

Low-Cost Intelligent Robot Car with Autonomous and Manual Wireless Navigation Systems

Swati Shelar^{1*}, Punam Warke²

¹Department of Computer Science, Dr. D. Y. Patil, Arts, Commerce and Science College Pimpri, Pune, Maharashtra, India

²Department of Electronics, Dr. D. Y. Patil, Arts, Commerce and Science College Pimpri, Pune, Maharashtra, India

DOI: <https://doi.org/10.51583/IJLTEMAS.2025.1413SP012>

Received: 26 June 2025; Accepted: 30 June 2025; Published: 23 October 2025

Abstract: This research paper introduces an abstract avoidance system controlled via Bluetooth and voice commands using an Arduino board. Ultrasonic sensors detect obstacles, providing real-time data for the system. Users can remotely control the system's movements and navigate through complex environments using a mobile device connected via Bluetooth. Additionally, voice commands enhance usability and convenience. The integration of hardware components, including ultrasonic sensors, an Arduino board, and a Bluetooth module, along with algorithm development for obstacle detection and communication protocols, enables the system's functionality. This versatile system finds applications in robotics, automation, and smart environments where obstacle avoidance is crucial. By combining Bluetooth and voice control, this project offers an efficient and user-friendly solution for enhancing control and safety in various real-world scenarios.

Keywords: Obstacle, Robot car, Bluetooth and voice control.

I. Introduction

Building an obstacle avoidance, Bluetooth, and voice control robot car using Arduino is an educational and captivating project that merges robotics, programming, and electronics. [1-5]. By assembling a robot car, integrating sensors, connecting them to an Arduino board, and developing code, users can create a versatile vehicle capable of autonomously evading obstacles while also being remotely controlled through Bluetooth and voice commands. This project offers several advantages, including fostering an understanding of robotics, honing programming skills, and nurturing problem-solving abilities. It also provides hands-on experience with hardware assembly, wiring, and troubleshooting, encouraging practical knowledge in electronics. The flexibility of an Arduino-based robot car opens up numerous applications. It can be utilized for surveillance, industrial automation, search and rescue missions, and home automation tasks. [6-9] In educational settings, it serves as a valuable tool for teaching robotics and electronics to students, promoting engagement and curiosity. Furthermore, it provides practical experience for aspiring engineers in fields such as autonomous vehicles, smart cities, and environmental monitoring. In summary, an obstacle avoidance, Bluetooth, and voice control robot car using Arduino offers an immersive exploration of the realm of robotics and automation. It nurtures creativity, innovation, and problem-solving skills while delivering practical proficiencies in programming and electronics. the robot car can be controlled via Bluetooth, allowing users to interact with it using a smartphone or other Bluetooth-enabled devices. Furthermore, integrating voice control into the system enhances the user experience, enabling hands-free operation and increased accessibility.

This project showcases the integration of different technologies, including microcontrollers, sensor systems, Bluetooth communication, and voice recognition, making it a comprehensive and practical example of modern robotic systems. It has potential applications in fields such as home automation, robotics research, and smart vehicles, where both autonomous navigation and remote-control capabilities are valuable. [10-13].

Existing System and Proposed System

The existing system for the obstacle avoidance robotic car focuses solely on obstacle detection and avoidance capabilities. It lacks versatility and interactivity. In contrast, the proposed system enhances the existing obstacle avoidance by incorporating Bluetooth and voice control. With Bluetooth integration, the car can be remotely controlled using a smartphone or computer, providing a convenient and wireless mode of operation. Voice control adds interactivity, allowing users to issue commands for movement. This comprehensive system combines obstacle avoidance, Bluetooth, and voice control, making the project more engaging and versatile for educational, practical, and recreational purposes.

