Mini Project Report On

Smart Bus Monitoring System

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

Bachelor of Technology

in

Electronics and Communication Engineering

By

A Kalaiyarasi (U2201001)
Aleena M B (U2201020)
Angelina Mary Giby (U2201034)
Architha B (U2201051)

Under the guidance of

Dr. Jisa David

Department of Electronics and Communication Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039

April 2025

CERTIFICATE

This is to certify that the mini project report entitled ***"Smart Bus Monitoring System"*** *is a bonafide record of the work done by* ***A Kalaiyarasi (U2201001)****, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Electronics and Communication Engineering" during the academic year 2024-2025.*

Dr. Jisa David	**Dr. Rithu James**	**Dr. Jisa David**
Project Guide	Project Co-ordinator	Head of Department
Associate Professor	Associate Professor	Associate Professor
Dept. of ECE	Dept. of ECE	Dept. of ECE
RSET	RSET	RSET

ACKNOWLEDGEMENT

I wish to express my sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **Dr. Jisa David**, Head of the Department of Electronics and Communication Engineering for providing me with the opportunity to undertake my project, Smart Bus Monitoring System.

I am highly indebted to my project co-ordinators, **Dr. Rithu James** and **Dr. Simi Zerine Sleeba**, Department of ECE, for their valuable support and guidance.

It is indeed my pleasure and a moment of satisfaction for me to express my sincere gratitude to my project guide **Dr. Jisa David** for her patience and all the priceless advice and wisdom she has shared with me.

Last but not the least, I would like to express my sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Abstract

Public transportation in India is a lifeline for millions, yet it faces persistent challenges such as overcrowding, lack of real-time information, and concerns over driver safety. While individual systems exist to address specific issues like location tracking, alcohol detection, and passenger counting, these solutions often function in isolation. Our project, Smart Bus Monitoring System stands out by integrating all these key features into a unified and practical solution. This system enables real-time tracking of buses, live passenger count monitoring, and ensuring driver sobriety all in one platform. Passengers can access information such as the number of buses on a particular route, their current locations, occupancy levels, and seat availability through a simple interface, such as a website linked via QR codes placed at bus stops. This allows commuters to make informed decisions and avoid overcrowded buses, enhancing travel experience, comfort and safety. In addition, the system includes an alcohol detection mechanism to ensure that drivers are fit to operate the vehicle, thereby promoting safety and responsible driving and preventing potential accidents. This holistic approach benefits not only daily commuters, but also school administrations, transport authorities, and the general public by improving operational transparency, passenger experience, and safety standards in public transport. By integrating multiple standalone systems into a single smart solution, this project contributes significantly to the development of a safer, smarter, and more efficient public transportation network.

Contents

List of Abbreviations

ESP - Espressif Systems Protocol

MCU - MicroController Unit

GPS - Global Positioning System

HW - Hardware

MQ3 - Metal Oxide Semiconductor Gas Sensor

HTML - HyperText Markup Language

IR - Infrared

QR - Quick Response

Wi-Fi - Wireless Fidelity

UART - Universal Asynchronous Receiver Transmitter

IDE - Integrated Development Environment

URL - Uniform Resource Locator

SDK - Software Development Kit

RISC - Reduced Instruction Set Computer

GPIO - General Purpose Input Output

SPI - Serial Peripheral Interface

I2C - Inter-Integrated Circuit

LED - Light Emitting Diode

NMEA - National Marine Electronics Association

List of Figures

List of Tables

Chapter 1

Introduction

Millions of people in India depend on public transportation system but problems like crowding and lack of real-time information make it less comfortable and inefficient particularly in urban areas. Without realizing that a less crowded bus might be only a few minutes away, commuters frequently board the bus that arrives first. This causes delay, discomfort, and possible safety hazards. In order to address these issues, we suggest a Smart Bus Monitoring System that offers real-time information on bus location crowd density and ensures driver sobriety. Passengers can make informed decisions via a web or mobile interface increasing safety, easing traffic and improving the travel experience.

1.1 Background

Buses are among the most popular forms of transportation, especially in urban and semi-urban areas in India. Students, office workers the general public rely on buses as an accessible and affordable mode of transportation for their daily commutes. However, there are several issues that affect the effectiveness, reliability, and safety of the public transportation system. Overcrowding is among the most prevalent issues. People often board the first bus that comes to their stop without realizing that there may be a more comfortable and less crowded bus on the same route just a short distance away resulting in longer boarding time and discomfort. The uneven distribution of passengers where some buses run with many empty seats while others are overcrowded leads to poor resource utilization and an uncomfortable commuting experience. In addition, passengers often have no way of knowing how many buses are available on a specific route at any given time, its current location and how full each bus is. Consequently, commuters end up making ill-informed choices which lead to inconvenience and deter regular reliance on public transportation. Safety issues are also a significant factor in determining how the general public views bus services. Reckless or irresponsible driving especially under the influence of alcohol also poses significant threats to passenger safety. Addressing such behaviour is crucial for ensuring a secure and reliable public transportation system.

1.2 Problem Definition

This project aims to develop an integrated system that provides real-time information about buses operating on a selected route to help passengers make safer and more informed travel decisions. This includes data on the current location of the bus, its occupancy levels, and the driver's sobriety. Figure 1.1 shows an overview of the Smart Bus Monitoring System. This system is designed to address key concerns related to safety, route visibility, and overcrowding in public bus transportation.

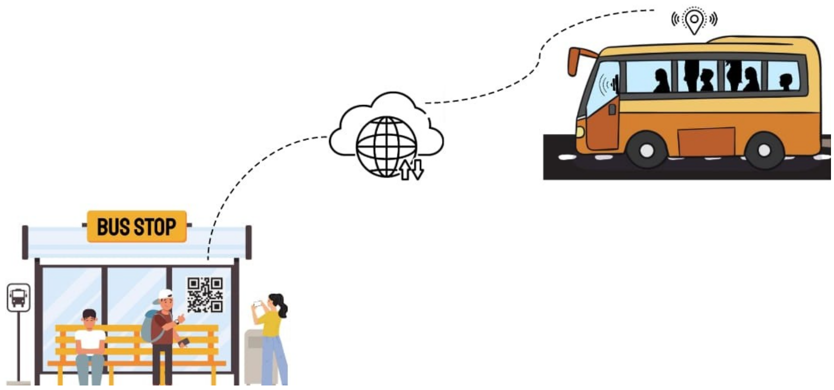


Figure 1.1: Overview of the Smart Bus Monitoring System

1.3 Scope and Motivation

The Smart Bus Monitoring System aims to improve public transportation by providing real-time data on bus location, occupancy, and availability. This information is easily accessible via QR codes at bus stops, leading to a user-friendly web interface. The motivation behind the project comes from common challenges faced by daily commuters such as overcrowded buses, lack of route visibility, and uncertainty in bus availability. The scope of the system includes real-time tracking, passenger counting, and driver sobriety monitoring all integrated into a single platform. Unlike existing solutions that address these problems through separate, isolated modules, our project offers a comprehensive and unified system, delivering a smarter, safer, and more efficient public transport experience.

1.4 Objectives

1. To provide real-time tracking of buses on a selected route
2. To implement a passenger counting system for live occupancy updates
3. To ensure passenger safety by monitoring the driver's sobriety
4. To display comprehensive route and bus details through a user-friendly web interface
5. To reduce overcrowding and improve distribution of passengers across buses
6. To enhance overall efficiency and trust in public transportation through a unified system combining real-time data and safety measures

1.5 Challenges

The Smart Bus Monitoring System faced challenges such as limited GPS accuracy in areas with weak signal coverage, which affected real-time location tracking. Accurate passenger counting depended on individual boarding and exiting one at a time, which did not always happen in practice. Additionally, maintaining consistent alcohol detection was difficult, as sensor thresholds varied due to environmental factors.

1.6 Assumptions

1. It is assumed that only one passenger boards or exits the bus at a time to ensure accurate counting through the IR sensor-based bi-directional counter.
2. The bus operates on a fixed route. The changes in routes can be updated through the software interface to reflect the new path.
3. It is also assumed that passengers will have access to a smartphone or compatible device to scan QR codes and access route and occupancy information.

1.7 Societal / Industrial Relevance

In a country like India, where a large portion of the population depends on public buses for daily commuting, the efficiency and safety of bus transportation directly impact the quality of life. However, current transportation systems often suffer from disorganization, overcrowding, lack of real-time information, and poor safety monitoring. This leads to unnecessary stress, delays, and even accidents, especially during peak hours. The Smart Bus Monitoring System addresses these issues by introducing a user-friendly solution that provides real-time updates on the number of buses on a particular route, their current locations, and the number of passengers on board. With access to such information, passengers can make better travel decisions and avoid boarding overcrowded buses, leading to a more balanced and comfortable commuting experience. Moreover, the system helps reduce the chances of dangerous situations caused by drunk driving through an alcohol detection feature that ensures drivers are fit to operate the vehicle. The system also helps prevent overcrowding by allowing better distribution of passengers across available buses. This not only increases passenger safety and comfort but also encourages greater public trust in the transport system. The project thus holds significant social value, offering benefits for everyday travelers, enhancing public transportation safety, and assisting public transport authorities in managing fleet operations more effectively. By promoting safer, smarter, and more efficient public transport, this system contributes to the development of a well-organized urban mobility infrastructure that is responsive to the needs of the people.

1.8 Targeted customers

The Smart Bus Monitoring System is designed to cater to a diverse range of customers within the public and institutional transportation sector. Daily commuters form a primary user group, benefiting from real-time bus location tracking and live occupancy data, enabling them to make informed travel decisions and avoid overcrowded buses. School management systems and parents could also benefit from the project, as the platform provides timely parental notifications and ensures safer transportation through driver sobriety monitoring. Furthermore, public transport authorities can utilize the system for emergency response, route optimization, thereby enhancing overall operational efficiency. By addressing the needs of these customer segments through an integrated and reliable solution, the system ensures smarter and safer travel for a wide audience.

This chapter introduced the key issues in public bus transportation, such as overcrowding, lack of real-time updates, and safety concerns. The proposed Smart Bus Monitoring System aims to address the issues by offering live tracking, passenger counting, and driver sobriety checks. The project's objectives and motivation were outlined, along with practical challenges and system assumptions. Its relevance was emphasized in the context of improving urban mobility, passenger safety, and commuter convenience. Overall, the chapter sets the groundwork for building a smarter and more reliable public transport solution.

Chapter 2

Literature Survey

Currently, some of the main challenges faced in public transportation are overcrowding of passengers in a vehicle, inefficient tracking, and driver intoxication, affecting safety and efficiency. Various studies have explored different approaches to solving these problems. These technologies improve monitoring and safety but come with limitations. This survey reviews solutions that have been already proposed eventually helping to justify the components and methods used in our project.

