PALESTINE POLYTECHNIC UNIVERSITY
College of Information Technology and Computer Engineering
Computer Engineering Department

Graduation Project

Intelligent Vehicle-to-Vehicle and Vehicle-to-Infrastructure Communication System

Submitted by

Yousef Hantash

Mostafa Tahboub

Mohammad Salameen

Supervisor

Dr.Elayan Abu Gharbyeh

2024 - 2025

Certification and Anti-Plagiarism Declaration

This is to declare that the graduation project produced under the supervision of **Dr. Elayan Abu Gharbyeh** having the title **"Intelligent Vehicle-to-Vehicle and Vehicle-to-Infrastructure Communication System"** was prepared by the students listed below in partial fulfillment of the requirements for the degree of Bachelor in **Computer Systems Engineering**.

No part of this project has been reproduced illegally (in particular: cut and paste), which can be considered as plagiarism. All referenced parts have been used to support and argue the idea and have been cited properly.

We certify that we will not commit any plagiarism, cheating, or any other academic integrity violations. We will be responsible and liable for any consequences if a violation of this declaration is proven.

Date: 26 , Jan 2025

Graduation Project Group's Students:

Name	**Signature**
Yousef Hantash	
Mostafa Tahboub	
Mohammad Salameen	

Abbreviations

Abbreviation	Meaning
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
IoT	Internet of Things
RFID	Radio Frequency Identification
DSRC	Dedicated Short Range Communications
Wi-Fi	Wireless Fidelity
LCD	Liquid Crystal Display
ESP32	Microcontroller by Espressif
UHF	Ultra High Frequency
HF	High Frequency
LF	Low Frequency
UART	Universal Asynchronous Receiver-Transmitter
ADC	Analog-to-Digital Converter
API	Application Programming Interface
IDE	Integrated Development Environment
GND	Ground
VCC	Voltage Common Collector
WPA	Wi-Fi Protected Access

Table 1: List of Abbreviations

Acknowledgement

In the name of ”Allah”, the most beneficent and merciful who gave us strength, knowledge and helped us to get through this project.

To the people that have inspired, supported, and molded us into the people that we are today, our families, friends and our supervisor. We would’ve never been able to reach this achievement without your support, care and encouragement.

We would like to express our sincere appreciation to our graduation project supervisor Dr. Elayan Abu Gharbyeh for his guidance, continuous encouragement and support throughout the semester, for being there to answer our questions and provide us with valuable suggestions and tips which helped us in choosing some system components.

Furthermore, we have to show our gratitude to Eng. Wael Takrouri for being there to answer our questions and provide us with valuable suggestions and tips which helped us in choosing some system components.

Moreover, it is our duty to thank our families for their generous encouragement and continuous support throughout our life. We are truly grateful for all your support throughout the whole education stage.

At last, we would like to thank all the people who helped, supported and encouraged us to successfully finish the graduation project.

الملخص

يقدم هذا المشروع نظامًا متكاملًا لمراقبة المركبات وتنبيه السائق يعتمد على تقنيات إنترنت الأشياء، ويهدف إلى تعزيز السلامة على الطرق وتقديم المساعدة للسائق من خلال معالجة البيانات في الوقت الفعلي وآليات استجابة سريعة وغير معيقة. يعتمد النظام على مجموعة من المستشعرات لمراقبة العقبات المحتملة، وحوادث الاصطدام، وموقع المركبة بشكل مستمر. يقوم مستشعر المسافة بقياس المسافات مع الأجسام القريبة بشكل ديناميكي، مع تنبيه السائق في حالة الاقتراب الشديد لتحسين الوقاية من الاصطدامات.

بالإضافة إلى ذلك، يكشف مستشعر الحوادث عن الحوادث فور وقوعها، مما يؤدي إلى تفعيل استجابات فورية مع الاستمرار في مراقبة وظائف النظام الأخرى بشكل متزامن. يعتمد النظام على التخزين السحابي لتخزين البيانات، مما يتيح سهولة الوصول إلى البيانات والتواصل بين الأجهزة بسلاسة. وتشمل الميزات الرئيسية للنظام تحديد موقع المركبة باستخدام تقنية تحديد الهوية بالموجات الراديوية، مما يتيح تحديد الموقع الحالي للمركبة بدقة وبشكل آلي، بالإضافة إلى إرسال إشعارات فورية عند وقوع الحوادث وتقديم تقارير عن حالة المركبة في الوقت الفعلي عبر الأوامر اللاسلكية.

تعزز واجهة العرض الإلكترونية من تفاعل المستخدم من خلال عرض تحديثات الحالة والتنبيهات، بينما تتضمن بنية النظام وظائف تأخير غير معيقة لضمان استجابة سريعة وتشغيل متزامن للعديد من المستشعرات.

بشكل عام، يجمع هذا المشروع بين بنية تحتية قوية وتقنيات حديثة لمعالجة البيانات في الوقت الفعلي، مما يساهم في إنشاء نظام مراقبة ذكي وموثوق للمركبات يهدف إلى تحسين سلامة السائق ودعم التدابير الوقائية ضد الحوادث. يمثل هذا النظام حلاً فعالاً وقابلاً للتطوير يمكن تكييفه مع التطورات المستقبلية في مجالات السلامة والأنظمة الذكية للمركبات.

Abstract

This project presents a comprehensive, IoT-integrated vehicle monitoring and alert system designed to enhance road safety and driver assistance through real-time data processing and non-blocking, responsive mechanisms. Utilizing a combination of sensors, the system continuously monitors potential obstacles, crash events and vehicle location. An ultrasonic sensor dynamically tracks distances to nearby objects, alerting the driver in cases of close proximity to enhance collision prevention. Additionally, a crash sensor promptly detects accidents, triggering immediate handling routines while simultaneously allowing ongoing monitoring of other system functions.

The system leverages Firebase for cloud-based data storage, facilitating seamless data retrieval and communication across devices. Key features include RFID-based vehicle location identification, enabling precise and automated selection of a vehicle's current location, remote crash alerts and real-time vehicle diagnostics accessible via Bluetooth commands. An LCD interface enhances user interaction, displaying status updates and notifications, while the system architecture incorporates non-blocking delay functions to ensure high responsiveness and concurrent operation of multiple sensors.

Overall, this project combines robust IoT infrastructure with real-time data handling to create a responsive and reliable vehicle monitoring system that aims to improve driver safety and support preventive measures against road incidents. This smart vehicle system not only exemplifies an efficient use of IoT in transportation but also offers a scalable solution adaptable for future advancements in automotive safety and automation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

This project aims to develop a V2V (Vehicle-to-Vehicle) and V2I (Vehicle-to-Infrastructure) communication system to enhance road safety, traffic efficiency and management. By enabling real-time data exchange, the system helps prevent accidents, optimize traffic flow and provide timely updates to drivers.

1.2 Motivations

The project leverages V2V and V2I communication technologies to improve transportation safety and efficiency. Real-time data exchange allows for proactive accident prevention, reduced congestion and better traffic management. This approach can significantly enhance road navigation, leading to a smarter and more sustainable transportation system.

1.3 Problem Statement

Current traffic systems face challenges such as frequent accidents, congestion and inefficient emergency response. These issues arise due to the lack of proactive collision avoidance, poor traffic management and limited real-time updates. A robust V2V and V2I communication system can address these problems by enhancing road safety, optimizing traffic flow and improving driver awareness. Effective implementation of this technology is crucial for reducing accidents and ensuring a more efficient transportation network.

1.4 Objectives

- Enhance road safety through real-time exchange of information between vehicles and infrastructure.
- Improve traffic flow and congestion management by optimizing communication and coordination among vehicles.
- Reduce accidents, injuries and fatalities by enabling proactive hazard detection and warning systems.
- Facilitate a rapid and efficient emergency response through direct communication between vehicles involved in accidents and emergency services stations.

1.5 Requirements

1.5.1 Functional Requirements

- **Vehicle Detection and Collision Avoidance System:** Implement a system to detect nearby vehicles and provide alerts to drivers using distance measuring sensors .
- **Communication System:** The system must enable seamless communication between vehicles and infrastructure components, such as traffic lights, using a WiFi-based communication model and cloud. This connectivity facilitates real-time data exchange to improve traffic flow, enhance road safety and support cooperative vehicle-infrastructure interactions.
- **Emergency Vehicle Detection:** Implement a basic mechanism to detect emergency vehicles and prioritize their passage through traffic.
- **Data Processing and Storage (Cloud):** Implement basic data processing and storage capabilities to handle vehicle data and communication logs.

1.5.2 Non-Functional Requirements

- **System Reliability:** Ensure basic reliability of the system to operate effectively under normal conditions.
- **Stable communication system:** Ensure that the system supports real-time data transfer and operates without delay in critical cases.
- **System scalability:** The performance of a system must not be affected by increasing the number of cars.
- **System security:** The system such as the cloud must be protected against unauthorized use.
- **Real-Time Responsiveness:** The communication system must provide real-time data exchange with minimal latency to ensure timely interactions between vehicles and infrastructure. Additionally, the system should achieve low response times to facilitate smooth traffic management and immediate safety responses.

1.6 Limitations

- **Environmental Factors:** Sensor performance, including ultrasonic and infrared sensors, may be impacted by adverse weather conditions like rain, fog, or extreme temperatures, potentially affecting obstacle detection accuracy.
- **Technology Dependency:** The effectiveness of the V2V and V2I communication systems may be limited by the availability and reliability of underlying technologies such as sensors and cloud-based infrastructure.
- **Components cost:** Budget constraints restrict our choice of components, leading to more affordable but potentially less efficient options, which may impact system performance and reliability.

1.7 System Description

The proposed V2V and V2I communication system consists of several key components

- **Vehicle Sensors:** Sensors are utilized for real-time detection of surrounding vehicles, obstacles, accident detection and road conditions.
- **Microcontrollers:** Microcontrollers manage data aggregation, communication protocols and control algorithms within vehicles.
- **Cloud-based Communication:** Cloud-based platforms enable secure and scalable data exchange between vehicles, infrastructure elements and centralized servers, facilitating real-time updates and coordination.
- **Infrastructure Components:** Infrastructure elements such as traffic lights equipped with communication modules to transmit critical information to vehicles.

1.8 Project Time Line

Task No.	Task Name	Duration (Weeks)
1.	Project planning	4
2.	Determination project requirements	4
3.	Project design and analyzing	8
4.	Project development	10
5.	Project testing and maintenance	4
6.	Documentation	28

Table 1.1: Project Tasks and Durations.

1.9 Report Outline

This report is organized as follows: Chapter 2 introduces some literature review including available technologies and related projects. It also goes briefly over the theoretical background of the project, hardware and software components. Chapter 3 discusses the conceptual design of the system, block diagrams, flowcharts and detailed hardware connections. Chapter 4 discusses the hardware and software implementations of the project. Chapter 5 shows the testing process which is made until we reach the final system design. Finally, Chapter 6 presents the final results of the project, future works and conclusion.

Chapter 2

Background and Literature Review

2.1 Overview

This chapter provides a general overview of the theoretical foundation of the project, including a literature review.

2.2 Theoretical Background

In this section, we will provide an overview and discuss the technologies and techniques that will be used in this project.

RFID (Radio-frequency identification)

Radio Frequency Identification (RFID) is a wireless technology that uses electromagnetic fields to identify and track tags on objects. An RFID system includes tags (which can be passive, active, or semi-passive), readers and antennas. Readers emit radio signals to activate tags, which respond by transmitting data back for applications such as inventory tracking or access control.

RFID operates across various frequency bands, each suited to specific applications. For example, Low Frequency (LF) bands are used for animal tracking and access control, High Frequency (HF) bands (like 13.56 MHz, used in this project) are common for smart cards and secure data transfer, Ultra-High Frequency (UHF) bands support inventory management and logistics and Microwave bands are utilized for industrial applications such as toll collection.