Arduino UnoATmega328P



Fig.1. Atmega328P

Arduino Uno, a popular microcontroller board used for electronics projects and prototyping. Here's a breakdown of its key components and features:

- **Microcontroller:** The heart of the Arduino Uno is the Atmel ATmega328P microcontroller (the large black chip). This is a small computer that can be programmed to control various electronic circuits and devices.
- **USB Connector:** The USB port is used to connect the Arduino to a computer for programming and power. It also allows for serial communication between the Arduino and the computer.
- **Power Connector:** The barrel jack allows the Arduino to be powered by an external power supply (typically 7-12V).
- **Digital I/O Pins:** These pins (labeled 0-13) can be configured as either inputs or outputs. They can be used to control LEDs, motors, relays, and other digital devices, or to read signals from sensors and switches. Some digital pins (marked with a ~) also support Pulse Width Modulation (PWM), which allows for analog-like control of digital devices (e.g., dimming LEDs or controlling motor speed).
- **Analog Input Pins:** These pins (labeled A0-A5) can read analog voltages from sensors and convert them into digital values that the microcontroller can understand. This is useful for reading data from potentiometers, temperature sensors, light sensors, and other analog devices.
- **Reset Button:** This button resets the microcontroller, restarting the program that is currently running.
- **Crystal Oscillator:** The crystal oscillator provides a precise clock signal for the microcontroller, ensuring accurate timing for its operations.
- **Voltage Regulator:** This component regulates the input voltage to a stable 5V, which is required by the microcontroller and other components on the board.
- **LED Indicators:** There are a few LEDs on the board:
 - **ON:** Indicates that the board is powered on.
 - **L:** Connected to digital pin 13. It is often used for simple testing or debugging.
 - **TX/RX:** These LEDs blink when data is being transmitted or received via the serial communication (usually through the USB connection).

Motor Driver Shield

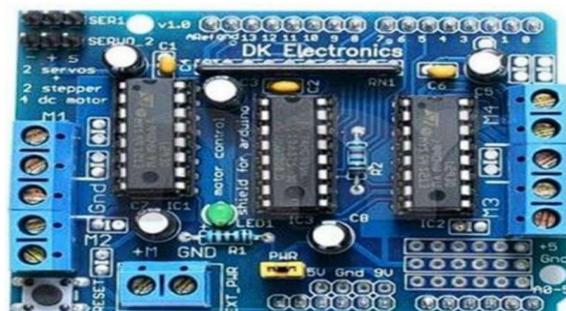


Fig.2 Motor shield

A motor driver shield is an electronic circuit board that connects to an Arduino microcontroller to control the speed and direction of DC motors. It utilizes H-bridge circuits for variable speed and bidirectional control. Motor driver shields offer convenience and additional features like current sensing and over-current protection. They are commonly used in robotics and automation projects for precise motor control. Popular motor driver shields include L293D, L298N, and TB6612FNG.

SG90 Servo Motor



Fig.3 Servomotor

The SG90 is a very popular and widely used micro-servo motor. It's favored for its small size, affordability and ease of use, making it a staple in hobby robotics, RC vehicles.

- Its small and lightweight, typically weighing around 9 grams, making it suitable for projects where space and weight are constraints.
- **Rotation:** The SG90 offers a rotation range of approximately 180 degrees. This means its output shaft can move within a half-circle arc.

