A Smart IR-Controlled Autonomous Vehicle Using Arduino Uno

Parastoo Shojaei Sarcheshmeh
University of Northampton
Northampton, UK
parastoo.shojaeisarcheshm25@my.northampton.ac.uk

Abstract— **This paper focuses on design and implementation of a smart IR-controlled autonomous vehicle using Arduino Uno. The system is designed to work in two different modes: manual mode and autonomous mode. In manual mode, the vehicle is controlled using an infrared (IR) remote control, allowing the user to move the robot forward, backward, left, right, or stop. In autonomous mode, the vehicle uses an ultrasonic distance sensor to detect obstacles and automatically change its movement to avoid collisions. A DC motor driver (L293D) is used to control two DC motors for differential drive movement. The ultrasonic sensor continuously measures the distance in front of the vehicle, and when an obstacle is detected, the system stops and performs a predefined avoidance routine. A push button is used to switch between manual and autonomous modes. An LCD module is added to display the current mode, movement direction, and distance information in real time.**

Keywords— autonomous vehicle, Arduino Uno, IR remote control, ultrasonic sensor.

I. INTRODUCTION

In recent years, small autonomous vehicles have been used in many projects to test basic ideas of movement, sensing, and control. These vehicles are usually designed to move in environment, detect obstacles, and react to them in a simple and safe way.

In many situations, a vehicle needs both manual control and automatic control. Manual control is useful when the user wants full control of the movement, while automatic control is helpful when the vehicle needs to move by itself without constant human input. For this reason, combining both modes in one system makes the vehicle more practical and flexible.

This project presents a simple autonomous vehicle built using Arduino Uno. The vehicle can be controlled manually using an IR remote, or it can move automatically using an ultrasonic sensor to detect obstacles. By switching between these two modes, the system can be used in different environments and for different tasks.

II. LITERATURE REVIEW

There are several examples of autonomous vehicles and robotic systems designed for various purposes, ranging from industrial applications to small-scale projects. Most of these systems use DC motors for movement and one or more sensors to detect obstacles and make basic movement decisions. In addition, many projects also include manual control methods such as Bluetooth or IR remote control to give the user direct control when needed.

Kuveskar et al. [1] developed an Arduino-controlled robotic vehicle with multiple operating modes, including Bluetooth control, voice control, and autonomous navigation. The system was based on an Arduino Uno and used an L293D motor driver to control DC motors. Ultrasonic and IR sensors were used for obstacle detection, while an ESP32-CAM module was added for real-time video streaming and basic object recognition. The robot was designed to operate in different scenarios such as surveillance, assistance for disabled users, and smart transportation. Although the system provides advanced features such as camera-based detection and mobile application control, it requires additional hardware modules and higher computational resources.

Baballe et al. [2] developed an obstacle avoidance robot using an Arduino Uno and an ultrasonic sensor. In this system, the ultrasonic sensor is used to detect objects in front of the robot, and based on the measured distance, the robot moves forward or backward to avoid collisions. Two DC motors are controlled using a motor driver, and LEDs are used to show the movement status of the robot. The control logic is simple and focuses only on autonomous movement without any manual control option.

Sati et al. [3] presented a human-following robot using Arduino Uno with ultrasonic and IR sensors. In this system, the ultrasonic sensor is used to measure the distance from a person or object, while IR sensors help in detecting movement direction and avoiding obstacles. The robot continuously follows a human target while maintaining a safe distance and stopping when an obstacle is detected. DC motors controlled by an L293D motor driver are used for movement, and all processing is done on the Arduino microcontroller. The system is designed mainly for assistance tasks such as carrying loads in hospitals, libraries, and public spaces. Although the robot performs well in following humans, it focuses only on autonomous operation and does not provide a manual control option for the user.

Jabines and Perin [4] designed a fully autonomous obstacle avoiding robot using Arduino Uno and the HC-SR04 ultrasonic sensor. In this system, the ultrasonic sensor continuously measures the distance in front of the robot, and when an obstacle is detected, the robot automatically changes its direction to avoid collision. DC motors are controlled using a motor driver module, and all control logic is implemented using the Arduino IDE. The robot is able to navigate in unknown environments without human input, and the hardware components used are low-cost and easy to replicate. However, the system only supports autonomous operation and does not provide any manual control option for the user.