The versatility of RFID supports applications in many fields. It enhances inventory accuracy and theft prevention in supply chains, enables automated checkout in retail, supports patient and equipment tracking in healthcare and improves asset management across sectors, enhancing operational efficiency and security [1].

Firebase Realtime Database

Firebase Realtime Database is a cloud-hosted NoSQL database that allows data to be stored and synchronized across clients in real time. It employs a JSON-based data structure, making it efficient for applications requiring live updates, such as chat apps or collaborative tools. Changes to the database are automatically propagated to connected clients, ensuring data consistency across platforms. Its serverless architecture reduces the need for backend management, enabling developers to focus on application logic. This makes Firebase Realtime Database a popular choice for real-time and scalable applications [2].

Internet of Things (IoT)

The Internet of Things (IoT) is a technological paradigm that enables physical devices to connect, communicate and exchange data over the internet. These devices, embedded with sensors, actuators and communication modules, can collect and transmit real-time information, facilitating automation and intelligent decision-making. IoT has revolutionized multiple industries, including healthcare, agriculture, smart homes and transportation.

In the field of transportation and intelligent mobility, IoT plays a vital role in enhancing road safety, optimizing traffic flow and improving the overall driving experience. By enabling connectivity between vehicles (V2V) and between vehicles and infrastructure (V2I), IoT allows for the seamless exchange of data related to traffic conditions, road hazards, weather updates and emergency alerts. [3].

2.3 Literature Review

In this section, we'll highlight previously proposed products and projects similar to this one, offering valuable points of view and ideas for our project.

2.3.1 V2V and V2I Communication in a Heterogeneous Wireless Network

- **Description**

  This project focuses on evaluating the performance of integrated Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication in heterogeneous wireless networks. The goal is to improve traffic management, enhance safety, and reduce delays by enabling efficient communication between vehicles and infrastructure. Using advanced wireless technologies, the system facilitates real-time information exchange [4].

- **Methods and Technologies Used in the Project:**

  - **DSRC (Dedicated Short-Range Communications):** A core technology for short-range communication between vehicles.
  - **Wi-Fi:** To extend the communication range and increase flexibility.
  - **LTE:** To connect vehicles to the infrastructure using cellular networks.

2.3.2 Vehicle To Vehicle Communication

- **Description**

  The main objective of this system is to enable vehicles to exchange real-time data for collision avoidance using Dedicated Short Range Communication (DSRC). DSRC allows vehicles to communicate with each other over short distances, providing critical information such as speed, location and braking status. Additionally, infrared communication is used to detect potential collisions, especially during emergency braking scenarios, helping to prevent accidents by providing accurate distance measurements. The system also utilizes a low-power wireless transceiver, which ensures 360-degree information sharing between vehicles, allowing them to exchange data from all directions while maintaining energy efficiency [5].

- **Methods and Technologies Used in the Project:**
  - **Infrared Communication:** Used for detecting potential collisions and measuring accurate distances during emergency braking scenarios.
  - **Real-Time Data Exchange Algorithms:** Algorithms designed to process and transmit critical data such as vehicle speed, position and braking status instantaneously.

2.3.3 The Future of Driving: How V2V and V2I Communications Are Paving the Way

- **Description**

  This project focuses on developing an Intelligent Traffic System (ITS) by integrating Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication technologies. The objective is to enhance road safety, optimize traffic flow and improve urban mobility by leveraging real-time data exchange between vehicles and infrastructure [6].

- **Methods and Technologies Used in the Project:**
  - **Smart Infrastructure Integration:** Incorporates intelligent traffic signals and road signs capable of communicating with vehicles to provide real-time updates on traffic conditions and hazards.
  - **Data Analytics:** Collects and analyzes traffic data to optimize traffic flow, reduce congestion and enhance overall urban mobility.
  - **Smart Infrastructure Integration:** Incorporates intelligent traffic signals and road signs capable of communicating with vehicles to provide real-time updates on traffic conditions and hazards.
  - **Data Analytics:** Collects and analyzes traffic data to optimize traffic flow, reduce congestion and enhance overall urban mobility.

The table below highlights the key features and limitations of the previously implemented projects.

Project Name	Features	Limitations
V2V and V2I Communication in a Heterogeneous Wireless Network	• The combination of DSRC, Wi-Fi, and LTE enhances communication efficiency. • The project supports applications such as early collision warnings and hazardous zones.	• DSRC technology has limited bandwidth, which restricts its use in dense traffic scenarios with many vehicles. • Interference between different technologies affects communication quality and reliability.
Vehicle To Vehicle Communication	• 360-degree situational awareness. • The system uses infrared communication to detect potential collisions.	• Affected by environmental factors (e.g., fog, rain). • Range and bandwidth limitations of the low-power transceiver.
The Future of Driving: How V2V and V2I Communications Are Paving the Way	• Real-time data exchange. • Collision prevention.	• Security and privacy concerns. • Challenges in technological adaptation for older vehicles.

Table 2.1: List of Features and Limitations of Previous Projects.

Our project differs from the previously mentioned projects by utilizing a cloud-based communication model as a mediator to enable seamless V2V and V2I interactions. While the earlier projects focus on specific communication technologies like DSRC and LTE, our system focus with cloud-based data processing to enhance scalability, reliability, and real-time responsiveness. Unlike projects limited to local communication, our system facilitates centralized data storage and processing, enabling advanced features like emergency vehicle prioritization, optimized traffic management and accidents alert. Additionally, our approach emphasizes cost-effective components and addresses critical non-functional requirements such as system reliability, scalability and security, setting it apart as a comprehensive and practical solution for modern traffic safety and efficiency challenges.

Chapter 3

Design

3.1 Overview

This chapter discusses the overall design of the system and the way its components are integrated together, showing the block diagram and schematic diagram for the design, in addition to some details about the scenarios we are going to implement. Additionally, it offers a brief overview of the hardware and software components used within the system.

3.2 Components

3.2.1 Hardware

Vehicle Components

represent the basic part for the project and each car has a set of components include:

- **DC Motors:** Enable movement.

Figure 3.1: DC Motor [7].

- **Motor Driver:** A motor driver is an electronic component or module that controls the speed, direction and operation of an electric motor. It acts as an interface between a microcontroller or control system and the DC motor, translating control signals into precise motor movements. Motor drivers are essential in robotics, automation and vehicle applications where precise motor control is required. This driver has many pins, power supply pins (12V, 5V), Ground pins, Output pins (OUT1,OUT2,OUT3,OUT4) and control pins, input pins (IN1, IN2,IN3,IN4) and enable pins (ENA, ENB).

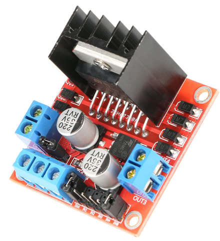


Figure 3.2: Motor Driver [8].

- **Power Bank:** Provide electrical power with a 10,000mAh capacity to the vehicle's systems, including the motors and onboard electronics.

Figure 3.3: Power Bank [9].

- **ESP32 Microcontroller:** The ESP32 is a dual-core microcontroller chip developed by Espressif Systems, equipped with Wi-Fi and Bluetooth connectivity, making it suitable for IoT (Internet of Things) and wireless communication applications. It features a rich set of peripheral interfaces, including GPIOs (General Purpose Input/Output), ADC (Analog-to-Digital Converter), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver-Transmitter) and more.

  ESP32 is a versatile microcontroller that plays a crucial role in V2V (Vehicle-to-Vehicle) and V2I (Vehicle-to-Infrastructure) communication systems due to its capabilities in wireless communication, data processing and interfacing with various sensors and peripherals.

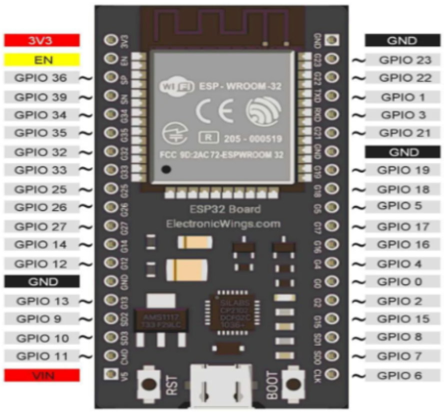


Figure 3.4: ESP32 Microcontroller [10].

Development Board	**Number of cores**	**Connectivity**	**Power consumption**	**Price**	**Number of GPIO**	**RAM**
ESP32	2	Wifi, Bluetooth	low	20$	38	512KB
ESP8266	1	Wifi	low	2-5$	17	160KB
Arduino uno R3	1	NO	low	4-10$	14	2KB

Table 3.1: Comparison Between ESP32 and Other Microcontrollers.

We choose this microcontroller instead of other microcontrollers because this development board is mainly designed for cost-effective, power-efficient and simple IoT-based applications because of integrated Wi-Fi and Bluetooth.

- **Crash sensor:** The crash sensor detects the collision and converts it to usable signals within milliseconds. The speeding-up forces acting on the sensors after a collision are high. For example, When a car is stopped abruptly by an impact, all bodies or objects that are not firmly fixed to the car will continue to move at the impact speed.
  This sensor has three pins : VCC pin for power supply and GND pin and signal pin that provides the analog or digital output signal indicating the detected impact intensity or collision.

Figure 3.5: Crash Sensor [11].

- **Ultrasonic sensor HC-SR04:** The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. This sensor reads from 2 cm to 400 cm (0.8 inch to 157 inch) with an accuracy of 0.3 cm (0.1inches), which is good for most hobbyist projects. In addition, this particular module comes with ultrasonic transmitter and receiver modules.
  Ultrasonic sensors operate by emitting high-frequency sound waves, which are inaudible to humans. These waves travel through the air until they encounter an object, at which point they bounce back to the sensor.
  The sensor then calculates the time it takes for the waves to return, using this information to determine the distance to the object. This process is based on the principle of echolocation, similar to how bats navigate. Ultrasonic sensors are commonly used in various applications due to their accuracy, reliability and ability to work in various environmental conditions.

Figure 3.6: Ultrasonic Sensor [12].

- **RFID Reader:** The RFID Reader RC522 is a widely-used module for reading and writing data to RFID tags, operating at 13.56 MHz. It is based on the MFRC522 chip from NXP and supports communication interfaces like SPI, I2C, and UART, making it compatible with microcontrollers such as Arduino and ESP32. Operating at 3.3V, the RC522 often includes level shifters to work with 5V systems and features a built-in antenna for straightforward integration. The RC522 is cost-effective, making it ideal for hobbyist and commercial use. Its compact design allows for easy incorporation into various projects, providing a reading distance of 2-5 cm and a data transfer rate of up to 10 Mbit/s. Its versatility allows it to work with a wide range of RFID tags and MIFARE cards.

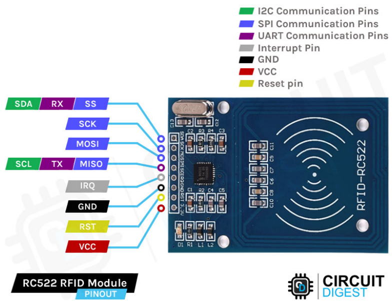


Figure 3.7: RFID Reader [13].

- **LCD 16×2 :** An LCD screen is an electronic display module that uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIYs and circuits. The 16×2 translates a display of 16 characters per line in 2 such lines. In this LCD, each character is displayed in a 5×7 pixel matrix.

Figure 3.8: 16×2 LCD Display [14].

- **Buzzer:** An electronic component that generates sound through the transmission of electrical signals. Its primary function is to provide an audible alert or notification and typically operates within a voltage range of 5V to 12V.

Figure 3.9: Buzzer [15].

- **Push Button:** The push button is a simple input device that triggers an action or event when pressed. In this project, it serves as a manual control mechanism, allowing users to initiate specific functions or emergency overrides within the traffic control system. This inclusion provides a straightforward way to interact with the system, ensuring quick and reliable human intervention when necessary.

Figure 3.10: Push Button [16].

Infastructure Components

- **Traffic Light:** Represents the status of traffic control at intersections. Receives signals from emergency vehicles to adjust light timings. Facilitates smooth traffic flow and prioritizes emergency vehicle passage by changing light signals accordingly.