Software Requirements

Arduino Ide

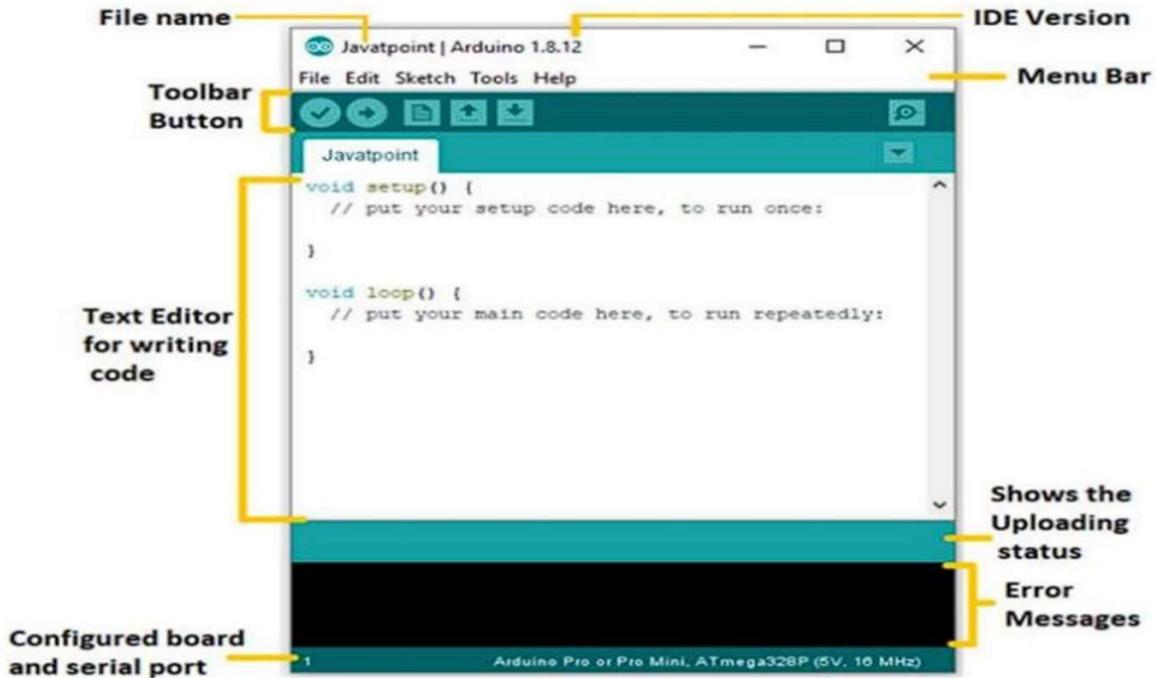


Fig. 4. Arduino IDE

The image shows the Arduino Integrated Development Environment (IDE), which is the software used to write, compile, and upload code to Arduino boards. Here's a breakdown of the key parts labeled in the image.

- **Setup () function:** This function is called once when the Arduino board starts up. It's typically used to initialize variables, set pin modes (input or output), and perform other setup tasks.
- **Loop () Function:** This function is called repeatedly after the setup () function has finished. It's where the main program logic resides, controlling the behaviour of the Arduino.

Fritzing

Fritzing is a popular open-source software tool designed for electronics enthusiasts, hobbyists, and designers. It provides a user-friendly platform for creating and documenting electronic circuits, prototyping.

□ **Open-source nature:** You mentioned Fritzing is open-source, which means the software code is freely available for anyone to see, modify, and contribute to. This allows for a constantly evolving program with features and components added by the community.

