KLE Technological University | Creating Value, Leveraging Knowledge

Dr. M. S. Sheshgiri Campus, Belagavi

2024-2025

Department of
Electronics and Communication Engineering

Mini Project Report

on

Smart Driving Assistant

By:

1. **Santoshi Uppin** USN:02FE22BEC085
2. **Vaishnavi Kadlokar** USN:02FE22BEC115
3. **Swaroop Patil** USN:02FE23BEC401
4. **Swayam Mane** USN:02FE22BEC113

Semester: V, 2024-2025

Under the Guidance of

Prof. Vinayak Dalavi

KLE Technological University

Creating Value, Leveraging Knowledge

Dr. M. S. Sheshgiri Campus, Belagavi

2024-2025

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that project entitled **“Smart Driving Assistant ”** is a bonafide work carried out by the student team of **” Santoshi Uppin (02FE22BEC085), Vaishnavi Kadolkar (02FE22BEC115), Swayam Mane (02FE22BEC113), Swaroop Patil (02FE23BEC401)”**. The project report has been approved as it satisfies the requirements with respect to the mini project work prescribed by the university curriculum for B.E. (V Semester) in Department of Electronics and Communication Engineering of KLE Technological University Dr. M. S. Sheshgiri CET Belagavi campus for the academic year 2024-2025.

Prof. Vinayak Dalavi **Guide**	**Dr. Dattaprasad A. Torse** **Head of Department**	**Dr. S. F. Patil** **Principal**

External Viva:

Name of Examiners	**Signature with date**
1.	
2.	

ACKNOWLEDGMENT

We would like to express our sincere gratitude to the following individuals and organizations who have played a significant role in the development of the Smart driving assistant system.

We would like to express our sincere gratitude to our guide, Prof. Vinayak Dalavi, whose invaluable guidance and unwavering support have been instrumental throughout the entirety of our project.

We take the opportunity to thank our project coordinator Dr. U. L. Naik, for providing us motivation and encouragement.

We are grateful to our Head of the Department Dr. Dattaprasad Torse, for granting us the privilege to work on this project and for his valuable support throughout.

We take the opportunity to thank our Principal Dr. S. F. Patil, for providing us the opportunity to undergo the project.

We acknowledge the contributions of all the team members who participated in this project. Each team member brought unique skills and perspectives, contributing to the development of different elements of the project.

We extend our thanks to KLE Dr. MSSCET for providing the necessary resources and infrastructure to carry out this project. Their support was vital in enabling us to conduct the required research, acquire relevant knowledge, and access computing resources.

We take this opportunity to thank all the staff of Electronics and Communication Department for their cooperation and suggestions during the project.

-The project team

ABSTRACT

The smart driving assistant system focuses on the integration of the ESP32 microcontroller into smart transportation systems, highlighting its role in advancing both autonomous driving and industrial automation. By managing real-time data from sensors such as ultrasonic detectors,IR as lane followers, the ESP32 enables efficient obstacle detection, collision avoidance, and precise navigation for autonomous vehicles. Ultrasonic sensors continuously monitor the surroundings, allowing the system to calculate distances to nearby objects and make real-time decisions to avoid collisions, ensuring safe and efficient operation.The microcontroller's real-time data processing capabilities allow these robots to avoid obstacles and operate safely in complex environments, ensuring smooth and efficient operation in industrial settings. Overall, the project demonstrates the ESP32's versatility in enabling cost-effective, scalable, and intelligent solutions for both transportation and industrial automation.

Contents

List of Figures

List of Abbreviations

Abbreviation	Description
ADAS	Advanced Driver Assistance Systems
AROI	Adaptive Region of Interest
cROI	Compact Reconfigurable Input/Output
FSM	Finite State Machine
GPS	Global Positioning System
HT	Hough Transform
IoT	Internet of Things
IR	Infrared
PPHT	Progressive Probabilistic Hough Transform
RANSAC	Random Sample Consensus
ROI	Region of Interest
V2I	vehicle-to-infrastructure
V2V	Vehicle-to-vehicle

Chapter 1

Introduction

The integration of ESP32 microcontrollers into smart transportation systems has significantly advanced the development of autonomous driving technologies. These systems employ the ESP32 to manage real-time data processing, sensor integration, and communication, making them a cost-effective and efficient solution for achieving traffic awareness and safety. Autonomous vehicles equipped with ESP32-based systems use ultrasonic sensors for obstacle detection, enabling collision avoidance by calculating the distance to objects and rerouting accordingly. GPS modules integrated with the ESP32 facilitate precise navigation and path planning.

Driver fatigue and negligence, common causes of accidents, can also be addressed through ESP32-powered systems. For instance, the ESP32 can be used in driver alertness monitoring systems, detecting lane departures or signs of drowsiness and issuing timely warnings. These features are essential for reducing traffic-related fatalities and ensuring safer roads.

In the industrial and logistics sectors, ESP32-powered line-following robots are becoming popular for their ability to autonomously transport objects while avoiding obstacles. Equipped with multiple sensors and wireless connectivity provided by the ESP32, these robots streamline workflows, enhance productivity, and reduce dependency on traditional rail or conveyor solutions. Moreover, the ESP32's IoT capabilities enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, fostering a connected ecosystem of smart vehicles. This real-time data sharing improves traffic flow, reduces congestion, and enhances overall safety. The ESP32's versatility and processing power make it an ideal choice for implementing cost-effective, scalable, and intelligent solutions in modern autonomous transportation systems.

