DR. B. C. ROY ACADEMY OF PROFESSIONAL COURSES DURGAPUR

DR. B.C. ROY SOCIETY
ACADEMY OF PROFESSIONAL COURSES

Obstacle Avoiding Arduino Car

A Project Report

Submitted in partial fulfilment of the requirements for the award of degree of

Approved by:

Prof. Pratap Sharma

Submitted by:

Sweta Singh (32301222033)

Sayan Golder (32301222031)

Kunal Gautam (32301222091)

Ashmita Chakraborty (32301222029)

Self-Certificate

This is to certify that the dissertation/project proposal entitled **“Obstacle Avoiding Arduino Car”** is done by us, and is an authentic work carried out for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Computer Application** under the guidance of **Prof. Pratap Sharma**.

The matter embodied in this project work has not been submitted earlier for the award of any degree, to the best of our knowledge and belief.

Name of the Student(s):
Sweta Singh : -
Sayan Golder : -
Kunal Gautam : -
Ashmita Chakraborty : -

Certificate by Guide

This is to certify that the project entitled "**Obstacle Avoiding Arduino Car**", submitted in partial fulfilment of the requirements for the degree of Bachelor of Computer Application by Sweta Singh, Kunal Gautam, Ashmita Chakraborty and Sayan Golder, students of Dr. B.C. Roy Academy of Professional Courses, Durgapur, is an authentic work carried out under my guidance and to the best of our knowledge and belief.

(Signature with date)

<u>Certificate of Approval</u>

This is to certify that the documentation of **Major Project 2025**, entitled **“Obstacle Avoiding Arduino Car”**, is a record of bona-fide work carried out by **Sweta Singh**, **Kunal Gautam**, **Sayan Golder**, and **Ashmita Chakraborty**, under my supervision and guidance.

In my opinion, the report in its present form fulfils all the requirements specified by **Dr. B.C. Roy Academy of Professional Courses, Durgapur**, and adheres to the regulations of the college. It has attained the standard necessary for submission. To the best of my knowledge, the results embodied in this report are original in nature and worthy of incorporation in the present version of the report for the **Bachelor of Computer Application** degree.

Asst. Prof. Pratap Sharma

Professor
Dr. B.C. Roy Academy of Professional Courses
Jemua Road, Fuljhore, Durgapur – 713206

Content

Acknowledgement

We would like to take this opportunity to thank our valued guide **(Prof. Pratap Sharma)** for their invaluable inspiration, astute direction, and tireless assistance, all of which made it possible for us to carry out and finish this effort.

We respect **(Prof. Rita Ganguly)** for her dedication to the job and the effort put forward, as well as for giving up her valuable time to hear our concerns and provide enlightening responses.

We sincerely thank **(Principal Dr. Rajib Roy)** for providing all the facilities necessary for the course of study.

We would like to take this opportunity to offer our sincere gratitude to all of the faculty members at (B.C.A) for their assistance and encouragement.

We also want to express our gratitude to all of our friends who directly or indirectly assisted us in finishing the dissertation.

Signature of Student(s):

1. Sayan Golder (32301222031)

2. Sweta Singh (32301222033)

3. Kunal Gautam (32301222091)

4. Ashmita Chakraborty (32301222029)

Date —

Chapter -1

Introduction

A unique form of technological innovation, the Obstacle Avoiding and Bluetooth Controlled Arduino Car is engineered to intelligently navigate environments by detecting and avoiding physical obstructions using real-time sensor data and embedded control logic. In an age where automation and robotics are transforming industries, developing autonomous mobile systems serves as an essential step toward smarter and safer applications in various domains, including smart surveillance, unmanned logistics, rescue missions, and educational tools.

Traditional robotic vehicles often rely on manual control and lack the capability to make autonomous decisions in dynamic surroundings. This results in navigation inefficiencies, increased chances of collision, and limitations in usability. Manual control systems also face several challenges, such as:

- Delayed Response: Manual inputs may not react swiftly enough to avoid sudden obstacles.
- Limited Precision: Human-controlled navigation often lacks the fine-tuned control achievable through sensors and microcontrollers.
- Restricted Autonomy: Continuous human interaction is required, making it impractical for remote or hazardous areas.
- Inconsistent Performance: Variations in user control can lead to unstable movement and inaccurate routing.

With the rapid progression of embedded systems, microcontrollers, and sensor technologies, the opportunity to build affordable and efficient autonomous systems has become more viable than ever. This project leverages the power of Arduino Uno, ultrasonic and IR sensors, and Bluetooth communication to create a car that can operate both autonomously and manually. In its automatic mode, the car uses sensors to detect obstacles and make decisions in real time, ensuring a collision free route. In manual mode, the car can be controlled via a smartphone through a Bluetooth interface (HC-05), providing user flexibility and control when needed.