Sojol et al. [1] proposed a solution which aims to measure the passenger counting in buses using pressure pads by installing them on the seats. This solution uses an Arduino Uno microcontroller to process data from pressure pads based on the change in weight and then transmits this data in real-time using a Bluetooth module to a mobile application. The application is the proposed front end of the system to try to improve transparency and operational efficiency in public transport. With the prototype achieving 90% accuracy, the solution offers a much simpler alternative compared to complex and costly ones that make use of infrared sensors, video imaging, or wireless signal tracking.A disadvantage of using pressure sensors on the sears is that it can count an additional passenger onboard if a heavy object has been placed on the seat. Similarly, it can also detect a count of passengers less than the actual number of passengers onboard in case the pressure threshold is not calibrated properly. Since the solution does not use IR or video-based systems, it fails to track how many passengers board or leave the bus, which hinders route optimization and passenger flow analysis.

Nair et al. [2] proposed a solution which is an Android application designed to allow real-time tracking of the bus and digital ticketing. It uses GPS to obtain the live location, QR codes for validating the bus passes of the passengers and a printer integrated with

Bluetooth to generate bus tickets on the go. This approach aims to build a system that aims to enhance user convenience and operational efficiency. To make the process more convenient, passengers can access the real-time location of the bus, as well as purchase tickets using e-wallets or cash. This also helps in avoiding formation of queues and ensures that a smooth flow of passengers, from entering the bus to exiting when their stop has arrived, is maintained. The proposed solution only focuses on tracking and ticketing, because of which the current number of passengers on the bus remains a mystery. This data could be extremely valuable for managing the capacity and how many passengers the bus can safely transport. It also assumes that users can download an application on the spot, which might not be possible due to various reasons such as shortage of storage capacity on their mobile device.

Hannan et al. [3] put forth a solution that uses a combination of elements such as Radio Frequency Identification (RFID), Global Positioning System (GPS), General Packet Radio Service (GPRS), and Geographic Information System (GIS). Here, RFID is used to identify the vehicle and GPS is used to track the location of said vehicle in real-time. Along with that, GPRS is used for transmission of data and GIS is used for mapping and visualization of data. The data is then processed and analyzed using a theoretical framework and rule-based decision algorithms. The authors have implemented a prototype of the solution and the results show that this kind of approach is effective and achieves the goal of monitoring and managing transportation systems well. The solution presented here does not discuss ways to monitor passenger activity such as boarding and exiting the bus, which is crucial information to manage crowd levels. Also, there is no front end designed for the passengers to access relevant data related to the bus, thus, limiting user engagement.

Astrain et al. [4] put forth a system to monitor electric buses in real-time using IoT (Internet of Things) technologies. Here, there is GPS to track the location of the bus, specific sensors to keep a check on vehicle diagnostics, and finally, wireless communication modules so that the data that is collected can be transmitted to a central platform. This setup enables improved passenger information services, efficient and effective management of multiple vehicles, and proper and timely maintenance of the same. Such solutions strive to contribute towards developing smarter and more sustainable urban transportation systems. Like other proposed solutions, this approach also pays minimal attention to keeping a track of the number of passengers currently on board, making it difficult to ensure that overcrowding does not occur. It also does not elaborate and how passengers can view information related to the bus, such as current location, seating capacity, etc., reducing user engagement.

Chapter 3

Methodology

This methodology highlights the systematic approach towards designing, developing, and implementing a real-time system that displays the current location of the bus, counts the number of current passengers, and ensures driver safety. This information is constantly updated to a website which can be accessed by users to obtain relevant details. The system is an integration of various sensors and communication modules which aims to achieve enhanced public transport efficiency, reliability, and safety.

3.1 Block Diagram of Smart Bus Monitoring System

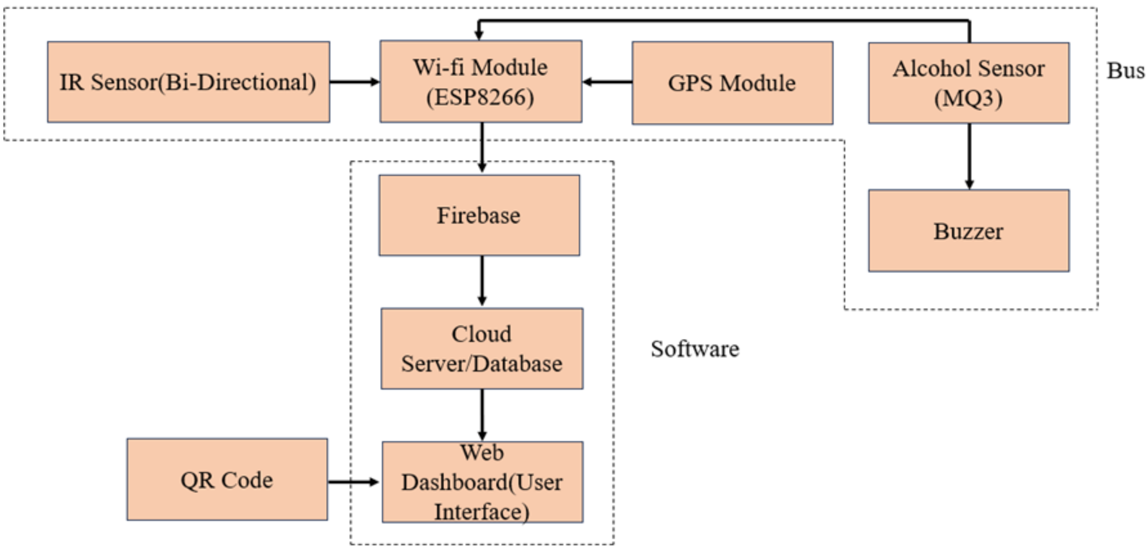


Figure 3.1: Block diagram of Smart Bus Monitoring System

Figure 3.1 shows the block diagram of the Smart Bus Monitoring System. Two IR sensors are placed at the bus entry or exit. When a passenger enters, the first sensor detects an object in its path and sends the light back to the receiver while the second sensor remains in its initial state. This increments the count of current passengers in the bus. Similarly, when a passenger exits, the second sensor detects an object in its path and sends the light back to the receiver while the first sensor remains in its initial state.this decrements the count.The GPS module, NEO-6M, sends the location data via UART. The microcontroller, ESP8266, then extracts the latitude and longitude data using the TinyGPS++ library in Arduino IDE. It then pushes the data received from the IR sensors and GPS module into the Firebase Realtime Database, which stores the current passenger count and location of each bus in it. A QR code placed at each bus stop can be scanned by a user to obtain relevant information about the available buses. This QR code will redirect them to the hosted web URL. Upon redirection, the website prompts users to select from a list of predefined routes. Once a route is, selected the users can choose a bus available in the drop down list, which only consists of the buses traveling in the selected route. The website connects to Firebase using JavaScript SDK. The website then displays the current location of the bus using a location pin on a map, the current passenger count and the seating capacity.

3.2 Components and Specifications

3.2.1 ESP8266 (NodeMCU)

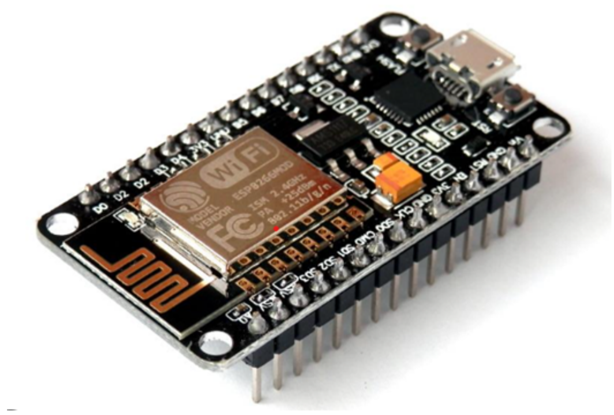


Figure 3.2: ESP8266 (NodeMCU)

Figure 3.2 shows the microcontroller, ESP8266. It is a low-cost, Wi-Fi-enabled microcontroller module developed by Espressif Systems, widely used in IoT (Internet of Things) applications. It features a 32-bit RISC processor, integrated Wi-Fi capabilities, and multiple GPIO (General Purpose Input/Output) pins, making it suitable for wireless communication and embedded system projects. The module supports various communication protocols such as UART, SPI, and I2C, allowing it to interface with sensors, actuators, and other peripherals.

One of the key advantages of the ESP8266 is its ability to connect to Wi-Fi networks and function as both a client and an access point, enabling remote monitoring and control of devices over the internet. It can be programmed using different platforms, including Arduino IDE, MicroPython, and AT commands, making it accessible to developers of all skill levels. Additionally, its low power consumption and compact size make it ideal for battery-powered applications in smart home automation, industrial monitoring, and remote sensing systems.

3.2.2 IR Sensor

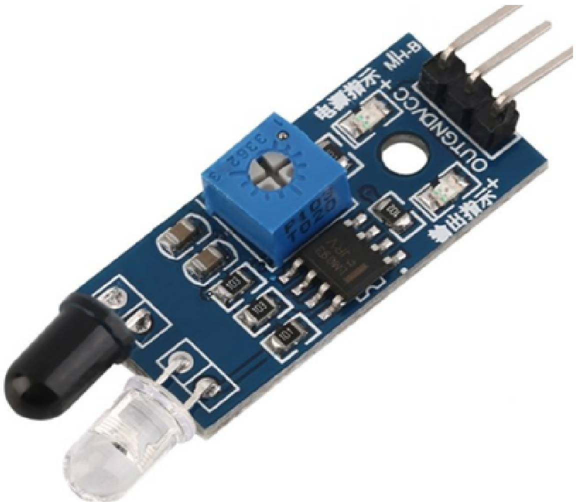


Figure 3.3: IR Sensor

Figure 3.3 shows an IR sensor module. It is a widely used electronic device designed for object detection, proximity sensing, and motion tracking. It works by emitting infrared light through an IR LED and detecting the reflected light using a photodiode or phototransistor. When an object comes within the sensor's range, it reflects the infrared light back to the receiver, which then generates an electrical signal indicating the presence of an object. The module is commonly used in automation systems, robotics, and security applications due to its reliability and cost-effectiveness.

The IR sensor module typically includes a comparator circuit to process the received signals and produce a digital or analog output. A potentiometer is often included in the circuit, allowing users to adjust the sensor's sensitivity and detection range. The output can be interfaced with microcontrollers such as Arduino, ESP8266, or Raspberry Pi for further processing, making it suitable for a wide range of embedded applications.

There are two main types of IR sensors: active IR sensors and passive IR sensors (PIR sensors). Active IR sensors emit their own infrared light and detect objects based on the reflection, while passive IR sensors detect infrared radiation naturally emitted by warm objects, such as human bodies. Because of this, active IR sensors are commonly used in obstacle detection and automation, while passive IR sensors are primarily used in motion detection and security systems.

Due to their compact size, low power consumption, and fast response time, IR sensor modules are widely used in line-following robots, automatic doors, touchless sanitizers, remote control systems, and industrial automation. Their versatility and ease of integration make them an essential component in modern electronic and IoT-based projects.