Figure 3.11: Traffic Light [17].

Component	Component Used	Alternative 1	Alternative 2	Reasoning
Microcontroller	ESP32	Arduino Mega 2560	Raspberry Pi Pico	Built-in Wi-Fi/Bluetooth, powerful dual-core processor and cost-effective compared to others.
Motor Driver	L298N	TB6612FNG	DRV8833	Reliable and widely available, with enough current handling for basic motor control tasks.
Buzzer	Piezo Buzzer	Active Buzzer Module	Passive Buzzer Module	Simple to integrate, easy to control directly from the microcontroller and cost-efficient.
LCD Display	LCD 16x2	OLED-Display 128x64	LCD 20x4	Readily available, easy to use with most microcontrollers and provides adequate space for basic data.
Ultrasonic Sensor	HC-SR04	JSN-SR04T	Parallax Ping	Economical, reliable for short distances and easy to program with Arduino/ESP32.
Crash Sensor	Limit Switch	Vibration Sensor Module	Force Sensitive Resistor	Simple mechanical structure, highly reliable for detecting physical contact and cost-effective.
RFID Reader	RC522	RV522	PN532	Robust, supports multiple card types and has reliable range and performance.
Traffic Lights	LED Arrays	Smart RGB LEDs	Individual LEDs	Easy to control, bright and can be customized for standard traffic signals with minimal wiring.
Power Supply	Power Bank	Li-ion Battery Pack	Portable Generator	Portable, reliable for on-the-go projects and easy to recharge, making it versatile for outdoor use.

Table 3.2: Comparison of Hardware Components with Alternatives and Reasoning.

3.2.2 Software

- **Arduino IDE**

  The Arduino Integrated Development Environment (IDE) is a software tool designed for writing, compiling, and uploading code to Arduino microcontroller boards. It simplifies code development by providing a user-friendly interface with built-in libraries and functions. The IDE allows developers to write code in the Arduino programming language (based on C and C++), compile it into machine-readable instructions and upload it to Arduino boards via USB or other interfaces. Key features include code editing, syntax checking, hardware configuration, serial communication monitoring, and library management. Overall, the Arduino IDE streamlines the development process for Arduino-based projects, making it accessible to beginners and experienced programmers alike.

- **C Language**

  The C programming language is essential for implementing the core functionalities of V2V systems, including microcontroller programming, algorithm development, hardware interaction, resource optimization and platform compatibility. Its versatility and efficiency make it a preferred choice for embedded software development in automotive and IoT applications.

- **Firebase for Cloud Integration**

  Firebase, developed by Google, is a cloud platform offering real-time databases, cloud functions and seamless integration with various platforms. It is designed for real-time data synchronization and scalable application support.

  In our project, the cloud will serve several key functions : data processing, data collection, acting as an intermediary for V2V and V2I communication and monitoring system performance.

  We chose Firebase because it provides real-time updates, ensuring instant data synchronization for accident alerts and traffic light changes. Its seamless integration with IoT devices like ESP8266 and ESP32 simplifies the development process. Moreover, Firebase is highly scalable, allowing us to handle increased data loads efficiently. It also offers a cost-effective solution, with a free tier for development.

- **React Library**

  A React library for building an interactive dashboard that takes real-time data from Firebase Realtime Database. The goal of this dashboard is to display data in an easy and user-friendly way.

  One of the main reasons to use React over other libraries is the fast interaction and high flexibility it provides, thanks to the Virtual DOM model. This model allows React to update small parts of the user interface without having to reload the entire page, which improves application performance and increases the speed of interaction with the user.

3.3 Detailed Design

V2V System (Vehicles Communicate with Each Other Through Cloud)

The vehicle communication system relies on the ESP32 microcontroller as the central unit for connecting with the cloud. The communication between vehicles occurs through the cloud in two directions:

- **Data Upload (Vehicle to Cloud):** The vehicle collects data from sensors such as crash sensors and RFID and uploads this data to the cloud to share critical information like accidents or road conditions.

- **Data Download (Cloud to Vehicle):** The vehicle receives updates from the cloud, including emergency alerts such as accidents or obstacles on the road, helping the vehicle make real-time decisions to improve safety and reduce risks.

V2I System (Vehicles Communicate with Traffic Infrastructure Through Cloud)

In the V2I system, the traffic infrastructure is equipped with traffic lights controlled by an ESP32 microcontroller. The microcontroller communicates with the cloud to receive real-time data, process it and adjust the traffic signals accordingly.

The traffic lights are managed by control circuits that change the signals (red, yellow, green) based on instructions received from the cloud. The timing and duration of these signals are adjusted according to traffic conditions, data from vehicles and emergency vehicle prioritization.

The cloud platform serves as the central system for processing and managing data from various sources, including vehicles and emergency services. It uses this data to make decisions that optimize traffic management, such as modifying traffic light signals to give priority to emergency vehicles.

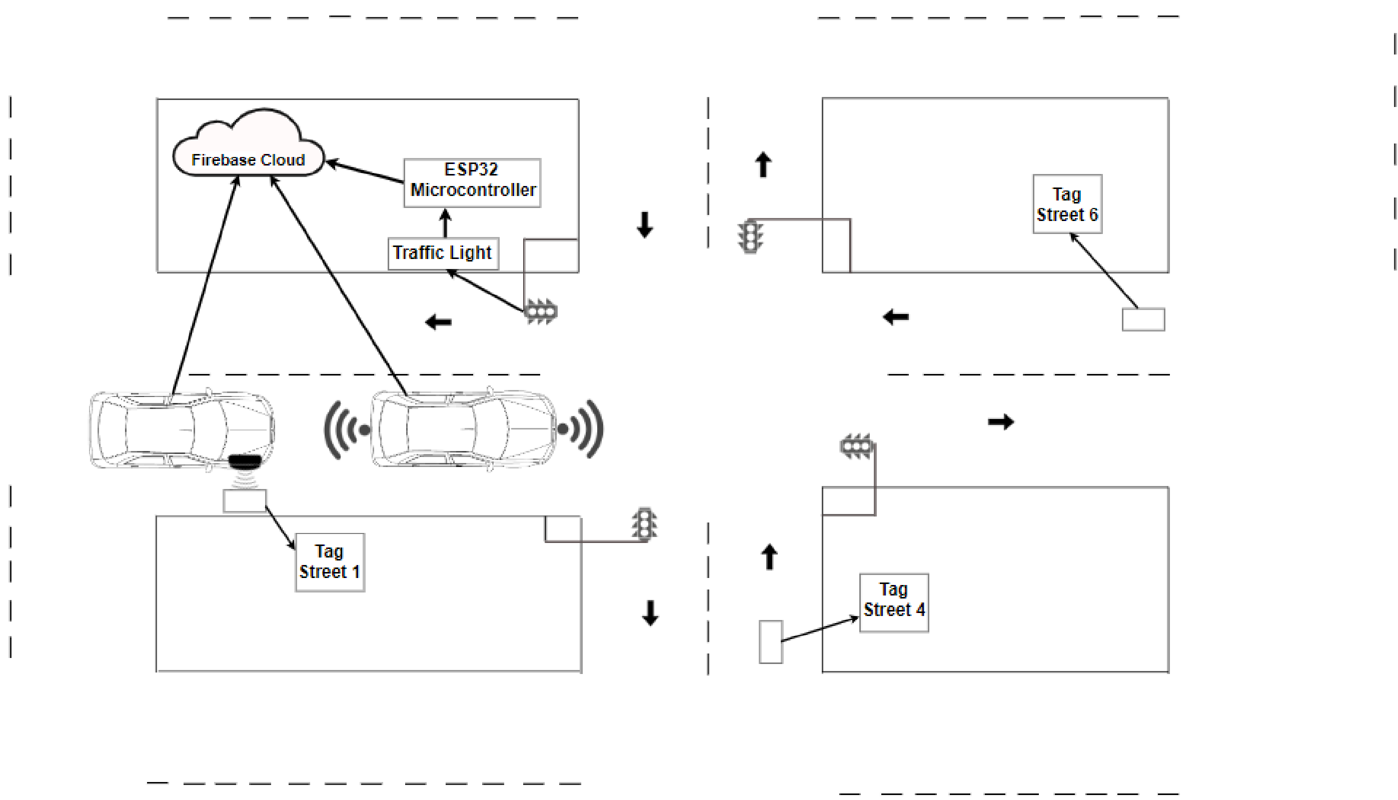
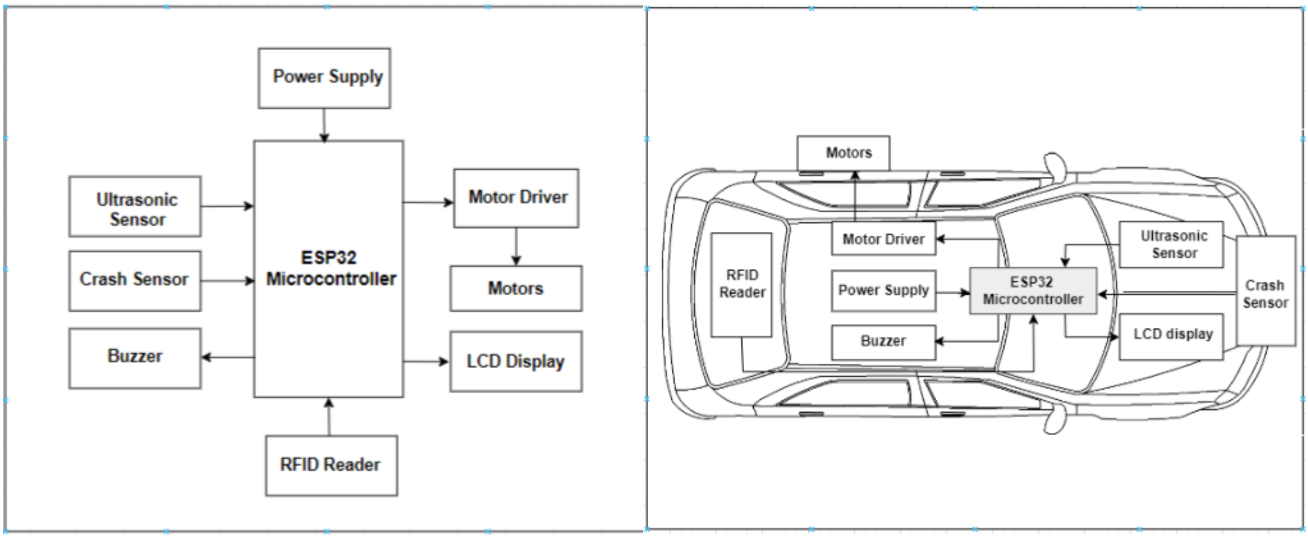


Figure 3.12: Conceptual Diagram of V2V and V2I Communication System.

3.4 Block Diagrams

The vehicle system includes several key components. The ESP32 Microcontroller serves as the central processing unit. An Ultrasonic Sensor at the front detects and measures the distance to nearby objects using high-frequency sound waves, crucial for collision avoidance. A Crash Sensor detects collisions by measuring sudden deceleration. The Buzzer, near the central control unit, provides audible alerts to the driver. The LCD displays the messages and notifications to the driver .The RFID Reader reads the RFID cards or tags that are placed on the streets and sends them to ESP32 to update location . The Motor Driver, acting as a link between the ESP32 and the vehicle's DC Motors, converts control signals into precise motor movements, regulating speed and direction.



(a) Vehicle Subsystem block diagram.
(b) Vehicle Subsystem design.

Figure 3.13: Vehicle Subsystem.

The infrastructure system, consisting of ESP32 microcontrollers and traffic lights integrated with a cloud platform, creates a smart traffic management network. By enabling real-time adjustments of traffic signals based on the presence of emergency vehicles, this system significantly enhances response times and overall road safety. The ESP32 microcontrollers execute cloud commands, ensuring efficient and safe traffic flow, particularly during critical emergency situations.