1.1 Motivation

This project is driven by the goal of creating a real-time, low-cost obstacle detection and avoidance system utilizing the ESP32 microcontroller, with a focus on enhancing efficiency and reliability. By employing ultrasonic sensors for accurate obstacle detection and GPS modules for precise navigation, the system is designed to operate seamlessly in various environments. The ESP32's powerful processing capabilities and IoT connectivity support the implementation of advanced algorithms for real-time decision-making, ensuring improved obstacle avoidance. This solution aims to provide a scalable, cost-effective system suitable for autonomous vehicles, industrial automation, and logistics, prioritizing safety and operational efficiency.

1.2 Objectives

- Obstacle Detection: Ensure timely detection of obstacles using ultrasonic sensors and prevent collisions through alerts and automated braking.

- Lane Detection: Maintain vehicle alignment within lanes using IR sensors or cameras, with automatic correction and alerts for lane departure.
- Control and Navigation: Utilize a robust FSM to handle vehicle movement, lane changes, and speed adjustments in real-time, ensuring smooth, autonomous navigation.
- To provide a scalable and efficient solution for applications in autonomous vehicles, industrial automation, and logistics.

1.3 Literature Survey

1.3.1 An IoT based Obstacle Detection and Line Follower Robot using LabVIEW

The document outlines the development of an IoT-based obstacle detection and line-following robot using the ESP32 microcontroller, designed for autonomous navigation. The project integrates ultrasonic sensors for obstacle detection, infrared sensors for line tracking, and a GPS module for real-time location tracking. It employs LabVIEW for visualization and control, with Firebase Cloud providing remote monitoring capabilities. The robot autonomously follows a predefined path, avoids obstacles, and transmits its position to the cloud, demonstrating its potential for industrial and logistical applications. The design emphasizes cost-effectiveness, efficiency, and adaptability to various environments.[6]

1.3.2 A Lane Tracking Method Based on Progressive Probabilistic Hough Transform

The paper "A Lane Tracking Method Based on Progressive Probabilistic Hough Transform" proposes a real-time, vision-based lane detection and tracking system for Advanced Driver Assistance Systems (ADAS), addressing challenges like varying lighting, occlusions, and diverse road types. The approach involves three main stages: pre-processing input images using the Sobel filter and Otsu's thresholding to handle noise and lighting variations, defining an Adaptive Region of Interest (AROI) using a horizon line to reduce computational complexity, and detecting and tracking lanes with the Progressive Probabilistic Hough Transform (PPHT) enhanced by Kalman filtering and K-means clustering. Tested on real-world datasets under diverse conditions, the system achieves a 93.82 detection rate and processes at 21.54 ms per frame, outperforming many existing methods in robustness and efficiency. The authors suggest further improvements through hardware-software integration for better performance in real-time applications.[4]

1.3.3 Detecting Obstacles Within the Driving Lane and Vehicle Speed Adjustment

This presents an algorithm for detecting obstacles in the driving lane and adjusting vehicle speed accordingly, with the goal of enhancing autonomous driving systems. The proposed solution relies on traditional computer vision techniques such as color filtering, Gaussian blur, Canny edge detection, and RANSAC for lane detection and obstacle identification. Developed using OpenCV and tested within the Carla simulator, the algorithm identifies lane boundaries, detects obstacles (vehicles and pedestrians), classifies them, and estimates their distances. Based on these inputs, the vehicle adjusts its speed or applies brakes to avoid collisions. [3]

1.3.4 Integration of Hough Transform and Inter-Frame Clustering for Road Lane Detection and Tracking

The paper titled "Integration of Hough Transform and Inter-Frame Clustering for Road Lane Detection and Tracking" presents a vision-based algorithm for detecting and tracking road lanes. It utilizes image processing techniques such as color space conversion, region of interest (ROI) selection, edge detection, Hough Transform (HT), slope filtering, and inter-frame clustering. A key feature is the use of the Kalman filter for lane tracking, enhancing robustness in various scenarios including shadows, illumination changes, and missing lane markings. The algorithm demonstrates high accuracy on the Caltech dataset, achieving a correct rate of 84.17. Implemented on a National Instruments cRIO processor using LabVIEW, it operates efficiently in real-time, with a processing time of 40 milliseconds per frame. This approach offers a balance between computational efficiency and accuracy, making it suitable for real-world driver assistance systems.[1]

1.3.5 The Complex Lane Line Detection Under Autonomous Driving

This paper proposes a robust method for detecting lane lines in challenging conditions to enhance the reliability of autonomous driving systems. The authors introduce a fusion-based image enhancement technique combining Single-scale Retinex (SSR) and Gamma correction to improve lane line visibility under adverse lighting conditions, such as shadows and low light. The process involves extracting feature points of lane lines using advanced image processing, fitting and filtering these points to accurately reconstruct lane lines, and applying conditional processing to address scenarios like intersections and curves. Experimental results on various datasets demonstrate the method's effectiveness in improving accuracy and robustness in complex driving environments, making it a valuable contribution to advanced driver assistance systems (ADAS).[5]

1.3.6 Autonomous Car Controller using Behaviour Planning based on Finite State Machine

The paper "Autonomous Car Controller using Behaviour Planning based on Finite State Machine" proposes a control mechanism for self-driving cars, focusing on decision-making, speed control, traffic sign detection, and obstacle avoidance using a Finite State Machine (FSM) model. The FSM framework, designed with JFLAP and implemented using Python's Pygame, consists of 28 states representing various traffic scenarios and vehicle actions. The model enhances the car's ability to make adaptive decisions, such as lane changes and speed adjustments, ensuring safe navigation on multilane highways. By utilizing sensor inputs like cameras, radars, and LiDAR, the FSM guides the car through real-time traffic conditions. Evaluation of the system demonstrates its effectiveness in collision avoidance, lane-keeping, and maintaining speed limits, highlighting FSM's simplicity, reliability, and computational efficiency for autonomous vehicle behavior planning. Despite its strengths, the paper acknowledges limitations, such as challenges in handling unforeseen scenarios, suggesting potential future improvements.[2]