Bisen et al. [5] developed a multi-functional obstacle avoidance robot car based on the Arduino platform. The robot uses ultrasonic sensors to detect obstacles and automatically change its movement to avoid collisions. In addition, the system integrates an ESP32 camera with YOLO object recognition to identify and classify objects in real time. DC motors and servo motors are used for movement and steering, and all control logic is implemented using the Arduino microcontroller. The system is designed for advanced autonomous navigation and includes complex algorithms for decision making. Although this project provides high accuracy in obstacle detection and visual perception, it requires additional hardware, higher processing power, and more complex software.

Shelke et al. [6] designed an obstacle avoidance vehicle using Arduino Uno and ultrasonic sensors. In this system, the ultrasonic sensor is used to detect obstacles in real time, and the Arduino processes the distance data to control DC motors through a motor driver shield. When an obstacle is detected, the vehicle stops, reverses slightly, and then turns left or right to find a new path. The system focuses on fully autonomous navigation and is designed to be low-cost, easy to implement, and suitable for educational projects. The authors also highlight the use of simple hardware components to achieve reliable obstacle avoidance in different environments. However, the system does not include a manual control option for the user and works only in autonomous mode.

Mahmud et al. [7] designed an ultrasonic sensor-based obstacle avoidance system for Arduino robots. In this project, ultrasonic sensors are placed at the front of the robot to continuously measure the distance from nearby obstacles. The collected data is processed by the Arduino microcontroller, which controls the motors and automatically changes the robot's direction to avoid collisions. The authors also used simple artificial intelligence and decision-making algorithms to improve the accuracy of obstacle detection, achieving an avoidance accuracy of about 84%. The system is designed for fully autonomous operation and is suitable for applications such as industrial automation, surveillance, and safe navigation in unknown environments.

Review of the existing systems shows that most of the projects focus mainly on fully autonomous obstacle avoidance using ultrasonic sensors, cameras, or advanced algorithms, and many of them do not include a simple manual control option. Some systems are designed with complex hardware such as cameras, AI modules, or multiple sensors, which increases the cost and complexity of implementation. Other projects provide only manual control or only automatic navigation, which limits their flexibility in different environments.

The proposed project focuses on combining both manual and autonomous control in a single low-cost system. The vehicle can be controlled manually using an IR remote, while also supporting autonomous obstacle avoidance using an ultrasonic sensor. Unlike many existing systems, this project allows the user to switch between modes depending on the task and environment. In addition, the system uses a simple and reliable avoidance routine and does not rely on complex hardware or heavy processing, making it suitable for educational use and basic real-world applications.

III. Design and Implementation

In this section, the design and implementation stages of the proposed vehicle are explained step by step. The process includes the control algorithm, working concept, hardware components, software design, and testing procedure. The system is implemented in an iterative way to make the vehicle work correctly in both manual and autonomous modes.

A. *Algorithm*

1. The system starts, and the vehicle enters autonomous mode automatically. The LCD displays the initial mode and distance.
2. The ultrasonic sensor continuously measures the distance in front of the vehicle to detect obstacles.
3. In autonomous mode, the vehicle moves forward automatically while monitoring the distance from obstacles.
4. If an obstacle is detected within a predefined safety distance, the system performs an avoidance routine in which the vehicle stops immediately, turns right to find a free path, and finally continues moving forward.
5. If no obstacle is detected, the vehicle continues moving forward in a straight direction.
6. When the mode selection button is pressed, the system switches to manual mode.
7. In manual mode, the vehicle is controlled using an IR remote control. The user can command the vehicle to move forward, backward, left, right, or stop.
8. While in manual mode, the ultrasonic sensor continues measuring distance. If an obstacle becomes too close, the vehicle stops automatically to prevent collision.
9. The LCD is updated continuously to show the current mode, movement direction, and distance information.

B. *Working Concept*

The goal of this project was to design a smart vehicle that can operate in both manual and autonomous modes using simple and low-cost hardware. The system was built using an Arduino Uno board with an ATmega328P microcontroller, DC motors, an L293D motor driver, an ultrasonic distance sensor, an IR receiver with a remote control, a mode selection button, and an LCD for status display.

The vehicle's main function is to move in an environment while detecting obstacles and reacting to them in a safe and controlled way. In autonomous mode, the ultrasonic sensor continuously measures the distance in front of the vehicle. When an obstacle is detected within a safe range, the vehicle stops, moves backward briefly, turns right to avoid the obstacle, and then continues moving forward. This routine allows the vehicle to navigate in unknown spaces without human input.