3.2.3 Alcohol Sensor

Figure 3.4: Alcohol Sensor (MQ-3)

Figure 3.4 shows an alcohol sensor, MQ-3. It is a widely used gas sensor designed to detect the presence of alcohol vapors in the air. It operates on the principle of chemiresistance, where a sensing material undergoes a change in electrical resistance when exposed to alcohol. The MQ-3 sensor is particularly sensitive to ethanol, making it useful for applications such as breathalyzers, alcohol detection systems, and safety monitoring devices.

The sensor consists of a tin dioxide semiconductor layer, which has low conductivity in clean air but exhibits a significant change in resistance when exposed to alcohol vapors. When alcohol molecules come into contact with the sensor's surface, they undergo an oxidation reaction, altering the resistance of the material. This change is then converted into an electrical signal, which can be read as an analog or digital output by microcontrollers like Arduino, ESP8266 or Raspberry Pi.

The MQ-3 module typically operates at 5V DC and has a built-in heating element to maintain the optimal operating temperature for gas detection. However, this heating element results in higher power consumption compared to other sensors. The module features both analog and digital outputs, with the analog output providing a varying voltage based on the alcohol concentration, while the digital output can be used with a preset threshold. A potentiometer on the module allows users to adjust the sensitivity of the sensor.

Due to its high sensitivity, fast response time, and ease of integration, the MQ-3 sensor is widely used in breath alcohol analyzers, vehicle safety systems, workplace safety enforcement, and industrial alcohol detection applications. However, it is important to note that factors like temperature, humidity, and other gases can affect its accuracy, requiring proper calibration for precise readings.

3.2.4 Buzzer (HXD)

Figure 3.5: Buzzer (HXD)

Figure 3.5 shows a buzzer. It is an audio signaling device used to produce sound in response to an electrical signal. It is commonly used in applications such as alarm systems, timers, notifications, and electronic devices where audible feedback is required. Buzzers are compact, energy-efficient, and easy to integrate into circuits, making them widely used in consumer electronics, automotive systems, industrial machines, and security systems.

There are two main types of buzzers: active buzzers and passive buzzers. Active buzzers have a built-in oscillator circuit that generates sound when powered, requiring only a simple DC voltage to operate. Passive buzzers, on the other hand, require an external signal (such as a pulse-width modulation (PWM) signal) to produce sound. This allows for greater control over the frequency and tone of the output, making passive buzzers suitable for applications that require variable tones or melodies.

Buzzers operate using either electromagnetic or piezoelectric mechanisms. Electromagnetic buzzers use a coil and a diaphragm to produce sound through vibration when an electric current passes through them. Piezoelectric buzzers, which are more common in modern electronics, use a piezoelectric ceramic disc that deforms when subjected to an electric field, generating sound waves. Piezoelectric buzzers are preferred for their low power consumption, durability, and ability to produce a wide range of frequencies.

Due to their simple operation and reliability, buzzers are widely used in warning systems, electronic devices, home appliances, and automation projects. They provide an effective way to alert users through audible signals, making them essential components in various embedded systems and IoT applications.

3.2.5 GPS Module (NEO-6M)

Figure 3.6: GPS Module (NEO-6M)

Figure 3.6 shows the GPS module, NEO-6M. It is a widely used satellite positioning device designed for accurate location tracking, navigation, and time synchronization. Developed by u-blox, it is based on Global Positioning System (GPS) technology, enabling real-time determination of latitude, longitude, altitude, and speed. The module communicates with satellites to provide precise positioning data, making it ideal for applications such as vehicle tracking, geolocation services, robotics, and IoT-based navigation systems.

The NEO-6M GPS module consists of a high-sensitivity GPS receiver, an integrated ceramic patch antenna, a flash memory chip for storing configurations, and a UART (Universal Asynchronous Receiver-Transmitter) interface for communication with microcontrollers like Arduino, ESP8266, Raspberry Pi, and other embedded systems. The module typically operates at 3.3V to 5V and supports NMEA (National Marine Electronics Association) protocols, which provide standard output formats for GPS data, including information such as coordinates, time, and satellite status.

A key feature of the NEO-6M GPS module is its high accuracy and fast position acquisition, even in challenging environments. It supports cold, warm, and hot start modes, allowing it to establish a connection with satellites at different speeds depending on previous usage and environmental conditions. Additionally, it includes an external backup battery (RTC battery) to retain location data and reduce the time required for reacquiring satellite signals after power loss.

Due to its compact size, low power consumption, and high reliability, the NEO-6M GPS module is widely used in automobile navigation, asset tracking, drone applications, and smart transportation systems. Its ability to work seamlessly with various microcontrollers and software platforms makes it a versatile choice for real-time positioning and geospatial applications.

3.2.6 Lithium-Ion Battery

Figure 3.7: Lithium-Ion battery

Figure 3.7 shows a lithium-ion battery. It is a rechargeable energy storage device widely used in consumer electronics, electric vehicles, and industrial applications due to its high energy density, lightweight design, and long lifespan. Unlike traditional lead-acid or nickel-based batteries, lithium-ion batteries offer higher efficiency, lower self-discharge rates, and minimal memory effect, making them ideal for applications requiring reliable and long-lasting power sources.

Lithium-ion batteries operate based on the movement of lithium ions between the anode and cathode during charging and discharging cycles. The anode is typically made of graphite, while the cathode consists of lithium metal oxides such as lithium cobalt oxide or lithium iron phosphate. An electrolyte solution, usually a lithium salt dissolved in an organic solvent, allows ion transfer between the electrodes. During discharge, lithium ions move from the anode to the cathode, generating electrical energy, while the process is reversed during charging.

One of the main advantages of lithium-ion batteries is their high energy-to-weight ratio, which enables compact and lightweight power storage solutions. They also have low self-discharge rates, meaning they retain their charge for extended periods when not in use. Additionally, modern lithium-ion batteries are designed with protection circuits to prevent overcharging, deep discharge, and overheating, improving safety and extending battery life.

Due to their versatility and reliability, lithium-ion batteries are commonly used in smartphones, laptops, power tools, medical devices, and renewable energy storage systems. They are also integral to electric vehicles (EVs) and smart grid applications, where their ability to deliver high power output and fast charging capabilities is crucial. However, proper handling and storage are essential, as lithium-ion batteries can be sensitive to extreme temperatures and physical damage, which may lead to thermal runaway or degradation over time.

3.2.7 HW-105 5V Booster

Figure 3.8: HW-105 5V Booster

Figure 3.8 shows a 5V Booster Module, HW-105. It is a DC-DC step-up (boost) converter designed to increase a lower input voltage to a stable 5V output. It is commonly used in battery-powered applications where a consistent 5V supply is needed but the available voltage from the power source is lower, such as in lithium-ion batteries, AA battery packs, or solar-powered circuits. This module ensures efficient voltage regulation, making it ideal for powering microcontrollers, sensors, wireless communication modules, and portable electronic devices.

The HW-105 module operates based on a boost converter topology, which uses an inductor, a switch (typically a MOSFET), a diode, and a capacitor to step up the input voltage. When power is applied, the circuit stores energy in the inductor during the "on" phase and releases it at a higher voltage during the "off" phase, maintaining a stable 5V output. This switching regulator design provides high efficiency, reducing power loss compared to traditional linear regulators.

One of the key features of the HW-105 booster module is its wide input voltage range, typically supporting 1V to 5V input, depending on the specific variant. This allows it to work effectively with low-voltage power sources, such as single-cell lithium-ion batteries (3.7V), NiMH batteries (1.2V), or even weak power supplies. It also includes built-in overcurrent and short-circuit protection, ensuring safe operation even under fluctuating loads.

Due to its compact size, low power consumption, and ease of integration, the HW-105 5V booster module is widely used in wearable electronics, IoT devices, DIY electronics projects, and emergency power solutions. Its ability to efficiently convert low voltages to a stable 5V makes it an essential component in battery-powered embedded systems and portable applications.

3.3 Design

3.3.1 NodeMCU (ESP8266)

NodeMCU has got in-built WiFi capability which enables real-time data transmission.it provides with multiple input output pins which can be used to connect the alcohol sensors, IR sensors and GPS module.Also its power efficiency makes it a best choice for this project.

3.3.2 Alcohol Sensor (MQ3)

Since it has got high sensitivity it can easily detect the presence of alcohol from the driver's breath.It provides a analog output which make it suitable to read using NodeMCU.The cost effectiveness also plays a huge role in making it the best option for the system.

3.3.3 Infrared Sensors

IR sensors satisfy the requirement of counting the passengers in the bus without the involvement of physical interaction since it is a non-contact sensor.it is accurate and also power effective.

3.3.4 NEO-6M GPS Tracker

Signal availability is a important constraint in case of implementing GPS in the bus.But NEO-6M has the advantage of working even in the regions having less signal strength.also it supports serial communication making it easier to interface with the NodeMCU.Its accuracy also adds to the above factors.

3.3.5 Lithium-Ion Battery

Absence of frequency recharging while charging NodeMCU and other components is one of the factor which contributes to its selection.longer lifecycle and high energy density also adds to its advantages.

3.4 Circuit Diagram Layout

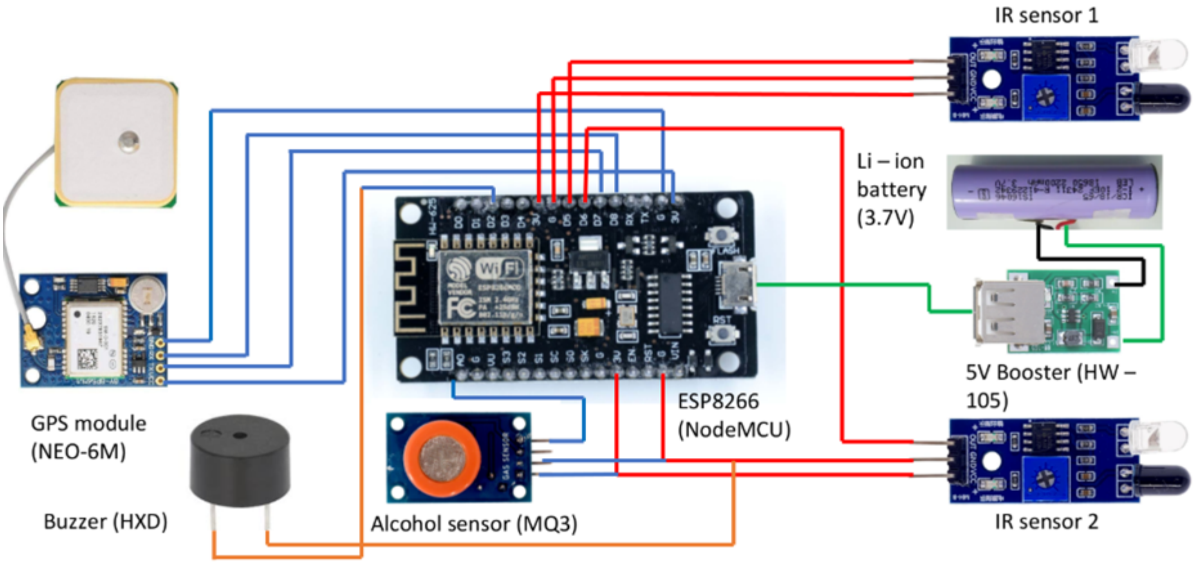


Figure 3.9: Circuit diagram of the Smart Bus Monitoring System

Figure 3.9 shows the circuit diagram of the Smart Bus Monitoring System. The VCC of both IR sensors are connected to the 3V pin of NodeMCU and the grounds of the IR sensors are connected to the ground of NodeMCU. The output pin of first IR sensor is connected to D5 and the output of second IR is connected to D6 of the microcontroller.The transmitter and receiver of the GPS module are connected to D7 and D8 of the microcontroller respectively.Similarly the VCC and GND pins of the GPS module are connected to the 3V and GND pins of the microcontroller respectively.Analog out of the alcohol sensor is given to the Analog pin of the NodeMCU and the VCC and GND pins of the sensor are connected to 3V and GND of the microcontroller just like the IR sensor and GPS module.The positive of the buzzer is connected to D2 of the microcontroller and the negative is grounded.The NodeMCU is powered using Lithium Ion battery connected through a 5V booster.