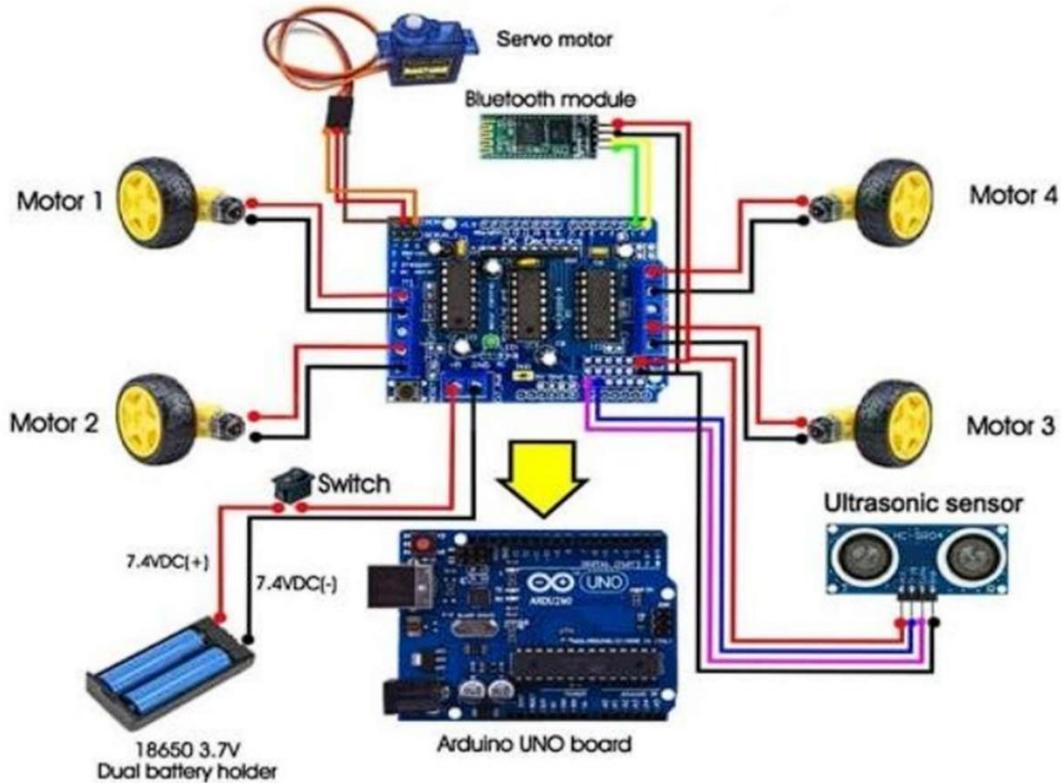


Fig. 4. Fritzing Model

□ **Learning Electronics:** Fritzing's breadboard view is particularly helpful for beginners in electronics. It allows them to experiment and visualize circuits without the risk of damaging components in the real world.

□ **Community and Sharing:** The ability to create and share project documentation with Fritzing fosters collaboration and learning within the electronics community. People can share their designs, troubleshoot issues, and inspire others.

□ **Limitations:** While powerful for prototyping, Fritzing might not be suitable for complex, high-frequency PCB designs. Specialized Electronic Design Automation (EDA) software might be needed for those projects.

Implementation and Design

Assemble the Hardware Components

□ **Assemble the Hardware:** Gather the chassis, motors, wheels, motor driver shield, ultrasonic sensor, Arduino, and battery. Follow instructions to assemble them.

□ **Connect the Motor Driver Shield:** Attach the motors to the motor driver shield, then connect the shield to the Arduino.

□ **Connect the Ultrasonic Sensor:** Connect the ultrasonic sensor to the Arduino to detect obstacles.

□ **Write the Code:** Program the Arduino for obstacle detection, avoidance, and movement control.

□ **Test the Robot Car:** Power on the robot and observe its reaction to obstacles and commands.

□ **Fine-Tune the Code:** Adjust the code for improved performance, accuracy, and reliability.

□ **Add a User Interface:** Optionally add an LCD display or Bluetooth for control and distance readings.

□ **Enclose the Robot Car:** Add a casing to protect the components and improve durability.

□ **Test and Debug:** Test thoroughly and debug any issues to ensure proper operation.