In manual mode, the vehicle is controlled by the user using an IR remote control. The user can move the vehicle forward, backward, left, or right, or stop it at any time. While the vehicle is under manual control, the ultrasonic sensor remains active as a safety feature. If an obstacle becomes too close, the system automatically stops the vehicle to prevent collision. A push button is used to switch between autonomous and manual modes, making the system flexible for different situations.

The LCD module provides real-time feedback by displaying the current operating mode, movement direction, and distance from obstacles.

C. Flowchart Design

The flowcharts were designed based on the control algorithm explained in the previous section.

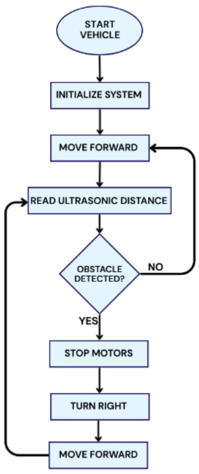

Fig. 1. Autonomous mode Flowchart Design

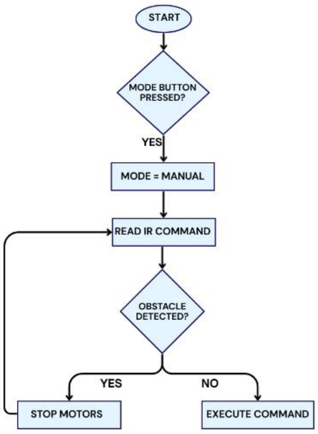

Fig. 2. Manual mode Flowchart Design

Fig. 1 shows the autonomous operation of the vehicle.

Fig. 2 shows the manual control mode and the mode switching logic.

D. Hardware Used

- Arduino Uno (ATmega328P)

The Arduino Uno board is used as the main controller of the system. It is based on the ATmega328P microcontroller and is responsible for handling all inputs and outputs in the vehicle. In this project, the Arduino reads data from the ultrasonic sensor, receives commands from the IR receiver, controls the DC motors through the motor driver, and updates the LCD display. The board provides 14 digital input/output pins and 6 analog input pins, which makes it easy to connect multiple components at the same time.[8] All connections were made according to the standard Arduino Uno pinout shown in Fig. 4.

Fig. 3. Arduino Uno board

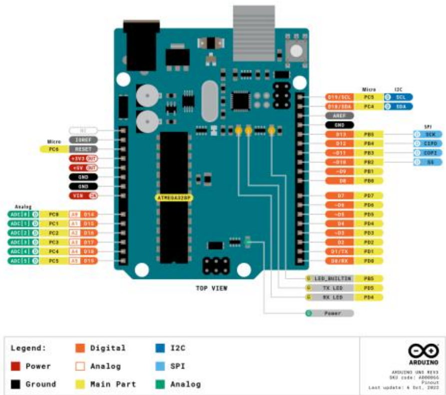

Fig. 4. Arduino Uno pinout

- DC Motors and H-Bridge Motor Driver (L293D)

DC motors convert electrical energy into mechanical motion by using electromagnetic principles. When voltage is applied to the motor terminals, the motor shaft begins to rotate, which allows the wheels of the vehicle to move. By changing the direction of the current, the direction of rotation can also be changed, making DC motors suitable for forward and backward motion.

In this project, two DC motors are used to drive the left and right wheels of the vehicle. To control these motors safely, an L293D H-bridge motor driver is used. The motor driver acts as an interface between the Arduino and the motors, because the Arduino cannot supply enough current to drive the motors directly. The L293D receives low-power control signals from the Arduino and amplifies them to drive the motors with sufficient current.[9] It also allows the direction of each motor to be controlled independently by changing the logic level of the input pins.

Using the L293D motor driver makes it possible to control the movement of the vehicle in different directions such as forward, backward, left, and right.

Fig. 5. DC Motors

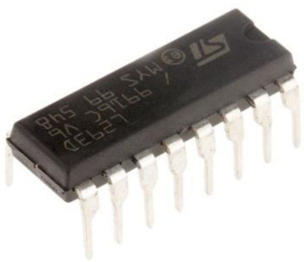

Fig. 6. H-Bridge Motor Driver (L293D)

- Ultrasonic Distance Sensor

An ultrasonic distance sensor is used to measure the distance between the vehicle and nearby objects. They work by emitting a high-frequency pulse and measuring the time it takes for it to bounce back after it hits an object. Based on the time it takes for the sound to return, it is possible to calculate the distance to the object. In this project, the ultrasonic sensor is mounted at the front of the vehicle and is used in both autonomous and manual modes. In autonomous mode, it helps the vehicle avoid obstacles by triggering the avoidance routine. In manual mode, it acts as a safety mechanism that stops the vehicle if an object is too close.

