Design and Implementation of an Autonomous Differential Drive Mobile Robot with Obstacle Avoidance Using Arduino Nano

Carl Danielle C. Andante., Selwyn James S. Jimenez., Janine P. Natalia., Kyla Marie G. Septimo., Engr. Bernard C. Fabro

Computer Engineering Department, Eulogio "Amang" Rodriguez Institute of Science and Technology, Nagtahan, Sampaloc, Manila, 1016 Philippines

DOI: https://doi.org/10.51244/IJRSI.2026.13010020

Received: 08 January 2026; Accepted: 13 January 2026; Published: 23 January 2026

ABSTRACT

This paper presents the design and implementation of an autonomous mobile robot capable of navigating unknown environments through maze-solving algorithms and dynamic obstacle avoidance. The primary objective is to engineer a low-cost Automated Guided Vehicle (AGV) that integrates environmental perception with real-time decision-making. The system architecture utilizes a differential drive chassis configuration, driven by two DC gear motors. Unlike servo-steering systems, directional control is achieved by varying the speed and direction of the independent rear motors (skid-steering). The central processing unit, an Arduino Nano, interfaces with a distributed sensor array consisting of three downward-facing infrared (IR) sensors for path tracking and one forward-facing IR sensor for collision prevention. The control software implements the "Left-Hand Rule" algorithm for traversing simply connected mazes and a "Smart Avoidance" subroutine that enables the robot to autonomously reroute when facing physical obstructions. Additionally, the system incorporates a Human-Machine Interface (HMI) utilizing an I2C-enabled Liquid Crystal Display (LCD) and LED indicators to provide diagnostic feedback during operation.

Keywords: Mobile Robotics, Arduino Nano, Obstacle Avoidance, Maze Solving, Left-Hand Rule, Embedded Systems, Differential Drive

INTRODUCTION

The rapid advancement of robotics technology has revolutionized industries ranging from manufacturing to logistics. Automated Guided Vehicles (AGVs) have become essential tools for material handling, surveillance, and hazardous environment exploration. A critical challenge in the development of such systems is autonomous navigation—the ability of a robot to perceive its environment, determine its location, and plan a path to its destination without human intervention. While Global Positioning Systems (GPS) are effective for outdoor navigation, indoor environments often require reliance on local sensor data, such as visual lines or physical barriers.

Traditional educational robots commonly use differential drive mechanisms, where turning is achieved by varying the speed and direction of the wheels on opposite sides of the chassis. This approach is mechanically simple and well suited for low-cost robotic platforms. In this project, a four-wheel-drive mobile robot is developed using a differential drive configuration, where directional control is achieved through independent motor control rather than mechanical steering. By regulating motor speed using Pulse Width Modulation, the robot is able to perform smooth turns and pivot maneuvers, enabling accurate navigation in maze environments and effective obstacle avoidance.

The objective of this research is to develop a robust embedded system capable of two distinct operational modes: (1) Maze Solving, utilizing the "Left-Hand Rule" algorithm to navigate a track with intersections and dead ends; and (2) Obstacle Avoidance, utilizing proximity sensors to detect and react to dynamic physical obstructions. By integrating these functions into a single platform controlled by an Arduino Nano, this paper demonstrates a cost-effective solution for autonomous indoor navigation.

Importance and Relevance of the Study

This was an essential study as it demonstrated through design and implementation how to develop an autonomous mobile robot that could navigate through mazes with obstacles detected with low cost and commercially obtainable materials. This study also provides a very hands-on approach to understanding autonomous navigation, integrating multiple sensors into one system, and making decisions in-line in an embedded system environment with Arduino Nano and Infrared (IR) sensors. The system implemented here is an excellent example of how simple rules such as the left-hand rule can be utilized to solve practical navigation issues.

The application of this study will benefit students and teachers in the fields of Computer Engineering and Electronics Engineering by providing a hands-on learning experience in the areas of Robotics, Control Systems, and Microcontroller Programming. This study may serve as an example or reference source for laboratory experiments, instructional demonstrations, and future studies in Automated Guided Vehicles and Mobile Robots. Additionally, the system is modularly designed and can be modified and expanded for additional experiments and enhancements to the system.

In addition to the contributions made by this study to the area of student and teacher education, the study has added to the developing body of knowledge in the field of Automation by providing a low cost method for indoor autonomous navigation. The concepts and techniques employed in this project are directly applicable to many real world applications; examples include Warehouse Automation, Service Robots, Educational Robotics, etc. The study demonstrates the large potential of Embedded Systems in creating Intelligent and Efficient Robotic Platforms for both practical and academic applications.