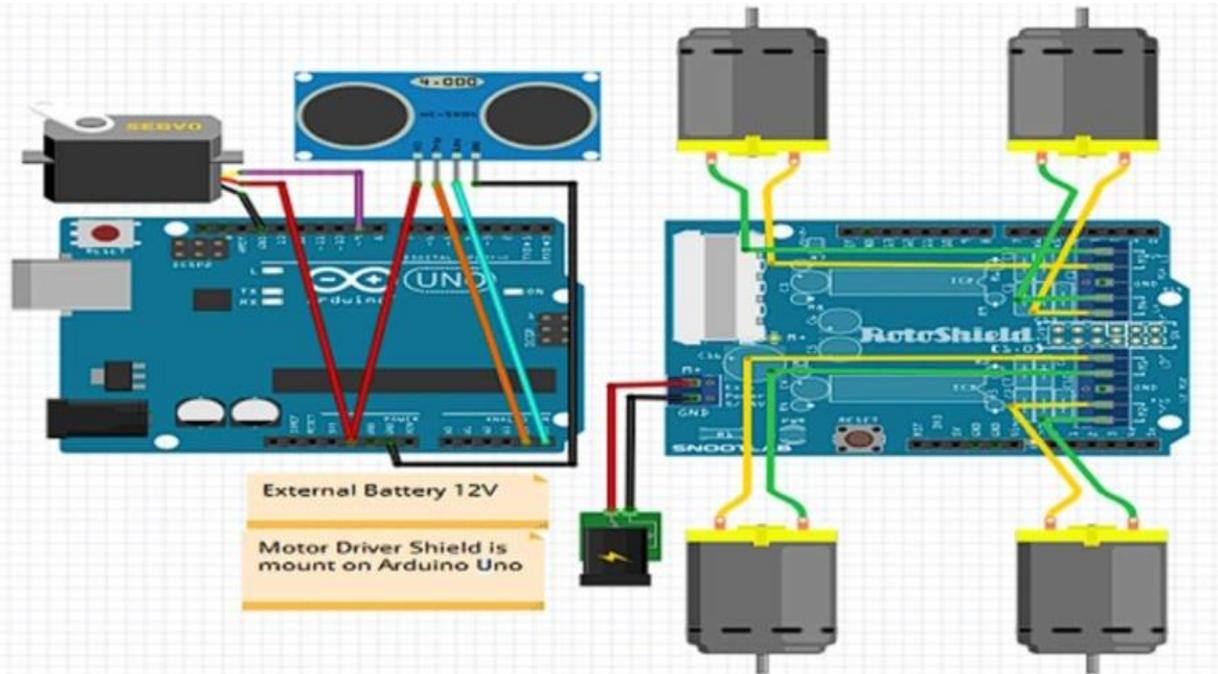


Fig. 5. Proposed Model

Conclusion:

An obstacle avoidance robot car using Arduino is an engaging and practical project with a wide range of applications. By integrating components such as DC gear motors, ultrasonic sensors, and motor driver shields with Arduino's programming environment, a self-navigating robot car capable of detecting and avoiding obstacles can be built. This project involves motor control, sensor integration, and software development, providing a hands-on opportunity for individuals to learn concepts in electronics, robotics, and coding.

As a result, it serves as an excellent educational tool for students and enthusiasts. The project challenges users to develop problem-solving and technical skills in the fields of robotics and programming. Its real-world applications extend to areas such as security, surveillance, and autonomous exploration.

References

1. M. A. Sadeeq, S. R. Zeebaree, R. Qashi, S. H. Ahmed, and K. Jacksi, "Internet of Things security: a survey," in 2018 International Conference on Advanced Science and Engineering (ICOASE), 2018, pp. 162–166.
2. H. I. Dino and M. B. Abdulrazzaq, "Facial Expression Classification Based on SVM, KNN and MLP Classifiers," in 2019 International Conference on Advanced Science and Engineering (ICOASE), 2019, pp. 70–75.
3. M. Jayakannan and J. Inbaraj, "A RECENT REVIEW ON GROWTH OF MOBILE GENERATIONS-CASE STUDY
4. A. M. Abdulazeez, S. R. Zeebaree, and M. A. Sadeeq, "Design and Implementation of Electronic Student Affairs System," *Acad. J. Nawroz Univ.*, vol. 7, no. 3, pp. 66–73, 2018.
5. M.W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*, 1st Edition, Jon Wiley & Sons Inc, 2005.
6. K. Ogata, *Modern control engineering*, Prentice Hall, 2009.
7. S. Ge Shuzhi, T. H. Lee and C. J. Harris, "Adaptive Neural Network Control of Robotic Manipulators". *World Scientific Series in Robotics and Intelligent Systems – Vol. 19*, 1998.
8. R. E. King, "Computational Intelligence in Control Engineering", Marcel Decker, New York, 1999.
9. Wang, Y.; Hou, J.; Li, C.; Wu, J.; Jiang, Y.; Liu, M.; Hung, J.Y. Ultrafast Mode Reversal Coriolis Gyroscopes. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 5969–5980. [[Google Scholar](#)] [[CrossRef](#)]
10. Zhang, T.; Zhao, S.; Luan, X.; Liu, F. Bayesian Inference for State-Space Models with Student-t Mixture Distributions. *IEEE Trans. Cybern.* **2022**, *53*, 4435–4445. [[Google Scholar](#)] [[CrossRef](#)]
11. Arduino, [en línea]. [Consulta: septiembre de 2013]. Disponible en web: <http://arduino.cc/>.
12. Wiki. "Ultra-Sonic range measurement module "[en línea]. 4 de mayo de 2013, [Consulta: September de 2013]. Disponible web: http://www.seeedstudio.com/wiki/index.php?title=Ultra_Sonic_range_measurement_module.
13. Wikipedia contributors, "Servomotor" [en línea]. Wikipedia, The Free Encyclopaedia. 25 de junio de 2013, [Consulta: septiembre de 2013]. Disponible en web: <http://es.wikipedia.org/wiki/Servomotor>