Fig. 7. Ultrasonic Distance Sensor

- IR Receiver and Remote Control

The IR receiver is a device that detects infrared signals transmitted from a remote control. It works by decoding the modulated infrared light sent by the remote and converting it into digital signals that can be read by a microcontroller.

In this project, the IR receiver allows the user to manually control the vehicle. Commands such as forward, backward, left, right, and stop are sent from the remote and processed by the Arduino Uno.

Fig. 8. IR Sensor

Fig. 9. Remote Control

- LCD 16×2 Display

A 16×2 liquid crystal display (LCD) is used to show information about the system during operation. The display can show 16 characters per line and two lines at a time, making it suitable for simple text-based information. In this project, the LCD is used to display the current operating mode, the movement direction, and the distance measured by the ultrasonic sensor.

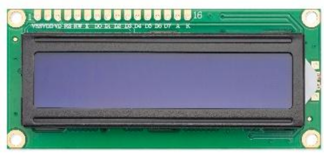

Fig. 10. LCD 16×2 Display

- Power Supply (9V Battery)

The system is powered using a 9V battery, which provides a portable and convenient power source for the vehicle.

Fig. 11. 9V Battery

- Push Button

A push button is used as a simple input device to allow the user to switch between autonomous and manual modes. When the button is pressed, a signal is sent to the Arduino, which then changes the operating mode of the vehicle. The button is connected using a pull-up configuration to prevent false triggering due to noise or unstable signals.

Fig. 12. Push Button

E. Circuit Description

The proposed system is controlled using an Arduino Uno. The Arduino reads sensor data, executes the control algorithm, and generates control signals for the motors and display in real time. The complete circuit schematic and hardware layout of the system are shown in Fig. 13 and Fig. 14.

The input pins of the motor driver are connected to the Arduino digital pins D6, D5, D10, and D9, which allows the direction of rotation of each motor to be controlled independently.

An ultrasonic uses a single signal pin that is connected to digital pin D7 of the Arduino Uno. The Arduino sends a short trigger pulse to the sensor and then measures the time of the returning echo to calculate the distance to an obstacle.

Manual control of the vehicle is achieved using an IR receiver connected to digital pin D11. The receiver detects infrared commands sent from a remote control, and these commands are decoded by the Arduino to control the movement of the vehicle.

The LCD control and data pins are connected to digital pins D13, D12, D8, D4, D3, and.

A push button is connected to analog pin A0 and is used to switch between manual and autonomous modes.

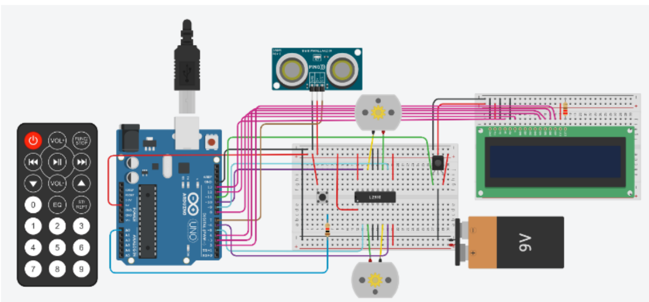

Fig. 13. Tinkercad simulation

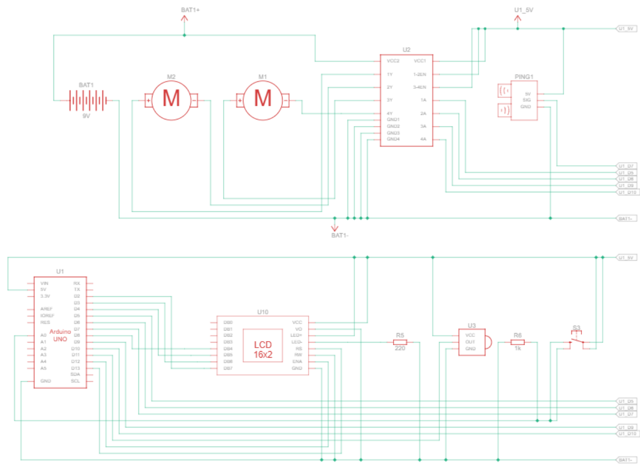

Fig. 14. circuit schematic diagram

F. Software Program

The code is divided into small parts, and each part does one specific job, such as reading sensors, moving the motors, or updating the display.

