Chapter 1

INTRODUCTION

1.1 Overview

An obstacle-avoiding robot using Arduino typically involves a microcontroller (such as an Arduino Uno or Nano), ultrasonic sensors, a motor driver, and DC motors with wheels. The ultrasonic sensor, usually an HC-SR04, measures the distance to nearby obstacles by emitting ultrasonic waves and receiving the echoes. The Arduino processes this distance data and, if an obstacle is detected within a certain threshold, it sends signals to the motor driver to alter the robot's movement, either stopping and turning to avoid the obstacle or continuing forward if the path is clear. The motor driver, often an L298N module, controls the speed and direction of the motors based on these signals. This setup allows the robot to navigate its environment autonomously by continuously monitoring for obstacles and adjusting its path accordingly. The project is a practical introduction to robotics and embedded systems, offering opportunities to learn about sensors, motor control, and programming.

An obstacle-avoiding robot using Arduino combines hardware and software to create an autonomous navigation system. The central component is the Arduino microcontroller, such as an Arduino Uno or Nano, which acts as the robot's brain. This microcontroller processes inputs from sensors and controls outputs to motors. The microcontroller platform that reads sensor inputs and controls the motors.

The using ultrasonic sensors for obstacle avoidance in robotic vehicles. We will explore how these sensors enhance the vehicle's ability to navigate complex environments autonomously, ensuring safe and efficient operation. The integration of ultrasonic sensors into robotic systems not only improves their functionality but also broadens their potential applications, paving the way for more advanced and intelligent robotic solutions.

Ultrasonic Sensor (HC-SR04): Measures the distance to obstacles using ultrasonic waves. It consists of a transmitter and receiver; the transmitter emits a pulse, and the receiver measures the time taken for the echo to return, allowing the calculation of the distance to an obstacle.

Motor Driver (L298N): Interfaces between the Arduino and the DC motors. It allows the Arduino to control the direction and speed of the motors by sending appropriate signals. DC Motors and Wheels: Provide movement for the robot. Typically, two DC motors are used for driving the wheels, and a third castor wheel is used for stability. Power Supply: Usually a battery pack that powers both the Arduino and the motors.

In recent years, the development of autonomous robotic systems has gained significant momentum, driven by advancements in sensor technology and artificial intelligence. Among these systems, obstacle avoidance robotic vehicles have emerged as crucial tools for various applications, ranging from industrial automation to consumer electronics. One effective approach to achieving reliable obstacle avoidance is through the use of ultrasonic sensors. These sensors offer precise distance measurement capabilities by emitting ultrasonic waves and detecting their reflections from nearby objects.

The obstacle avoidance robotic vehicle using ultrasonic sensors is a foundational project in robotics that combines sensor technology, microcontroller programming, and motor control. It serves as an excellent introduction to autonomous systems and has wide-ranging applications in various fields. By understanding and implementing this project, one can gain valuable insights into the principles of robotic navigation and obstacle avoidance

1.2 Problem Statement

In many applications, autonomous navigation is crucial for robots, whether for household cleaning, warehouse automation, or exploration in hazardous environments. To overcome this we design and implement an obstacle-avoiding robot using Arduino. The robot should autonomously navigate an environment, detect obstacles using sensors to avoid collisions. The goal is to create a robust system that demonstrates effective obstacle detection and avoidance capabilities using Arduino-based hardware and programming.

1.3 Objectives

- To develop a robust algorithm on Arduino to enable real-time detection of obstacles using ultrasonic sensor.
- To integrate sensor data with motor control to achieve precise and reliable navigation around obstacles in varying environments.
- To evaluate the system's effectiveness through testing in controlled environments and iterate on the design to improve obstacle detection accuracy and response speed.

1.4 Motivation of the Project

Understanding and solving the challenges of autonomous navigation is crucial for developing systems that can operate independently in dynamic environments, such as self-driving cars. the motivation behind creating an obstacle-avoiding robot using Arduino is multifaceted, encompassing educational benefits, practical applications, personal development, and broader societal impacts.

Chapter 2

LITERATURE REVIEW

Creating an obstacle-avoiding robot using Arduino involves integrating hardware and software components for autonomous navigation and remote control. Literature on such projects highlights the use of ultrasonic sensors for accurate distance measurement, crucial for detecting obstacles. Arduino microcontrollers serve as the central processing unit, executing obstacle avoidance algorithms to steer the robot away from detected objects while aiming towards its goal. Effective design considers mechanical stability and software optimization to enhance the robot's reliability and performance in dynamic environments, emphasizing the interdisciplinary approach necessary for successful implementation.

1. **“Obstacle avoidance robotic vehicle using Ultrasonic sensor, Arduino controller” R. Vairavan, S. Ajith Kumar, L. Shabin Ashiff, and C. Godwin Jose - 2018**

This paper focuses on developing an obstacle avoidance robotic vehicle utilizing ultrasonic sensors and an Arduino controller. It contributes to the field by integrating these technologies for autonomous navigation. The study likely reviews existing methodologies in obstacle avoidance robotics, discusses sensor placement strategies, and evaluates algorithmic approaches for effective obstacle detection and avoidance. It addresses challenges such as sensor accuracy and algorithm efficiency, aiming to enhance the reliability and performance of such systems in dynamic environments.