The Obstacle Avoiding Arduino Car project aims to build a low-cost, versatile, and intelligent robotic system. Through the integration of a servo motor for sensor scanning, DC gear motors for motion, and a motor driver (L293D) for navigation logic, the system achieves smooth and responsive movement across varying terrain and conditions.

This project not only enhances safety and automation in robotic systems but also serves as a powerful learning platform for understanding embedded design, sensor interfacing, and real-world implementation. By bridging the gap between theoretical knowledge and practical robotics, it inspires innovation and contributes to a future where autonomous technology is accessible and impactful across diverse sectors.

Moreover, as automation becomes increasingly important in domains such as smart cities, agriculture, security, and industry, scalable and intelligent systems like this can offer tremendous value. The approach adopted in this project opens pathways to more advanced enhancements, including AI integration, GPS-based navigation, and cloud connectivity, making it a stepping stone toward the next generation of intelligent robotics.

Chapter - 2

Project Category

The Obstacle Avoiding and Bluetooth Controlled Arduino Car falls under the Embedded Systems and Robotics category, specifically focusing on intelligent autonomous navigation and remote-controlled mobility. This project showcases the integration of hardware and software in a compact, real-time environment, solving navigation and obstacle detection challenges through microcontroller-based logic and sensor fusion. Embedded systems and robotic technologies are the building blocks of modern automation. These systems are designed to execute dedicated tasks efficiently and with high precision. In this project, the car is built to operate both autonomously (using sensor feedback) and manually (via smartphone Bluetooth control), making it a versatile and educational prototype for real-world robotic applications.

EMBEDDED SYSTEM OVERVIEW:

The Obstacle Avoiding Arduino Car exemplifies an embedded robotics solution, using **Arduino Uno** as the central controller interfacing with a range of components such as **ultrasonic sensors, IR proximity sensors, servo motors, and DC gear motors.** It responds to its environment by detecting obstacles in real time, adjusting movement accordingly, and allowing user interaction via Bluetooth for manual operation.

Key features of embedded robotic systems include:

- **Automation:** Automates mobility and navigation by detecting obstacles and dynamically adjusting movement without human intervention.
- **Dual-Mode Operation:** Offers both autonomous and manual control, allowing flexibility in different scenarios.

- **Adaptability:** Adjusts behavior based on sensor input, including real-time scanning using servo motor mounted sensors. 4. **Real-Time Control:** Processes environmental data instantly, making decisions within milliseconds for safe navigation.
- **Scalability:** Serves as a base platform for advanced applications such as surveillance, patrol robots, and delivery systems.
- **Cost-Effective Design:** Utilizes open-source hardware and software, making it affordable and ideal for academic projects and DIY robotics.

Embedded systems like this are essential for students, hobbyists, and professionals who want to explore real-time processing, sensor fusion, and autonomous mobility. The project also provides hands-on experience with circuit design, C/C++ programming, and Bluetooth communication protocols.

FUNCTIONALITIES PROVIDED:

The Obstacle Avoiding Arduino Car offers the following key functionalities:

1. **Obstacle Detection and Avoidance**
   • Utilizes HC-SR04 ultrasonic sensors to measure distance and avoid collisions.
   • IR sensors provide additional close-range obstacle detection for refined movement.
   • Ensures dynamic rerouting when an obstacle is detected within a preset threshold.
2. **Manual Control via Bluetooth**
   • Integrates HC-05 Bluetooth module for wireless control using smartphone applications.

• Users can manually drive the car forward, backward, left, right, or stop through simple commands.

3. **Servo-Based Environmental Scanning**
   • A servo motor mounted with an ultrasonic sensor performs a 180° sweep for enhanced obstacle awareness.
   • Improves field of vision and obstacle detection capability.
4. **Differential Steering System**
   • Employs DC gear motors controlled via L293D motor driver for differential motion, providing better turning and control.
5. **User-Friendly and Interactive**
   • Easy to assemble and operate for students and beginners in embedded systems.
   • Supports simple command inputs for manual operation via Bluetooth apps.
6. **Simulation and Prototyping Support**
   • Can be designed and tested in simulation platforms like Tinkercad or Proteus before hardware implementation.
7. **Educational Value**
   • Ideal for teaching core concepts of microcontroller programming, robotics, and automation.
   • Demonstrates real-world applications of embedded system design.
8. **Upgrade Potential**
   • Can be expanded with GPS modules, cameras, AI-based vision systems, and IoT for advanced functionalities.
   • Acts as a foundational project for building more complex robotic platforms.
9. **Power-Efficient Design**

   • Operates using a 7V rechargeable battery, enabling long-duration demonstrations without frequent charging.
   • Efficient motor and sensor selection helps conserve energy.

Chapter - 3

Tools and Platforms

In the development of the **Obstacle Avoiding and Bluetooth Controlled Arduino Car**, a combination of hardware tools and software platforms was utilized to ensure smooth design, testing, and implementation. The project required accurate real-time processing, sensor interfacing, motor control, and wireless communication. This chapter outlines the tools and platforms that enabled the successful execution of the system.