3.5 Estimation of the Total Product Cost

COMPONENT	NUMBER OF UNITS	PRICE (in Rs)
ESP8266	1	400
GPS NEO-6M	1	500
ALCOHOL SENSOR (MQ3)	1	200
IR SENSOR	2	90 (45 X 2)
BUZZER	1	20
LITHIUM BATTERY	1	50
BATTERY CASE	1	10
BATTERY CHARGER	1	120
BATTERY CONNECTOR	1	60
PCB BOARD	1	60
WIRES	2 meters	60
TOTAL COST		**1540**

Table 3.1: Total Component Cost

Table 3.1 shows an estimate of the total cost of the Smart Bus Monitoring System.

The methodology is a complete integration of passenger counting , real time location updating and alcohol sensing and buzzer system it a single unit of NodeMCU. The passenger counting can be efficiently implemented using IR sensors arranged in bidirectional manner. Alcohol sensing is efficiently accomplished using highly sensitive MQ3 alcohol sensor. Real time location is updated with the help of NEO-6M GPS Module. Efficiency, scalability and compatibility are the guiding principles behind the selection of the components used. Scalability and cost effectiveness are taken into consideration while designing the above methodology.

Chapter 4

Results and Discussions

This chapter presents the results obtained from the development and implementation of the Bus Monitoring System and discusses the key outcomes, features, and challenges encountered throughout the process. The section highlights the final working status of the prototype, elaborates on the individual functionalities achieved, and reflects on how closely the system aligns with the original objectives. The chapter also includes photographic documentation of the demo, a breakdown of the difficulties faced during the course of the project, and a clear division of work among the team members.

4.1 Final Working Model status

The final working model of the bus monitoring system was successfully developed as a fully functional prototype that demonstrated real-time tracking, passenger monitoring, and safety enforcement using IoT technology. The core objective of the system was to provide users with access to the live location of buses along with the current number of passengers onboard, through a simple and accessible web interface. This was accomplished through the seamless integration of both hardware and software components. Figure 4.1 shows the model we have developed to show the working demo of the Smart Bus Monitoring System while Figure 4.2 shows the internal circuitry of Smart Bus Monitoring System.

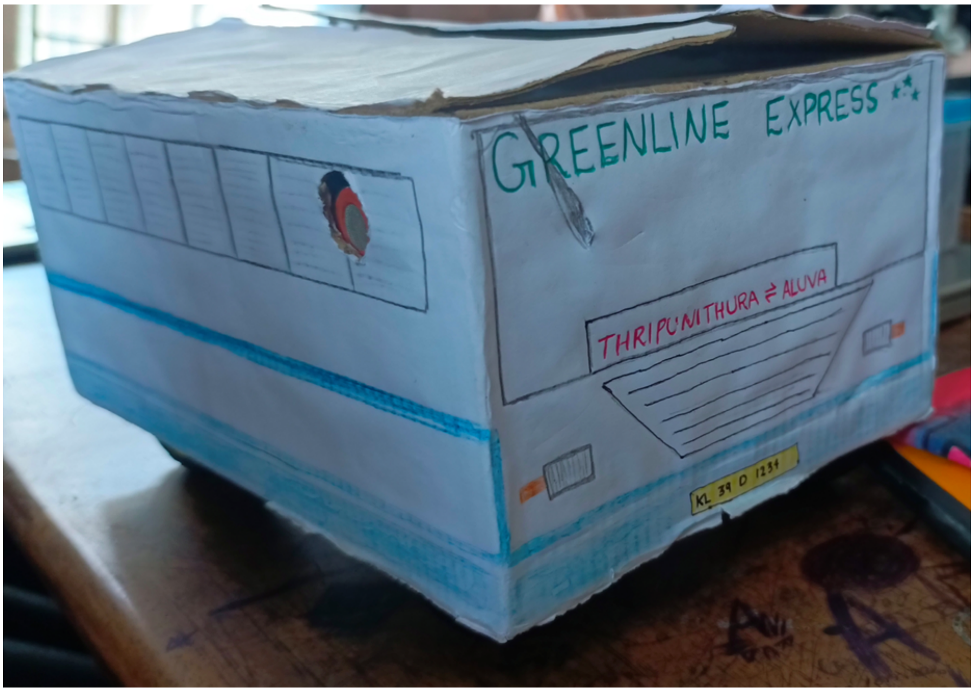


Figure 4.1: Model of the Smart Bus Monitoring System

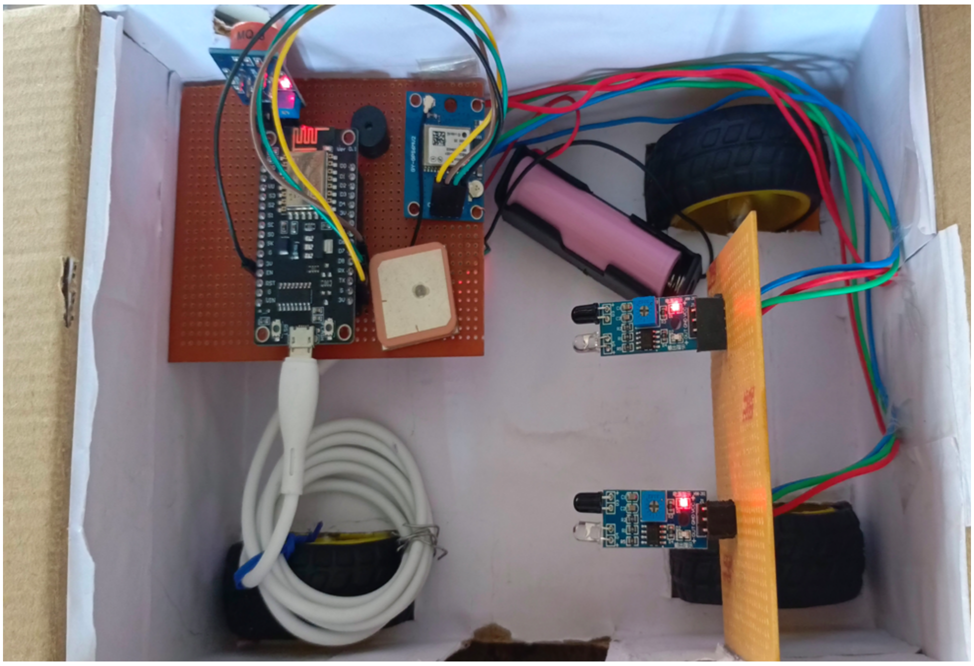


Figure 4.2: Internal Circuitry of the Smart Bus Monitoring System

On the hardware side, the system utilized the ESP8266 microcontroller, selected for its built-in Wi-Fi capabilities and compatibility with cloud-based applications. The GPS6MV2 module was interfaced with the ESP8266 to obtain accurate location coordinates, which were periodically sent to the Firebase Realtime Database. In addition to location tracking, a method to monitor passenger count was included using IR sensors or other suitable counting mechanisms. This value was also transmitted to Firebase to be displayed on the web interface. Figure 4.3 shows the bi-directional counting using IR sensors and the simulataneous updation of the webpage

A key enhancement to the model was the integration of an alcohol detection feature for driver monitoring. An alcohol sensor was deployed within the bus cabin to detect the presence of alcohol vapors. If the sensor reading exceeded a predefined threshold, a buzzer was automatically triggered to alert surrounding personnel of possible intoxication. This feature added a critical safety component to the system, ensuring that vehicle operation is not compromised due to driver impairment. The successful deployment and testing of this module demonstrated the system's potential in promoting safer public transportation. Figure 4.4 shows the alcohol detector placed at the driver's side which will continuously monitor the driver's sobriety.