2. **Obstacle Avoiding Robot By Faiza Tabassum, Susmita Lopa, Muhammad Masud Tarek & Dr. Bilkis Jamal Ferdosi -2017**

The paper details the development and implementation of an autonomous robot capable of avoiding obstacles using an ultrasonic sensor and Arduino Uno. The implementation involves integrating an ultrasonic sensor to detect obstacles in the robot's path. The sensor continuously measures the distance between the robot and potential obstacles, sending this data to the Arduino Uno microcontroller. The robot's movement is controlled using motor drivers connected to the Arduino, which adjusts the speed and direction of the wheels. When an obstacle is detected within a certain range, the robot stops, assesses alternative paths, and navigates around the obstacle to continue its course. This implementation combines sensor data processing, motor control, and real-time decision-making to achieve effective obstacle avoidance.

3. **“Implementation of Obstruction Avoiding Robot using Ultrasonic Sensor and Arduino UNO” Arjun Varma, Ashwath A, Ayush Verma, A. Bagubali, Kishore V Krishnan-2019**

This paper focuses on implementing an obstruction-avoiding robot using an Ultrasonic Sensor and Arduino UNO. It likely reviews existing literature on similar projects, emphasizing the role of ultrasonic sensors in obstacle detection and Arduino UNO's capabilities in controlling robotic movements. The study may discuss algorithmic approaches for obstacle avoidance, sensor placement strategies for optimal performance, and the integration of Arduino programming for autonomous navigation. It aims to enhance the robot's ability to navigate complex environments autonomously, addressing challenges in sensor accuracy and algorithm efficiency for reliable operation.

4. **Obstacle Avoidance Robot using an ultrasonic Sensor with Arduino Uno Muhammad Ahmad Baballe , Mukhtar Ibrahim Bello , Shehu Hassan Ayagi , Umar Farouk musa - 17 Oct. 2023**

The paper presents the design and implementation of an autonomous robot capable of navigating around obstacles using an ultrasonic sensor. The system leverages an Arduino Uno as the central microcontroller, interfaced with an HC-SR04 ultrasonic sensor to detect obstacles. The sensor measures the distance to obstacles by emitting ultrasonic waves and receiving the reflected signal. When an obstacle is detected within a predefined range, the Arduino processes the data and commands the motor driver (L298N) to alter the robot's path, thereby avoiding collisions. The robot is powered by a battery pack and uses DC motors for movement, which are controlled by the motor driver.

Chapter 3

METHODOLOGY

The overall circuit diagram of the proposed undergraduate mini-project work is shown in the figure. The Arduino Uno microcontroller processes inputs from an ultrasonic sensor to detect obstacles. The sensor, mounted on a rotating servo motor, emits ultrasonic waves and measures the return time, providing distance data to the Arduino. The servo allows the sensor to scan the environment. When an obstacle is detected within a certain range, the Arduino processes this data and sends commands to motor drivers (L298N) that control the DC motors. These drivers amplify the Arduino's signals to power the motors.

Four DC motors manage the robot's movement, enabling it to move forward, backward, and turn to avoid obstacles. A battery pack of two 3.7V 18650 lithium-ion batteries powers the system, ensuring adequate voltage and current. The Arduino is programmed to interpret sensor data, make navigation decisions, and control the motors, resulting in an efficient, autonomous obstacle-avoiding robot.