REVIEW OF RELATED LITERATURE

In the present chapter, the literature review concerning the autonomous mobile robots, line-following systems, and obstacle-avoidance tactics on Arduino-based system platforms has been conducted in a holistic manner. This goal of the review will provide a sound theoretical and technical basis to the present investigation by analyzing the works in the field of investigation that include: the mechanism of differential drives, sensor guided navigation as well as the use of microcontrollers to control robotic structures. The practical studies have once again supported the effectiveness of infrared sensors in detecting lines and avoiding obstacles, as well as the implementation of Arduino microcontrollers in decision making in real-time and motor control. As a result of this, there is a literature review of the sources that are used to design and implement an autonomous differential drive mobile robot and outline relevant methodological strategies that fit the objectives of the current study.

Mobile Robotics

As described in Deshpande and Mazumdar (2016), the design and execution of a line-following robot was a form of mobile robotic platform, which has the capability of performing autonomous movement in a given path after a predefined route based on sensor feedback. Their exploration highlighted the combination of perceivers, microcontrollers and actuators to support the decision making under real-time and pose control. The usefulness of these results to the present research is the fact that similar techniques of the embedded system and sensor-based control methods are applied to an autonomous, differential-drive mobile robot. Although the prototype developed by Deshpande and Mazumdar only focused on line following, the current study implements the use of obstacle detection, maze solving through the Left-Hand Rule, and dynamic route re-planning, thus demonstrating how the main ideas of mobile-robotics may be applied in order to facilitate more advanced and adaptive autonomous operations.

Arduino Nano

Arduino Nano is a small board grounded on ATmega328 with numerous applications and is used in smaller applications like triggering the shutter of a digital camera or driving an LCD screen. Banzi and Shiloh (2014) mention that the Arduino boards are designed to be easily used in embedded systems and electronics

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

prototyping. These allow the users to read sensor values, switch devices and run programming code in real time. The Nano model is specifically applicable in the micro-scales projects due to its small size and low power usage as well as the availability of digital and analog I/O pins. These features make it perfect to use mobile robotics where space efficient and usefulness is the main concern.

As discussed in the framework of the current project, the Arduino Nano serves as the primary control unit of an autonomous mobile robot with a differential drive configuration. It processes signals from infrared line sensors and an obstacle detection module to control the DC motors directly. The simplicity and accessibility of the Arduino Nano, as emphasized by Banzi and Shiloh (2014), allow the implementation of control algorithms such as the Left-Hand Rule for maze navigation while integrating sensor feedback and motor actuation within a single embedded platform. Therefore, the Arduino Nano plays a crucial role in achieving autonomous navigation and reliable obstacle detection in the proposed system.

Obstacle Avoidance

Katona, Neamah and Korondi (2024) explored different obstacle avoidance and path planning strategies of autonomous mobile robots with real-time navigation in dynamic worlds. They focused their research on discussing the application of sensor arrays, such as infrared and ultrasonic sensor, to determine the presence of obstacles and paths around them, which are safe. Among the algorithms mentioned by the authors, the possibility to allow the robots to respond to the moving obstacles and correct their path without compromising their efficiency and stability is worth mentioning. This study is applicable to the given research since the autonomous differential drive mobile robot also uses infrared sensors to scan the surrounding environment and determine avoidance maneuvers. The similarity of real-time decision-making principles implies that the current project will guarantee the robot will move safely along unknown paths and combine the capacity to follow the lines and solve a maze.

Embedded Systems

The study by Hughes (2016) is focused on practical application of the Arduino and open-source hardware in designing embedded systems in a variety of projects. It is stressed in the text that microcontrollers can be programmed to receive sensor readings, to drive mechanical devices, and to do live data calculation, and thus comprise the heart of an embedded system. These principles directly apply to the present research since the Arduino Nano is the core controller of the autonomous differential-drive mobile robot; it allows the robot to respond to line-followers and obstacle-detector modules. Utilizing the ideas of Hughes, the current research incorporates elements of sensors, motors, and controls logic so as to produce an entire embedded system through which the robot will move autonomously, avoid obstacles, and carry out decision-making algorithms including the Left-Hand Rule.

Table 1. Comparison Matrix of Related Studies and Current Research

Study	Sensors Used	Controller	Main Outputs	Scope	Key Features	Gap Addressed by This Study
Banzi & Shiloh (2014)	IR Sensor ,light (LDR), temperature Sensor	Arduino nano	LEDs, motors, displays	Educational & prototyping	Easy-to-use, open-source, integrates sensors & actuators	No autonomous robot, no obstacle avoidance or maze solving
Deshpande & Mazumdar (2016)	3x IR line sensors	Arduino Uno	DC motors	Line-following robot	Autonomous line tracking using sensor feedback	Only line following, no dynamic obstacle avoidance or