Hardware Specification

The hardware components used in the project were chosen for their compatibility, performance, and affordability. Each plays a crucial role in executing core functionalities like obstacle detection, motion control, and wireless communication.

Content	**Specification**
Arduino Uno	Microcontroller: ATmega328P Operating Voltage: 5V Clock Speed: 16 MHz
Ultrasonic Sensor (HC-SR04)	Range: 2 cm – 400 cm Accuracy: ±3 mm Power: 5V
IR Proximity Sensors	Range: 2 – 30 cm Output: Digital Power: 3.3V – 5V
HC-05 Bluetooth Module	Bluetooth Version: 2.0 Range: 10 meters Baud Rate: 9600
L293D Motor Driver	Voltage: 4.5V – 36V Dual H-Bridge Max Current: 600 mA per channel

DC Gear Motors	Operating Voltage: 6V – 12V Speed: ~100 RPM Torque: ~1.5 kg/cm
Servo Motor	Operating Voltage: 4.8V – 6V Rotation Angle: 0°–180°
Power Source	7V battery with connector
Chassis and Wheels	2WD robotic car frame with plastic/acrylic build
Jumper Wires and Breadboard	For temporary connections during testing and prototyping
Soldering Kit	For permanent assembly

Software Specifications

Software platforms and tools were used for writing code, simulating circuits, compiling programs, and uploading code to the microcontroller. Below is a breakdown of the key software components used:

Tool / Platform	**Purpose**
Arduino IDE	Online simulation of Arduino circuits and component testing, as well as compiling and uploading code
Embedded C/C++	Core programming languages used to control sensors, motors, and logic flow
Circuito.io	Circuit design and simulation platform for visualizing and validating Arduino-based circuits
Circuit design and simulation platform for visualizing and validating Arduino-based circuits	Mobile app to send commands via Bluetooth (e.g., “Arduino Bluetooth Controller”)
Serial Monitor (Arduino IDE)	Debugging and communication with Arduino board

Chapter - 4

System Design

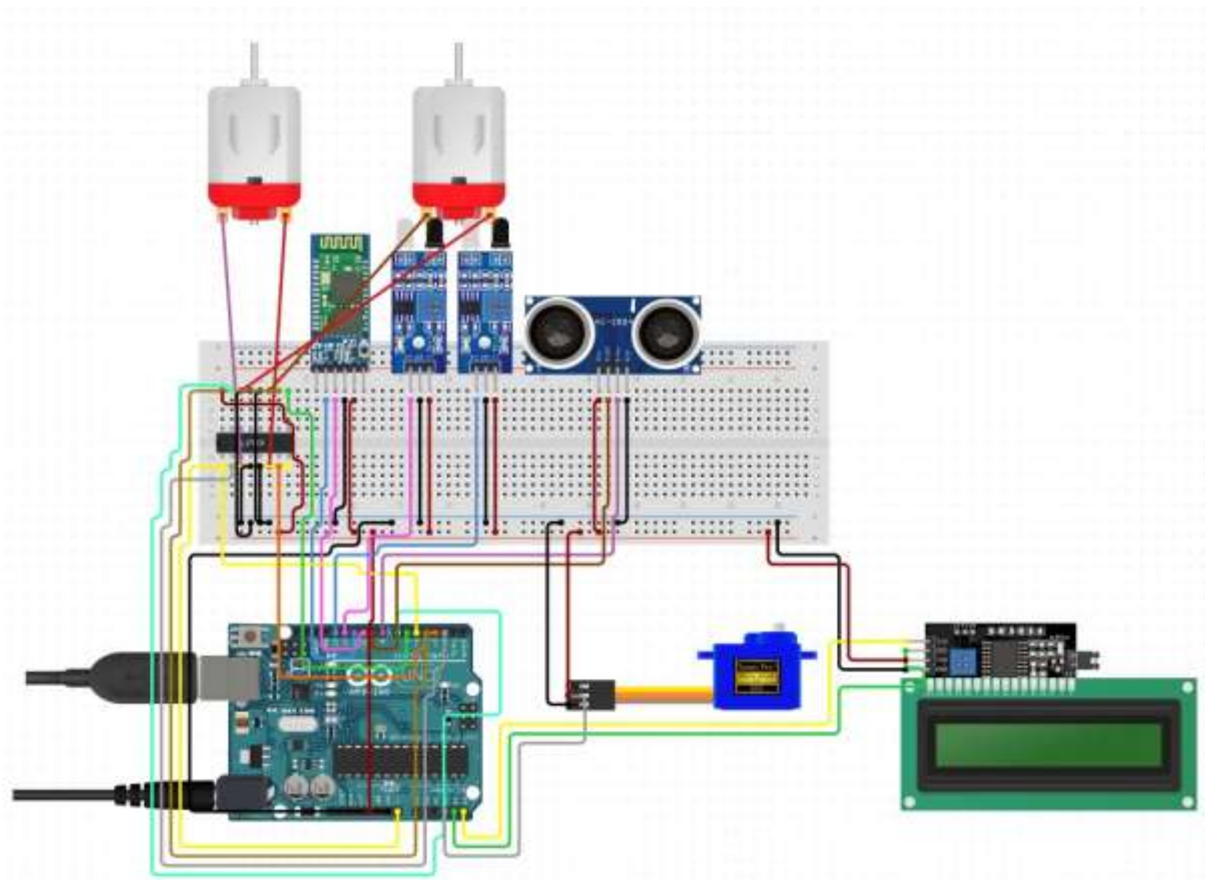
The **Obstacle Avoiding and Bluetooth Controlled Arduino Car** is designed to operate efficiently in both autonomous and manual modes by combining hardware and software components in a well-structured embedded system. The system is capable of analysing its environment, making real-time decisions, and navigating safely using sensor input and control logic.