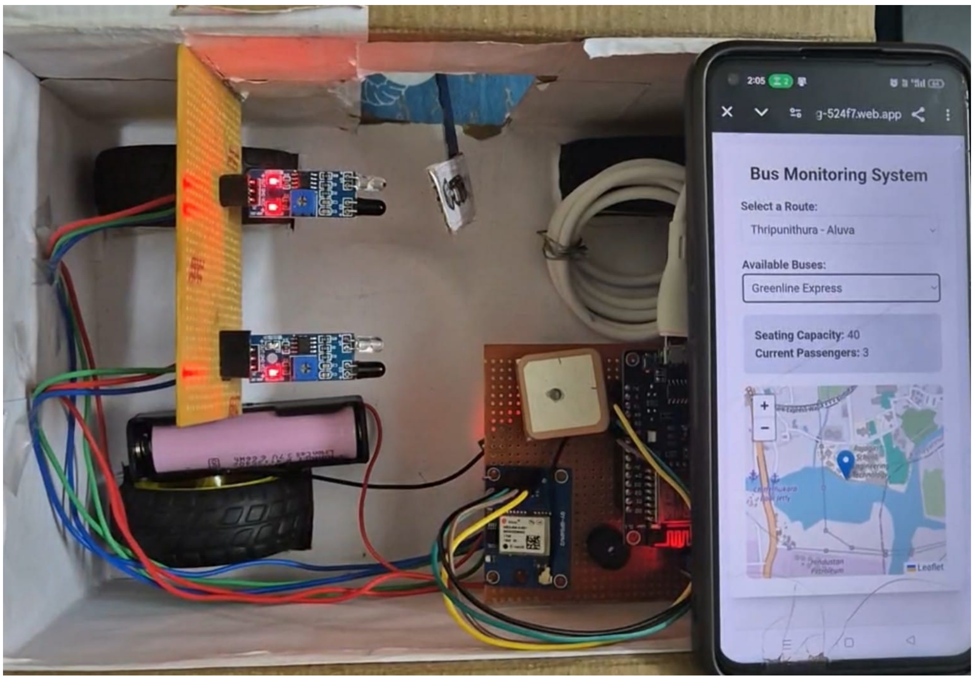


Figure 4.3: IR sensor Bi-directional Counting and Webpage updation

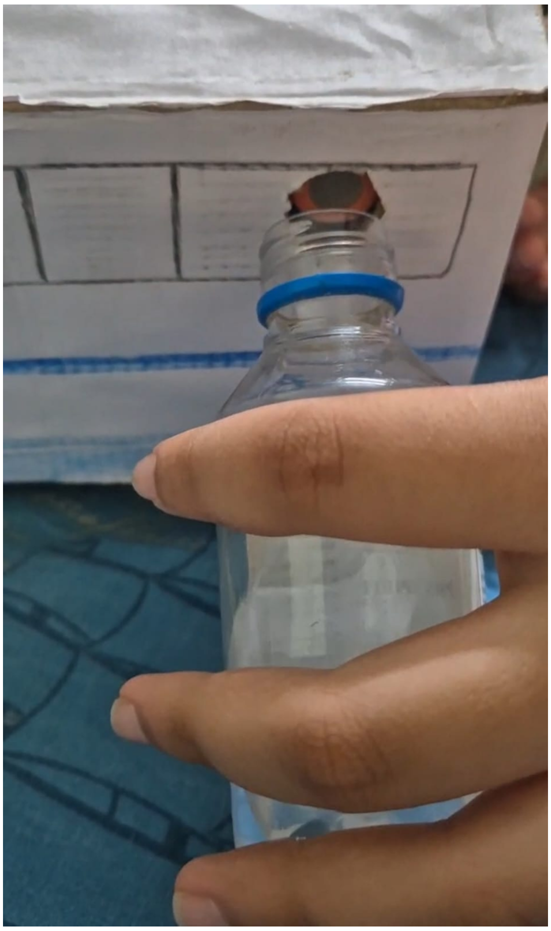


Figure 4.4: Alcohol detection at Driver's Side

For the front end, a responsive web application was developed using HTML. When users scan the QR code placed at the bus stop, they are redirected to a webpage with a dashboard where they can select from a list of predefined routes and available buses. Figure 4.5 shows the scanning of QR code placed at the bus stop. They will be able to access the website using this QR code. Upon selecting a bus, users are presented with its live location displayed on an interactive OpenStreetMap interface. Leaflet.js, an open-source JavaScript library for interactive maps, was used to dynamically display the position of buses. Alongside the map, the current passenger count was shown, helping users decide which bus to board based on crowd levels. Figure 4.6 shows the webpage layout in the user's phone when route and bus are selected.

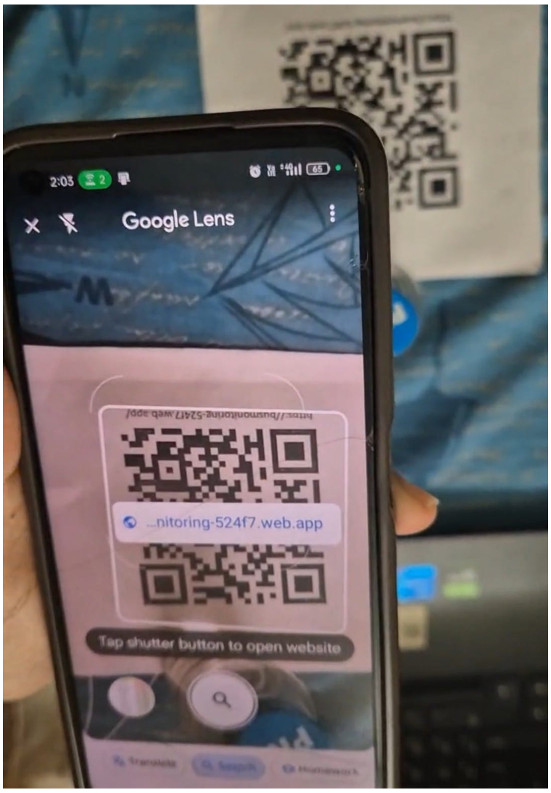


Figure 4.5: Scanning of QR Code placed at Bus Stop

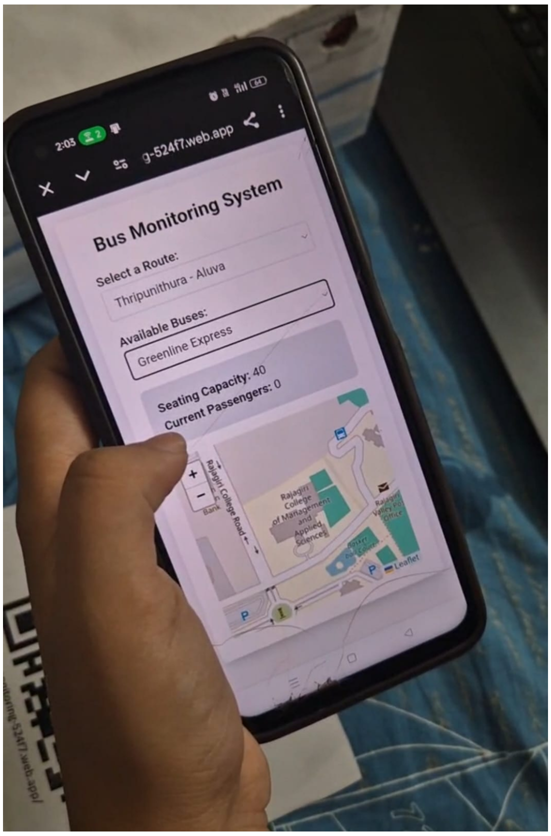


Figure 4.6: Webpage:Display of live location of the bus

One of the initially planned features was the display of a timestamp indicating the last time the data was updated. Although the system was capable of pushing live data to Firebase, this feature was not implemented in the final version due to time limitations and prioritization of other functionalities. Despite this, the core goals of real-time location tracking, passenger monitoring, and safety enhancement were successfully achieved. The project demonstrates a practical application of IoT in public transportation, showcasing how affordable hardware and cloud-based technologies can be combined to build scalable, informative, and safety-conscious monitoring solutions.

4.2 Challenges faced during the Project Work

Throughout the development of the bus monitoring system, the team encountered several technical and practical challenges. One of the primary issues involved the GPS6MV2 module, which initially faced difficulties in acquiring a stable satellite lock. This was particularly problematic in indoor environments or areas with obstructions, which delayed the process of obtaining accurate location data. To mitigate this, testing was moved to open areas where the module could function more reliably, though it remained a time-consuming part of the project.

Another significant challenge was ensuring seamless and real-time communication between the ESP8266 microcontroller and Firebase. Wi-Fi connectivity issues sometimes caused interruptions in data updates, requiring the team to implement retry logic and verify data consistency in the database. Similarly, during the integration of the passenger counting mechanism, fine-tuning the sensor placement and sensitivity proved challenging, as false triggers could skew the count, especially when multiple people entered or exited the bus simultaneously.

Incorporating OpenStreetMap into the front-end interface using Leaflet.js was also a learning experience. Real-time updating of bus location markers without using WebSocket protocols required creative solutions using polling and efficient JavaScript logic. Designing a user-friendly, mobile-responsive interface with Tailwind CSS also required multiple iterations and careful layout planning to accommodate different screen sizes while maintaining clarity and accessibility.

An additional challenge arose during the integration of the alcohol sensor and buzzer system. While the sensor was successfully calibrated to detect alcohol presence above a threshold, the sensitivity of the module had to be carefully adjusted to avoid false alarms. Ensuring timely activation of the buzzer and managing the power supply without affecting the performance of other modules required hardware-level troubleshooting. Debugging and coordinating this feature with the rest of the ESP8266-based system introduced some delay but ultimately added a critical safety layer to the project.

Lastly, one of the initially planned features:a timestamp to display the last data update could not be implemented due to time constraints. While Firebase supports timestamp data, implementing and syncing it correctly on both the database and front-end side was deprioritized in favor of completing the core features. Despite these challenges, the team was able to successfully deploy a working model that fulfilled the main objectives of the project, with added emphasis on safety through alcohol detection.

4.3 Work Division

Kalaiyarasi worked on setting up and testing the GPS module. She also handled all the soldering work for the hardware. In addition, she helped collect the required components, supported the integration of the alcohol and IR sensors, and contributed to writing the report, making the presentation, and building the model.She was also involved in setting up website software of the project.

Aleena also helped with GPS module testing and She was involved in collecting components, doing simulations, writing the report, preparing the presentation, and assembling the final model.She was also involved in setting up website software of the project.

Architha took charge of integrating the alcohol sensor and IR sensor into the system. She helped gather components and worked on the simulation part. She also contributed to the report writing, presentation, and building the model.

Angelina worked along with Architha on integrating the alcohol and IR sensors. She helped collect components, assisted with simulations, and took part in preparing the report, making the presentation, and assembling the final working model.

Chapter 5

Conclusion & Future Scope

The Bus Monitoring System developed as part of this project successfully demonstrated the integration of IoT technology into public transportation management. The system achieved its primary objectives by enabling real-time tracking of buses, displaying live passenger count, and incorporating a safety feature through alcohol detection. The combination of ESP8266, GPS6MV2 module, IR sensors, and an alcohol sensor, along with Firebase and an interactive web interface, created a cost-effective and functional prototype suitable for practical use.

Despite certain limitations such as the absence of a timestamp feature, the prototype provided valuable insights into building smart transport solutions. The successful implementation of core features like location updates, passenger monitoring, and safety alerts proves the viability of IoT-based systems in enhancing commuter experience and safety in public transport.

The bus monitoring system can be further enhanced by adding a timestamp feature to indicate the last update time for location and passenger data. Real-time communication can be improved using WebSocket technology instead of regular polling. The accuracy of passenger counting can be increased by integrating camera-based detection or machine learning algorithms. A dedicated mobile app and admin panel could also be developed for better usability and fleet management. Additionally, features like push notifications and power optimization would make the system more reliable and scalable for real-world deployment.

References

[1] J. I. Sojol, N. F. Piya, S. Sadman, and T. Motahar, "Smart bus: An automated passenger counting system," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 3169–3177, 2018. [Online]. Available: https://www.researchgate.net/publication/323027620_Smart_Bus_An_Automated_Passenger_Counting_System

[2] V. Nair, A. Pawar, D. L. Tidke, V. Pagar, and N. Wani, "Online bus tracking and ticketing system," *MVP Journal of Engineering Sciences*, vol. 1, no. 1, 2018. [Online]. Available: https://www.researchgate.net/publication/354523203_Online_Bus_Tracking_and_Ticketing_System

[3] M. A. Hannan, A. M. Mustapha, A. Hussain, and H. Basri, "Intelligent bus monitoring and management system," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, 2012, pp. 24–26. [Online]. Available: https://www.iaeng.org/publication/WCECS2012/WCECS2012_pp1049-1054.pdf

[4] J. J. Astrain, F. Falcone, A. J. Lopez-Martin, P. Sanchis, J. Villadangos, and I. R. Matias, "Monitoring of electric buses within an urban smart city environment," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 11 364–11 372, 2021. [Online]. Available: https://academica-e.unavarra.es/bitstreams/1b95d37b-b7d4-472f-8215-18922a42fba6/download

Appendix A: Presentation

SMART BUS MONITORING SYSTEM

PROJECT GUIDE: Dr. Jisa David

MEMBERS:
A Kalaiyarasi (U2201001)
Aleena M B (U2201020)
Angelina Mary Giby (U2201034)
Architha B (U2201051)

CONTENTS

2

INTRODUCTION

3

- Public buses in cities across India are often overcrowded.
- Lack of awareness about bus routes and real-time locations leads to passengers boarding overcrowded buses.
- Overcrowding increases the risk of accidents and makes commuting uncomfortable.
- Provide a simple and effective solution for comfortable and safe bus travel.
- Helps passengers make informed decisions and reduces overcrowding.