						maze-solving
Hughes (2016)	IR, light, temperature, touch, ultrasonic	Arduino Uno and Arduino Nano	LEDs, motors, buzzers, displays	Practical Arduino projects	Step-by-step Arduino projects; integrates sensors & actuators	No fully autonomous navigation or decision-making robot
Monk (2016)	IR, ultrasonic, photoresistors, buttons	Arduino Uno or Nano	Motors, LEDs, displays	Advanced Arduino programming projects	Advanced C++ programming, sensor-actuator integration	No combined line-following, maze-solving, and obstacle avoidance system
Katona, Neamah & Korondi (2024)	IR, ultrasonic	Microcontroller board	DC motors	Mobile robot obstacle avoidance & path planning	Real-time obstacle detection, dynamic path planning	Focused only on obstacle avoidance; no line-following or maze-solving
Current Study(Autonomous Differential Drive Mobile Robot with Obstacle Avoidance Using Arduino Nano)	3x IR line sensors, IR obstacle sensor	Arduino Nano	DC motors, LEDs, LCD, buzzer	Autonomous differential drive mobile robot	Combines line-following, Left-Hand Rule maze solving, and obstacle avoidance	Integrates all navigation functions in one embedded mobile robot

METHODOLOGY

The development of the mobile robot involves the integration of mechanical structure, electronic circuitry, and software algorithms. The system is designed as a closed-loop control system where sensor inputs are processed to determine actuator outputs.

- The robot operates on a modular hardware architecture centered around the Arduino Nano microcontroller.
- Chassis and Actuation: The physical structure uses a differential drive configuration. Movement is produced by two DC gear motors that independently drive the wheels. Directional control is achieved through differential or skid steering, where the microcontroller adjusts the speed and direction of each motor using Pulse Width Modulation signals. By controlling the motors independently, such as running one motor forward while reversing the other, the robot can perform left and right turns as well as zero-radius pivot turns. This method allows the robot to maneuver effectively in narrow and enclosed environments.
- Motor Driver: To interface the microcontroller with the motors, an L298N Dual H-Bridge Driver is utilized. This module draws power from the battery to drive the motors, while the Arduino regulates speed via Pulse Width Modulation (PWM) signals.
- Sensor Array: Perception is achieved through specific infrared modules:

- Navigation: Three IR Line Sensors are mounted on the underbelly to detect path contrast (Black vs. White).
- Safety: A Flying Fish IR Obstacle Sensor is mounted on the front chassis (Top position facing forward). It acts as a hardware interrupt for the safety system.
- Human-Machine Interface (HMI): A 16x2 LCD serves as the system dashboard, displaying real-time logic states. Additionally, a toggle switch serves as a hardware kill-switch to safely cut logic execution.

Software Implementation The control logic is developed in C++ within the Arduino IDE. The software architecture utilizes a non-blocking loop structure, employing the millis() function to manage LED indicators without halting the main navigation logic.

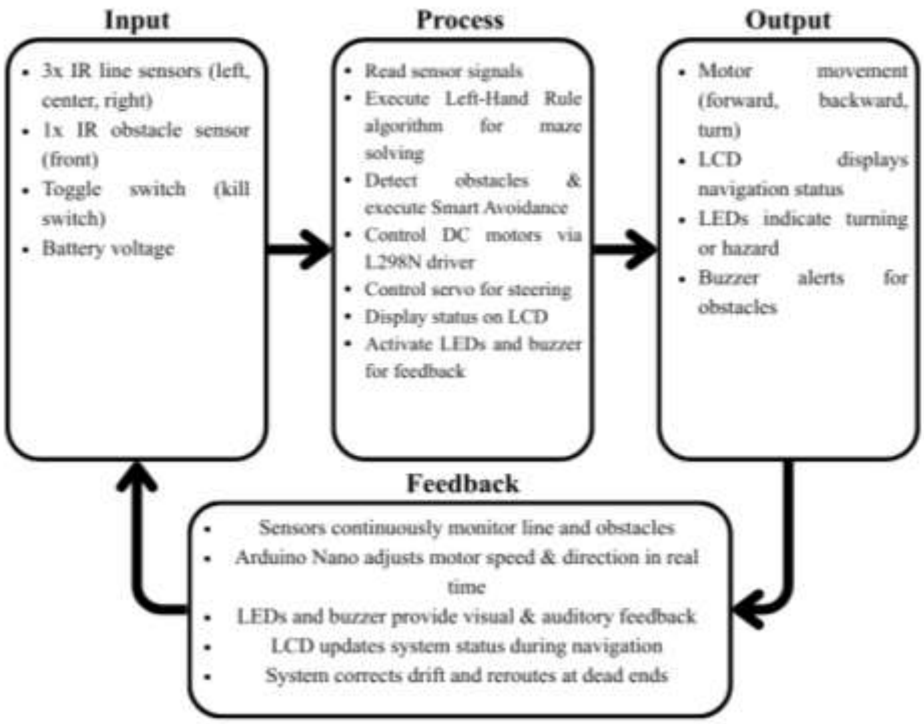
The navigation algorithm implements a "Smart Avoidance" strategy combined with the Left-Hand Rule. The decision hierarchy is prioritized as follows:

1. Obstacle & Goal Analysis: The system first checks the forward obstacle sensor. If triggered (LOW):

- Goal Check: If all three floor sensors also detect black lines, the robot identifies this as the "Finish Zone" and triggers a 15-second celebration routine.
- Smart Avoidance: If it is not the finish line, the robot scans the side sensors. If a left or right path is open, it executes a differential turn (spinning wheels in opposite directions) to bypass the obstacle. If no path is found, it executes a U-Turn to reroute.