Circuit Diagram

The circuit diagram represents the physical layout and connections between the core components of the system. It includes the Arduino Uno microcontroller, ultrasonic sensor, IR proximity sensors, L293D motor driver, servo motor, Bluetooth module, DC gear motors, and the power supply.

Key Connections:

- **Ultrasonic Sensor (HC-SR04):** Connected to digital I/O pins for Trigger and Echo signals.
- **IR Sensors:** Connected to digital pins to detect obstacles at close range.
- **HC-05 Bluetooth Module:** Connected via TX and RX pins for serial communication.
- **L293D Motor Driver:** Connected to motor control pins and powered via the battery.
- **Servo Motor:** Connected to a PWM pin on Arduino for angle-based control.
- **DC Motors:** Driven through the motor driver for left-right and forward-backward movement.

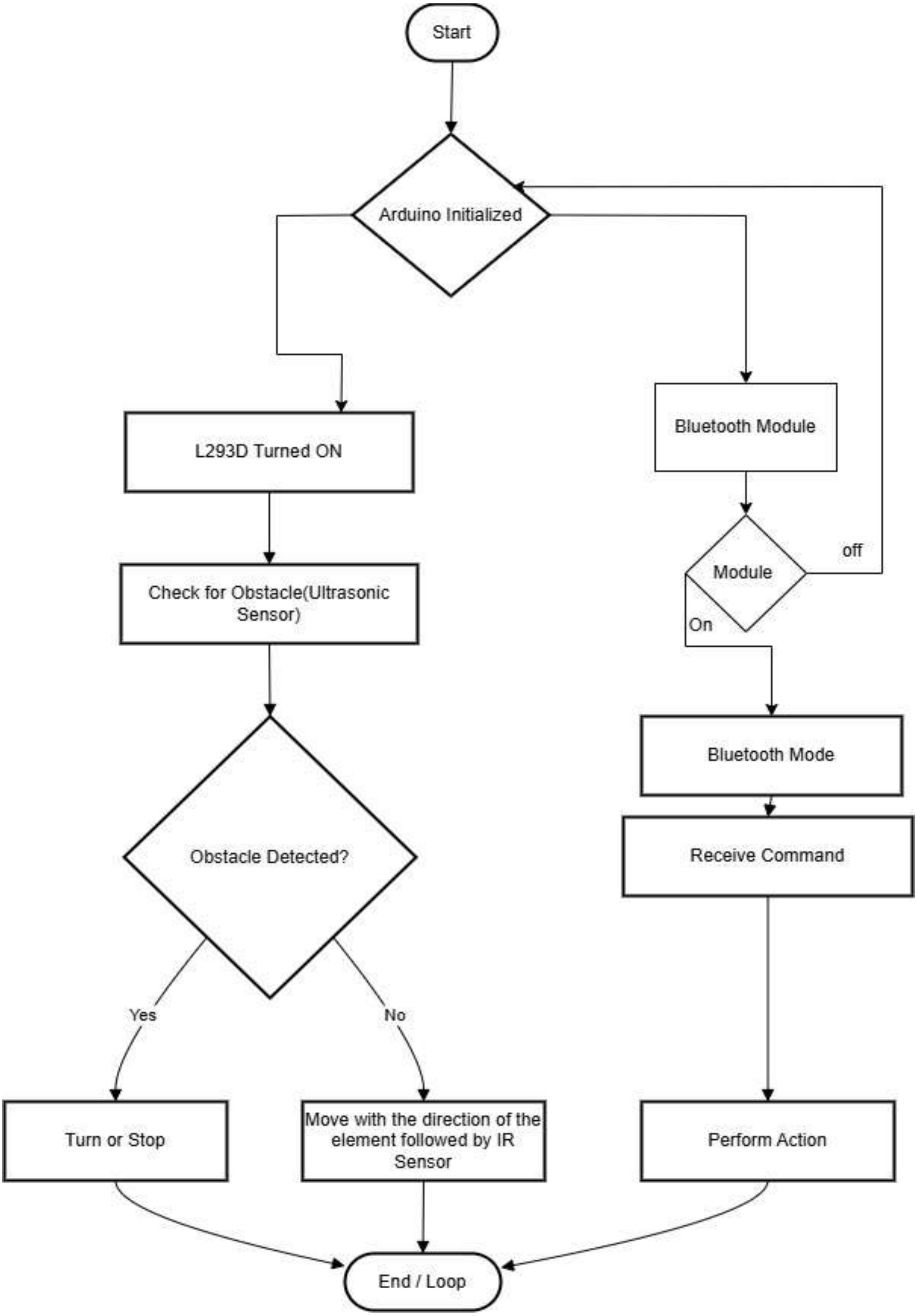


Flow Chat of Proposed System

Flow Chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

Flowcharts are used in analysing, designing, documenting or managing a process or program in various fields.



Flow Chat of Proposed System

Chapter - 5

Technical Overview

The Obstacle Avoiding and Bluetooth Controlled Arduino Car demonstrates the integration of embedded systems, real-time sensor processing, and robotic mobility. This chapter explores the core functionality, component interaction, and logic behind the system's autonomous and manual operations. The car is designed to navigate dynamically, make intelligent decisions based on sensory input, and provide interactive manual control through wireless communication.

Working Principle

The system operates in two distinct modes, both managed by the Arduino Uno microcontroller:

- **Autonomous Obstacle Avoidance Mode**
  This mode allows the car to move independently by detecting and responding to obstacles in its environment without any human intervention.
    - **Ultrasonic Sensor (HC-SR04):**
      Continuously emits ultrasonic waves and calculates the time taken for the echo to return to determine the distance of objects. If an object is detected within a predefined range (e.g., 15 cm), it triggers an avoidance routine.
    - **Servo Motor Integration:**
      The ultrasonic sensor is mounted on a servo motor, which performs a sweeping motion (0° to 180°). This allows the car to scan its surroundings to the left and right to determine the clearer path, rather than simply reversing.
    - **IR Proximity Sensors:**

Placed on the sides or front of the car, these detect nearby objects or surface edges to enhance close-range detection, especially helpful in edge or corner detection.

- **Motor Control via L293D:**