The main loop runs continuously and checks the system state without stopping the program.
Two different modes are handled by the software. In manual mode, the vehicle follows commands from the IR remote. However, before moving forward, the program always checks the distance to make sure the path is safe. If an obstacle is too close.
In autonomous mode, the vehicle moves on its own and follows a fixed sequence of actions when an obstacle is detected. The software first stops the vehicle, then turns, and finally continues moving forward.

G. Testing

- *Autonomous Mode Test*

When the simulation was started, the system entered AUTONOMOUS mode by default, as defined in the program. The LCD displayed the initial mode and the vehicle started moving forward automatically.
During this test, no obstacles were placed in front of the vehicle. The motors rotated in the forward direction and the vehicle moved straight without interruption.

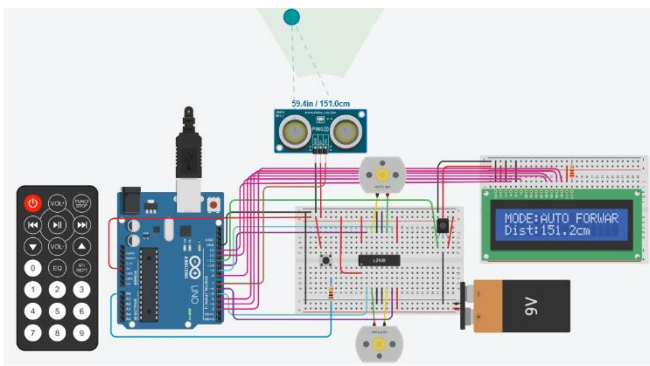

Fig. 15. System in autonomous mode

- *Autonomous Mode Obstacle Detection and Test*

In this test, a virtual obstacle was placed in front of the ultrasonic sensor while the vehicle was moving in autonomous mode. When the measured distance dropped below 25 cm, the system detected the obstacle and immediately stopped the motors.
After stopping, the vehicle turned right for a fixed time and then continued moving forward.

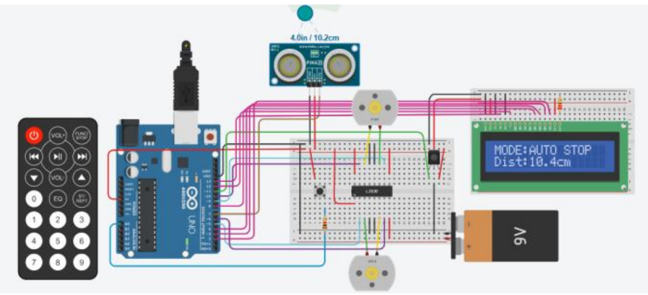

Fig. 16. Obstacle detection in autonomous mode

- *Mode Switching from Autonomous to Manual Test*

The mode switching function was tested by pressing the push button in the simulation. When the button was pressed, the system switched from autonomous mode to manual mode. The motors stopped immediately, and the internal states were reset to avoid unexpected movement.
The LCD display was updated instantly and showed that the system was now operating in MANUAL mode.

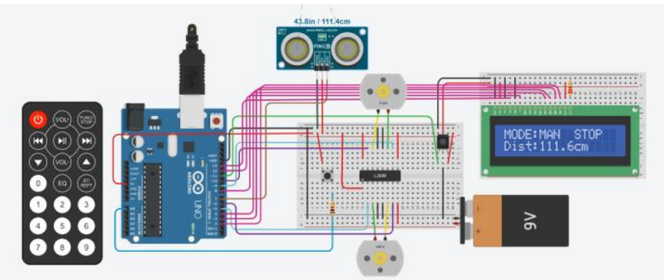

Fig. 17. Switching from autonomous to manual mode using push button

- *Manual Mode IR Control and Safety Test*

After switching to manual mode, the vehicle was controlled using the simulated IR remote. Commands such as forward, backward, left, right, and stop were sent to the system.
The vehicle responded correctly to all commands. To test the safety function, a virtual obstacle was placed in front of the ultrasonic sensor while the forward command was active. When the distance became less than 25 cm, the safety lock stopped the motors automatically.

