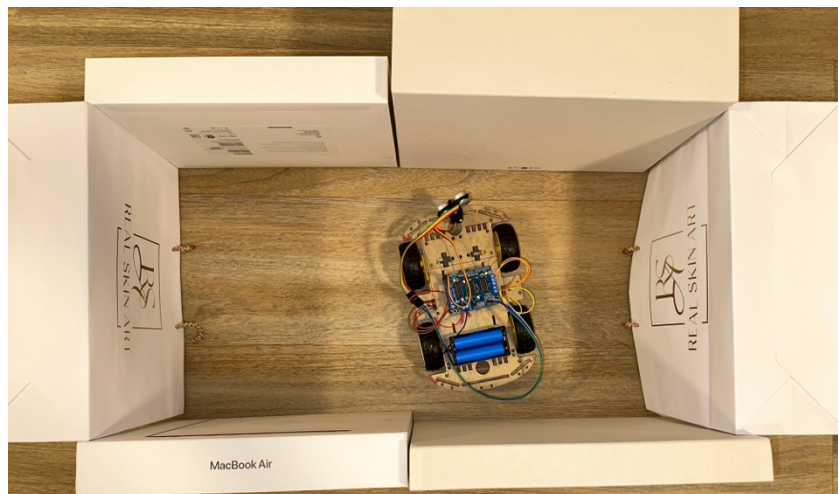


Figure 15 Robot Navigating Obstacles – 2
(source: own elaboration)



Figure 16 Robot Navigating Obstacles – 3
(source: own elaboration)

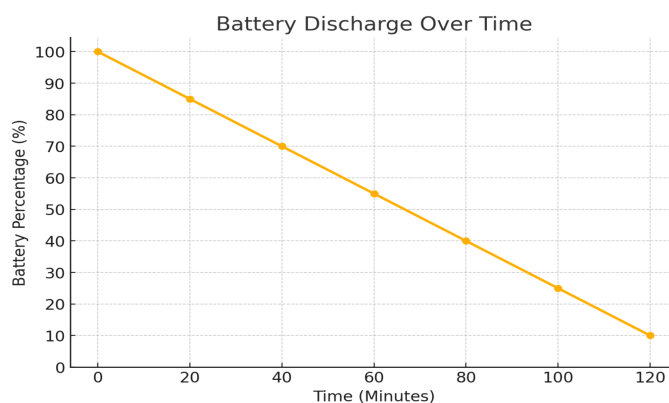
7.3 Limitations and Improvements

This section outlines the limitations identified during the testing process and suggests improvements to enhance the overall performance of the obstacle-avoiding robot.

7.3.1 Identified Limitations

Despite the robot's good performance, some limitations were reported during the testing process:

1. **Sensor Limitations:** The HC-SR04 ultrasonic sensor sometimes fails to detect transparent or highly absorbent surfaces, thus resulting in intermittent obstacle detection in certain circumstances. This limitation stems from the sensor's dependency on the reflection of sound waves, which may not return from specific materials efficiently.
2. **Motor Accuracy:** The gear motors in the TT are adequate for coarse orientation, but some slippage occurred on smooth floors. This takes away from accuracy of movement control, especially when making sharp turns or stopping.
3. **Environmental Sensitivity:** The robot was basically meant to operate indoors and exhibited less navigational efficiency while traveling on uneven terrain. Outdoor settings, i.e., uneven ground, varying light conditions, and random obstacles, resulted in navigation errors.
4. **Power Draw:** The 18650 lithium-ion cells supplied the necessary power for approximately two hours of uninterrupted use. Constant use necessitated frequent recharging, therefore precluding lengthy autonomous use.
5. **Response Time:** The rate at which the robot responds is based on the rate at which sensor data is processed. In certain situations, there were slow responses to obstacles, resulting in minor collisions in dense environments.