  Based on sensor feedback, the Arduino sends directional control signals to the motor driver, which controls the rotation of the left and right DC gear motors. The car adjusts its path accordingly — stopping, turning left/right, or reversing when an obstacle is encountered.
- **Navigation Loop:**
  This logic runs in a continuous loop, enabling real-time decision-making and responsive navigation in dynamic environments.

- **Manual Bluetooth Control Mode**
  This mode allows the user to control the movement of the car using a mobile device via Bluetooth communication.
  - **HC-05 Bluetooth Module:**
    Connected to the Arduino’s TX and RX pins (serial communication), this module receives commands sent from a smartphone Bluetooth controller app.
  - **Command Mapping:**
    Specific characters sent from the phone are interpreted by the Arduino as control commands:
    - F → Move Forward
    - B → Move Backward
    - L → Turn Left
    - R → Turn Right
    - S → Stop
  - **Motor Execution:**

Upon receiving a command, the Arduino signals the L293D motor driver to rotate the motors in the desired direction, enabling responsive manual control.

- **Application Flexibility:**
  This mode is useful for demonstrations, obstacle-free navigation, or environments where human direction is preferred.

System Interaction Flow

Component	Role in the System
Arduino Uno	Central controller that processes input from sensors and outputs motor commands
Ultrasonic Sensor	Detects distance to obstacles and sends real-time data to Arduino
IR Sensors	Provides edge or close-object detection for refined control
Servo Motor	Rotates the ultrasonic sensor to perform directional scanning
HC-05 Module	Enables manual control by receiving commands from smartphone via Bluetooth
L293D Motor Driver	Drives the DC motors based on Arduino's directional instructions
DC Gear Motors	Responsible for the actual movement and steering of the car

Key Functional Highlights

- **Dual Operational Modes:**
  Offers flexibility with autonomous decision-making and remote user control.
- **Real-Time Response:**

Immediate obstacle detection and course correction ensure efficient navigation.

- **Sensor Fusion:**
  Combination of ultrasonic and IR sensors increases the system’s accuracy and reliability.
- **Environmental Awareness:**
  Servo scanning allows for better situational analysis compared to static sensors.
- **Low-Cost, Open-Source:**
  Built with affordable, easily available components, ideal for educational and prototyping purposes.

This chapter highlights how a basic microcontroller-based robotic system can be made intelligent and interactive through simple yet effective integration of sensors and motors. The system acts as a stepping stone for developing more advanced robotics applications, paving the way for future enhancements like AI navigation, GPS-based routing, and camera-based object recognition.