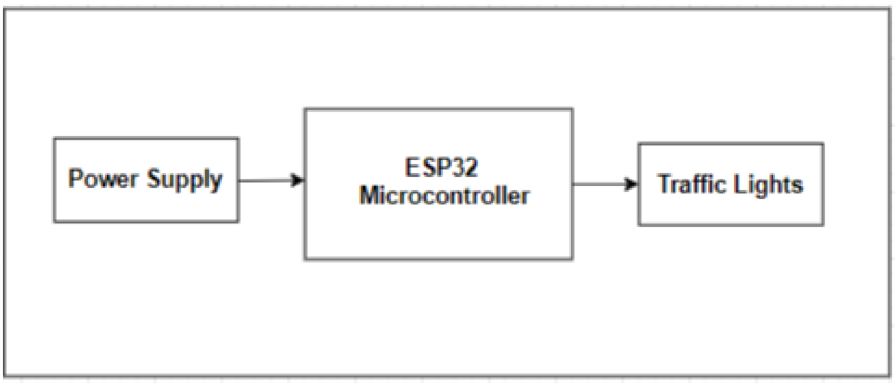


Figure 3.14: Infrastructure Subsystem Block Diagram.

Cloud systems form the backbone between V2V and V2I communications, and provide seamless and efficient communication between vehicles and services. By collecting, processing, and analyzing

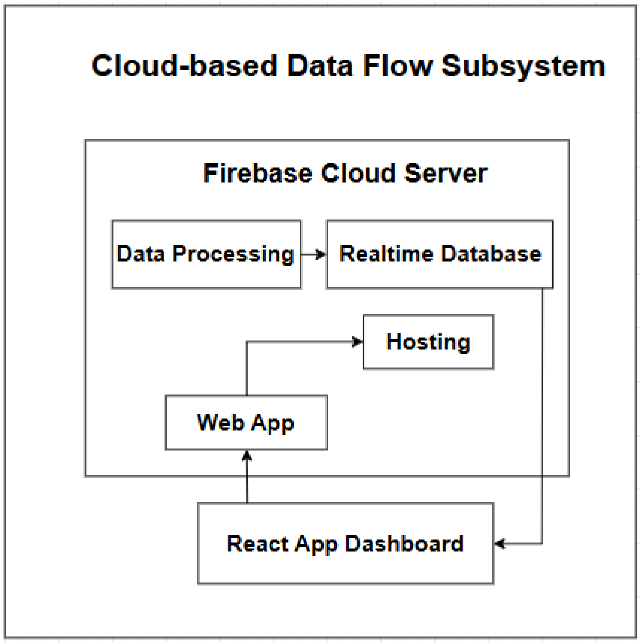


Figure 3.15: Cloud-based Data Flow Block Diagram.

real-time data, the cloud makes appropriate decisions to improve traffic management, enhance road safety, and fly emergency response times effectively.

3.5 Sequence Diagrams

This section illustrates and explains each scenario in the system by designing a sequence diagram for each scenario.

Emergency in V2I (Emergency Vehicle to Infrastructure Communication)

In this scenario, the sequence diagram shows the interaction between an emergency vehicle, a cloud server and a traffic light controller to ensure the vehicle moves quickly through traffic. When the emergency vehicle approaches an intersection, a button in the vehicle sends a signal with its current location. The cloud server processes this information to determine the vehicle's route and communicates with the traffic light controller. The controller then changes the signal for the target street to green and keeps other signals red, allowing the emergency vehicle to pass through the intersection without stopping. This process reduces response time and enhances the efficiency of emergency services.

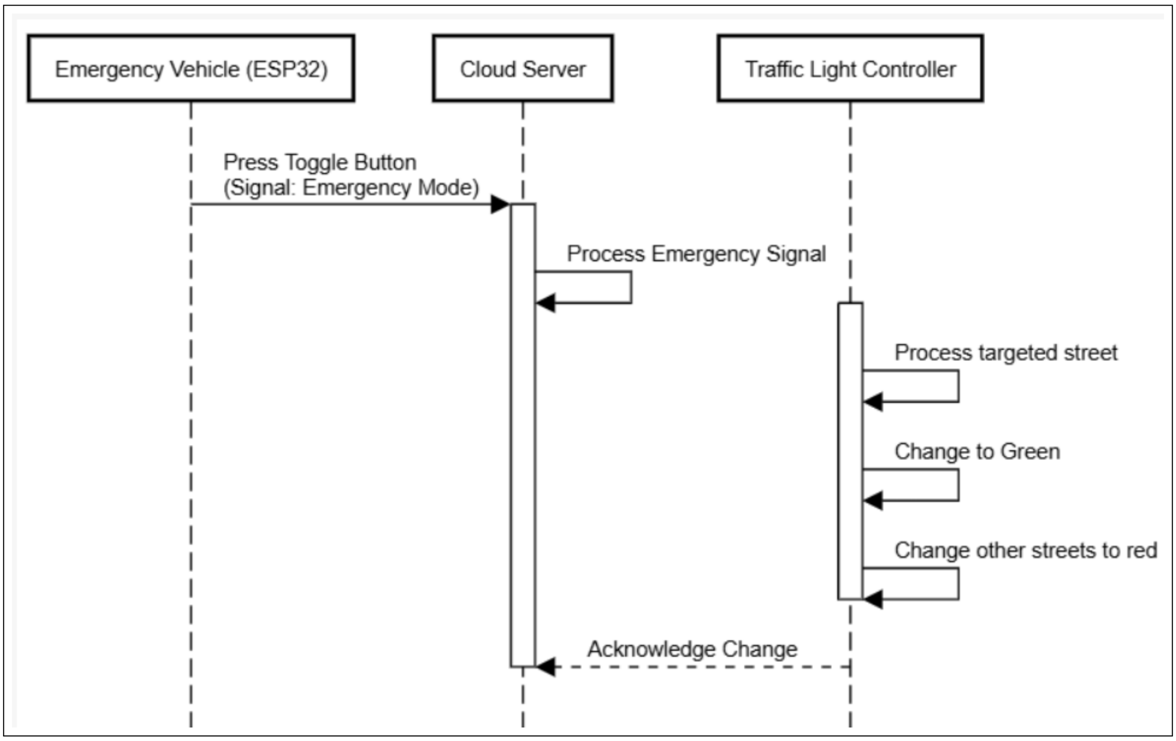


Figure 3.16: Emergency in V2I Communication Sequence Diagram.

Accident Detection in V2V (Vehicle-to-Vehicle Communication)

In this scenario, the sequence diagram illustrates how cars and a cloud system respond to a road accident. When an accident occurs, the car with the activated crash sensor sends the accident location and details to the cloud server. The cloud processes this information to assess the situation and minimize traffic disruption.

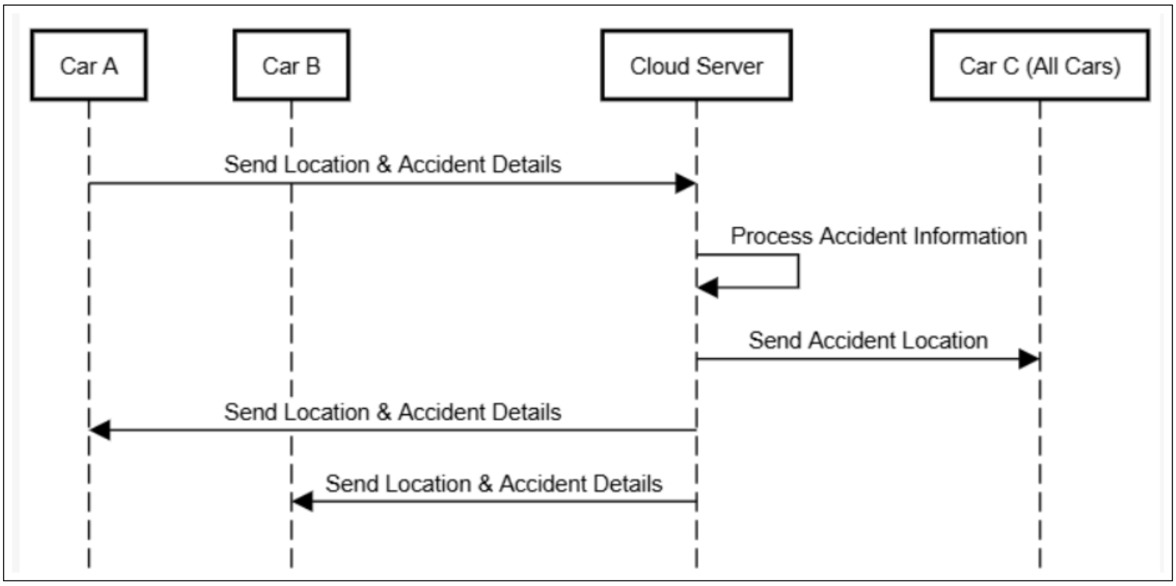


Figure 3.17: Accident Detection in V2V Sequence Diagram.

The cloud server then sends updates to nearby vehicles, such as Car C, informing them of the accident's location. Car C can use this information to take an alternate route, avoiding the accident site. This process helps reduce traffic congestion, ensures safety and maintains smoother traffic flow around the accident scene.

Accident Avoidance and Alert Activation in Vehicles

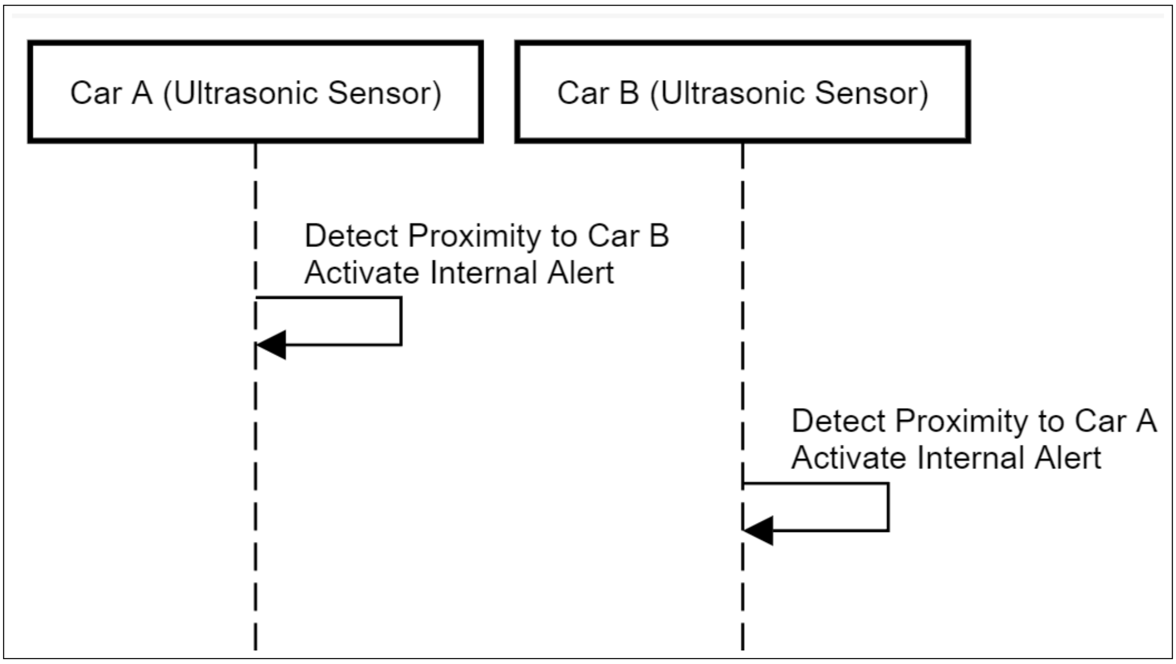


Figure 3.18: Accident Avoidance Sequence Diagram.