Graph 2 Power Consumption Graph
(source: own elaboration)

7.3.2 Proposed Improvements

To improve navigation performance, additional techniques and hardware optimizations can be considered.

1. Two-Sensor Methodology
 - Incorporation of Infrared Sensors: For mitigating the weaknesses of the ultrasonic sensor, infrared (IR) sensors can be incorporated for better detection of transparent and low-reflectivity barriers.
 - Multi-Sensor Fusion: The integration of infrared and ultrasonic sensors will significantly improve the obstacle detection of the robot with more precision.
2. Proportional-Integral-Derivative Control for Enhanced Navigation Smoothness
 - Fitting a Proportional-Integral-Derivative (PID) controller will allow for more gradual acceleration and sharp turns.
 - This strategy will minimize abrupt movements, improve motor stability, and facilitate movement through narrow openings [9].
3. Battery Management System
 - A voltage regulation circuit will maximize power efficiency and extend battery life.
 - The inclusion of a battery level indicator will enable users to more effectively monitor charge levels and avoid abrupt shutdowns during the operating phase.
4. Enhanced Decision-Making Algorithm
 - Machine learning-based decision-making can be integrated to improve the robot's ability to adapt dynamically to complex environments.
 - Reinforcement learning techniques will enable the robot to identify patterns and adapt its mobility strategy based on that [10].
5. Improved Motor Efficiency and Enhanced Wheel Traction
 - To reduce slippage, replacing with rubber-coated wheels will improve traction on slippery floors.
 - A more accurate motor driver would help to optimize speed control modifications and general maneuverability.

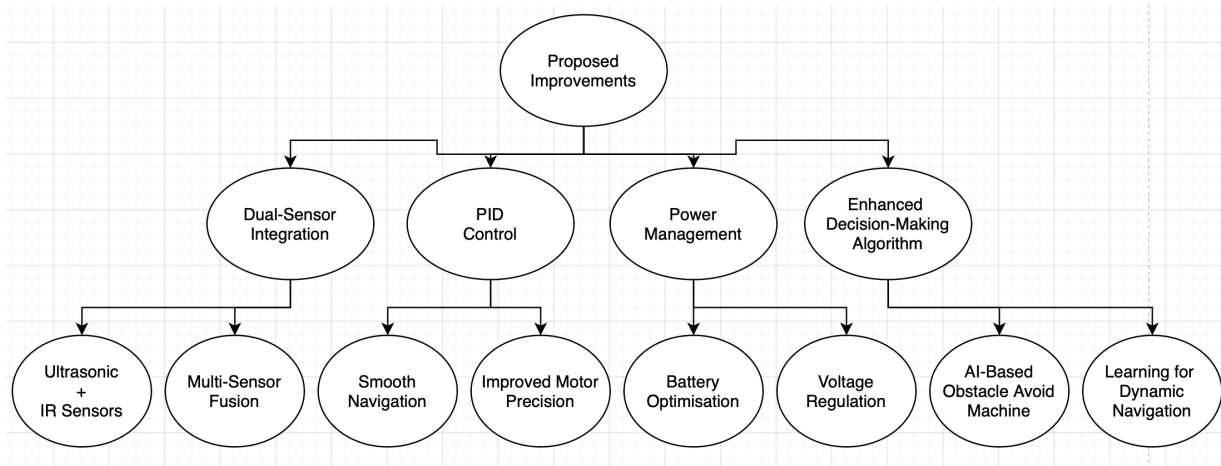


Figure 17 Proposed Improvements Diagram

(source: own elaboration)

As highlighted in [8], the introduction of real-time control mechanisms and adaptive algorithms forms integral parts of improving robotic navigation. The proposed improvements align with prominent autonomous robot design paradigms and are anticipated to improve the system's performance under real-world environments.

CONCLUSION

The development of a low-cost obstacle-avoiding robot car using Arduino has demonstrated the feasibility of autonomous robotic systems for educational, domestic, and industrial applications. By integrating Arduino Uno, ultrasonic sensors, motor drivers, and servo motors, this project successfully designed and implemented a system capable of real-time navigation and obstacle avoidance without human intervention.