Implementation Details

The implementation phase of the Obstacle Avoiding and Bluetooth Controlled Arduino Car focuses on integrating all the hardware and software components to bring the designed system to life. This chapter outlines the step-by-step process for assembling, programming, and executing both autonomous and manual functionalities. The implementation is divided into key functional segments to provide a clearer understanding of each system's working.

1. Sensor Integration and Configuration

Ultrasonic Sensor (HC-SR04) Setup:

- Connected to the Arduino through two digital I/O pins: Trigger and Echo.
- The sensor emits an ultrasonic pulse via the Trigger pin and waits for the reflected signal on the Echo pin.
- The time taken for the echo to return is used to calculate the distance from an object.
- A distance threshold (e.g., 15 cm) is defined; if an obstacle is detected within this range, the car takes corrective action.

Servo Motor Scanning Mechanism:

- The ultrasonic sensor is mounted on a servo motor.
- The servo rotates between 0° and 180°, allowing the sensor to scan the area in front of the car.
- Based on the distance readings from different angles, the Arduino determines the optimal direction for movement (left or right).

IR Proximity Sensors:

- IR sensors are placed on the sides and/or front of the car to detect short-range obstacles or wall edges.
- Digital output (LOW when obstacle is detected) is read by the Arduino to supplement the ultrasonic readings.

2. Motor Control Logic

DC Gear Motor Configuration:

- Two gear motors are connected to the wheels of the car and controlled via the **L293D motor driver**.
- The L293D H-bridge driver receives logic signals from the Arduino to rotate the motors in the required direction (forward, backward, left, right).
- Each wheel can be independently controlled to enable smooth turns and differential steering.

L293D Integration:

- Four Arduino pins are used to control the direction of the motors.
- Additional pins are used for enabling motor A and B channels.
- The power supply (7V battery) is connected to the L293D to drive the motors.

3. Bluetooth-Based Manual Control

HC-05 Bluetooth Module Configuration:

• The Bluetooth module is connected to the Arduino’s TX and RX pins.

• Paired with a smartphone using a Bluetooth controller app (e.g., “Bluetooth Terminal”).

• The app sends predefined character commands to the Arduino, such as:

  • F – Forward

- B – Backward
- L – Turn Left
- R – Turn Right
- S – Stop

Command Interpretation:

- The Arduino receives the characters via serial communication and maps them to motor control logic.
- The car responds immediately, performing the desired action.

4. Programming and Logic

- **Arduino IDE** is used to write and upload the program to the Arduino Uno.
- The program consists of:
    - Sensor reading functions (ultrasonic and IR)
    - Servo motor scanning function
    - Motor control logic (L293D)
    - Bluetooth serial communication handling
- The code runs in a loop, constantly checking sensor inputs and acting based on the current mode (autonomous/manual).

5. Testing and Calibration

- Initially tested on a breadboard with jumper wires for quick debugging.
- Threshold distances for obstacle detection were calibrated based on sensor accuracy and desired stopping distance.

- Servo angle range and delay were optimized for smooth scanning.
- After successful testing, components were soldered and fixed on the robotic chassis.

6. Implementation Outcome

- The car successfully detects and avoids obstacles using sensor feedback.
- In manual mode, the car responds accurately to Bluetooth commands.
- The system is stable, reusable, and serves as a practical foundation for further developments in robotics.

Chapter - 7

Components and Modules

The successful execution of the Obstacle Avoiding and Bluetooth Controlled Arduino Car relies on the integration of several key hardware modules. Each component plays a specific role in achieving autonomous obstacle detection, real-time control, and wireless communication. This chapter provides a detailed description of the components used, along with their purpose and specifications.

1. Arduino Uno (Rev3)

Role: Central microcontroller responsible for controlling the entire system.

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (Recommended): 7–12V
- Digital I/O Pins: 14 (6 provide PWM output)
- Analog Input Pins: 6
- Clock Speed: 16 MHz
- Purpose: Reads sensor inputs, processes logic, and controls actuators (motors, servo).

2. HC-05 Bluetooth Module

Role: Enables wireless communication between the Arduino and a smartphone.

- Operating Voltage: 3.3V – 6V
- Bluetooth Version: 2.0
- Communication Mode: Serial (UART)

• Range: Up to 10 meters

• Baud Rate: 9600 (default)

• Purpose: Receives character commands (F, B, L, R, S) to control the car in manual mode.

3. Ultrasonic Sensor (HC-SR04)

Role: Detects the distance of obstacles ahead and provides real-time feedback to avoid collisions.

• Operating Voltage: 5V DC

• Range: 2 cm – 400 cm

• Accuracy: ±3 mm

• Frequency: 40 kHz

• Interface: Trigger and Echo pins

• Purpose: Measures obstacle distance and helps in rerouting the car.

4. IR Proximity Sensors

Role: Detects close-range obstacles or surface edges (e.g., table ends, walls).

• Operating Voltage: 3.3V – 5V

• Detection Range: 2 cm – 30 cm (adjustable)

• Output: Digital (LOW when obstacle detected)

• Purpose: Refines obstacle detection for narrow or edge detection.