1.4 Problem statement

Design and implement a Smart Driving Assistant System using ESP32

1.5 Application in Societal Context

Designing a smart driving assistant system using ESP32 with sensors has several important applications in a societal context, enhancing the performance and reliability of various electronic and automative systems:

1. **Enhanced Road Safety:** By integrating automatic obstacle detection and lane keeping, your project can contribute to reducing road accidents, especially in challenging driving conditions, thereby improving overall traffic safety

2. **Advanced Driver Assistance Systems (ADAS):** The project can be used in ADAS applications, providing real-time alerts for obstacles and lane deviations, which enhances vehicle safety and assists drivers in making better decisions.

3. **Public Transportation:** In buses, taxis, and other public transportation systems, the automatic obstacle and lane detection system can help improve driver awareness and reduce accidents, contributing to safer and more reliable public transportation networks

4. **Cost-effective Safety Solutions:** Using the ESP32, a low-cost microcontroller, makes this system affordable, which could make it accessible for use in budget-friendly vehicles, thereby extending safety features to a wider audience.

5. **Driver Fatigue Prevention:** By detecting lane departures and obstacles, your system can assist in reducing driver fatigue and distraction, providing warnings when attention is needed, thus promoting safer driving behaviors

6. **Improved Road Infrastructure:** Improved Road Infrastructure: The data gathered from such systems can also be used to assess and improve road infrastructure, contributing to better road design and maintenance..

Chapter 2

Project planning

Project planning is critical for ensuring project success by clearly defining objectives, identifying tasks, and setting timelines. It enables efficient resource allocation, risk management, and helps to maintain focus and alignment among team members. Additionally, it enhances communication, allowing for smooth coordination, and ensures that milestones are met within budget and scope.

2.1 Gantt chart

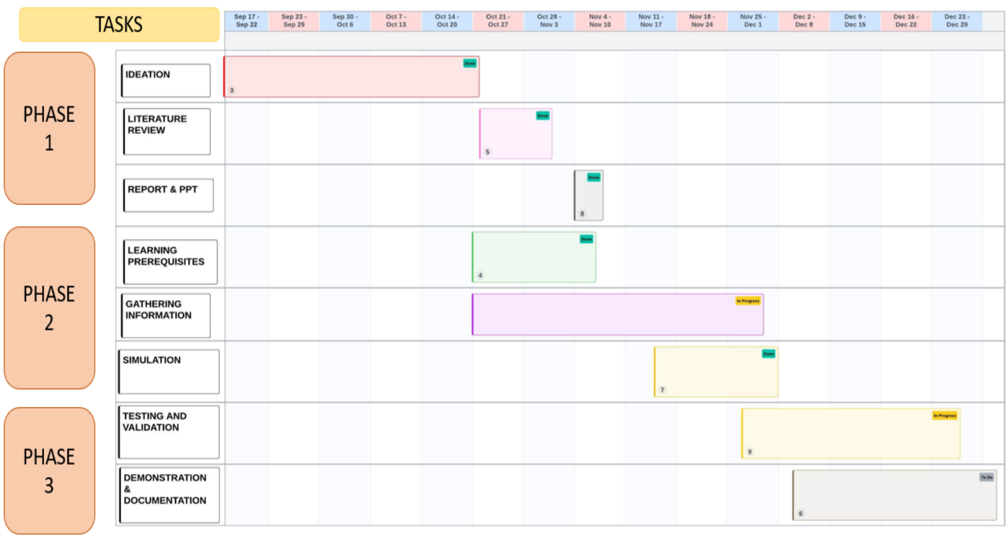

Figure 2.1: Gantt chart

A Gantt chart is a visual project management tool that represents the timeline of a project. It displays tasks or activities as horizontal bars along a timeline, with the length of each bar indicating the duration of a task.The Gantt chart helps project managers track progress, manage deadlines, and visualize how tasks overlap or depend on one another. It is widely used for scheduling, resource allocation, and ensuring timely completion of projects.