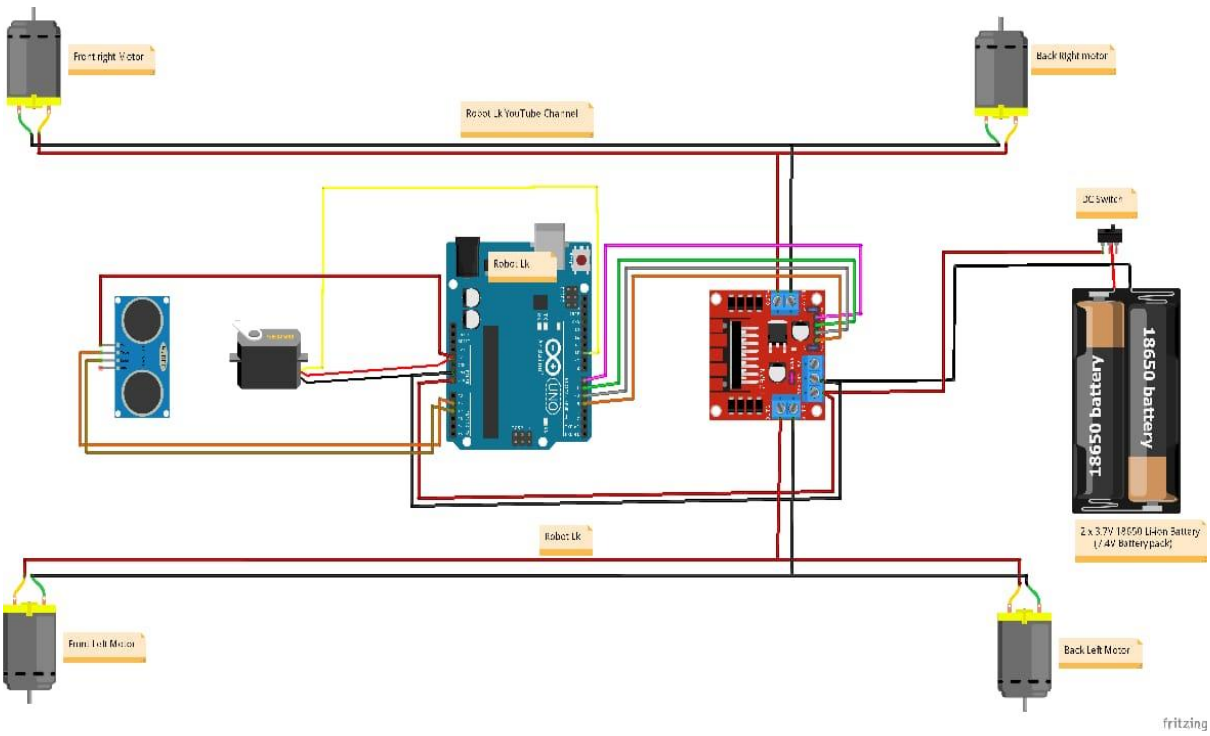

Figure 3.1: The Circuit diagram of the obstacle avoiding robot

Working Principle

The working principle of the obstacle-avoiding robot involves a coordinated interaction between its components to achieve autonomous navigation. The ultrasonic sensor, mounted on a servo motor, emits ultrasonic waves and measures the time it takes for the waves to bounce back from obstacles. This time data is used to calculate the distance to the obstacles. The servo motor allows the sensor to scan the environment in different directions. The Arduino Uno microcontroller processes the distance data and, based on predefined thresholds, decides the robot's movement. It sends control signals to the L298N motor drivers, which amplify these signals to drive the high-power DC motors.

These motors are responsible for the robot's movement, enabling it to move forward, backward, and turn. When an obstacle is detected within a certain range, the Arduino processes the data and decides to stop or change the robot's direction to avoid the obstacle. The motor drivers adjust the speed and direction of the motors accordingly. Powered by a battery pack of two 3.7V 18650 lithium-ion batteries, the system ensures adequate voltage and current supply. This continuous scanning and real-time processing allow the robot to navigate its environment effectively, avoiding obstacles autonomously.