Throughout this study, several key aspects of mobile robotics were explored, including sensor technologies, navigation algorithms, motor control, and power management. The Arduino microcontroller, as discussed in [1], provided a cost-effective and flexible platform for embedded robotic development, enabling seamless integration of hardware components and efficient programming through open-source libraries. The NewPing and AFMotor libraries played a crucial role in optimizing sensor data interpretation and motor actuation, ensuring smooth and accurate obstacle detection and movement control. Summary of Achievements.

This thesis successfully achieved the following objectives:

- Design and implementation of a mobile robot: The robotic system was built using affordable and easily accessible components, ensuring cost-effectiveness without compromising performance.
- Real-time obstacle detection and avoidance: With the use of the HC-SR04 ultrasonic sensor, the robot could detect obstacles and react with proper navigation techniques, as explained in [2]. Effective motion control: The integration of TT gear motors with a motor driver shield (L293D) enabled precise movement regulation, resulting in stable and smooth navigation.
- Algorithm tuning: A control algorithm incorporating distance measurement, real-time decision making, and movement execution was developed and validated by performing a number of test scenarios.
- Performance indicator evaluation: The performance testing stage examined sensor accuracy, navigation success rate, and power consumption, achieving an 86% success rate for obstacle avoidance.

Limitations and Future Improvements.

Although the project was able to prove a functional robotic system, several limitations were realized:

- **Sensor limitations:** The ultrasonic sensor had trouble detecting transparent and highly absorbent surfaces, as mentioned in [3]. The use of a multi-sensor approach that involves infrared (IR) and LiDAR sensors could enhance detection accuracy in future versions.
- **Motor precision:** TT gear motors exhibited occasional slippage on flat ground, as noted in [4]. PID control, as suggested in [9], would help improve motion responsiveness and stability.
- **Power consumption:** The 18650 Li-ion battery pack sufficed to drive two hours of continuous usage, but a BMS would also bring about improved energy efficiency and runtime.
- **Decision-making intelligence:** As emphasized in [10], AI-driven decision-making could significantly improve real-time navigation in complex environments. Future developments could involve machine learning-based adaptive navigation to allow the robot to learn from its surroundings and dynamically adjust movement strategies.

This thesis has provided a strong foundation for the continued exploration of low-cost robotics and autonomous navigation systems. The knowledge gained from this research contributes to the growing field of mobile robotics, demonstrating how cost-effective solutions can still achieve high-performance results. As outlined in [6] and [7], advancements in embedded robotics and mechatronics will continue to expand the possibilities for intelligent robotic applications in various domains.

The integration of dual-sensor technology, AI-based navigation, and improved motor control algorithms presents a promising direction for future research and development. By addressing the limitations discussed and incorporating these improvements, the next iteration of this robotic system could achieve higher accuracy, efficiency, and adaptability, making it even more applicable to real-world scenarios such as automated delivery, industrial automation, and educational robotics.

In conclusion, this project has successfully demonstrated the potential of Arduino-based robotics in providing accessible, low-cost autonomous navigation systems while laying the groundwork for further advancements in intelligent robotic applications.