2. Intersection Analysis: If the path is clear, the system reads the floor sensors. If a cross-junction is detected (Left, Center, and Right active), the algorithm defaults to a left turn to maintain wall-following behavior.
3. Dead End Recovery: In the absence of any line detection, the robot identifies a "Dead End" state. It then executes a pivot maneuver—reversing the direction of one motor relative to the other—to re-orient the chassis 180 degrees.

Figure 1. Input–Process–Output (IPO) Model



ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

The autonomous mobile robot is designed around the Input-Process-Output (IPO) model, as shown in Figure 1. This framework guides the flow of data from sensing the environment to physical movement and feedback.

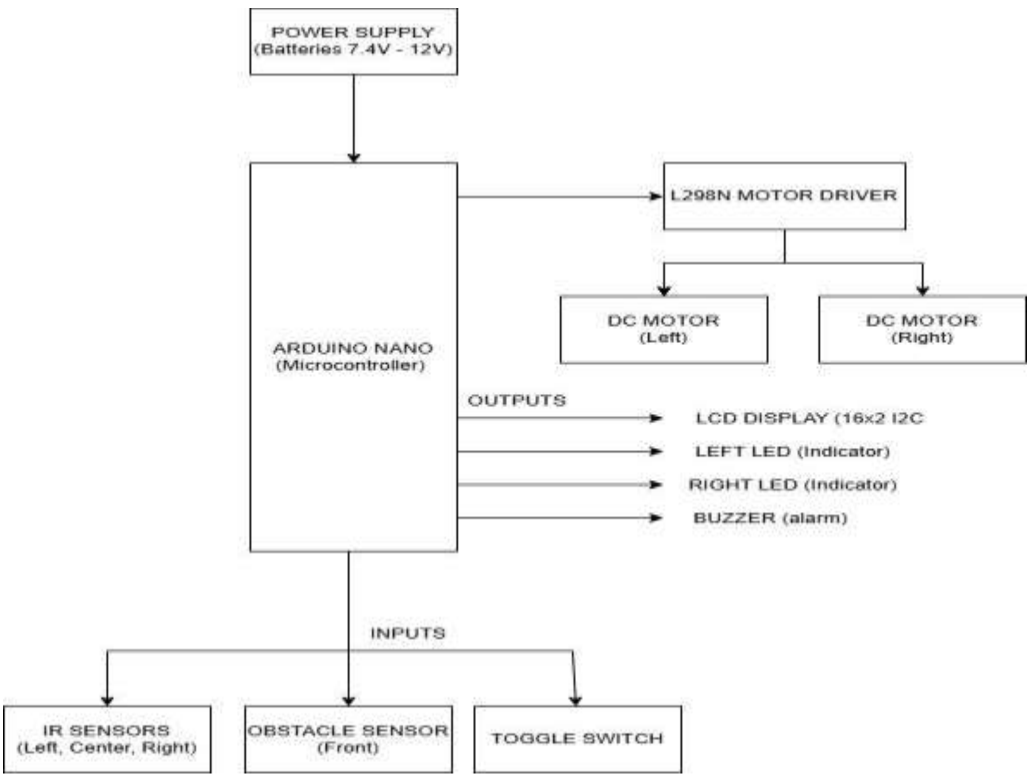
The Input stage handles how the robot perceives its surroundings. It relies on three infrared (IR) line sensors to detect the contrast between the black navigation line and the white floor, sending digital signals to the microcontroller. To ensure safety, a forward-facing 'Flying Fish' IR sensor acts as a proximity detector, triggering an immediate signal if an obstacle is found within range. Additionally, a manual toggle switch allows the user to start or stop the autonomous system, while the battery supplies the necessary power to drive the logic and motors.

The embedded C++ program on the Arduino Nano microprocessor manages the process stage. The microprocessor reads the status of the sensors and creates a current snapshot of where the robot is on the track (position) in real-time. Once it has this information, it will run a dual-layer program: Layer One uses the "Left Hand Rule" to solve mazes by using a left turn first at intersections, while Layer Two performs "Smart Avoidance" to find an alternate route around the obstacle if it is present, such as turning to the side or making a U-turn. Based on the results from both layers, the microprocessor determines how much differential rotational speed and/or direction will be needed for each motor to make the movement (i.e., stop one motor while running the other to perform a pivot turn).