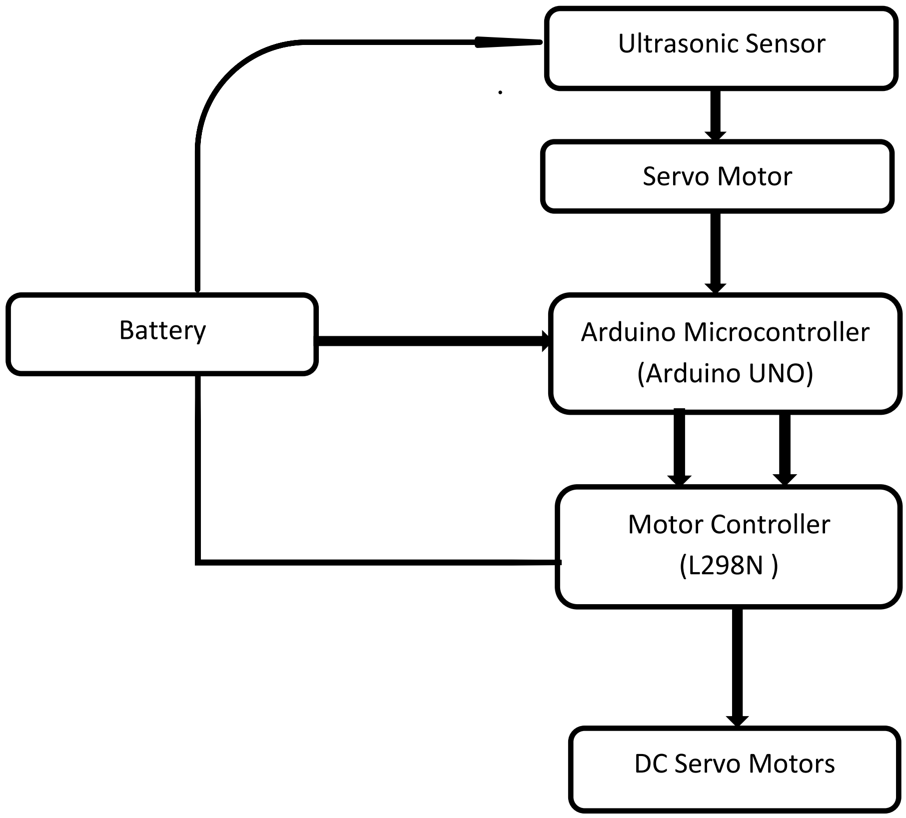

Figure 3.2: Flowchart

Chapter 4

HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements

4.1.1 Arduino uno board

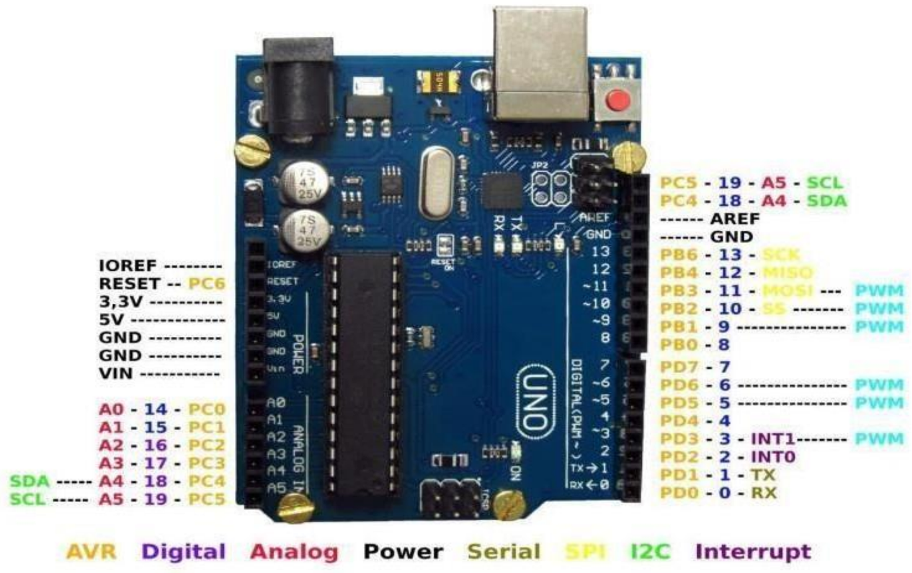

Figure 4.1: Pin Description of Arduino uno

Arduino Uno is a popular open-source microcontroller board based on the ATmega328P microcontroller. It is widely used by makers, hobbyists, and students to build various electronic projects and prototypes. The board has a simple design and is easy to use even for beginners.

- The Arduino Uno board has 14 digital input/output pins, 6 analog input pins, and a USB interface for programming and power supply. The board also has a reset button, an ICSP header, and a power jack. The digital pins can be configured as either input or output and canbe used to connect to sensors, actuators, LEDs, and other electronic components. The analoginput pins allow the board to read analog signals from sensors such as temperature, light, andsound sensor.
- The board is powered by a 5V DC power supply, which can be supplied through the USB connector or the power jack. The board has an onboard voltage regulator that can regulate theinput voltage to 5V, making it easy to power the board using a wide range of power sources.

- The Arduino Uno board can be programmed using the Arduino IDE (Integrated DevelopmentEnvironment), which is available for Windows, Mac, and Linux. The IDE provides an easy to-use interface for writing and uploading code to the board. The code is written in C/C++ programming language and can be easily modified and customized to suit your project's needs.

4.1.2 Ultrasonic sensor

Figure 4.2: Ultrasonic sensor

Ultrasonic sensors typically detect objects within ranges from a few centimeters to several meters, depending on the model and environment. They use sound waves to measure distances by calculating the time taken for the waves to reflect back.

Specifications:

- **Operating Voltage**: 5V DC
- **Operating Current:** 15mA
- **Frequency:** 40kHz
- **Max Range:** 4 meters
- **Min Range:** 2 cm
- **Measuring Angle:** 15 degrees
- **Resolution:** 0.3 cm

Pin Descriptions:

- **VCC:** Power supply pin (5V DC)
- **GND:** Ground pin
- **Trig:** Trigger pin (initiates the ultrasonic pulse with a 10µs HIGH signal)

- **Echo:** Echo pin (outputs a HIGH signal proportional to the distance to the detected object)

4.1.3 L298N motor driver

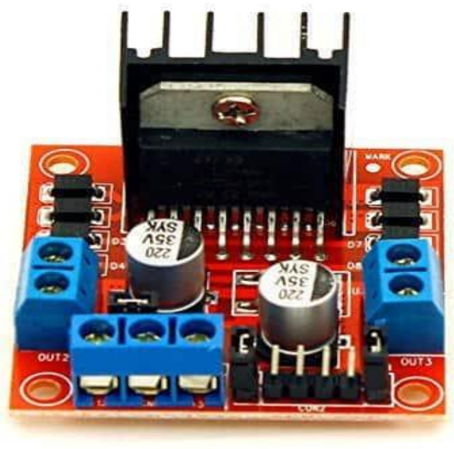

Figure 4.3: L298N motor driver

The L298N motor driver is a dual H-bridge motor driver IC capable of driving two DC motors or a single stepper motor. It allows bidirectional control of each motor independently, making it suitable for various robotic and motor control applications.