In this scenario, the sequence diagram illustrates how two cars, Car A and Car B, use ultrasonic sensors to prevent collisions. Car A's sensor detects when Car B is too close, triggering an alert system to inform the driver or adjust the vehicle to maintain a safe distance. Similarly, Car B's sensor activates its alert system if Car A gets too close. Both cars continuously monitor their surroundings, activating safety measures when necessary to ensure drivers are aware of proximity and can avoid potential accidents.

3.6 Flow Chart and Pseudocode

The following pseudocode illustrate the process of emergency vehicle to infrastructure communication.

Algorithm 1 Emergency Vehicle to Infrastructure Communication

```
Start
Emergency vehicle arrives at the intersection.
Driver pushes the button to trigger the cloud system.
if Emergency vehicle detected then
    Intersection microcontroller reads emergency location.
    Save the current traffic light state.
    Change the traffic light for the emergency street to green.
    Change all other traffic lights to red.
    while Emergency vehicle is in the intersection do
        Wait for the vehicle to exit.
    end while
    Restore the previous traffic light state.
else
    End
end if
End =0
```

The following flowchart illustrate the process of accident detection and alert system.

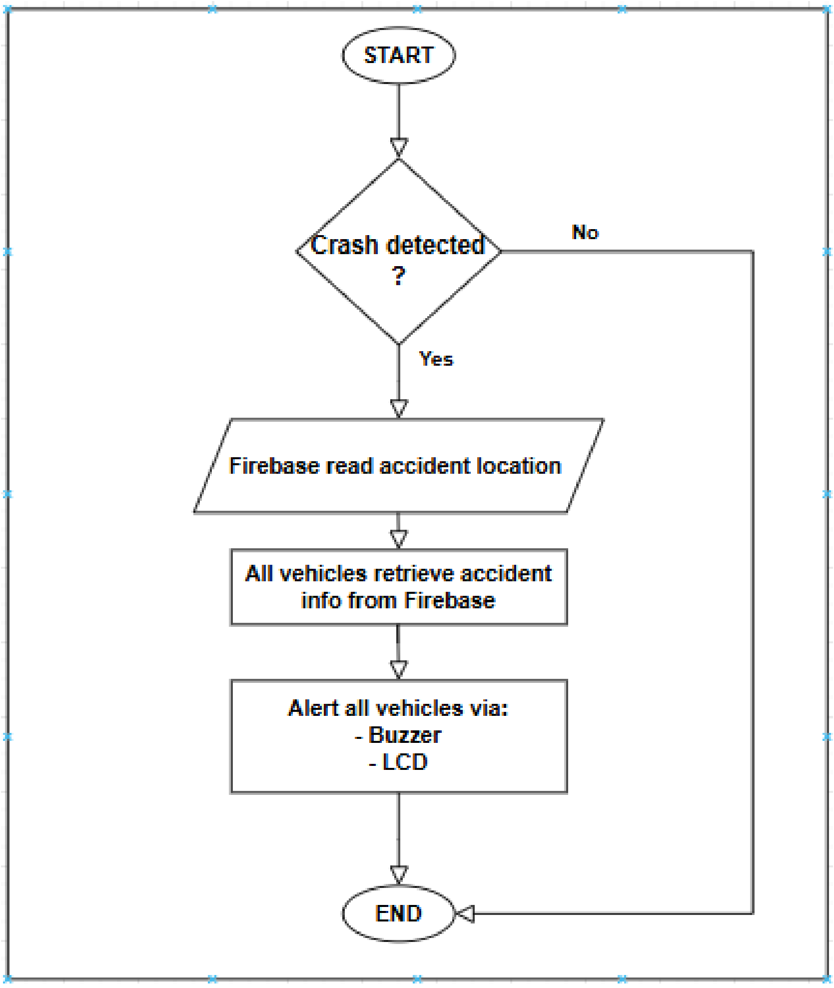


Figure 3.19: Flowchart Representing the Accident Detection.

3.7 Schematic Diagrams

In figure 3.20, the schematic diagram represents the components of the vehicle and their connection with the microcontroller.

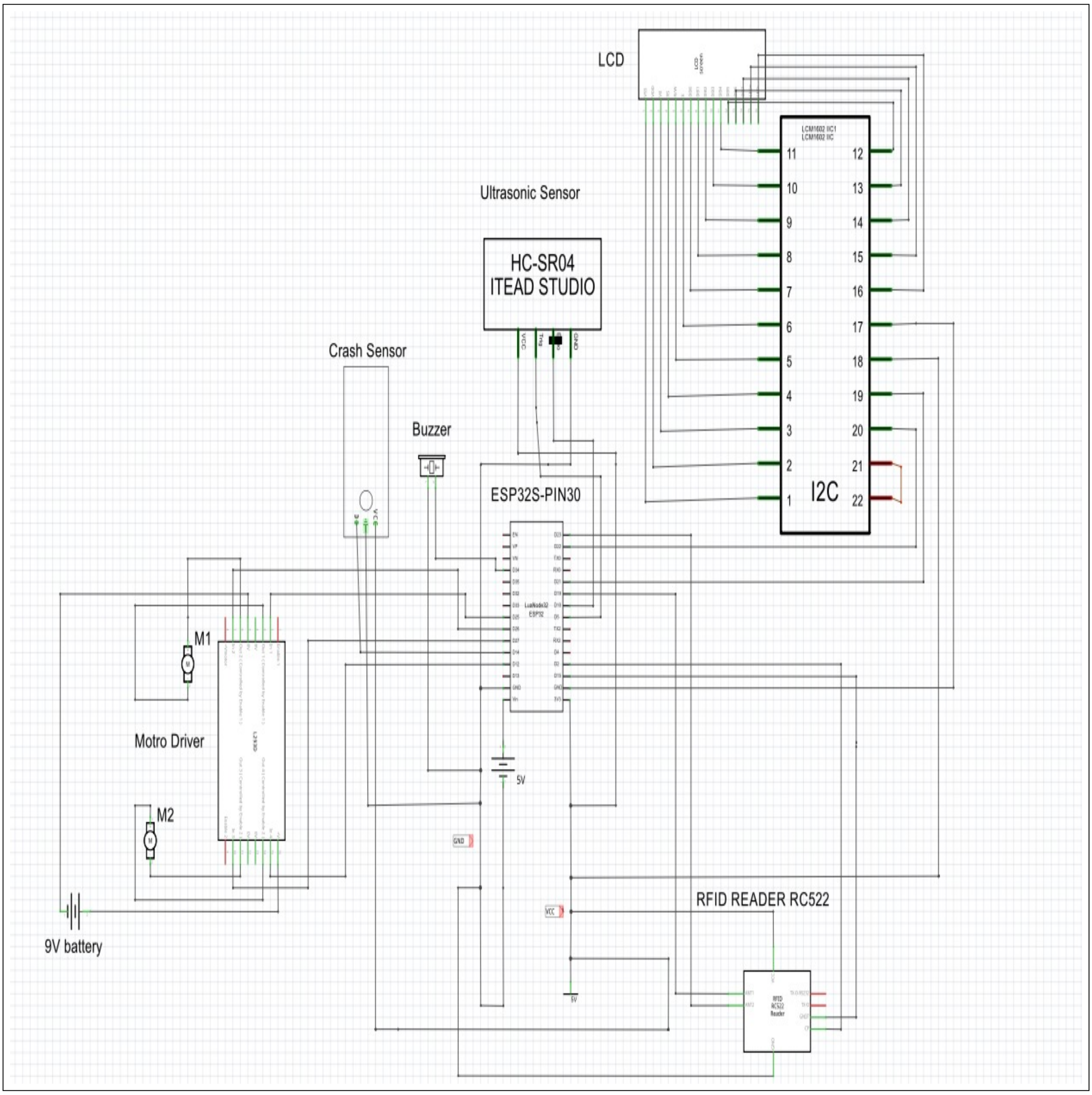


Figure 3.20: Vehicle Schematic Diagram.

In figure 3.21, the schematic diagram represents the components of the infrastructure and their connection with the microcontroller on the street.

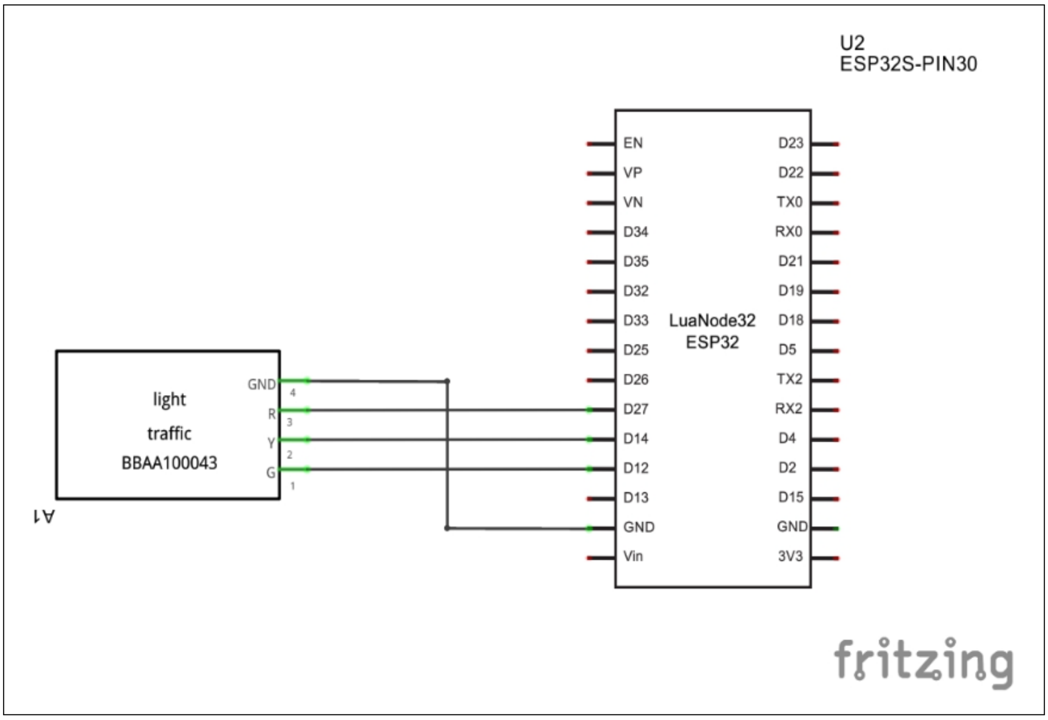


Figure 3.21: Infrastructure Schematic Diagram.

Chapter 4

Implementation and Testing

4.1 Overview

This chapter provides a detailed overview of the implementation phase of our project. It thoroughly explores the various hardware components of the system and delves into its software, covering all its modules comprehensively and discuss the testing of all components of the system.

4.2 Implementation Issues

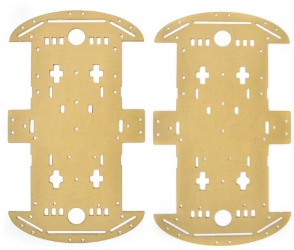
4.2.1 Hardware

In the previous chapter, we viewed the system's schematic diagram, illustrating the interconnection of its components. In this section, we will detail the process of assembling and organizing these components. We will discuss two key subsystems of the project: the vehicle and the Infrastructure, focusing on how they were constructed and assembled step by step.

Vehicle Implementation

Building the Body and Initial Setup The four motors shown in Figure 4.1 (a) were mounted onto the plastic base as illustrated in Figure 4.1 (b).

(a) Vehicle Motors and Wheels.



(b) Plastic Base.

Figure 4.1: Vehicle Parts.

We then proceeded to attach each wheel to its respective motor , After completing the assembly, the vehicle's structure appeared as shown in Figure 4.2.

Figure 4.2: Completed Vehicle Body Structure.

Positioning the Motor Driver

We mounted the motor driver centrally on the chassis to minimize wiring length and reduce potential interference. It was connected to the microcontroller via digital pins, which we used to control the motors' direction and speed. Proper insulation was applied to avoid accidental shorts.
To ensure synchronized movement, we connected the terminals of each pair of motors to the same motor driver output as shown in Figure 4.3. This configuration allows the left-side motors to rotate together in the same direction, and the right-side motors to do the same, ensuring smooth and coordinated motion for all four wheels.