The output stage converts the digital instructions generated by the software into tangible responses and visual feedback. The actuation is accomplished via two DC geared motors that utilize a motor driver (L298N), which receives pulse-width modulation (PWM) input signals from the Arduino to control the amount of torque and speed generated. For observing the system's operation, a 16 x 2 character LCD with I2C connectivity displays in real time the system's current status using text such as "Navigating," or "Rerouting." In addition to the above-mentioned methods, the system utilizes LEDs and an active buzzer to generate immediate visual and audible notifications of the system performing a turn, detecting a hazard, and/or other operational events for use by the observer.

To complete the cycle, the system functions as a closed-loop feedback mechanism. The robot's physical movement alters its position, directly changing the sensor readings. This constant stream of data enables the Arduino Nano to make real-time adjustments, correcting for drift and ensuring the robot stays on course.

Figure 2. Block Diagram



ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

A system block diagram graphically depicts the connections between a robot's different functional subsystems and clearly defines where information and energy flow within the system. This will separate the subsystems into four basic categories (processing sub-system; power sub-system; input sub-system; output sub-system), and it will show how these four basic categories are connected so that both information and energy can flow. The total system's energy comes from an unregulated DC power source composed of lithium-ion batteries. This power source will provide both the high current necessary for the motor driver and the 5V logic voltage needed to run the microprocessor and the lower power sensor circuitry.

At the center of this system is the Arduino Nano micro-controller, which acts as the central processor. As such, it is responsible for all decisions; taking in digital input from the sensors and utilizing those inputs with the built-in "Left-Hand Rule" and "Smart Avoidance" algorithms to produce precise output commands for the actuators based on its decision-making process. In addition to the micro-controller, the Input Sub-system utilizes a sensor array to provide real-time environmental information to the micro-controller. This sensor array includes an array of IR Line Sensors (right, left, and center) to measure the track contrast to determine where the robot is in relation to the navigation line. Additionally, there are two other sensors included in the sensor array: a forward-facing Obstacle Sensor to provide a hardware interrupt signal when a physical obstruction is detected, and a toggle switch to act as a manual user interface, allowing the user to manually disable or enable the auto-logic loop, thus enabling a kill-switch functionality to be used to disable the system.

The Output and Actuation Subsystem is responsible for the actual motion of the robot and for the feedback needed by the user. The L298N motor driver serves as the link between the low-voltage logic from the Arduino board and the higher current requirements of the dc gear motors. The L298N accepts pulse-width modulation (PWM) input to enable the independent control of the speed and direction of rotation of each of the two motors, thereby enabling differential steering. A 16x2 LCD display with I2C enabled displays the current state of the robot in text format, e.g., "Navigating", "obstacle detected", etc. and can also use active LEDs and/or buzzer to alert the user visually and/or audibly during turns and obstacle detection.