2.2 Work Breakdown Structure(WBS)

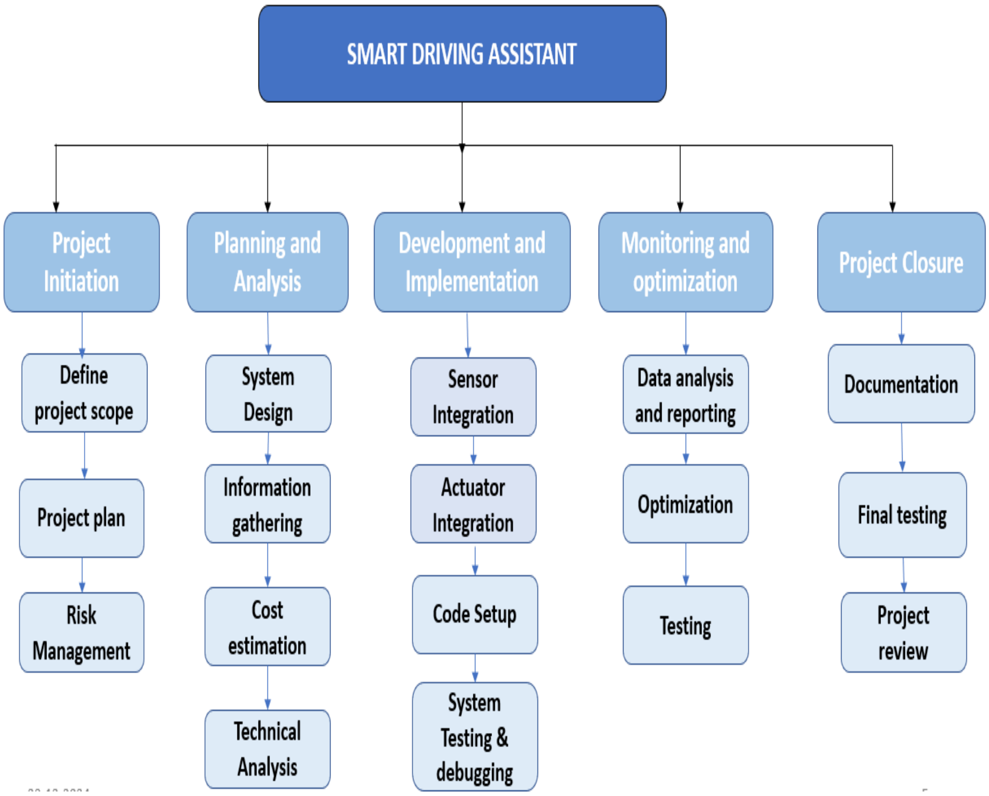

Figure 2.2: Work breakdown structure

A Work Breakdown Structure (WBS) is a project management tool that divides a project into smaller, manageable components or tasks. It organizes the work into a hierarchy, making it easier to plan, assign resources, and track progress. The WBS helps clarify the scope of the project, ensuring that all required work is identified. It also enhances communication among team members and stakeholders, assigns responsibilities, and improves time and resource management. By breaking down complex projects into manageable parts, it makes it easier to estimate costs, schedule timelines, and monitor progress.

Chapter 3

Design specifications

3.1 Input specifications

3.1.1 ESP32 :-

Central processing unit for data collection and processing. The ESP32-WROOM is a highly integrated and versatile Wi-Fi and Bluetooth microcontroller module, based on the ESP32 dual-core processor. It features a Tensilica LX6 CPU, operating at up to 240 MHz, with built-in 2.4 GHz Wi-Fi and Bluetooth 4.2/LE connectivity, making it ideal for a wide range of IoT and embedded applications. The module has a rich set of peripherals, including 34 GPIO pins, 12-bit ADCs, DACs, SPI, I2C, UART, PWM, and touch sensors, allowing it to interface with various sensors and actuators. The ESP32-WROOM supports up to 16 MB of flash memory, with 520 KB of SRAM, offering ample space for complex applications and data storage. It has built-in hardware encryption for secure data transmission, along with low-power modes, making it suitable for energy-efficient applications. Its robust RF performance and ability to operate in various environments further enhance its reliability. The module's small size, powerful performance, and extensive connectivity options make it a popular choice for IoT projects, smart devices, and automation systems.

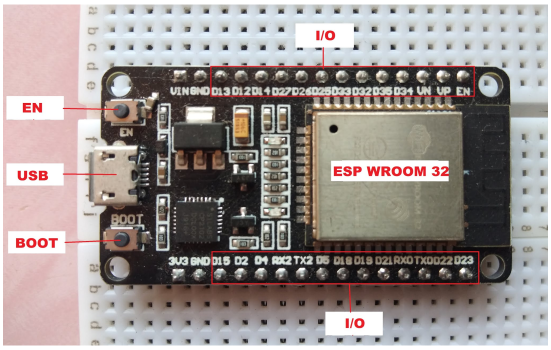

Figure 3.1: ESP-32 WROOM

3.1.2 ULTRASONIC SENSOR(HC-SR04) :-

The HC-SR04 ultrasonic sensor is a widely used distance measuring sensor that operates using ultrasonic sound waves to detect objects and measure their distance from the sensor. It consists of two main components: a transmitter and a receiver. The transmitter emits ultrasonic waves at a frequency of 40 kHz, which travel through the air until they hit an object. The reflected waves are then picked up by the receiver, and the time taken for the signal to return is used to calculate the distance based on the speed of sound.

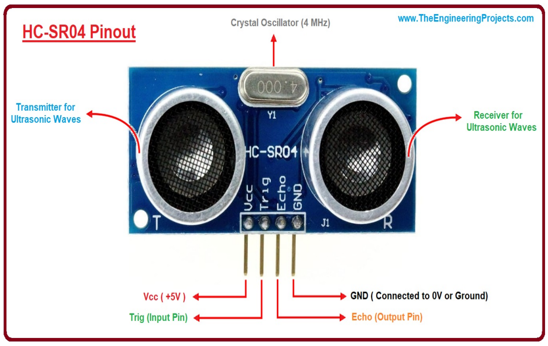

Figure 3.2: ULTRASONIC SENSOR(HC-SR04)

The sensor provides highly accurate distance measurements in the range of 2 cm to 400 cm with an accuracy of about 3 mm. It operates with a 5V DC power supply and consumes very low current, typically around 15 mA. The HC-SR04 uses a trigger pin to send a pulse and an echo pin to receive the reflected signal, and it communicates with microcontrollers like the ESP32 through simple digital signals. Its compact design, low cost, and reliable performance make it a popular choice for robotics, obstacle detection, and distance measurement applications.

3.1.3 INFRARED SENSOR:-

An infrared (IR) sensor is an electronic device that uses infrared light to detect objects and measure distances or proximity. It typically consists of two main components: an IR LED (emitter) that emits infrared light and a photodiode (receiver) that detects the reflected light from nearby objects. When an object comes within the sensor's detection range, the emitted infrared light reflects off the object and is captured by the photodiode, triggering a response.