Figure 4.3: Motor Driver Positioning on the Chassis.

Placing the Microcontroller

The microcontroller, which acts as the brain of the system, was placed near the center of the chassis for easy access to power sources and wiring. It was fixed using screws and standoffs to ensure stability as shown in Figure 4.4. The microcontroller's input and output pins were carefully wired to the motor driver, sensors and other peripherals to facilitate data flow and control.

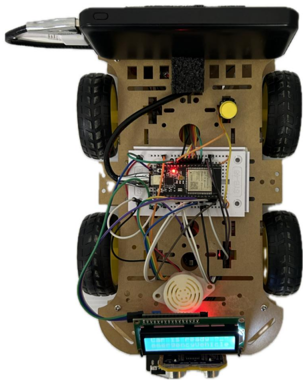


Figure 4.4: Microcontroller Placement on the Chassis.

Placing the Crash and the Ultrasonic Sensor

We mounted both the ultrasonic and crash sensors at the front of the vehicle, as shown in Figure 4.5. The ultrasonic sensor was positioned on a small angled mount to detect obstacles and measure distance, while the crash sensor was fixed to the front bumper to detect collisions.

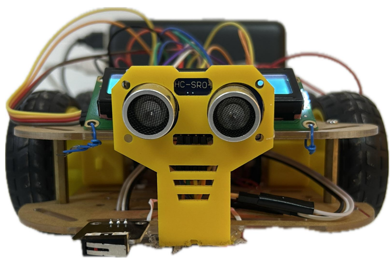


Figure 4.5: Placement of Ultrasonic and Crash Sensors at the Vehicle Front.

Positioning the Buzzer and LCD Display

A small buzzer was mounted near the microcontroller to alert users of potential obstacles or warnings as shown in Figure 4.6. The LCD display was positioned on the top panel, angled for clear visibility as shown in Figure. The display was connected to the microcontroller via I2C communication to reduce the number of wires needed and simplify the setup.

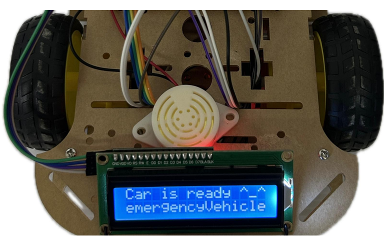


Figure 4.6: Buzzer and LCD Display Placement on the Vehicle.

Mounting the RFID Reader

The RFID reader was positioned at the bottom of the vehicle as shown in Figure 4.7, near the center, to facilitate card scanning as the vehicle passed over designated RFID cards.

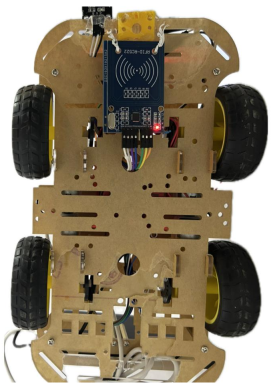


Figure 4.7: RFID Reader Mounted at the Front Underside of the Vehicle.

Infastructrue Implementation

One-Way Street Model

We designed a custom model for the project using a piece of leather, which we structured to resemble a network of one-way streets. The layout included a single intersection equipped with two traffic lights to manage the flow of vehicles. This model provided a practical representation of the system's functionality, as shown in Figure 4.8.

We placed an RFID card at the start of each street. Each card was programmed with a unique ID and street name to identify the location.

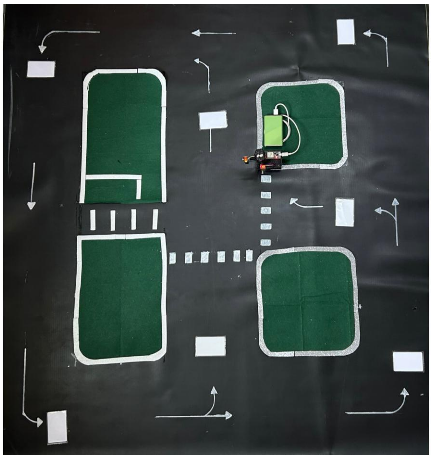


Figure 4.8: Custom One-Way Street Model with RFID Cards.

Traffic Light

We mounted the microcontroller onto a base and connected it to two traffic lights. Each traffic light was securely fixed onto a small pole attached to the base, ensuring stability and proper visibility, as illustrated in Figure 4.9.

Figure 4.9: Traffic Light Setup with Microcontroller Base.

4.2.2 Software

This section describes the implementation details of software components of the system. It also explains the features and functions of each software component in the system.

Tools

We utilized the Arduino IDE and Visual Studio Code as the primary software tools for developing and uploading the code to the microcontroller and implementing the dashboard. The Arduino IDE offers a straightforward and user-friendly environment for coding in C/C++ and managing hardware interfaces, while VS Code, which we use for dashboard development and building with React, provides a powerful and versatile coding platform.

Libraries Installation

We used various libraries to enhance the system's functionality. The MFRC522 library handled RFID card detection, while the Firebase ESP8266/ESP32 library enabled communication with Firebase for data storage. The Wi-Fi library provided internet connectivity, and the Wire and LiquidCrystal-I2C libraries supported I2C communication and LCD control. The BluetoothSerial library facilitated wireless command transmission. For the dashboard, we utilized React Hooks for state management, Toastify for alerts and React Router DOM for efficient navigation.

Firebase Configuration

For Firebase implementation, we used the project's API and token for seamless integration with the code. A Real-Time Database was utilized, designed as a non-relational structure to efficiently store and manage project data. The database includes the address and designated name of each street, along with detailed information for every vehicle, such as its current location, last recorded location and, in the event of an accident, the location of the vehicle responsible. Additionally, the database records the time and date of any accidents. Lastly, it tracks the status of the intersection, represented as a boolean value.

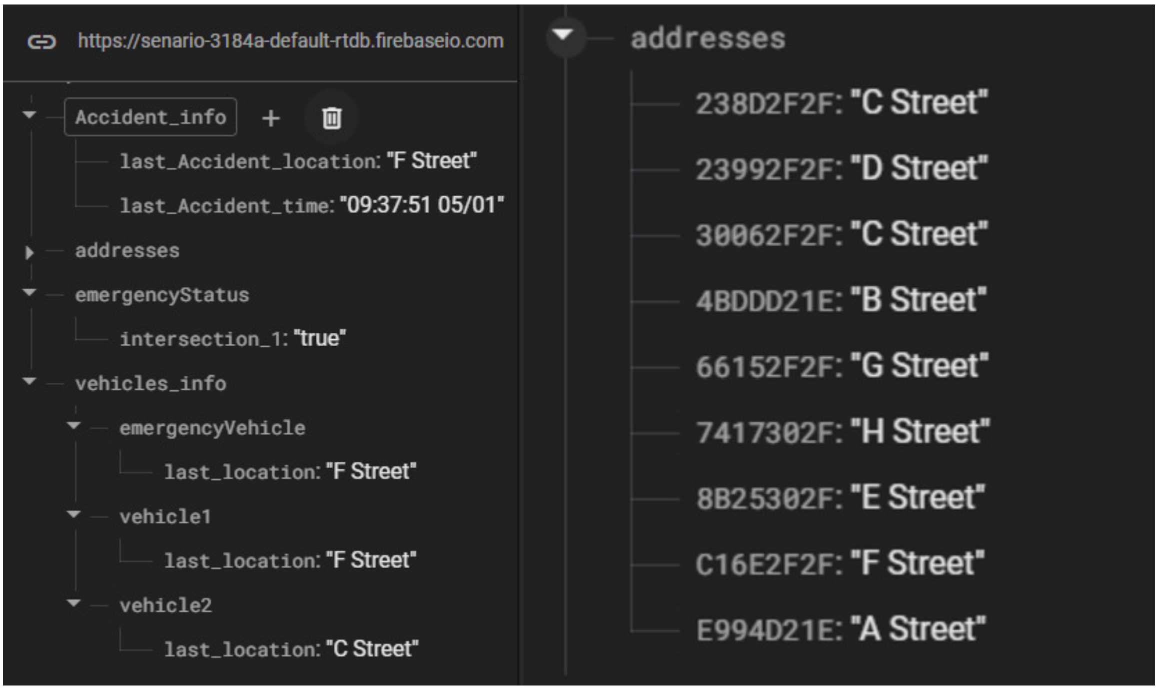


Figure 4.10: Real-time Database Configuration.

Dashboard Web Application

We designed a dashboard web application using React that fetches data from a Firebase Realtime Database and displays it in an organized manner. The application includes alerts for specific events, admin login functionality and Firebase Authentication to secure access.

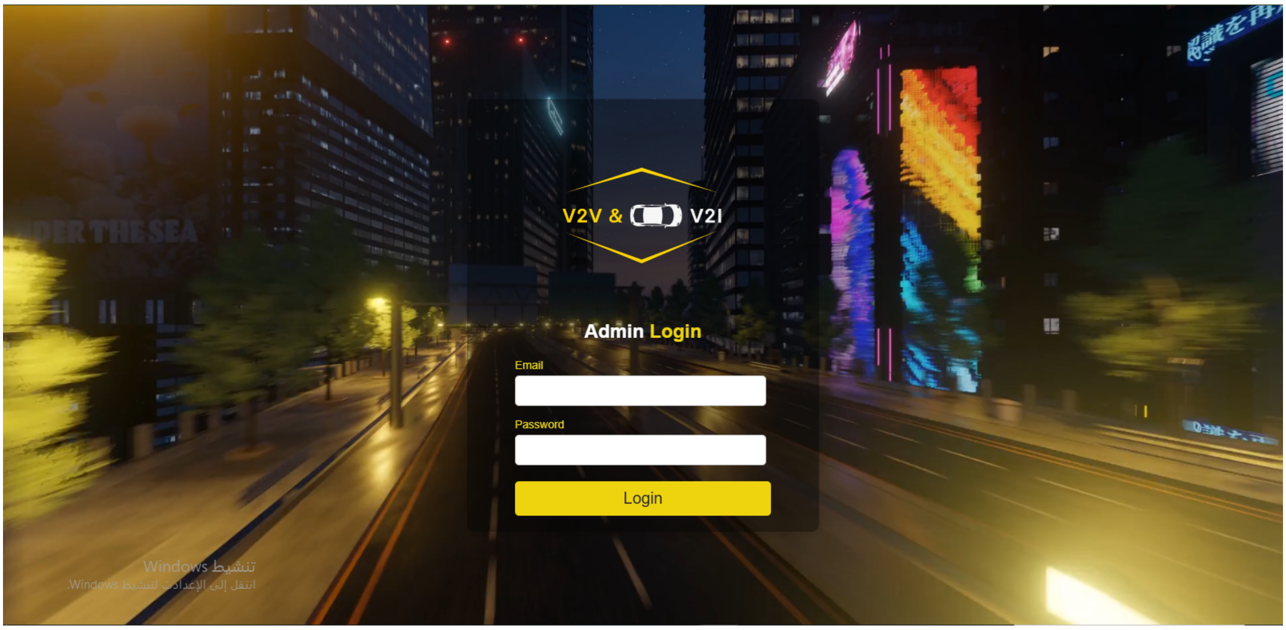


Figure 4.11: Dashboard Login Page.