OBJECTIVES AND KEY FEATURES

4

Objectives:
- Objective 1: Real-Time GPS Tracking of the Bus
- Objective 2: Passenger Counting Using IR Sensors
- Objective 3: Detection of Presence of Alcohol at Driver's Side

Key Features:
- QR codes at bus stations to access real-time bus information
- A website showing:
  - List of buses on the chosen route
  - Current location of buses
  - Passenger count on each bus

LITERATURE SURVEY

Author	Paper Title	Publication	Technology Used	Limitation
Sojol, J. I., Piya, N. F., Sadman, S., & Motahar, T.	Smart bus: An automated passenger counting system (2018)	International Journal of Pure and Applied Mathematics	Microcontroller, Bluetooth, Potentiometer, Pressure pad, Touch display	Passengers cannot view the number of vacant seats on board
Nair, V., Pawar, A., Tidke, D. L., Pagar, V., & Wani, N.	Online bus tracking and ticketing system (2018)	MVP Journal of Engineering Sciences	GPS, Mobile, Processor, Cloud	Seating capacity is not monitored
Hannan, M. A., Mustapha, A. M., Hussain, A., & Basri, H.	Intelligent bus monitoring and management system (2012)	Proceedings of the World Congress on Engineering and Computer Science	RFID, GPS, GPRS, GIS	Does not allow the passenger to view buses on their route

LITERATURE SURVEY

Author	Paper Title	Publication	Technology Used	Limitation
Darsena, D., Gelli, G., Iudice, I., & Verde, F.	Sensing technologies for crowd management, adaptation, and information dissemination in public transportation systems: A review (2022)	IEEE Sensors Journal	IoT, RFID, Wi-Fi, IR Sensors, Pressure Sensors, Camera	The location of the bus cannot be viewed
Astrain, J. J., Falcone, F., Lopez-Martin, A. J., Sanchis, P., Villadangos, J., & Matias, I. R.	Monitoring of electric buses within an urban smart city environment (2020)	IEEE Sensors Journal	LPWAN, Cloud, ML, Sensors	Seating capacity is not monitored

Table 1 shows Literature survey of Smart Bus Monitoring System

OVERVIEW

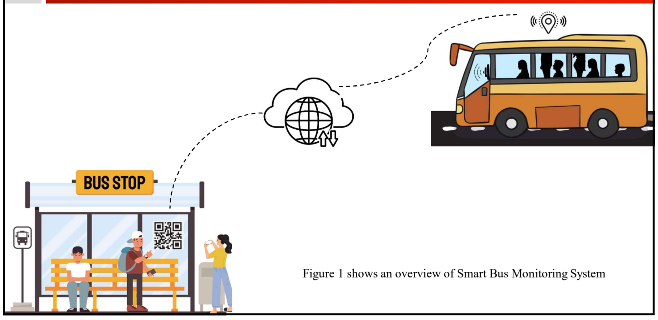


Figure 1 shows an overview of Smart Bus Monitoring System

BLOCK DIAGRAM OF SMART BUS SYSTEM

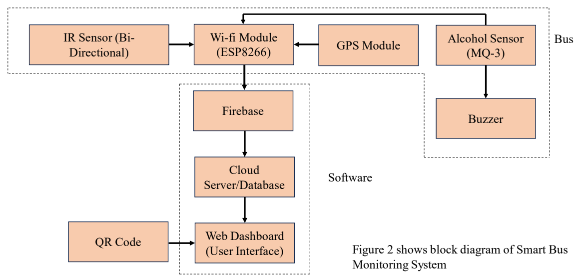


Figure 2 shows block diagram of Smart Bus Monitoring System

METHODOLOGY

1. Hardware Data Acquisition Flow (Sensors + GPS to Firebase)

Step 1: Passenger Detection via IR Sensors

- Two IR sensors are placed at the bus entry/exit.
- The microcontroller (ESP8266) reads digital input from these sensors.
- Based on the order of triggering:
  - Entry → Increments the passenger count.
  - Exit → Decrements the passenger count (ensuring it doesn't go below zero).

METHODOLOGY

Step 2: Location Tracking via GPS

- A GPS module (NEO-6M) sends location data via **UART**.
- The microcontroller parses latitude and longitude using the TinyGPS++ library.
- If GPS is unavailable, only passenger data is updated.

Step 3: Detection of Alcohol and Ringing of Buzzer

- A threshold value is detected upon powering up the system
- If alcohol is present, the value produced will be above threshold
- If so, the buzzer rings

METHODOLOGY

Step 4: Firebase Integration

- The microcontroller pushes data to Firebase Realtime Database:
  - Passengers → Current count
  - Location/Latitude & Location/Longitude → Live GPS coordinates
- Data is updated every 5 seconds (configurable delay).

METHODOLOGY

2. Web Application Flow (QR Code to Live Bus Info)

Step 1: QR Code Scan

- Users scan a QR code (placed at bus stops).
- The QR redirects to the hosted Web Application URL.

Step 2: Route Selection

- The web app prompts users to select from a list of predefined routes (e.g., Thripunithura to Aluva).

Step 3: Bus Selection

- Once a route is selected, all buses on that route are listed.
- User selects a specific bus (e.g., Greenline Express).

METHODOLOGY

Step 4: Display of Live Information

- The web app connects to Firebase using JavaScript SDK.
- It reads:
  - Live passenger count
  - Current GPS location
- Data is shown on:
  - Live Map (e.g., OpenStreetMap or Leaflet)
  - Textual Info (e.g., "24 passengers onboard")

FLOWCHART OF SMART BUS SYSTEM

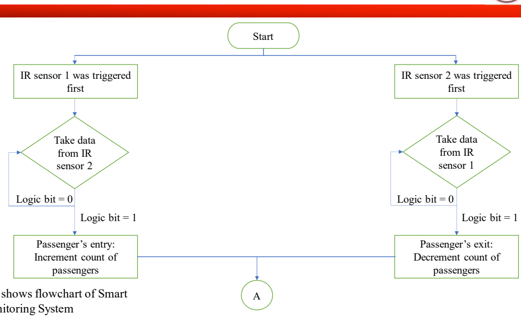


Figure 3 shows flowchart of Smart Bus Monitoring System

FLOWCHART OF SMART BUS SYSTEM

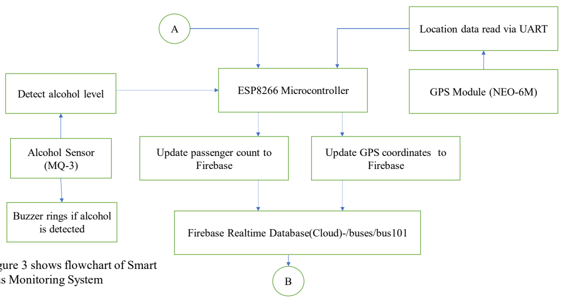


Figure 3 shows flowchart of Smart Bus Monitoring System

FLOWCHART OF SMART BUS SYSTEM

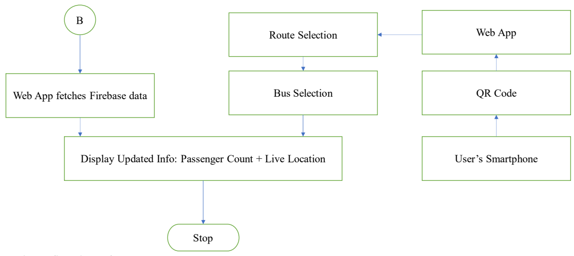


Figure 3 shows flowchart of Smart Bus Monitoring System

APPLICATIONS

Applications	Expected consumers
Real-Time Location Tracking and Bus occupancy information	Daily commuters
Parental Notifications for school buses	School Management
Emergency Response	Public
Route Optimization and Schedule Adherence	Public transport authorities

Table 2 shows applications of Smart Bus Monitoring System

LIST OF COMPONENTS

1. Connectors
2. Wire
3. PCB board
4. Connector cable
5. Lithium Ion Battery
6. Battery case
7. Battery to USB Connector
8. IR sensor (2)
9. Wi-Fi module: New NodeMCU v3(ESP8266)
10. GPS module: GY-GPS6MV2(NEO-6M)
11. Alcohol Sensor (MQ3)
12. Buzzer

SIMULATIONS

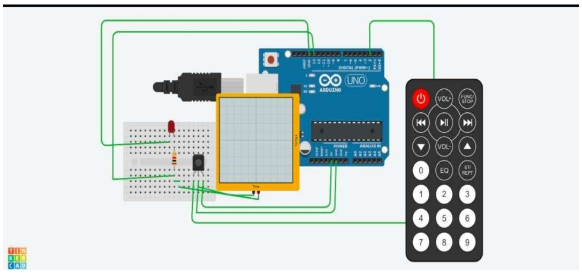


Figure 4 shows simulation of IR sensor data using Arduino

SIMULATIONS

Figure 5 shows simulation of GPS data using Arduino

SIMULATIONS

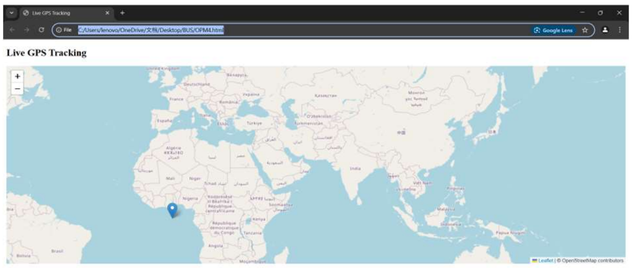
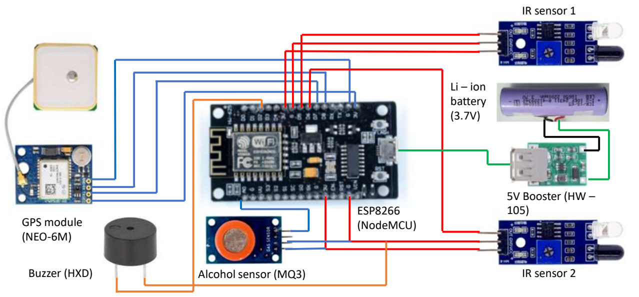


Figure 6 shows Simulation of Website Prototype (HTML File)

CIRCUIT DIAGRAM OF SMART BUS SYSTEM



HARDWARE

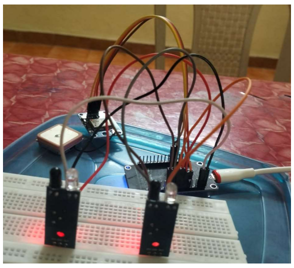
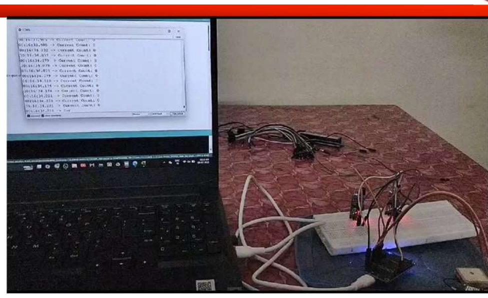