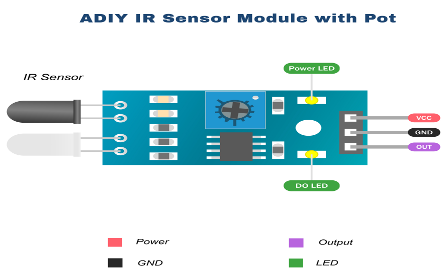

Figure 3.3: INFRARED SENSOR

IR sensors are commonly used for proximity detection and line-following applications. They typically operate on a 5V DC power supply and have a detection range of 2 cm to 30 cm, though the range may vary depending on the object's surface and ambient lighting conditions. The sensor outputs a simple digital signal (HIGH or LOW) when an object is detected, making it easy to interface with microcontrollers like the ESP32. IR sensors are widely used in robotics, automation, and obstacle detection due to their simplicity, low cost, and fast response times. Their ability to differentiate between light and dark surfaces makes them especially useful in line-following robots.

3.2 Output specifications

3.2.1 BUZZER:-

A buzzer is an electronic sound-producing device used in a variety of applications to provide audible signals or alerts. It operates by converting electrical energy into sound through mechanical vibrations. Buzzers are typically of two types: piezoelectric and electromechanical. The piezoelectric type uses a ceramic material that vibrates when an electric signal is applied, while the electromechanical type uses a magnetic coil to produce sound.

Buzzers commonly run on low voltage, usually 3V to 12V DC, and can be controlled by microcontrollers such as the ESP32 using a digital output pin. When activated, the buzzer produces a loud, distinct sound that can be used as an alert for events such as obstacle detection, alarms, or status notifications. The simplicity, low power consumption, and ease of integration make buzzers a popular choice in systems requiring auditory feedback, such as robotics, security systems, and automated alerts.

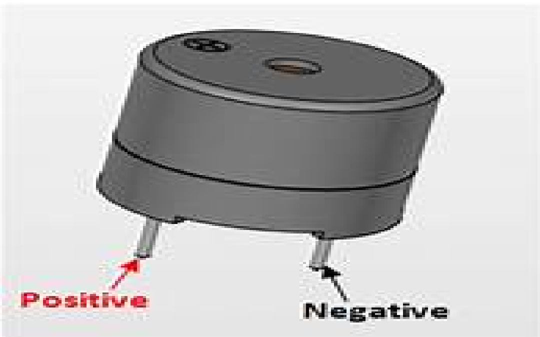

Figure 3.4: BUZZER

3.2.2 MOTOR DRIVER(L298) :-

.

A motor driver, such as the L298, is an integrated circuit used to control the direction and speed of DC motors or stepper motors. The L298 is a dual H-bridge motor driver, which means it can independently control two motors simultaneously. Each H-bridge within the driver consists of four switches that can control the flow of current through the motor, allowing it to spin in both directions.

The L298 operates on a 5V logic supply and can handle motor supply voltages up to 46V, with a maximum current of 2A per channel. It is ideal for controlling motors in robotics, automation, and embedded systems applications, especially when working with microcontrollers like the ESP32. The L298 receives input signals from the microcontroller through its input pins, which determine the direction of motor rotation, while PWM signals control the speed.

For protection, the L298 includes diodes to prevent back EMF (electromotive force) from damaging the circuit. It can be easily interfaced with different types of motors, such as DC motors or bipolar stepper motors, making it versatile for various projects. Its ability to control multiple motors with independent direction and speed makes the L298 a popular choice for robotics, motorized vehicles, and automation projects.

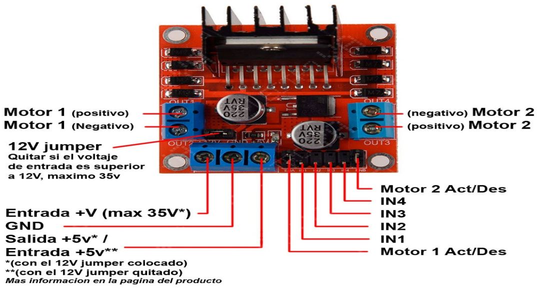

Figure 3.5: MOTOR DRIVER(L298)

3.2.3 DC MOTOR:-

A DC motor is an electric motor that runs on direct current (DC) electricity. It consists of a stator, which provides a magnetic field, and a rotor (or armature), which is a coil of wire that rotates within the magnetic field when an electric current is passed through it. The current flows through the rotor, generating a magnetic force that causes the rotor to turn.

DC motors are widely used for their simplicity, ease of control, and ability to provide variable speed and torque. The direction of rotation can be changed by reversing the polarity of the applied voltage. The speed of the motor is typically controlled using Pulse Width Modulation (PWM), which adjusts the amount of time the motor is powered during each cycle, thereby controlling the average voltage and speed.

DC motors are commonly found in a variety of applications, such as in robotics, appliances, and small machinery. They are popular in systems requiring precise control over movement, as they offer smooth, continuous rotation and can easily be integrated with controllers like microcontrollers or motor drivers for various tasks, including automation and robotics.

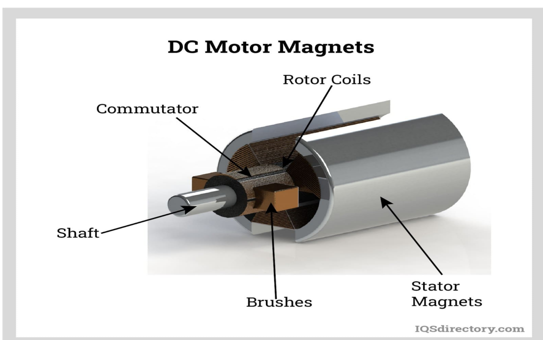

Figure 3.6: DC MOTOR

3.3 Software specifications

3.3.1 Aurdino IDE

The Arduino IDE (Integrated Development Environment) is a software platform used to write, compile, and upload code to Arduino boards. It supports various microcontroller boards, with the primary focus on the Arduino series, and enables users to develop projects easily using a simplified programming environment. The IDE uses the C/C++ programming language, providing a user-friendly interface for writing code, managing libraries, and uploading programs directly to the Arduino board via USB.