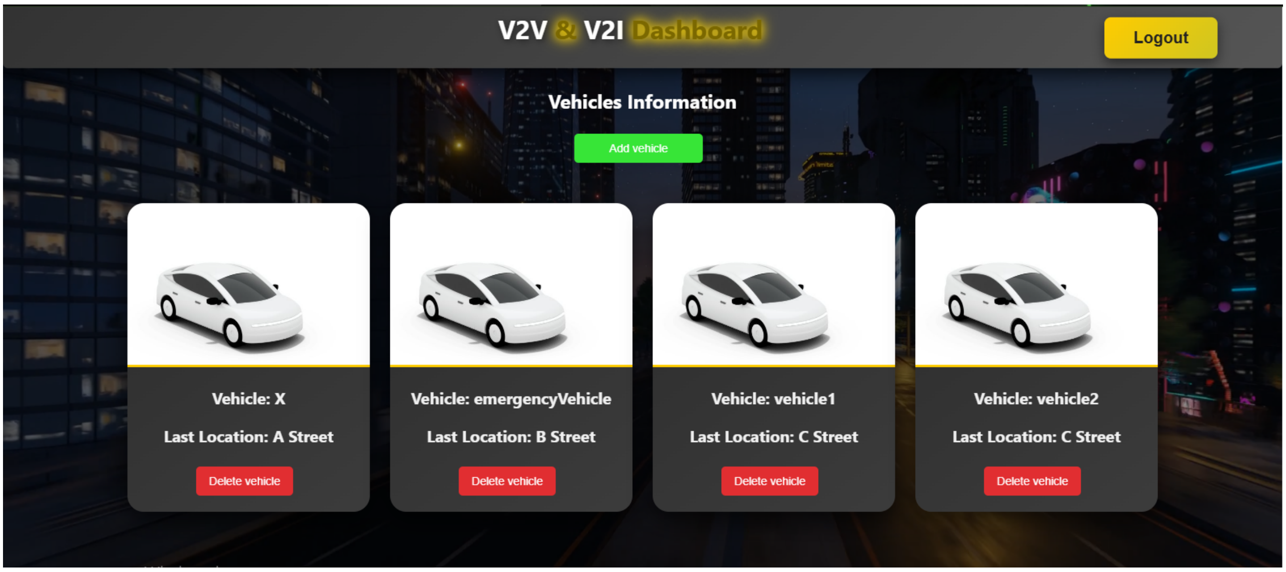


Figure 4.12: Dashboard Home Page.

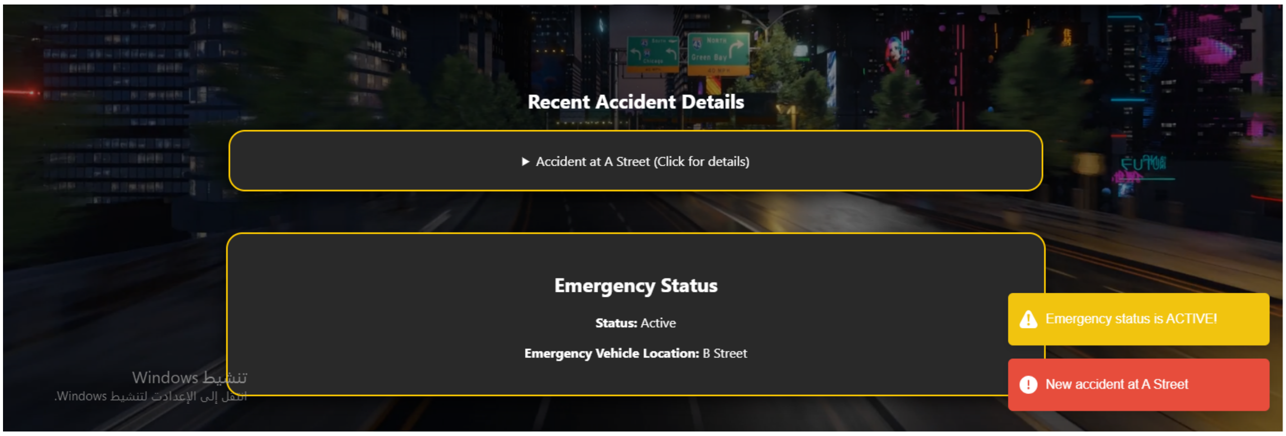


Figure 4.13: Dashboard Notifications.

Coding

Under this topic, we'll describe setup, communication and other functions.

Setup

First, we initialize several parts:

- **Hardware Initialization:** Configured sensors, motors, LCD, RFID reader and buzzer for operation.
- **Wi-Fi and Bluetooth Setup:** Connected to Wi-Fi using WiFi.begin() for Firebase communication and initialized Bluetooth with SerialBT.begin().
- **Firebase Connection:** Integrated FirebaseESP32 library and initialized Firebase with Firebase.begin() for real-time data updates.

Loop

The loop function contains several operations that execute periodically to monitor and control the vehicle's and infrastructure actions. These include accident avoidance, location updates, and periodic crash detection, ensuring real-time response and safety.

Vehicle Movement

The car's movement is controlled via Bluetooth, with each vehicle having a unique identifier for pairing. Commands sent from a dedicated app enable real-time actions like moving forward, backward, turning, or stopping.

Accident Avoidance and Alert Activation in Vehicles

The system continuously checks the distance between the vehicle and any potential obstacles using an ultrasonic sensor. In each iteration of the loop, a pulse is sent to the sensor to measure the distance. If the distance is less than a predefined safe distance (10 cm), the system activates a warning. This includes displaying a ”Too Close! Be Careful!” message on the LCD and triggering the buzzer for an alert. This process runs repeatedly to ensure constant monitoring and timely alerts for accident avoidance.

Vehicle Location Update

This operation involves using an RFID reader to scan the ID of a card that represents the street name. The RFID reader, mounted under the vehicle, detects the card and reads its unique identifier (UID). The UID is then converted into a string and used to fetch the corresponding street location from the Firebase database. Once the street location is retrieved, it is sent to the cloud and the vehicle's last known location is updated in the Firebase database.

Accident Detection and Alert

When a crash is detected, the system displays ”Crash Detected!” along with the vehicle's current location and crash time on the LCD. This information, including the location and time, is sent to Firebase and a buzzer is activated to provide intermittent alerts.

Traffic Light Control for Emergency Vehicle at Intersection

The code controls traffic lights at an intersection based on the presence of an emergency vehicle, using Firebase to monitor the emergency status and the vehicle's location. The system continuously checks for an active emergency by reading the status from Firebase. When an emergency is detected, the code saves the current state of the traffic lights, then adjusts the lights based on the location of the emergency vehicle. If the vehicle is on Street A or Street B, the corresponding traffic light is changed to give priority to the emergency vehicle. Once the vehicle exits the intersection and its location is recorded as ”G Street” in Firebase, the system restores the traffic lights to their previous states. The Firebase status is updated accordingly to reflect the end of the emergency.

4.3 Validation and Testing

This section focuses on the process of testing each of our hardware and software components.

4.3.1 Hardware

Unit Testing

Unit testing involves testing individual components in isolation before integrating them into the complete system. Below are the specific tests conducted for each hardware element.

Ultrasonic Sensor

- The ultrasonic sensor was tested by measuring distances to various objects placed at different ranges.
- Validation involves comparing the measured distance with a tape-measured value to ensure accuracy within an acceptable error margin (e.g., ±1 cm).
- The sensor was also tested for responsiveness when objects were moved closer and farther dynamically.

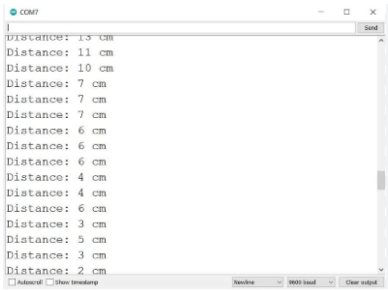


Figure 4.14: Serial Monitor Output Showing Ultrasonic Sensor Distance Measurements.

RFID Reader

- The RFID reader was tested using multiple RFID cards, ensuring that each card's ID was read correctly and transmitted to the system.

- Tests included scenarios with different orientations and distances between the card and the reader to confirm reliable detection.

Crash Sensor

- The crash sensor was pressed manually to ensure it detected physical contact and sent the correct signal to the microcontroller.
- Multiple tests simulated various intensities of impact to verify consistent triggering.

LCD Display

- The LCD screen was tested by displaying predefined messages.
- Both lines of the screen were checked for clarity, proper formatting and readability under different lighting conditions.

Integration Testing

After individual components were validated, they were tested together to ensure smooth integration

RFID Reader and LCD

- When an RFID card was detected, the corresponding location name was displayed on the LCD screen.
- Multiple cards were scanned in sequence to confirm the correct location update for each card.

Ultrasonic and Buzzer

- When the ultrasonic sensor detected an object within 10 cm, the buzzer turned on and the LCD displayed the message ”Too Close! Be Careful.”
- Moving the object beyond 10 cm turned off the buzzer and restored the LCD message to ”Car is ready.”

4.3.2 Software

This section discusses the testing process of each of our software testing.

Ultrasonic Sensor for Obstacle Detection

The ultrasonic sensor's functionality was tested to evaluate its ability to detect obstacles and trigger alerts when the distance fell below the safe threshold of 10 cm. Tests were conducted by placing obstacles at varying distances of 5 cm, 10 cm and 20 cm and the system's responses were observed.

RFID Card Detection and Location Updates

The RFID card detection system was tested to verify its ability to correctly read RFID cards and update the vehicle's location in the Firebase database. During the test, an RFID card was placed beneath the reader to ensure that the UID was accurately read and processed. The system successfully retrieved the corresponding location from Firebase and updated it in real-time.

Accident Detection and Alerts

The accident detection system was tested to ensure it could detect crashes and send alerts accurately. A crash was simulated by triggering the crash sensor and the system's responses were monitored. The LCD displayed the message "Crash Detected!", the buzzer activated for intermittent alerts and the Firebase database was updated with the crash location and time.

Bluetooth-Controlled Vehicle Movement

The Bluetooth-based vehicle control was evaluated to validate its responsiveness to commands. The vehicle was paired with a Bluetooth application and commands including forward, backward, left, right and stop were sent.

Firebase Communication

The Firebase communication was tested to ensure accurate and efficient data storage and retrieval. A test entry was written to the Firebase database and subsequently retrieved to verify its accuracy and consistency with the input data. The results demonstrated that the system effectively maintained data integrity with no observed delays.

Dashboard Testing

The dashboard was tested to ensure real-time responsiveness to changes in the Firebase Realtime Database. It successfully retrieves and synchronizes data from the database, displaying updates promptly and accurately to reflect the latest information.

In addition, the dashboard was tested to add or delete a new vehicle . The figures below shows the add and delete forms.

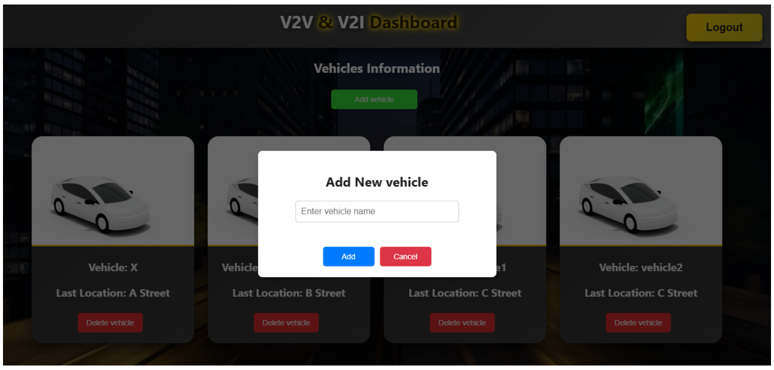


Figure 4.15: Vehicle Add Form.

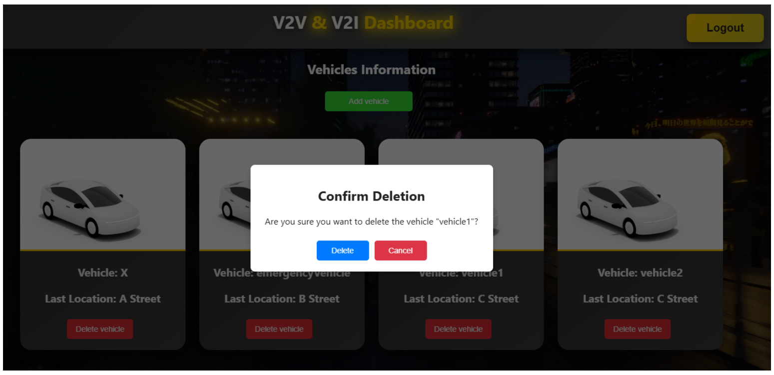


Figure 4.16: Vehicle Delete Form.