5. L293D Motor Driver Module

Role: Controls the direction and movement of the DC gear motors.

• Operating Voltage: 4.5V – 36V

• Channels: Dual H-Bridge (for 2 motors)

• Peak Output Current: 600 mA per channel

• Purpose: Converts Arduino logic signals into high-current signals for driving motors forward, backward, left, and right.

6. Servo Motor

Role: Rotates the ultrasonic sensor to scan the surrounding area.

• Operating Voltage: 4.8V – 6V

• Angle of Rotation: 0° – 180°

• Control Signal: PWM

• Torque: ~2 kg/cm

• Purpose: Expands obstacle detection range by scanning left and right directions.

7. Power Source (Battery)

Role: Powers the Arduino and motor driver module.

• Type: 7V Battery with connector

• Purpose: Supplies consistent power to all modules during movement and operation.

8. Chassis, Wheels & Assembly Tools

Role: Provides the physical structure and mobility for the robotic car.

• Chassis Material: Acrylic or plastic

• Wheels: 2 or 4 wheels based on motor configuration

• Tools: Screwdrivers, soldering iron, glue gun, jumper wires

- Purpose: Mounts all components securely for stable motion and structure.

Each module in the system has been selected for compatibility, ease of use, and cost-effectiveness, making the entire build modular and easy to replicate or extend.

Chapter - 8

Advantages and Disadvantages

The Obstacle Avoiding and Bluetooth Controlled Arduino Car is an efficient and scalable embedded system project that blends autonomous navigation with manual control. Like all systems, it offers significant advantages but also has some limitations that could be addressed in future developments.

Advantages

- **Autonomous Navigation**
  - The car can move independently without human intervention, making real-time decisions to avoid obstacles using ultrasonic and IR sensors.
- **Dual Operating Modes**
  - The system supports both automatic (sensor-based) and manual (Bluetooth controlled) operation, offering flexibility and broader applicability.
- **Cost-Effective Design**
  - Built using low-cost components such as Arduino Uno, HC-SR04, and L293D, making it ideal for students, hobbyists, and educational institutions.
- **Expandable Architecture**
  - The modular design allows future integration of additional modules such as cameras, GPS, or AI-based object recognition systems.
- **Real-Time Decision Making**
  - Quick and efficient obstacle detection and response ensure smooth and safe navigation.
- **Educational Value**

    - Introduces learners to core concepts of embedded systems, sensor integration, wireless communication, and motor control.
- **Low Power Consumption**
    - Operates efficiently on a 7V battery while still performing all required tasks for basic navigation and control.
- **Hands-on Practical Learning**
    - Encourages students to assemble, program, and troubleshoot real embedded hardware and understand real-time processing.

Disadvantages / Limitations

- **Limited Obstacle Detection Range**
    - The ultrasonic sensor has a limited effective range (up to 4 meters), which may not be suitable for high-speed or large-scale navigation.
- **No Terrain Adaptation**
    - The car performs well on flat surfaces but struggles on uneven or outdoor terrain due to basic motor and chassis design.
- **Lack of GPS or Path Planning**
    - The system doesn't include GPS or a predefined navigation path, limiting it to reactive navigation only.
- **Manual Mode Requires Line of Sight**
    - Bluetooth control range is limited to about 10 meters and may be affected by physical obstructions.
- **Sensor Blind Spots**
    - Depending on placement, there can be areas around the car where sensors do not detect obstacles effectively, leading to potential collisions.
- **6. No Real-Time Feedback to User**

  - The system doesn't provide feedback (e.g., live distance data or current status) to the user in manual mode.

Despite these limitations, the system achieves its primary goals effectively and lays the foundation for more complex robotics systems. With further enhancements, the project can evolve into a more intelligent and adaptable robotic vehicle.

Chapter - 9

Application and Testing

The Obstacle Avoiding and Bluetooth Controlled Arduino Car is not only a prototype for embedded systems and robotics but also a practical model that can be adapted to a variety of real-world scenarios. This chapter outlines the potential application areas where this system can be useful, along with insights into how the system was tested for functionality and performance.

Application Areas

The Obstacle Avoiding and Bluetooth Controlled Arduino Car can be applied in many basic and real-world situations, especially for learning, small projects, and controlled environments. Below are some simple and practical areas where this project can be useful:

1. **Basic Surveillance in Small Areas**
    - This car can be used to move around in small indoor spaces like rooms, labs, or corridors to check for objects or movements. Since it avoids obstacles on its own, it can work without someone manually driving it all the time.
2. **Mini Project for Self-Driving Concept**
    - It can be used to show how basic self-driving vehicles work. The car uses sensors to avoid objects and makes decisions based on its surroundings, similar to how actual smart vehicles function.
3. **Simple Search Tasks**
    - In a classroom or project lab, this car can be sent to move around small blocked areas. It can help in understanding how a robot can navigate tight spaces or even carry small items.
4. **Learning and Education**

- This project is great for students learning electronics, robotics, or embedded systems. It includes everything from coding to connecting wires and sensors, making it a perfect hands on learning experience.