The Arduino IDE features a built-in editor that highlights syntax and helps manage code, along with an integrated serial monitor for debugging and real-time data communication with the board. It supports a wide range of libraries for various sensors, actuators, and communication protocols, allowing developers to access pre-built functions and simplify their projects. The platform also enables the use of external boards and chips with custom settings, making it versatile for different hardware.

Compatible with Windows, macOS, and Linux, the Arduino IDE is open-source, allowing developers to contribute to the community and extend its capabilities. Its simplicity and broad range of features make it an ideal tool for both beginners and advanced users working on embedded systems, IoT projects, and electronics.

3.3.2 Proteus

Proteus is a popular electronic design automation (EDA) software suite used for circuit design, simulation, and PCB (printed circuit board) layout. It allows users to create and simulate electronic circuits using a wide range of components, including microcontrollers, sensors, and power devices. The software offers a powerful simulation engine that can model both analog and digital circuits, enabling users to test and verify the behavior of their designs before physically building them.

One of the standout features of Proteus is its ability to simulate embedded systems, allowing for the integration of real-world microcontroller programming. It supports various programming languages, including assembly, C, and C++, and can directly simulate the code written for microcontrollers like Arduino, PIC, and ARM-based systems. This enables users to test their code and circuit interactions in a virtual environment, reducing the need for physical prototypes during the early stages of development.

3.4 Resource specifications

1. OS - Window's 11th generation
2. System (Laptop) configuration -
   - Processor: Intel® Core™ i5-13420H (up to 4.6 GHz, 8cores, 12 threads)
   - Memory: 16 GB DDR4-3200 MT/s (2 x 8 GB)
   - Storage: 512 GB PCIe® Gen4 NVMe™ M.2 SSD
   - Graphics: NVIDIA® GeForce RTX™ 2050 Laptop GPU (4 GB GDDR6 dedicated)
3. Continuous power supply of 12V to ESP-32

3.5 List of tools used

1. Arduino IDE
2. Proteus professional
3. Circuit.io
4. Wokwi

3.6 Programming languages used

1. C++ Programming Language

   The core language used is C++, a general-purpose, object-oriented programming language. Arduino sketches (programs) are written in a subset of C++, which is one of the most widely used languages for embedded systems and microcontroller programming.

2. Arduino IDE / Framework

   The Arduino platform provides a framework on top of C++ to simplify microcontroller programming. It abstracts many lower-level details, making it easier for developers to interact with the hardware components like sensors, motors, and buzzers.

3. Embedded Systems Programming

   The code is written for embedded systems, where the program interacts directly with hardware components such as sensors, motors, and buzzers.

Chapter 4

Methodology

4.1 Methodology

4.1.1 System Design and Component Selection:

Select the ESP32 microcontroller as the main controller for processing data. Choose ultrasonic sensors for real-time obstacle detection and infrared (IR) sensors or a line tracking sensor array for lane detection.

4.1.2 Hardware integration

Interface the ESP32 with the ultrasonic sensors to measure distance and detect obstacles. Connect IR sensors to the ESP32 to detect lane boundaries by measuring the contrast between the road surface and lane markings. Integrate auditory alerts (buzzer) to signal obstacles and lane deviations.The ESP32 is programmed using the Arduino IDE and framework, enabling multitasking for efficient motor management, sensor inputs, obstacle avoidance, and line following. DC motors are controlled via L298N motor drivers connected to the ESP32's GPIO pins, using PWM for speed and direction. Infrared sensors for lane following and an ultrasonic sensor for obstacle detection are also connected to the GPIO pins.A stable power source is essential for uninterrupted operation, with proper circuit connections ensuring reliable performance.

4.1.3 Simulation

Write code for the ESP32 using arduino ide to process sensor data in real-time and control the output devices (motors, buzzer) and simulate using proteus. Implement obstacle detection logic using the distance data from the ultrasonic sensors. Implement lane detection by analyzing input from the IR sensors to determine lane positioning.

4.1.4 Testing and Calibration

Test the system by placing it in controlled environments to verify obstacle detection accuracy and lane-following capabilities.Calibrate the sensors to ensure reliable detection under different conditions (e.g., lighting, road surface).

4.1.5 Optimization and Scalability

Optimize the system for power efficiency and responsiveness. Expand the system with additional sensors or modules if necessary, making it scalable for autonomous vehicles and robotics applications

This methodology ensures a systematic approach to developing an efficient, low-cost obstacle and lane detection system.Select the ESP32 microcontroller as the main controller for processing data.

4.2 Functional block diagrams

A Functional Block Diagram (FBD) is a graphical representation of the functions and operations within a system, showing how the various components interact to achieve a specific objective.