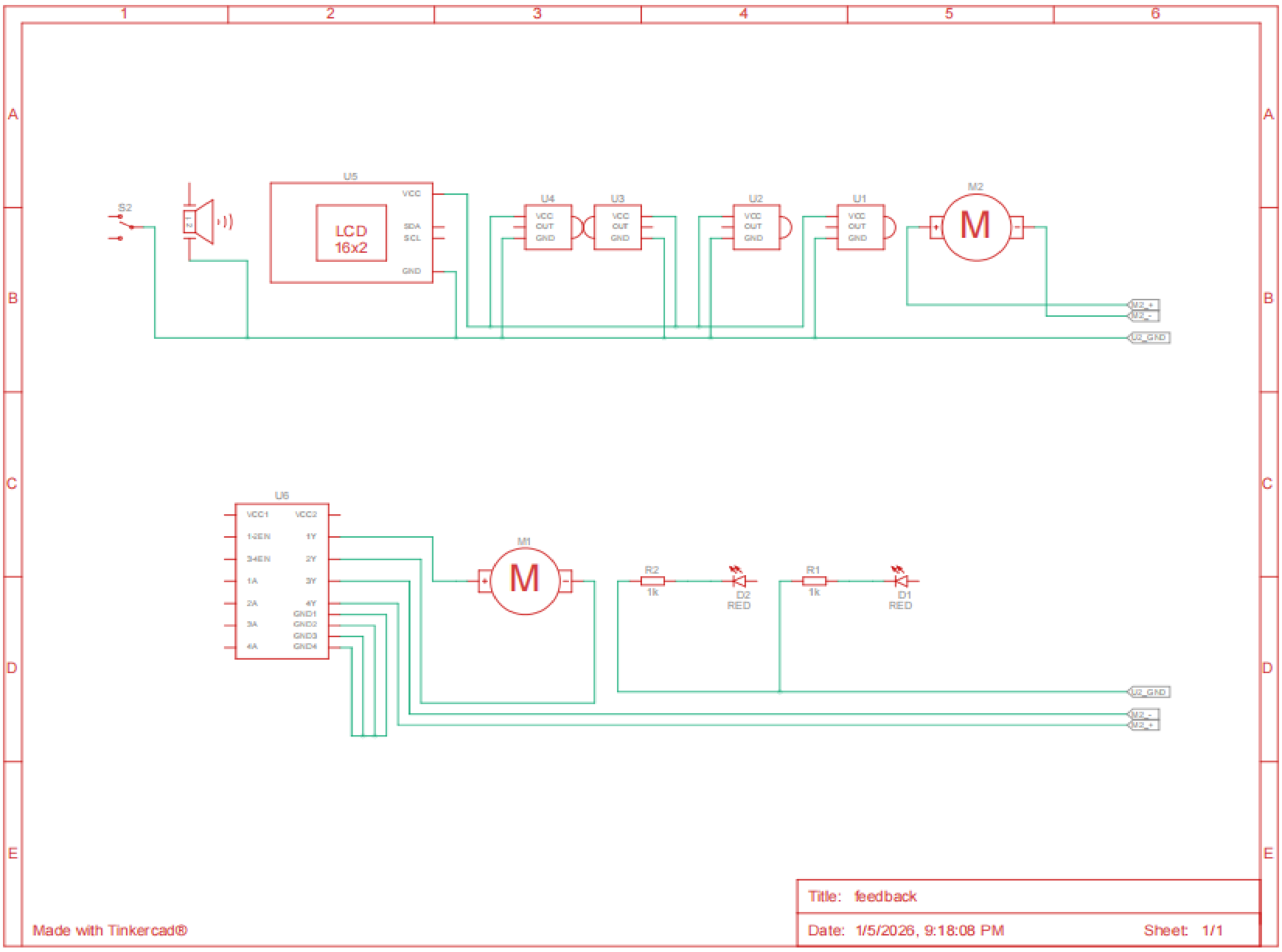


Figure 3. Schematic Diagram

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

The full schematic diagram displays all components that will be used in the Autonomous Mobile Robot System; it also demonstrates how each will work together to provide navigation, control and feedback of the system. The I2C-enabled LCD display is at the top of the diagram and serves as the robot's "dashboard" to show the user real time information about what the robot is doing (i.e., which way it is turning and if it detects an object). The push button shown on the right side of the diagram allows the user to manually enter a start/stop command to reset the system. The four Infrared sensors are located on either side of the diagram and are to be used to detect a line as well as detect obstacles along the path that the robot will follow.

The Arduino Nano (at the bottom) serves as the central processor unit that reads inputs from sensors and runs the navigation algorithms, controlling the motor driver circuit to propel the vehicle by operating the rear DC Motors. The two Resistors and Red LEDs (located at the bottom) are for visual indication of turn directions or Hazard States, while the multiple Ground Connections will keep the entire system stable during its operation. Overall this Schematic is an example of an embedded system where sensor inputs make the decisions, the Microcontroller makes the control decisions, and the Actuators respond accordingly; the responses also include LED and LCD Display for Feedback.