The module can handle a wide range of voltages (up to 46V) and deliver up to 2A of continuous current per channel, with proper heatsinking. It includes built-in diodes for protection against back EMF and features like over-temperature and over-current protection, enhancing its reliability in demanding environments. The L298N is commonly used with microcontrollers like Arduino for controlling motors in projects ranging from small robots to automated system.

Key Features:

- **Dual H-Bridge Motor Driver:** Capable of controlling two DC motors independently.
- **Operating Voltage:** Typically 5V for logic and 5-35V for the motor supply.
- **Current Capability:** Each H-Bridge can deliver up to 2A continuous current per channel, with a peak current of 3A per channel.
- PWM Control: Supports pulse-width modulation (PWM) to control motor speed.
- **Overtemperature Protection:** Protects the driver from overheating.
- **Enable Pins:** Separate enable pins for each H-Bridge to enable or disable motor control.

Pin Configuration:

- **IN1, IN2:** Control the direction of Motor A.
- **IN3, IN4:** Control the direction of Motor B.
- **ENA, ENB:** Enable pins for Motor A and Motor B, respectively. These can be connected to a PWM signal for speed control.
- **OUT1, OUT2:** Motor A outputs.
- **OUT3, OUT4:** Motor B outputs.
- **VCC:** Motor power supply (5-35V).
- **GND:** Ground.
- **5V:** Logic voltage supply, typically 5V.

4.1.4 SG90 servo motor

Figure 4.4: SG90 servo motor

The SG90 servo motor is designed to provide precise control of angular position, making it ideal for applications requiring accurate movement and positioning of mechanical components. Despite its small size, the SG90 offers sufficient torque and speed for a variety of tasks in robotics, radio-controlled models, and other DIY projects.

The SG90 servo motor operates at 4.8V to 6V DC, delivering 1.8 kg/cm of torque with a speed of 0.1 sec/60 degrees. It rotates up to 180 degrees (90 degrees in each direction) and uses pulse-width modulation (PWM) for precise control. Its compact size (22mm x 12mm x 22.5mm) and lightweight build (approximately 9 grams) make it ideal for small-scale robotics and hobbyist projects requiring accurate motion control.

Key Features:

- **Torque:** Typically around 1.8 kg/cm at 4.8V.
- **Speed:** Approximately 0.1 seconds per 60 degrees at 4.8V.
- **Operating Voltage:** 4.8V to 6.0V.
- **Control System:** Pulse Width Modulation (PWM).
- **Weight:** Approximately 9 grams.
- **Rotation Angle:** 0 to 180 degrees.
- **Dimensions:** About 22.2 x 11.8 x 31 mm.

Wiring and Connections:

- **Orange Wire (PWM Signal):** Receives the control signal from a microcontroller or a servo tester.
- **Red Wire (Vcc):** Power supply (4.8V to 6.0V).
- **Brown Wire (Ground):** Connected to the ground of the power supply.

4.1.5 4-wheel chassis kit with DC motor

Figure 4.5: 4-wheel chassis kit with DC motor

A four-wheel chassis kit for an obstacle-avoiding robot is designed to provide a robust and versatile platform for building autonomous robots. The chassis frame, often crafted from durable materials like aluminum or high-strength plastic, serves as the foundation, ensuring stability and support for the entire assembly.

The kit includes four high-torque DC or geared motors, which are essential for propelling the robot and navigating various terrains. These motors are securely mounted on the chassis using brackets and are connected to rubberized wheels that offer excellent traction and maneuverability. The kit also includes a battery holder, typically designed to accommodate rechargeable batteries, providing the necessary power to the motors and electronic components. Additional components in the kit may include jumper wires, screws, and other hardware needed for assembly. Some kits may come with integrated sensors, such as ultrasonic or infrared sensors, which are crucial for obstacle detection and avoidance. Overall, a four-wheel chassis kit offers a comprehensive and flexible solution for hobbyists and developers looking to build obstacle-avoiding robots, combining essential mechanical and electronic components into a single, easy-to-assemble package.

4.1.6 Two 18650 Rechargeable Battery 3.7V

Figure 4.6: Two 18650 Rechargeable Battery 3.7V

A rechargeable battery for an obstacle-avoiding robot is essential for providing a reliable and sustainable power source. Common types include Lithium-ion (Li-ion), Lithium Polymer (Li-Po), and Nickel Metal Hydride (NiMH) batteries, with Li-ion and Li-Po being popular due to their high energy density, lightweight, and longer lifespan. The voltage and capacity of the battery, typically 7.4V or 11.1V and 1000mAh to 3000mAh respectively, should match the robot's power requirements.