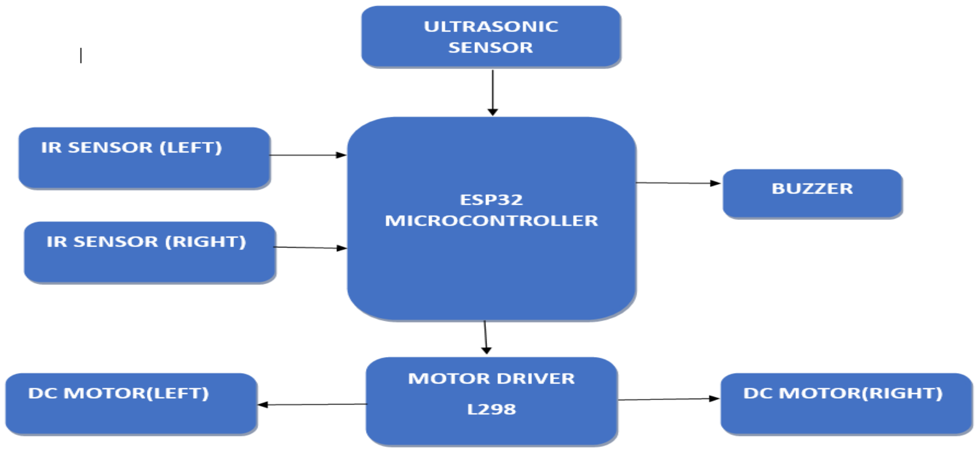

Figure 4.1: Functional block diagram

4.2.1 Flow chart

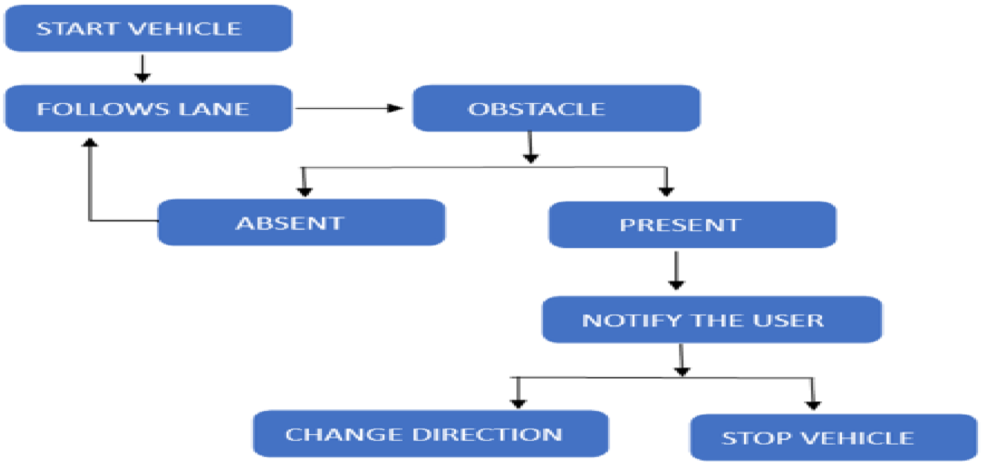

Figure 4.2: Flow chart

A flowchart is a graphical representation of a process, system, or algorithm, outlining the sequence of steps or operations to achieve a particular goal. It uses standardized symbols to visually depict each step and the flow of control from one step to the next, making it easy to understand and analyze complex processes.

4.2.2 Bill of material

Table 4.1: Bill of material

SL NO.	COMPONENTS	QUANTITY	PRICE(Rs.)
01.	ESP32	1	450
02.	INFRARED SENSOR	2	140
03.	ULTRASONIC SENSOR	2	120
04.	DC MOTOR	2	300
05.	BATTERY	1	50
06.	BREADBOARD	1	30
07.	JUMPER WIRES	1 SET	30
08.	CHASSIS	1	150
09.	BUZZER	1	10
10.	MOTOR DRIVER	1	150
	TOTAL		**1450**