Figure 7 shows hardware integration of IR Sensor and GPS Module (Phase 1)

HARDWARE



Video 1 shows Hardware Integration of IR Sensor and GPS Module Working (Phase 1)

11-04-2025

SOFTWARE

25

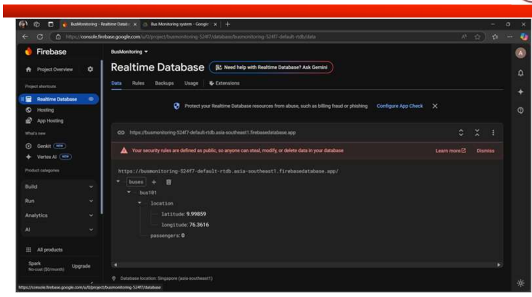


Figure 8 shows Firebase Updation

SOFTWARE

26

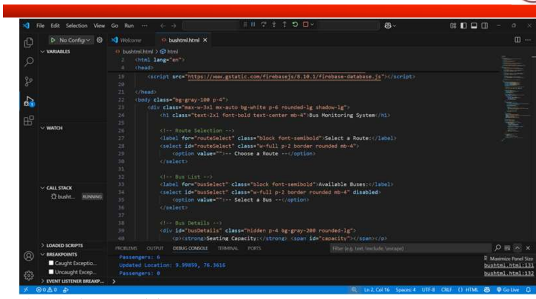


Figure 9 shows data in HTML updation

SOFTWARE

27

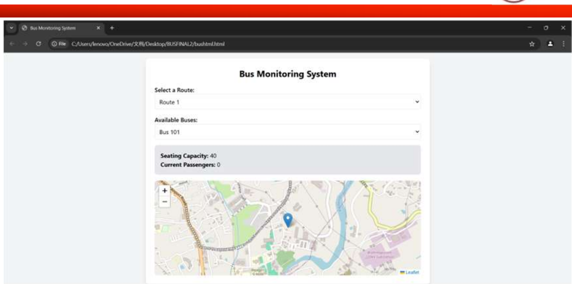
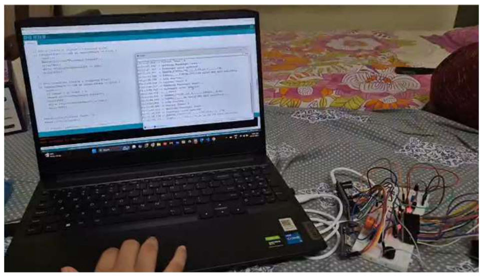


Figure 10 shows Web App updation

SOFTWARE

28



Video 2 shows Software Integration of IR Sensor and GPS Module Working (Phase 2)

SOFTWARE

https://busmonitoring-524f7.web.app/

Figure 11 shows QR for Web App

STATUS OF THE PROJECT

1. Components were identified
2. Specifications were decided
3. Components were acquired
4. Work was divided into two units : GPS Module and Sensor Module
5. Working condition of components were verified
6. Basic circuit diagram was decided
7. Simulation was done using Arduino for understanding purposes
8. Program codes were formulated separately for different components

STATUS OF THE PROJECT

9. Verification & Integration of both units was done using ESP8266
10. Software integration of both units was done successfully onto the website
11. Web App was successfully hosted
12. QR Code was generated for the Web App
13. Alcohol sensor and buzzer were configured
14. A basic model for displaying the circuit was made
15. The components were soldered onto a PCB
16. The circuit was arranged inside the model
17. The model was made wireless using a lithium ion battery

PROBLEMS IDENTIFIED & CHANGES MADE

- Placing IR sensor will force us to impose the constraint that entry and exit should be through separate doors which is practically not efficient. Thus, we move on to the decision of using bidirectional sensor at both doors.
- The use of both Arduino Uno Board and NodeMCU will not be function efficient. Hence, Arduino board was avoided and the two units are to be integrated onto the NodeMCU

MODEL OF THE PROJECT

33

Figure 12 shows outer view of model

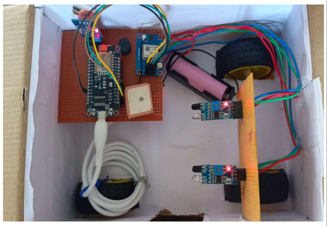
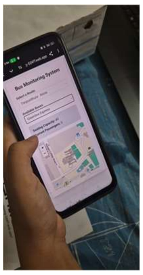


Figure 13 shows inner view of model

WORKING DEMO OF THE PROJECT

34



Video 3 shows Complete Demo of Project

REFERENCES

35

1. J. I. Sojol, N. F. Piya, S. Sadman, and T. Motahar, "Smart Bus: An Automated Passenger Counting System," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 3169–3177, 2018. [Online]. Available: https://www.researchgate.net/publication/323027620_Smart_Bus_An_Automated_Passenger_Counting_System
2. V. Nair, A. Pawar, D. L. Tidke, V. Pagar, and N. Wani, "Online Bus Tracking and Ticketing System," *MVP Journal of Engineering Sciences*, vol. 1, no. 1, 2018. [Online]. Available: https://www.researchgate.net/publication/354523203_Online_Bus_Tracking_and_Ticketing_System

REFERENCES

36

3. D. Darsena, G. Gelli, I. Iudice, and F. Verde, "Sensing Technologies for Crowd Management, Adaptation, and Information Dissemination in Public Transportation Systems: A Review," *IEEE Sensors Journal*, vol. 23, no. 1, pp. 68–87, 2022. [Online]. Available: https://arxiv.org/abs/2009.12619
4. J. J. Astrain et al., "Monitoring of Electric Buses Within an Urban Smart City Environment," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 11364–11372, 2021. [Online]. Available: https://academica-e.unavarra.es/bitstreams/1b95d37b-b7d4-472f-8215-18922a42fba6/download
5. M. A. Hannan, A. M. Mustapha, A. Hussain, and H. Basri, "Intelligent Bus Monitoring and Management System," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, pp. 24–26, Oct. 2012. [Online]. Availablehttps://www.iaeng.org/publication/WCECS2012/WCECS2012_pp1049-1054.pdf

Appendix B:
Programs

B.1 ESP8266 Code

Listing 5.1: ESP8266 Code

```
#include <FB_Const.h>
#include <FB_Error.h>
#include <FB_Network.h>
#include <FB_Utils.h>
#include <Firebase.h>
#include <FirebaseESP8266.h>
#include <FirebaseFS.h>
#include <MB_File.h>
#include <LiquidCrystal.h>
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <SD.h>
#include <TinyGPSPlus.h>
#define MQ3_ANALOG_PIN A0          // Analog pin on ESP8266
#define BUZZER_PIN      4          // D2 on NodeMCU = GPIO4

const int THRESHOLD = 400;         // Analog value above which buzzer
    will sound
// GPS Pins
#define RXPin D7
#define TXPin D8
#define GPSBaud 9600

// IR sensors
int sensor1 = D5;  // Sensor 1
int sensor2 = D6;  // Sensor 2

// WiFi Credentials
```

```

  // Connect to WiFi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);
  }
  Serial.println("\nConnected to WiFi!");

  // Firebase Configuration
  config.host = FIREBASE_HOST;
  config.api_key = FIREBASE_API_KEY;
  config.signer.tokens.legacy_token = FIREBASE_AUTH;

  // Initialize Firebase
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
  Serial.println("Firebase Connected!");

}

void loop() {
  Serial.println("Loop started...");
  int analogValue = analogRead(MQ3_ANALOG_PIN);  // Read sensor analog output

  Serial.print("Alcohol Sensor Value: ");
  Serial.println(analogValue);

  if (analogValue > THRESHOLD) {
    digitalWrite(BUZZER_PIN, HIGH);  // Alcohol detected   buzzer ON
  } else {
```

```
    digitalWrite(BUZZER_PIN, LOW);    // No alcohol    buzzer OFF
  }

  // Read IR sensor states
  int sensor1State = digitalRead(sensor1);
  int sensor2State = digitalRead(sensor2);

  // Entry Condition (Sensor 1 triggered first)
  if (sensor1State == LOW && sensor2State == HIGH) {
    count++;
    Serial.println("Passenger Entered");
    delay(600);
    while (digitalRead(sensor2) == LOW);
    delay(600);
  }

  // Exit Condition (Sensor 2 triggered first)
  if (sensor2State == LOW && sensor1State == HIGH) {
    count--;
    if (count < 0) count = 0;
    Serial.println("Passenger Exited");
    delay(600);
    while (digitalRead(sensor1) == LOW);
    delay(600);
  }

  Serial.print("Current Count: ");
  Serial.println(count);

  //Update passenger count to Firebase
  if (Firebase.ready()) {
    Serial.println("Updating Passenger Count...");
    if (Firebase.setInt(firebaseData, "/buses/bus101/passengers",
   count)) {
```

```
      Serial.println("Passenger count updated!");
    } else {
      Serial.println("Firebase Error (Passenger): " + firebaseData.errorReason());
    }
  } else {
    Serial.println("Firebase not ready for passengers update!");
  }

  //Check if GPS data is available before sending it
  bool gpsUpdated = false;
  while (gpsSerial.available() > 0) {
    char c = gpsSerial.read();
    Serial.write(c);  // Print raw GPS data for debugging
    gps.encode(c);
    gpsUpdated = true;
  }

  if (gpsUpdated && gps.location.isValid()) {
      float latitude = gps.location.lat();
      float longitude = gps.location.lng();
      Serial.print("Latitude: ");
      Serial.print(latitude, 6);
      Serial.print(" | Longitude: ");
      Serial.println(longitude, 6);

      //Update GPS coordinates to Firebase separately
      if (Firebase.ready()) {
        Serial.println("Updating GPS location...");
        if (Firebase.setFloat(firebaseData, "/buses/bus101/location/latitude", latitude) &&
            Firebase.setFloat(firebaseData, "/buses/bus101/location/longitude", longitude)) {
              Serial.println("GPS location updated!");
```

```
        } else {
          Serial.println("Firebase Error (GPS): " + firebaseData.errorReason());
        }
      } else {
        Serial.println("Firebase not ready for GPS update!");
      }
  } else {
    Serial.println("No valid GPS data available...");
  }
  delay(500); // Wait before next update
}

void sendDummyBusData() {
  if (Firebase.ready()) {
    // SMART RIDE 2 - bus102
    Firebase.setFloat(firebaseData, "/buses/bus102/location/latitude", 10.030657);
    Firebase.setFloat(firebaseData, "/buses/bus102/location/longitude", 76.336030);
    Firebase.setInt(firebaseData, "/buses/bus102/passengers", 44);
    // BLUE BIRD - bus201
    Firebase.setFloat(firebaseData, "/buses/bus201/location/latitude",10.018991);
    Firebase.setFloat(firebaseData, "/buses/bus201/location/longitude", 76.343589);
    Firebase.setInt(firebaseData, "/buses/bus201/passengers", 14);
    Serial.println("Dummy data for bus102 and bus210 sent to Firebase.");
  } else {
    Serial.println("Firebase not ready!");
  }
}
```