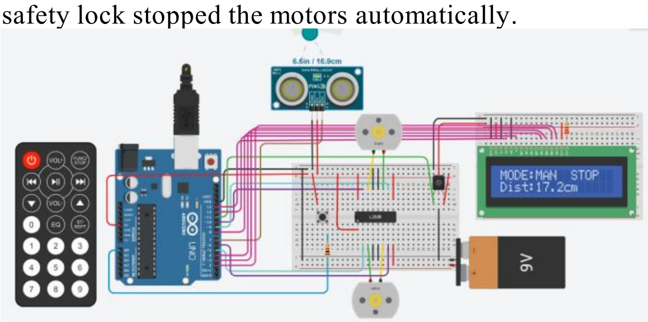

Fig. 18. Manual control and safety stop in simulation

IV. Observations And Limitations

During the Tinkercad simulation, the system worked correctly, but some practical limitations were observed:

1) The ultrasonic sensor used in this project measures distance only in a straight line in front of the vehicle. During simulation, it was observed that obstacles placed at sharp angles were not always detected correctly. This means that the vehicle can only react to obstacles that are directly in its forward path. Using multiple ultrasonic sensors or adding side sensors could improve obstacle detection and allow more accurate navigation.

2) In autonomous mode, the avoidance routine always follows the same sequence and always turns right. This is simple and works in basic situations, but in more complex layouts it may not be the best choice. For example, if the right side is blocked, the vehicle will still turn right and may need several cycles to escape.

V. Conclusion

The prototype worked correctly during the simulation and behaved as expected in most situations. The vehicle could move in both autonomous and manual modes, detect obstacles, and react to user commands without major problems. This shows that the basic idea of the project is working and the system logic is correct.

The project is simple and low cost. Some parts of the system can still be improved. More sensors or better control logic can be added later to make the movement more accurate and flexible.

Thinkercad Link (Function based):

https://www.tinkercad.com/things/76TRJFdnHKa-projectfunction?sharecode=mWa-bHeXG_40UwNrSl-SeItGsu9nSKFOS0ILSpXFmFc

References

[1] M. Kuveskar, K. Somani, O. Kore, and S. Pawar, “Arduino controlled vehicle,” International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS), vol. 7, no. 5, pp. 2517–2521, May 2025.

[2] M. A. Baballe, M. I. Bello, S. H. Ayagi, and U. F. Musa, “Obstacle avoidance robot using an ultrasonic sensor with Arduino Uno,” Global Journal of Research in Engineering & Computer Sciences, vol. 3, no. 5, pp. 14–25, Oct. 2023, doi: 10.5281/zenodo.10015177.

[3] D. Sati, S. Avkirkar, R. Pandey, and A. Somnathe, “Human following robot using Arduino,” International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), vol. 4, no. 2, pp. 347–350, Apr. 2021, doi: 10.48175/IJARSCT-1025.

[4] J. A. L. Jabines and M. A. D. Perin, “Obstacle avoiding robot using Arduino Uno and HC-SR04 ultrasonic sensor,” Research Project Presentation for Bachelor of Science in Computer Engineering, Bohol Island State University, Philippines, 2022.

[5] D. Bisen, A. Makode, J. Itiwale, S. Warade, and T. Malewar, “Multi-functional obstacle avoidance Arduino robot car,” International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol. 13, no. 1, pp. 205–211, Jan. 2024, doi: 10.17148/IJARCCE.2024.13128.

[6] S. M. Shelke, A. Nevase, A. Mail, and K. Mandhare, “Obstacle avoidance vehicle using Arduino,” International Research Journal of

Modernization in Engineering, Technology and Science (IRJMETS), vol. 7, no. 5, May 2025.

[7] T. Mahmud, I. Ara, R. Chakma, K. Barua, D. Islam, M. S. Hossain, A. Barua, and K. Andersson, “Design and implementation of an ultrasonic sensor-based obstacle avoidance system for Arduino robots,” in Proc. 2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, Sep. 2023, pp. 1–5.

[9] Arduino.cc. (2026). Available at: https://docs.arduino.cc/libraries/l293d/#Releases [Accessed 20 Jan. 2026].

[8] Arduino (2025). UNO R3 | Arduino Documentation. [online] docs.arduino.cc. Available at: https://docs.arduino.cc/hardware/uno-rev3/.