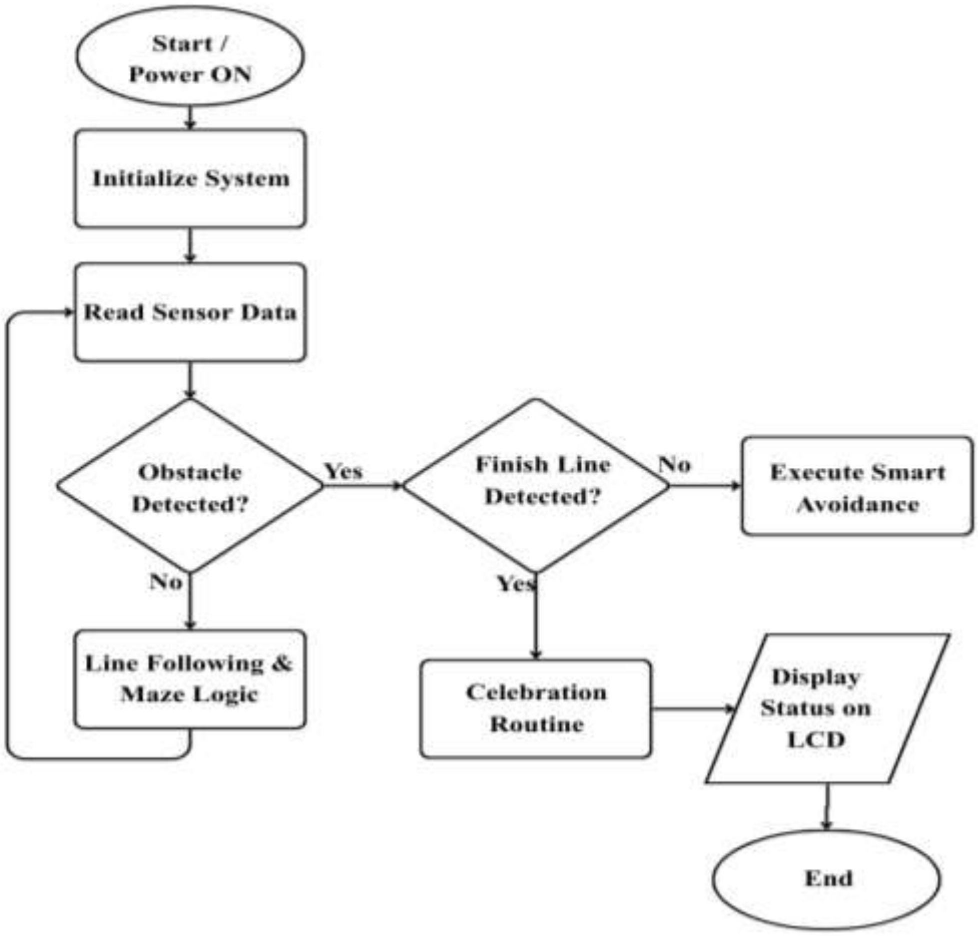


Figure 4. Flow Chart

The flowchart shows the operation of the autonomous mobile robot: how the system continuously senses the environment, decides on actions, and performs those actions. The process starts upon turning on the power supply to the robot, where Arduino Nano reads data from the IR line sensors and the front obstacle sensor. The first decision in the flow is if an obstacle has been detected. If there is, the robot enters into the obstacle avoidance routine by choosing a side turn or U-turn depending on the path available and completes the maneuver. It then updates its feedback through LCD, LEDs, and a buzzer back to reading sensors for a continuous closed-loop operation.

If no obstacle is detected, enter the maze navigation sequence. It will check the conditions: if it reached the finish zone, reached an intersection, or reached a dead end. Depending on this input, it goes forward, turns left,

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue I January 2026

or does a 180-degree pivot. All of them again pass through feedback to update the operator on the status of the robot and loop back to the sensor-reading stage.

In this view, the flowchart recognizes that the robot operates on a priority basis, first detecting obstacles over navigating a maze. Various operations are represented by shapes: rectangles for actions, diamonds for decisions, and a parallelogram for feedback. This helps to clearly distinguish the operation types. This representation effectively shows how the sensing, decision-making, and actuation capabilities of an embedded system are tied together in pursuit of fully autonomous navigation.

RESULTS AND DISCUSSION

The AMR (Autonomous Mobile Robot) was utilized to conduct experiments for the line-following task, maze solution by means of the Left-Hand Rule, and obstacle avoidance. During the majority of the trials, the robot was able to travel through intersections and avoid obstacles with the assistance of the IR sensors. Specifically, the center sensor assisted in maintaining course direction; while the left and right sensors compensated for drift from course. Additionally, the front IR sensor for obstacle detection allowed the robot to perform a right turn or a "U" turn when encountering an obstacle. All feedback components including the LCD, LED, and buzzer provided immediate indication of actions performed during the experiment.

The duration of each successful trial varied from 44 to 50 seconds. Many failed attempts demonstrated the need for precise calibration to achieve the desired result. Therefore, the proposed solution provides a seamless link of many tasks including line tracing, maze solving, and dynamic object avoidance, which has not been previously accomplished in other solutions that have limited their research to solve one function at a time. Therefore, the proposed solution utilizing the Arduino Nano module is capable of performing autonomous navigation throughout a predetermined setup.

Table 2. Variables and Conditions of the Autonomous Differential Drive Mobile Robot Using Arduino Nano

Variable / Component	Type (Input / Output)	Parameter Measured / Controlled	Condition or Range	System Response / Action
Left IR Line Sensor (A0)	Input	Line detection	High / Low (on black/white line)	Adjust motor speed for left turn or drift correction
Center IR Line Sensor (A1)	Input	Line detection	High / Low	Maintain forward movement if on line
Right IR Line Sensor (A2)	Input	Line detection	High / Low	Adjust motor speed for right turn or drift correction
Front IR Obstacle Sensor (Pin 4)	Input	Obstacle detection	LOW = obstacle detected, HIGH = clear	Execute Smart Avoidance (side turn or U-turn)
Toggle Switch	Input	System ON / OFF	ON / OFF	Start or stop the system safely
Arduino Nano	Processor	Decision making	N/A	Process sensor inputs and control outputs
DC Motors	Output	Wheel rotation	Forward / Backward / Stop	Move robot according to navigation algorithm
LCD 16x2	Output	Display status	Text messages (Turning Left,	Show real-time navigation

			Obstacle!)	info
LEDs (Red/Amber)	Output	Visual signaling	Blink Left / Blink Right / Both	Indicate turning direction or hazard
Buzzer	Output	Sound alert	Continuous / Beeping	Warn for obstacle detection or reverse action