Traffic Light Control for Emergency Vehicles

The traffic light control system was tested for its ability to prioritize emergency vehicles at intersections. The test involved simulating an emergency by setting the vehicle's status to "Emergency" in Firebase. The system adjusted the traffic lights based on the vehicle's location, giving priority to the emergency vehicle. Once the vehicle exited the intersection and the emergency ended, the traffic lights returned to their original state.

Chapter 5

Results and Discussion

5.1 Overview

This chapter provides a detailed analysis of the experimental results, supported by tables where applicable. It includes error and success rate calculations and offers justifications for the outcomes in relation to the project objectives.

5.2 Detailed analysis of the results/experiments

The experimental phase of this project aimed to evaluate the performance and functionality of the system. Comprehensive testing was conducted on individual components, subsystems and the overall integrated system. Response time was measured using a software timer to ensure accurate performance evaluation. The results obtained are presented in Figure 5.1, supported by relevant tables for better clarity.

Criterion/Scenario	Accident Avoidance (No Vibrations)	Vehicle Location Update	Accident Detection (Good Connection)	Accident Detection (Poor Connection)	Emergency State (Good Connection)	Emergency State (Poor Connection)
Real-Time Response	10-15 ms	10-20 ms	0.5-1.2 s	1.5-2 s	0.5-1 s	2-2.5 s
Stability	Very Stable	Stable	Stable	Stable with minor delay	Stable	Stable with slight impact
Impact of Connection Quality	None	None	None	Slight Delay	None	Slight Delay

Table 5.1: Performance Analysis with Detailed Metrics.

Table 5.2 presents the system's performance across various scenarios, emphasizing the effect of internet connectivity on real-time updates.

Scenario	Environment	Expectation	Results	Performance Analysis
Accident Avoidance	Obstacles present, no vibrations	Ultrasonic sensor detects obstacles and activates the necessary alarms	Issue alerts to successfully avoid collisions	Excellent performance, accurate with no false alarms
Accident Avoidance	Obstacles present, with vibrations	Sensor reading is slightly affected with no false alarms	Stable performance with no false alarms	System maintained accuracy even with vibrations
Vehicle Location Update	Dynamic vehicle movements	The system updates vehicle's real-time location	Real-time updates provided accurately	Very good performance in ideal connection
V2V Accident Detection	Two-Vehicle Crash (Good internet connection)	Real-time alert is sent to vehicles	Collision detected and alerts sent to nearby vehicles	Excellent performance with no delays in alerts
V2V Accident Detection	Two-Vehicle Crash (Poor Internet Connection)	Send alerts with slight delays	Alerts sent with minor delays	Acceptable performance; slight delays observed
V2I Emergency State	Priority Activation (Good internet connection)	Real-time traffic signal response and priority given to emergency vehicles	Traffic signals prioritized emergency vehicles	Accurate and fast system response
V2I Emergency State	Priority Activation (Poor Connection)	Signals optimized for emergency vehicles with slight delays	Given priority with minor delay	System managed to handle connection adequately

Table 5.2: Results and Performance Analysis.

5.3 Error / Success Rate Calculations

This section provides an analysis of success and error rates, calculated based on the number of attempts and supported by equations, for each scenario in the project. The error rate is calculated using the formula:

$$\text{Error Rate} = \left(\frac{\text{Number of Failed Trials}}{\text{Total Number of Trials}} \right) \times 100\%$$

Similarly, the success rate is derived from the error rate using the equation:

$$\text{Success Rate} = 100\% - \text{Error Rate}$$

Scenario Name	Number of Experiments	Success Rate	Error Rate	Reasoning
Accident Avoidance	50	98%	2%	When there is vibration and an open space in front of the ultrasonic sensor, the signal may become dispersed.
Vehicle Location Update	70	92%	8%	Improper placement or orientation of the RFID cards on the street might have made it difficult for the reader to detect them.
V2V Accident Detection	50	80%	20%	Poor internet connectivity delays the transmission of accident alerts and hinders real-time communication between vehicles.
V2I Emergency State	50	90%	10%	Weak Wi-Fi signal negatively impacts real-time communication between vehicles and infrastructure.

Table 5.3: Error / Success Rate Calculations.

These calculations help evaluate the system's performance by determining the proportion of failures and successes for each scenario. The error rate depends on not reaching the ideal time, affecting overall system reliability and efficiency.

5.4 Justifications of the Obtained Results

The results align well with the objectives of the project, demonstrating the efficacy of the system in real-world scenarios. Key justifications include:

1. **Sensor Performance:** The choice of ultrasonic and crash sensors ensured high precision in obstacle detection and collision recognition. These sensors operate efficiently within the defined parameters, contributing to the system's overall reliability.

2. **Cloud Integration:** Leveraging Firebase enabled real-time data exchange and processing. This was evident in the seamless operation of V2V and V2I communication during testing.

3. **Traffic Light Management:** The rapid adjustment of traffic signals in response to emergency scenarios validated the effectiveness of the implemented algorithms.

4. **Scalability and Reliability:** The system's architecture proved capable of handling multiple vehicles and intersections without significant performance degradation.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This project successfully developed an intelligent vehicle system that integrates hardware and software for autonomous navigation. The system includes components like motors, sensors, an RFID reader and communication modules, all working together to enable real-time interaction with the environment. The software, built with the Arduino IDE, handled critical tasks such as reading sensor data, processing commands and updating location information via Firebase. Comprehensive testing confirmed the system's accuracy in reading RFID cards, detecting obstacles and providing timely feedback, making it a reliable prototype for autonomous vehicle operations.

6.2 Challenges

Throughout the development of the intelligent vehicle system, several challenges were encountered:

1. **Hardware Integration:** Connecting various components like motors, sensors and the RFID reader required careful calibration and testing to avoid signal interference.

2. **Real-Time Performance:** Achieving timely responses from sensors (ultrasonic, RFID) was challenging, requiring optimization for faster data processing.

3. **Communication Issues:** Maintaining stable Wi-Fi communication for real-time data updates with Firebase was occasionally disrupted by network delays.

4. **Power Management:** Ensuring efficient power usage to keep the system running without frequent battery changes was a challenge.

5. **Debugging and Testing:** Debugging complex systems with multiple components required extensive testing and adjustments for seamless integration.

6.3 Future Work

Future improvements could focus on enhancing sensor integration by adding cameras or LiDAR for better environmental awareness. Additionally, introducing vehicle autonomy through advanced algorithms could allow for fully autonomous navigation. The scalability of the cloud system could also be expanded to handle more complex environments and traffic management could be made smarter through dynamic signal control based on real-time data. Moreover, energy efficiency could be improved with low-power solutions and a mobile app could provide better user interaction and monitoring. These enhancements would push the project toward a more sophisticated and efficient autonomous vehicle system.

References

[1] A. Sarango, G. Vásquez, and R. Torres, “A vehicle-to-vehicle communication system using zigbee,” *IOP Conference Series: Earth and Environmental Science*, vol. 1370, p. 012008, 2024.

[2] Google, *Firebase Realtime Database*, n.d., accessed: 2025-01-16. [Online]. Available: https://firebase.google.com/docs/database

[3] A. Name, “Iot in transportation: Enhancing road safety and efficiency,” *Transportation Journal*, vol. 5, pp. 25–40, 2021.

[4] K. Dey, A. Rayamajhi, M. Chowdhury, P. Bhavsar, and J. Martin, “Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communication in a heterogeneous wireless network–performance evaluation,” *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 168–184, 2016.

[5] G. S. Patil and N. V. Shirsat, “Vehicle to vehicle communication,” Bachelor’s project, Department of Electronics & Telecommunication, JSPM’s Rajarshi Shahu College of Engineering, Tathawade, Pune-33, 2016.

[6] P. Kumar and K. B. Ali, “Intelligent traffic system using vehicle to vehicle (v2v) & vehicle to infrastructure (v2i) communication based on wireless access in vehicular environments (wave) std,” 2022, [Online; accessed January 11, 2025], Available at: 2022 International Conference on Computing and Network Communications (CoCoNet), pp. 1-6.

[7] Electronic Spices, “Electronic spices dual shaft dc motor - plastic,” 2016, [Online; accessed April 27, 2024], https://www.amazon.in/Electronic-Spices-Shaft-Motor-Plastic/dp/B08RS48GQD.

[8] AZ-Delivery, “Dual h-bridge dc motor controller board module az-l298n,” 2020, [Online; accessed April 27, 2024],https://www.az-delivery.de/en/products/dual-h-bridge-dc-motor-controller.

[9] Giffits Powerban, “Loop 10.000 mah powerbank aus recycel,” 2019, [Online; accessed April 27,2024],https://www.giffits.de/loop-10-000-mah-powerbank-aus-recyceltem-kunststoff-werbeartikel-502300.htm.

[10] A.Arvind Kumar,S. Rajasekaran, et al., “Design and development of smart metering system for energy monitoring using esp32 and iot,” 2019, [Online; accessed April 27, 2024],https://www.researchgate.net/figure/PIN-description-of-ESP32-1-Power-Supply-Pins-VCC-This-pin-serves-as-the-power-supply_fig1_374504846.

[11] IndiaMART, “Crash collision sensor impact switch module,” 2019, [Online; accessed April 27, 2024],https://www.indiamart.com/proddetail/crash-collision-sensor-impact-switch-module-19967183112.html?mTd=1.

[12] Banggood (brand/manufacturer), “Ultrasonic distance measuring transducer,” 2015, [Online; accessed April 27, 2024],https://www.amazon.in/Banggood-Ultrasonic-Distance-Measuring-Transducer/dp/B01I1ZTPJC.

[13] Mybotic, “Rfid card reader module rc522,” 1948, [Online; accessed April 27, 2024],https://www.mybotic.com.my/rfid-card-reader-module-rc522.

[14] TinyTronics, "Lcd display 16*2 characters with white text and blue backlight," 1968, [Online; accessed April 27, 2024],https://www.tinytronics.nl/en/displays/lcd/lcd-display-16*2-characters-with-white-text-and-blue-backlight.

[15] RoboComp, "Piezo Electronic Buzzer Continuous Sound 9V-12V Buzzer Electromagnetic," 1831, [Online; accessed April 27, 2024],https://robocomp.in/product/piezo-electronic-buzzer-continuous-sound-9v-12v-buzzer-electromagnetic/.

[16] HIT (Palestinian Industrial Electronics Shop), "Tactile momentary push button red 12mm," 2022, [Online; accessed April 27, 2024],https://hit.ps/product/tactile-momentary-push-button-red-12mm/.

[17] ResearchGate, "Working principle of ir sensor traffic light module," 2022, [Online; accessed April 27, 2024],https://www.researchgate.net/figure/Working-principle-of-IR-sensor-Traffic-Light-Module-Traffic-Light-Module-is-a_fig4_366025155.

[18] K. Dey, A. Rayamajhi, M. Chowdhury, P. Bhavsar, and J. Martin, "Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communication in a heterogeneous wireless network—performance evaluation," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 168–184, 2016.

[19] P. S. Perumal, M. Sujasree, S. Kedarisetty, and K. Gokul, "Lidar based intelligent obstacle avoidance system for autonomous ground vehicles," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, pp. 2466–2474, 2020.

[20] J.-y. Zhuang, L. Zhang, S.-q. Zhao, and J. Cao, "Radar-based collision avoidance for unmanned surface vehicles," *China Ocean Engineering*, vol. 30, no. 6, pp. 867–883, 2016.

[21] H. M. Ali and Z. S. Alwan, "Car accident detection and notification system using smartphone," *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 4, pp. 620–635, April 2015.

[22] S. R. Aishwarya *et al.*, "An iot based accident prevention and tracking system for night drivers," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 4, April 2015.

[18] [1] [19] [20] [21] [22]