These batteries require specific chargers, such as balance chargers for Li-ion and Li-Po, to ensure safe charging. The discharge rate (C-rating) indicates the battery's ability to deliver current quickly, crucial for powering motors. Safety features like overcharge, over-discharge, and short-circuit protection are important to prevent damage and ensure safe operation. The size and weight of the battery should be compatible with the robot's design, and connectors like JST, Tamiya, or XT60 must match the robot's power input. Overall, the right rechargeable battery enhances the robot's performance and safety.

4.2 Software Tools Requirements

4.2.1 Arduino IDE software

Arduino IDE (Integrated Development Environment) is a software platform used for programming Arduino microcontrollers. It provides an easy-to-use interface for writing, compiling, and uploading code to Arduino boards. The IDE supports a simplified version of the C++ programming language and includes a set of libraries and examples to help beginners get started with electronics and robotics projects. It's widely favored for its accessibility, community support, and compatibility with various Arduino board models, making it a versatile tool for both hobbyists and professionals in the field of embedded systems development.

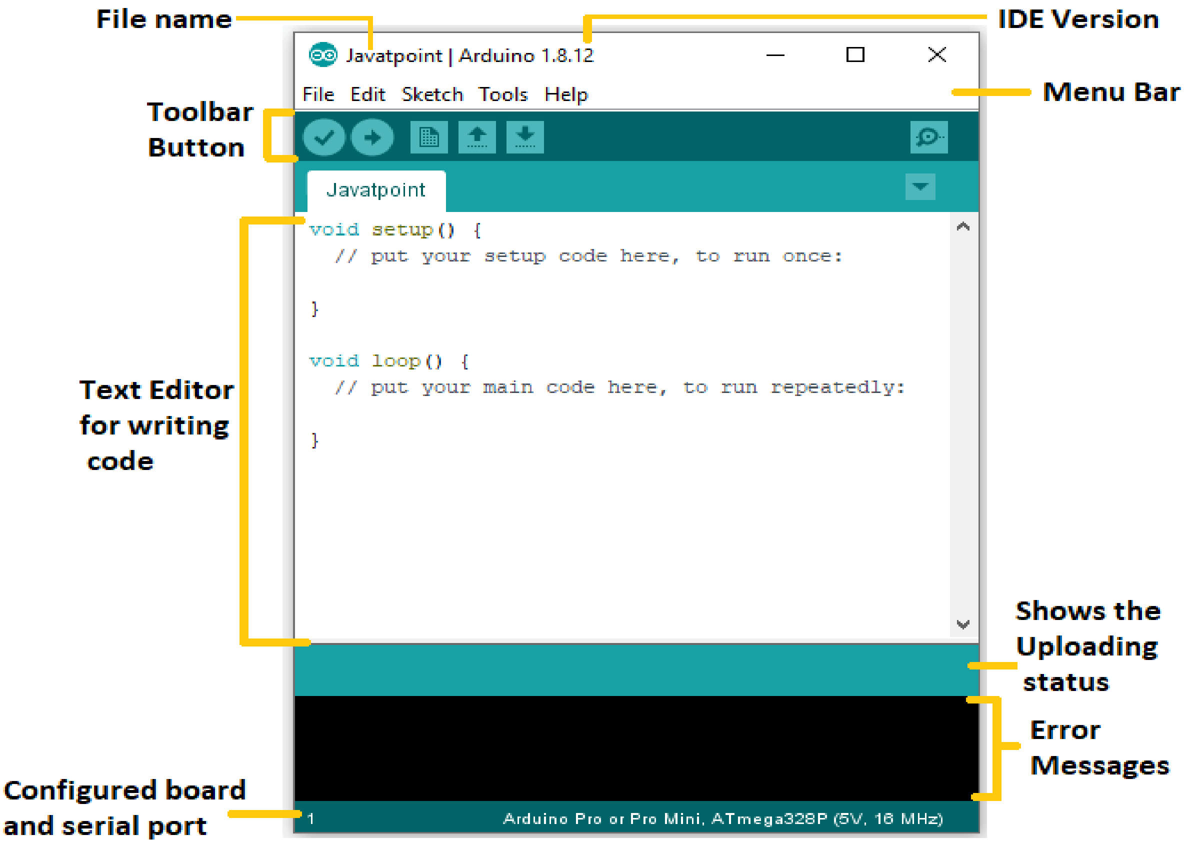

Figure 4.7: Arduino IDE software

The Arduino IDE (Integrated Development Environment) is a software tool that facilitates the programming of Arduino microcontroller boards. Here's how it works:

- **Code Writing:** Users write code in the Arduino IDE using a simplified version of the C++ programming language. The IDE includes a text editor with features like syntax highlighting and auto-completion to assist in writing code efficiently.
- **Compiling:** Once the code is written, the user compiles it within the IDE. Compilation is the process of translating the human-readable code into machine-readable instructions (binary code) that the Arduino board can understand and execute.
- **Uploading:** After successful compilation, the compiled code (also known as a sketch) is uploaded to the Arduino board via a USB cable connected between the computer and the board. The IDE handles the communication and transfer of the sketch to the board.
- **Execution:** Once uploaded, the Arduino board runs the uploaded sketch. The sketch instructs the board on how to interact with its peripherals (sensors, actuators, displays, etc.) based on the programmed logic and instructions.
- **Debugging and Serial Monitor:** The IDE includes tools for debugging, such as the Serial Monitor, which allows users to send and receive data between the Arduino board and the computer. This feature is crucial for troubleshooting and verifying the behavior of the code during runtime.
- **Libraries and Examples:** The Arduino IDE comes with a collection of libraries and example codes that users can utilize to implement various functionalities without starting from scratch. These resources cover a wide range of applications, from basic LED blinking to complex sensor integration and communication protocols.