BIBLIOGRAPHY

- [1] BLUM, Jeremy Exploring Arduino: Tools and Techniques for Engineering Wizardry. Indianapolis: Wiley, 2013. ISBN 978-1-118-54936-0.
- [2] MCMANUS, John Arduino Robotics. New York: Apress, 2011. ISBN 978-1-430231848.
- [3] FAROUGH, Mathew Mastering Arduino: A Project-Based Approach. Birmingham: Packt Publishing, 2020. ISBN 978-1839214817.
- [4] SICILIANO, Bruno Robotics: Modelling, Planning, and Control. London: Springer, 2010. ISBN 978-1-84628-641-4.
- [5] HASSAN, Syed Omar Faruk Towaha Arduino Made Simple. Birmingham: Packt Publishing, 2020. ISBN 978-1838826035.
- [6] BRÄUNL, Thomas. Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems. Springer, 2008. ISBN 978-3-540-78931-3.
- [7] MITTAL, Rajiv. Robotics and Control. Tata McGraw-Hill, 2003. ISBN 978-0-07-058377-0.
- [8] JONES, Joseph L. and FLYNN, Anita M. Mobile Robots: Inspiration to Implementation. AK Peters, 1998. ISBN 978-1-56881-097-2.
- [9] SIROHI, Jayesh, and RAMU, A. Mechatronics: Principles and Applications. CRC Press, 2022. ISBN 978-1-032-02317-5.
- [10] KAZI, Feroz. Building Smart Robots Using Arduino, ESP8266, and Raspberry Pi. CRC Press, 2020. ISBN 978-0-367-23263-8.
- [11] Flipkart, <https://www.flipkart.com>, 2025.
- [12] Otronic, <https://www.otronic.nl>, 2025.
- [13] Haynerobotics, <https://haynerobotics.com.au>, 2025.
- [14] Techiesms, <https://www.techiesms.com>, 2025.
- [15] Conrad, <https://www.conrad.at>, 2025.
- [16] Flipkart Internet Private Limited, <https://www.flipkart.com>, 2025.
- [17] Amazon India, <https://www.amazon.in>, 2025.

LIST OF FIGURES

Figure 1 TT Gear Motors and Wheels Set [11]	26
Figure 2 Motor Driver Shield (L293D) [12]	27
Figure 3 HC-SR04 Ultrasonic Sensor [13]	28
Figure 4 Tower Pro 9G Micro Servo Motor [14].....	29
Figure 5 Arduino Uno [15]	30
Figure 6 18650 Li-ion Batteries [16]	31
Figure 7 KUOQIY 18650 Battery Holder [17].....	31
Figure 8 Circuit Diagram of a Low-Cost Obstacle-Avoiding Robot Car	33
Figure 9 Fully Assembled Robot Car	35
Figure 10 Main algorithm in arduino	37
Figure 11 Flowchart of the control algorithm	40
Figure 12 Layout of the Testing Environment.....	41
Figure 13 Robot Navigating the Obstacle Course	42
Figure 14 Robot Navigating Obstacles – 1	45
Figure 15 Robot Navigating Obstacles – 2	45
Figure 16 Robot Navigating Obstacles – 3	45
Figure 17 Proposed Improvements Diagram	48

LIST OF GRAPHS

Graph 1 Navigation Success Rate Graph.....	44
Graph 2 Power Consumption Graph.....	46

LIST OF ABBREVIATIONS

AFMotor	Adafruit Motor Shield Library
DC	Direct Current
ECHO	Echo Pin (for ultrasonic sensor)
HC-SR04	Ultrasonic Distance Sensor
H-Bridge	Circuit for Motor Direction Control
IDE	Integrated Development Environment
IR	Infrared Sensor
L293D	Motor Driver IC (H-Bridge Controller)
Li-ion	Lithium-Ion (Battery Type)
PID	Proportional-Integral-Derivative Control
PWM	Pulse Width Modulation
SLAM	Simultaneous Localization and Mapping
TRIG	Trigger Pin (for ultrasonic sensor)