Table 3. Performance Evaluation of the Autonomous Mobile Robot

Trial #	Time (s)	Intersection Handling	Obstacle Avoidance	Outcome
1	45s	Successful (Left Turn)	Avoided (Side Turn)	Success
2	--	Failed (Missed Line)	N/A	Fail
3	--	Failed (Missed Line)	N/A	Fail
4	50s	Corrected drift	Avoided (Side Turn)	Success
5	--	Failed (Missed Line)	N/A	Fail
6	44s	Successful (Left Turn)	Avoided (U-Turn)	Success
7	--	Failed (Missed Line)	N/A	Fail
8	46s	Successful (Left Turn)	Avoided (Side Turn)	Success
9	49s	Successful (Left Turn)	Avoided (U-Turn)	Success
10	45s	Successful (Left Turn)	Avoided (Side Turn)	Success

The mobile robot with an autonomous differential drive was evaluated over 10 experiments run in a controlled environment (indoor maze). The tests examined the systems' ability to follow a path on a line, use the Left-Hand Rule algorithm to navigate intersections, and utilize the front mounted IR sensor to avoid physical obstacles. Out of 10 experiments 6 ran successfully and at a rate of 60%. Successful experiment times ranged from 44-50 seconds. For all of the successful trial runs the robot was able to identify intersections properly and make the correct left turn. Obstacle avoidance occurred by performing a side turn or a "U" turn maneuver and avoiding collisions.

4 experiments failed due to the loss of line detection by the robot, this prevented the robot from continuing to follow the maze path. Loss of line detection demonstrates how sensitive the IR line sensors are to the position of the sensors, the reflective properties of the surfaces being navigated, and ambient light. It's possible some minor mechanical drift during the robot's turns also affected the accuracy of the sensor readings. These results clearly show that accurate sensor calibration, and enhanced path recognition will be important to increase system reliability.

Although the above problems were evident, the obstacle avoidance system operated properly in each of the experiments run. The front mounted IR sensor successfully recognized obstructions and initiated the proper course correction action. The LCD display and LED indicators offered real-time visual feedback supporting the operator's ability to observe the robots operating status. The results of the experiments clearly demonstrated the Arduino Nano based system is capable of autonomous maze navigation, and obstacle avoidance and therefore represents a viable low cost model for indoor Automated Guided Vehicles.

CONCLUSION

The autonomous differential drive robot, powered by an Arduino Nano, was successfully implemented for maze navigation and obstacle avoidance. By following the Left-Hand Rule, the robot managed intersections efficiently and used its front IR sensor to steer clear of obstructions. Trial results confirmed that it can perform left turns, U-turns, and side turns, showing it is capable of operating in unfamiliar settings. Real-time monitoring was made easy through the LCD display and LED indicators. Ultimately, this project proves that a budget-friendly Arduino system can achieve reliable autonomous navigation for indoor applications.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our research advisor for their continuous guidance, enthusiastic encouragement, and useful critiques of this research work. We also wish to thank the Department of Computer Engineering at the Eulogio "Amang" Rodriguez Institute of Science and Technology for providing the necessary laboratory facilities and resources required to complete this project. Finally, we extend our appreciation to our families and peers for their support and motivation throughout the development of this prototype.

REFERENCES

1. Banzi, M., & Shiloh, M. (2014). *Getting started with Arduino: The open source electronics prototyping platform* (3rd ed.). Maker Media, Inc.
2. Deshpande, A., & Mazumdar, A. (2016). *Design and implementation of line follower robot.* International Journal of Scientific and Research Publications, *6*(3), 321–324. http://www.ijsrp.org/research-paper-0316.php?rp=P515159
3. Hughes, J. (2016). *Practical Arduino: Cool projects for open source hardware*. Apress. https://doi.org/10.1007/978-1-4302-2478-5
4. Katona, K., Neamah, H. A., & Korondi, P. (2024). *Obstacle avoidance and path planning methods for autonomous navigation of mobile robot. Sensors*, *24*(11), 3573. https://doi.org/10.3390/s24113573
5. Monk, S. (2016). *Programming Arduino next steps: Going further with sketches*. McGraw-Hill Education.
6. Siegwart, R., & Nourbakhsh, I. R. (2011). *Introduction to autonomous mobile robots* (2nd ed.). MIT Press. https://doi.org/10.7551/mitpress/8421.001.0001

ABOUT THE AUTHORS

Engr. Bernard C. Fabro is a Professional Computer Engineer and A Professor at Eulogio "Amang" Rodriguez Institute of Science and Technology. He has over 15 years of teaching experience in computer engineering, specializing in robotics, programming, and control systems. His research interests include automation, deep learning applications, and smart systems, with several published works in international conferences and journals.

The authors, [1]Carl Danielle C. Andante, [2]Selwyn James S. Jimenez, [3]Janine P. Natalia, and [4]Kyla Marie G. Septimo, are currently pursuing their Bachelor of Science degrees in Computer Engineering at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST), Manila, Philippines.

As a research group, their collective interests focus on the development of autonomous systems, embedded logic control, and the practical application of robotics in solving navigational problems. This project serves as a partial fulfillment of their degree requirements.