4.3 Simulation

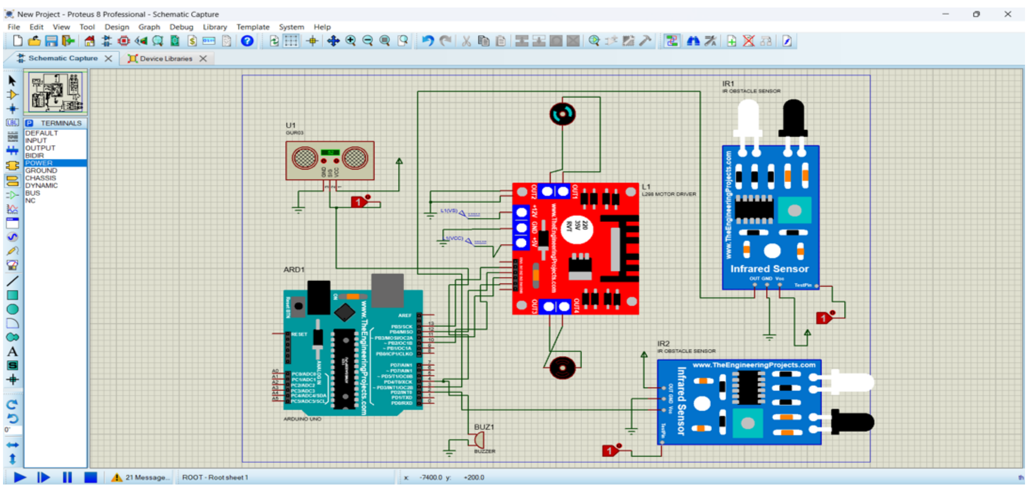

Figure 4.3: Simulation

Chapter 5

Results and discussions

5.1 Smart Driving Assistant System

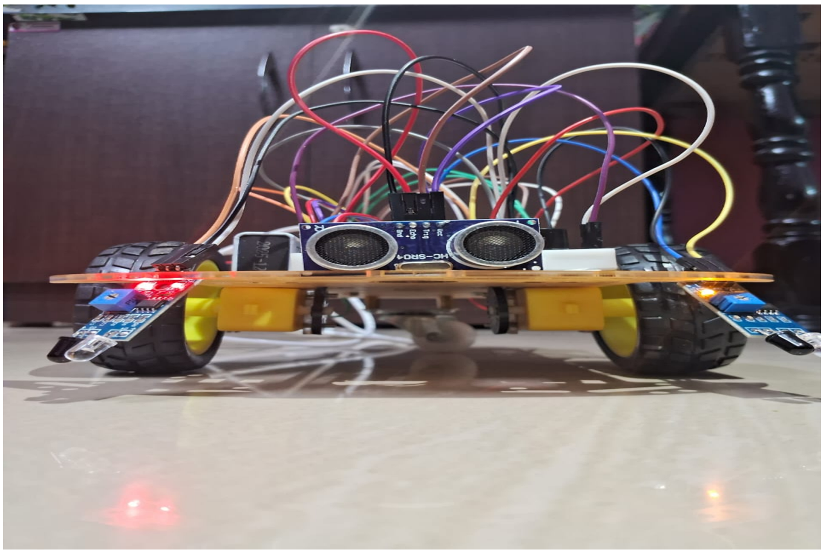

Figure 5.1: Result

The Smart Driving Assistant system, implemented with the ESP32 microcontroller and ultrasonic and infrared sensors, demonstrated effective real-time performance in detecting obstacles and identifying lane deviations. Ultrasonic sensors reliably detected obstacles within 2 cm to 40 cm, triggering prompt auditory alerts. Infrared sensors accurately identified lane boundaries and deviations on high-contrast surfaces. The system showed efficient real-time processing, with minimal delays, and consumed approximately 150 mA, making it suitable for battery-powered applications. Its scalability allows for easy integration of additional sensors or wireless communication, enhancing adaptability for various use cases.

Chapter 6

Conclusions and future scope

6.1 Conclusion

In conclusion, the ESP32-based obstacle and lane detection project showcases a powerful yet cost-effective solution for real-time navigation and safety enhancement in autonomous vehicles or robotic systems. By leveraging the processing capabilities of the ESP32 along with carefully integrated sensors, the system effectively detects obstacles and identifies lane boundaries, making it a crucial component for improving autonomous movement and decision-making.

The project's design focuses on providing a high level of accuracy and responsiveness, addressing the challenges of obstacle detection and lane tracking in dynamic environments. The integration of various sensors, such as ultrasonic or infrared, with the ESP32 microcontroller ensures efficient data collection and processing, allowing for real-time feedback that is essential for autonomous systems to operate safely and effectively.

The affordable hardware utilized in the project makes it an ideal solution for both prototype development and real-world applications, where cost constraints are often a consideration. Additionally, the ESP32's connectivity options enable potential future enhancements, such as remote monitoring or integration with other autonomous systems. This makes the project not only viable for smaller-scale or hobbyist applications but also scalable for larger, more complex systems.

Overall, the successful implementation of this obstacle and lane detection system demonstrates the feasibility of using cost-effective hardware to achieve reliable and efficient performance in real-world scenarios. The project lays the groundwork for further advancements in autonomous navigation technology and can be expanded to incorporate additional sensors, improve processing algorithms, or optimize energy consumption for long-term operation.

6.2 Future scope

The proposed design for our project holds considerable potential and serves as a solid foundation for further advancements aimed at improving its overall performance. Future research could explore several key areas to enhance the system's efficiency, scalability, and adaptability. One crucial direction is optimizing the system's robustness against external disturbances, such as environmental noise or interference. While the current design has demonstrated resilience, fine-tuning certain components to further minimize these effects could result in even more stable and reliable operation under varied conditions. Additionally, power optimization remains a key priority, particularly for applications in battery-operated devices. Exploring techniques for reducing power consumption without sacrificing performance, such as incorporating low-power components or utilizing energy harvesting methods, could substantially extend operational time and reduce operational costs.

Another important area for future development is the scalability of the design for large-scale integration. Investigating ways to minimize the system's footprint while maintaining functionality and efficiency will be essential for adapting the design to more compact platforms, particularly for consumer electronics and IoT devices. Moreover, enhancing the system's flexibility to support a broader range of applications and environments will be crucial. Implementing adaptive mechanisms or modular features could enable the design to seamlessly integrate into various use cases, from high-performance computing to low-power, resource-constrained devices. By focusing on these strategic improvements, future work can refine the design, ensuring its long-term relevance and widespread applicability across diverse fields.

Bibliography

[1] Sandeep Bisht, N Sukumar, and Parasuraman Sumathi. Integration of hough transform and inter-frame clustering for road lane detection and tracking. In *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6. IEEE, 2022.

[2] Mandava Jaswanth, Namburi KL Narayana, Sreedharreddy Rahul, and M Supriya. Autonomous car controller using behaviour planning based on finite state machine. In *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 296–302. IEEE, 2022.

[3] Sran Ljepić, Ratko Grbić, Jelena Kovačević, and Momčilo Krunić. Detecting obstacles within the driving lane and vehicle speed adjustment. In *2021 Zooming Innovation in Consumer Technologies Conference (ZINC)*, pages 225–230. IEEE, 2021.

[4] Mehrez Marzougui, Areej Alasiry, Yassin Kortli, and Jamel Baili. A lane tracking method based on progressive probabilistic hough transform. *IEEE access*, 8:84893–84905, 2020.

[5] Huan Shen. Complex lane line detection under autonomous driving. In *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pages 627–631. IEEE, 2020.

[6] R Phani Vidyadhar, Domakonda Sai Mallik, Kavali Uday Kumar, and Bhukya Renuka. An iot based obstacle detection and line follower robot using labview. In *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pages 852–857. IEEE, 2024.