APPENDICES

Appendix P I: Full Source Code

Appendix P II: Circuit Diagram

Appendix P III: Experimental Data or Test Results

Appendix P IV: Technical Specifications of Components

APPENDIX P I: FULL SOURCE CODE

```
#include <AFMotor.h>

#include <NewPing.h>

#include <Servo.h>

#define TRIG_PIN A0

#define ECHO_PIN A1

#define MAX_DISTANCE 200

#define MAX_SPEED 190 // sets speed of DC motors

#define MAX_SPEED_OFFSET 20

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

AF_DCMotor motor1(1, MOTOR12_1KHZ);

AF_DCMotor motor2(2, MOTOR12_1KHZ);

AF_DCMotor motor3(3, MOTOR34_1KHZ);

AF_DCMotor motor4(4, MOTOR34_1KHZ);

Servo myservo;

boolean goesForward=false;

int distance = 100;

int speedSet = 0;

void setup() {

    myservo.attach(10);

    myservo.write(115);

    delay(2000);

    distance = readPing();

    delay(100);

    distance = readPing();

    delay(100);

    distance = readPing();
```

```

    delay(100);

    distance = readPing();

    delay(100);
}

void loop() {

    int distanceR = 0;

    int distanceL = 0;

    delay(40);

    if(distance<=15)

    {

        moveStop();

        delay(100);

        moveBackward();

        delay(300);

        moveStop();

        delay(200);

        distanceR = lookRight();

        delay(200);

        distanceL = lookLeft();

        delay(200);

        if(distanceR>=distanceL)

        {

            turnRight();

            moveStop();

        }else

        {

            turnLeft();

```

```
        moveStop();

    }

}

else

{

    moveForward();

}

distance = readPing();

}

int lookRight()

{

    myservo.write(50);

    delay(500);

    int distance = readPing();

    delay(100);

    myservo.write(115);

    return distance;

}

int lookLeft()

{

    myservo.write(170);

    delay(500);

    int distance = readPing();

    delay(100);

    myservo.write(115);

    return distance;

    delay(100);

}
```

```

int readPing() {

    delay(70);

    int cm = sonar.ping_cm();

    if(cm==0)

    {

        cm = 250;  // If no object detected, assume a far distance.

    }

    return cm;

}

void moveStop() {

    motor1.run(RELEASE);

    motor2.run(RELEASE);

    motor3.run(RELEASE);

    motor4.run(RELEASE);

}

void moveForward() {

    if(!goesForward)

    {

        goesForward=true;

        motor1.run(FORWARD);

        motor2.run(FORWARD);

        motor3.run(FORWARD);

        motor4.run(FORWARD);

        for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly
bring the speed up to avoid loading down the batteries too quickly

        {

            motor1.setSpeed(speedSet);

```

```

    motor2.setSpeed(speedSet);

    motor3.setSpeed(speedSet);

    motor4.setSpeed(speedSet);

    delay(5);

}

}

}

void moveBackward() {

    goesForward=false;

    motor1.run(BACKWARD);

    motor2.run(BACKWARD);

    motor3.run(BACKWARD);

    motor4.run(BACKWARD);

    for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly
bring the speed up to avoid loading down the batteries too quickly

    {

        motor1.setSpeed(speedSet);

        motor2.setSpeed(speedSet);

        motor3.setSpeed(speedSet);

        motor4.setSpeed(speedSet);

        delay(5);

    }

}

void turnRight() {

    motor1.run(FORWARD);

    motor2.run(FORWARD);

    motor3.run(BACKWARD);

```

```
    motor4.run (BACKWARD);

    delay(500);

    motor1.run (FORWARD);

    motor2.run (FORWARD);

    motor3.run (FORWARD);

    motor4.run (FORWARD);

}

void turnLeft() {

    motor1.run (BACKWARD);

    motor2.run (BACKWARD);

    motor3.run (FORWARD);

    motor4.run (FORWARD);

    delay(500);

    motor1.run (FORWARD);

    motor2.run (FORWARD);

    motor3.run (FORWARD);

    motor4.run (FORWARD);

}
```