5. **School/College Exhibitions or Competitions**
   - The car is ideal for showcasing at science exhibitions or inter-college robotics competitions. You can even modify it slightly to follow lines or avoid edges for more challenges.
6. **Fun Home Projects**
   - It can be used as a fun robot that moves around the house avoiding walls and furniture. You can control it with your phone using Bluetooth, which makes it more interactive and exciting to use.

This project helps in building knowledge and confidence in basic robotics, and also opens the door to explore more advanced versions in the future.

Testing and Evaluation

The system was tested in both autonomous and manual modes to validate its performance across different environments:

1. **Autonomous Mode Testing:**
   - The car was tested in a room with cardboard walls, obstacles, and barriers.
   - Result: It successfully detected and avoided objects placed at varying distances.
   - Servo Scanning: Improved directional decision-making, reducing the chance of collision.
2. **Bluetooth Manual Mode Testing:**
   - Paired with a smartphone via HC-05 module using a Bluetooth Controller App.

- Commands were sent and received instantly within a 10-meter range.
- Result: The car responded smoothly to all directional inputs (F, B, L, R, S).

3. **Power and Runtime Testing:**
    - Operated on a 7V battery with stable voltage supply to all components.
    - Runtime: Approximately 30–40 minutes of continuous operation on a full charge.
4. **Real-World Limitations Observed:**
    - In dimly lit or reflective environments, sensor accuracy slightly dropped.
    - Performance was best on smooth, flat surfaces; struggled slightly on uneven terrain.

Testing Outcome Summary

Test Case	Input Conditions	Expected Behavior
1. Human Detection and Following	- Ultrasonic sensor detects object (human) at 30–60 cm - IR sensor confirms presence (e.g., for motion or direction)	- Car moves forward to follow - Display shows: "Following Human"
2. Obstacle Detected During Following	- Human detected at 40 cm ahead - Obstacle detected at 15 cm (ultrasonic or IR side sensors)	- Car stops or navigates around the obstacle - Then resumes following - Display shows: "Avoiding Obstacle"
3. No Human Detected, Obstacle Present	- No object detected in human range (ultrasonic > 100 cm) - Obstacle detected at 15 cm	- Car stays idle - Display shows: "Idle - No Target" or "Standby"
4. Human Lost After Following Starts	- Human initially within 40 cm, then out of range (>100 cm) - IR sensor detects no motion	- Car stops movement - Display shows: "Target Lost" or "Searching..."

The car passed all major test cases, validating the project's reliability, flexibility, and adaptability for small-scale autonomous navigation and manual control.

Chapter - 10

Conclusion and Future Scope

Conclusion

The Obstacle Avoiding and Bluetooth Controlled Arduino Car is a simple yet powerful example of how embedded systems and robotics can work together to create smart and interactive machines. The project successfully meets its goals by allowing the car to:

- Move automatically by detecting and avoiding obstacles using ultrasonic and IR sensors.
- Be controlled manually through Bluetooth using a smartphone app.
- Make basic decisions in real-time with the help of the Arduino microcontroller.

Through this project, we gained hands-on experience in sensor integration, circuit building, coding in Arduino IDE, and understanding how real-time data is used for decision-making in robotics. It also helped us understand how simple components like motors, sensors, and a microcontroller can be combined to build a working autonomous system.

Overall, this project was very useful for learning the basics of robotics, embedded systems, and wireless communication. It is cost-effective, easy to understand, and can be a great learning platform for beginners.

Future Scope

Although the project works well in its current form, there is still room for improvement and expansion. Here are some ideas that can be added in the future:

1. Adding a Camera or AI Vision Module

- A camera can be added to recognize faces, objects, or QR codes using AI.
- It can help the robot "see" better and perform more complex tasks.

2. GPS or Location Tracking
    - A GPS module can be added to track the robot's location in outdoor areas.
    - Useful for projects like delivery bots or outdoor navigation.
3. Control via Wi-Fi or IoT
    - Instead of Bluetooth, the car can be controlled over Wi-Fi using a mobile app or website.
    - This allows long-distance control and monitoring.
4. Solar Charging System
    - A small solar panel can be added to charge the battery.
    - It helps make the system eco-friendlier and more self-sustaining.
5. Swarm Robotics
    - Multiple robots can be programmed to work together as a team.
    - This is useful in fields like rescue missions, exploration, or patrolling.
6. Line Following and Maze Solving
    - By adding line sensors, the car can follow a marked path or solve mazes.
    - This feature is often used in robotics competitions.

With these upgrades, the same robot can be transformed into a more intelligent and advanced system. This project is just the beginning — it builds a strong foundation for exploring more complex robotics projects in the future.