Chapter 5

RESULTS AND DISCUSSION

The obstacle avoiding robot demonstrated effective performance in detecting and navigating around obstacles. The ultrasonic sensor accurately measured distances, enabling the robot to adjust its path promptly and avoid collisions. This functionality was consistent across various environments, including different lighting conditions and surface types, highlighting the sensor's reliability. The navigation algorithm facilitated smooth and continuous movement, allowing the robot to make precise maneuvers around obstacles without getting stuck or requiring manual intervention. The robot's successful obstacle detection and avoidance showcase the effectiveness of the implemented algorithms and sensor integration. However, there are areas for improvement, such as enhancing the algorithm to handle more complex environments with dynamic obstacles. Additionally, future work could focus on integrating multiple sensors to improve obstacle detection accuracy and developing more sophisticated decision-making algorithms to further enhance the robot's autonomy and efficiency.

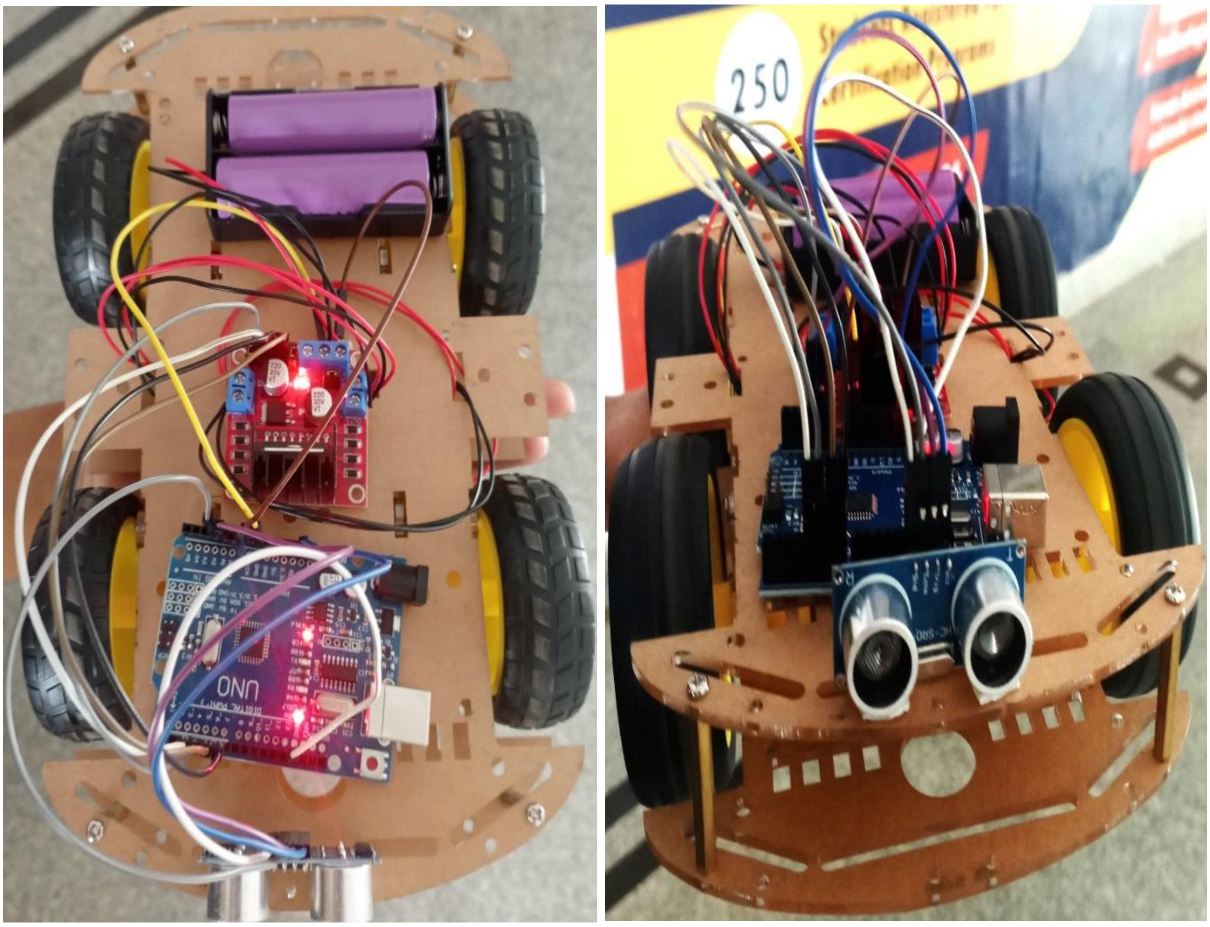

Figure 5.1: Obstacle avoiding robot using Arduino

Chapter 6

ADVANTAGES, DISADVANTAGES AND APPLICATIONS

6.1 Advantages

- Arduino is user-friendly and well-documented, making it accessible for beginners and hobbyists. Its simple programming environment and extensive community support facilitate quick learning and troubleshooting.
- Arduino boards and compatible sensors are generally affordable, making it a cost-effective option for building obstacle-avoiding robots.
- Arduino can be easily integrated with a variety of sensors and modules, allowing for customization and scalability of the robot's capabilities.
- Arduino has a vast collection of libraries for various sensors and modules, simplifying the process of writing code for obstacle detection and avoidance.
- Being open-source, Arduino provides extensive resources, tutorials, and forums where users can find guidance and share projects, fostering a collaborative learning environment.
- Arduino is capable of real-time data processing, which is crucial for timely detection and response to obstacles, ensuring smooth navigation.
- Arduino can interface with various components such as motors, servos, and communication modules, making it versatile for different robotic applications beyond obstacle avoidance.
- Arduino boards typically have low power requirements, which is beneficial for battery-operated robots, extending operational time.

6.2 Disadvantages

- Arduino is user-friendly with extensive documentation and community support.
- Arduino boards and components are inexpensive and budget-friendly.
- Arduino allows easy integration of various sensors and modules.
- The open-source nature promotes innovation and collaboration.
- Many compatible sensors and modules are available.
- Arduino IDE offers a simplified programming environment.
- Arduino handles real-time processing for immediate sensor responses.

- It provides hands-on experience in electronics and robotics.
- A large, active community offers resources and troubleshooting help.