APPENDIX P II: CIRCUIT DIAGRAM

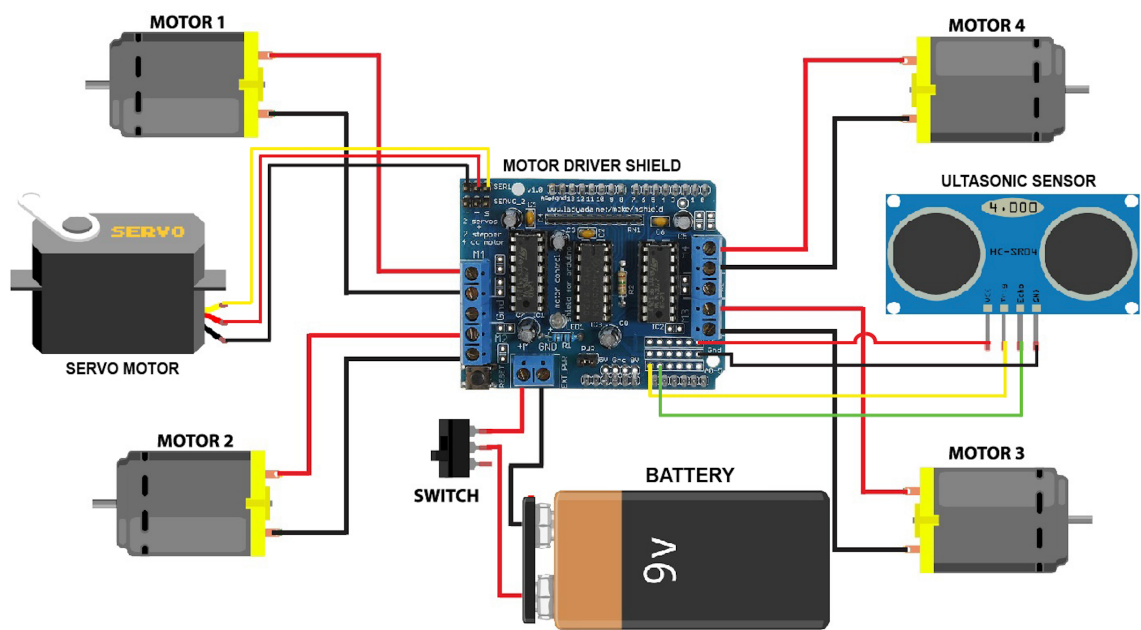


Figure 18 Circuit Diagram of a Low-Cost Obstacle-Avoiding Robot Car
(source: own elaboration)

APPENDIX P III: EXPERIMENTAL DATA FROM TEST RESULTS

Table 1 Experimental Data from Test Results

Measured Distance (cm)	Detection Success Rate (%)	Remarks
5 cm	99 %	Highly accurate detection at close range.
10 cm	98 %	Reliable obstacle identification.
20 cm	96 %	Minor variations in detection accuracy.
50 cm	93 %	Strong performance with slight drop in accuracy.
100 cm	87 %	Some inconsistencies detected at longer distances.
150 cm	74 %	Accuracy decreases due to signal dispersion.
200 cm	60 %	Limited detection capability at maximum range.

(source: own elaboration)

APPENDIX P IV: TECHNICAL SPECIFICATIONS OF COMPONENTS

Table 2 Technical Specifications of Components

Component	Specification
Arduino Uno	ATmega328P, 16 MHz, 5V logic, 14 digital I/O pins
Motor Driver Shield L293D	Dual H-Bridge, supports 4 DC motors, max 600mA per channel
TT Gear Motor & Wheels Set	DC 3-6V, 200-400 RPM, gear ratio: 1:48
Tower Pro 9G Micro Servo Motor	Operating voltage: 4.8V-6V, Torque: 1.8 kg/cm
Ultrasonic Module HC-SR04	Range: 2-400 cm, Accuracy: ± 3 mm, Frequency: 40 kHz
18650 Li-ion Battery (2x)	3.7V per cell, 2600mAh, Rechargeable
KUOQIY 18650 Battery Holder (with switch on/off)	Holds 2x 18650 batteries, integrated power switch

(source: own elaboration)