- Arduino supports diverse motor and driver options.

6.3 Applications

- Used for self-driving and navigation in robots and drones.
- Powers home automation tasks like lighting and temperature control.
- Measures and reports data on air quality, temperature, and humidity.
- Controls and monitors machinery and processes in manufacturing.
- Provides hands-on learning experiences in electronics and programming.
- Manages movement and operations of robotic arms for various tasks.
- Integrates into smartwatches and fitness trackers for health monitoring.
- Controls irrigation systems and monitors crop conditions.
- Records and stores data from various sensors for analysis.
- Creates responsive art and installations using sensors and actuators.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The integration of an ultrasonic sensor with the Arduino Uno platform provides a robust solution for real-time obstacle detection and avoidance. The system is designed to navigate through complex environments by utilizing the ultrasonic sensor to measure distances and the Arduino microcontroller to process the data and control the robot's movements. The successful implementation of this approach not only enhances the robot's ability to maneuver around obstacles but also significantly improves its operational efficiency and reliability. The use of the Arduino Uno simplifies the programming and control processes, making it an accessible choice for both educational and practical applications. Arduino's affordability and ease of integration make it a preferred choice for hobbyists and professionals alike. It enables the creation of autonomous vehicles, smart home systems, and data logging solutions, among many others.

7.2 Future Work

- **Sensor Integration:** Incorporate additional sensors, such as infrared or LiDAR, to enhance obstacle detection and navigation accuracy.
- **Advanced Algorithms:** Implement machine learning or adaptive control algorithms for improved path planning and obstacle avoidance.
- **Communication Enhancements:** Add Bluetooth or Wi-Fi modules for remote control and monitoring capabilities.
- **Hardware Upgrades:** Upgrade components like motors and the chassis to increase the robot’s agility and robustness.
- **Real-Time Processing**: Develop real-time data analysis and decision-making systems for more responsive navigation.
- **Autonomous Features:** Implement autonomous charging and multi-robot coordination to extend functionality and operational efficiency.

REFERENCES

1. **Obstacle avoidance robotic** vehicle **using** ultrasonic sensor, android controller. R.VAIRAVAN, S.AJITH KUMAR, L.SHABIN ASHIFF, C.GODWIN JOSE- Int. Res. J. Eng. Technol, 2018 - academia.edu.
2. **Youtube link- https://bit.ly/3IhfCMc** -Multi-function Arduino robot.
3. **Obstacle Avoiding Robot** By Faiza Tabassum, Muhammad Masud Tarek Dr. Bilkis Jamal Ferdosi -2017
4. **Implementation of Obstruction Avoiding Robot using Ultrasonic Sensor and Arduino UNO.** Arjun Varma, Ashwath A, Ayush Verma, A. Bagubali, Kishore V Krishnan-2019
5. **Obstacle Avoidance Robot using an ultrasonic Sensor with Arduino UNO** -Arjun Varma, Ashwath A, Ayush Verma, A. Bagubali, Kishore V Krishnan-2019