B.2 HTML Code of Website

Listing 5.2: HTML Code

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale
=1.0">
    <title>Bus Monitoring System</title>

    <!-- Tailwind CSS -->
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="stylesheet" href="busstyle.css"/>

    <!-- Leaflet CSS and JS -->
    <link rel="stylesheet" href="https://unpkg.com/leaflet/dist/
leaflet.css" />
    <script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>

    <!-- Firebase v8 (Correct version for your code) -->
    <script src="https://www.gstatic.com/firebasejs/8.10.1/firebase-
app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.10.1/firebase-
database.js"></script>

</head>
<body class="bg-gray-100 p-4">
    <div class="max-w-3xl mx-auto bg-white p-6 rounded-lg shadow-lg">
        <h1 class="text-2xl font-bold text-center mb-4">Bus
Monitoring System</h1>

```

```
        <!-- Route Selection -->
        <label for="routeSelect" class="block font-semibold">Select a Route:</label>
        <select id="routeSelect" class="w-full p-2 border rounded mb-4">
            <option value="">-- Choose a Route --</option>
        </select>

        <!-- Bus List -->
        <label for="busSelect" class="block font-semibold">Available Buses:</label>
        <select id="busSelect" class="w-full p-2 border rounded mb-4" disabled>
            <option value="">-- Select a Bus --</option>
        </select>

        <!-- Bus Details -->
        <div id="busDetails" class="hidden p-4 bg-gray-200 rounded-lg">
            <p><strong>Seating Capacity:</strong> <span id="capacity"></span></p>
            <p><strong>Current Passengers:</strong> <span id="passengers"></span></p>
        </div>

        <!-- Live Map -->
        <div id="map" class="hidden w-full h-64 mt-4 rounded-lg"></div>
    </div>

    <script>
        document.addEventListener("DOMContentLoaded", () => {
            const routeSelect = document.getElementById("routeSelect");
```

```
        const busSelect = document.getElementById("busSelect");
        const busDetails = document.getElementById("busDetails");
        const capacityEl = document.getElementById("capacity");
        const passengersEl = document.getElementById("passengers");
        const mapDiv = document.getElementById("map");
        let map, marker;

        // Sample route and bus data (to be replaced with dynamic data fetching)
        const routes = {
    "Thripunithura - Aluva": [
        { id: "bus101", name: "Greenline Express", capacity: 40 },
        { id: "bus102", name: "SmartRide 2", capacity: 38 }
    ],
    "Edapally - Kakkanad": [
        { id: "bus201", name: "BlueBird", capacity: 35 }
    ]
};


        // Populate route options
        Object.keys(routes).forEach(route => {
            const option = document.createElement("option");
            option.value = route;
            option.textContent = route;
            routeSelect.appendChild(option);
        });

        routeSelect.addEventListener("change", () => {
            busSelect.innerHTML = "<option value=''>-- Select a Bus --</option>";
            busSelect.disabled = !routeSelect.value;

```

```
            if (routeSelect.value) {
                routes[routeSelect.value].forEach(bus => {
                    const option = document.createElement("option
");
                    option.value = bus.id;
                    option.textContent = bus.name;
                    busSelect.appendChild(option);
                });
            }
        });

        busSelect.addEventListener("change", () => {
            if (busSelect.value) {
                const selectedRoute = routeSelect.value;
                const selectedBus = routes[selectedRoute].find(
bus => bus.id === busSelect.value);

                capacityEl.textContent = selectedBus.capacity;
                busDetails.classList.remove("hidden");
                mapDiv.classList.remove("hidden");

                // Initialize the map if not already done
                if (!map) {
                    map = L.map("map").setView([10, 76], 12); //
Default location
                    L.tileLayer("https://{s}.tile.openstreetmap.
org/{z}/{x}/{y}.png").addTo(map);
                    marker = L.marker([10, 76]).addTo(map);
                }

                // Start listening to Firebase updates
                listenForBusUpdates(busSelect.value);
            }
        });
```

```
                        // Update the map marker
                        updateMap(latitude, longitude);
                    } else {
                        console.log("No location data available");
                    }
                });
            }

            // Function to update the map with new location
            function updateMap(lat, lon) {
                if (marker) {
                    marker.setLatLng([lat, lon]);
                } else {
                    marker = L.marker([lat, lon]).addTo(map);
                }
                map.setView([lat, lon], 15); // Center the map on the new location
            }
        });
    </script>

</body>
</html>
```

Appendix C:
Poster

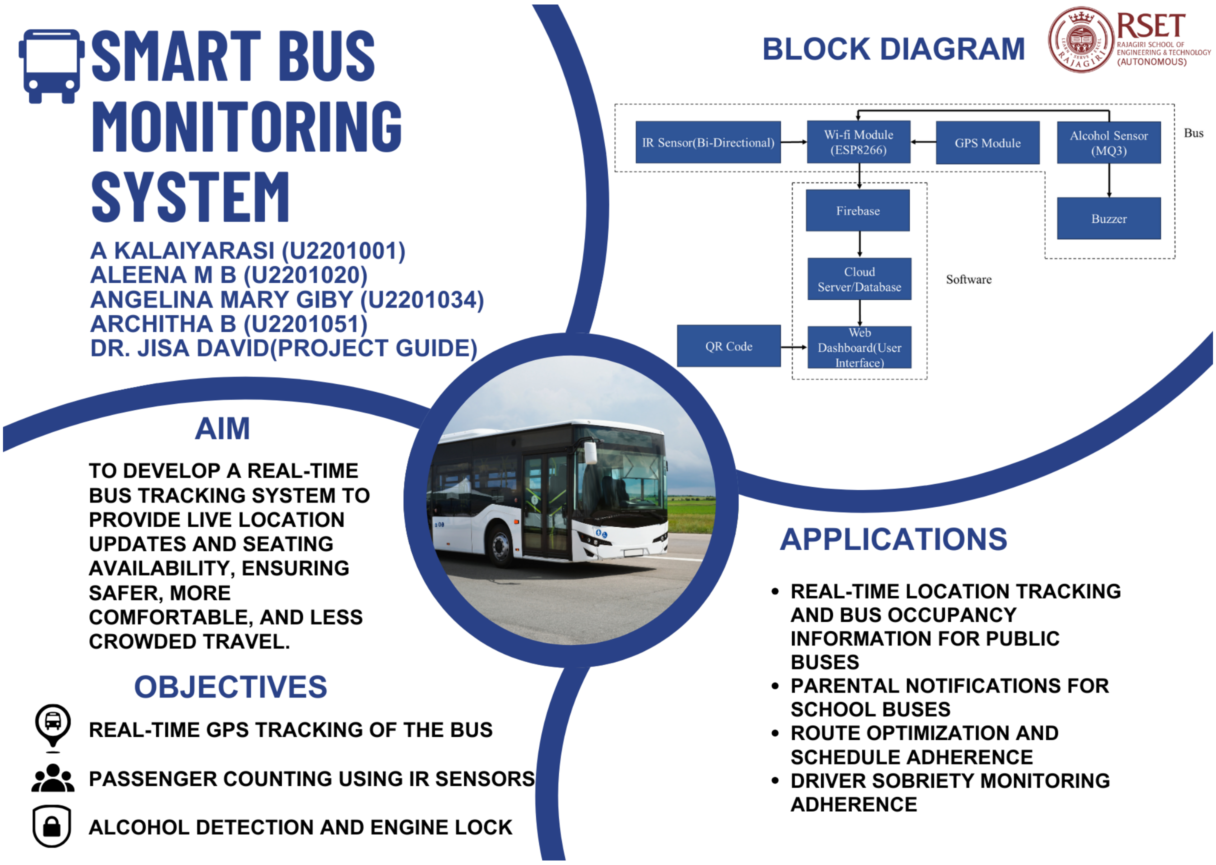


Figure 5.1: Poster of Smart Bus Monitoring System

Appendix D: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To evolve into a Center of Excellence in Electronics Communication Engineering,moulding professionals having Inquisitive,Innovative and Creative minds with sound practical skills who can strive for the betterment of mankind.

Department Mission

To impart state-of-the-art knowledge to students in Electronics Communication Engineering and to inculcate in them a high degree of social consciousness and a sense of human values, thereby enabling them to face challenges with courage and conviction.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

Engineering students will be able to:

PSO1: Demonstrate their skills in designing, implementing and testing analogue and digital electronic circuits, including microprocessor systems, for signal processing, communication, networking, VLSI and embedded systems applications.

PSO2: Apply their knowledge and skills to conduct experiments and develop applications using electronic design automation (EDA) tools.

PSO3: Demonstrate a sense of professional ethics, recognize the importance of continued learning, and be able to carry out their professional and entrepreneurial responsibilities in electronics engineering field giving due consideration to environment protection and sustainability.

Course Outcomes (CO)

After the completion of the course,the student will be able to

Course Outcome 1: Be able to practice acquired knowledge within the selected area of technology for project development.

Course Outcome 2: Identify, discuss and justify the technical aspects and design aspects of the project with a systematic approach.

Course Outcome 3: Reproduce, improve and refine technical aspects for engineering projects.

Course Outcome 4: Work as a team in development of technical projects.

Course Outcome 5: Communicate and report effectively project related activities and findings.

Project Outcomes for Smart Bus Monitoring System(P)

After the completion of the course,the student will be able to

P1: Apply knowledge in embedded systems, IoT, and communication technologies to develop a real-time Smart Bus Monitoring System.

P2: Identify, analyze, and justify the hardware and software design decisions such as GPS integration, Firebase communication, and web interface for efficient system implementation.

P3: Refine and optimize the functionality of subsystems including passenger counting, location tracking, and data synchronization across platforms.

P4: Collaborate effectively as a team to design, test, and deploy the complete Smart Bus Monitoring System.

P5: Document, present, and communicate technical concepts, implementation details, and project findings through reports and presentations.

Appendix E: CO-PO-PSO Mapping

CO - PO Mapping

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
1	3	3	3	2	3	3	2					2
2	3	3	3	2	3	3	2			3	2	2
3	3	3	3	2	3	3	2			3	2	2
4					2			3	3	3	2	2
5					2			3	3	3	2	2

CO - PSO Mapping

CO	PSO 1	PSO 2	PSO 3
1	3	3	2
2	3	3	2
3	3	3	2
4	2	2	3
5	2	2	3

P-PO Mapping

P	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
P1	3	3	3	2	3	1	2					2
P2	2	2	1	2	2	3	2			3	2	2
P3	2	3	3	2	3	3	1			3	1	2
P4					2			3	3	3	2	3
P5					2			3	3	1	2	2

P - PSO Mapping

P	PSO 1	PSO 2	PSO 3
1	2	2	2
2	3	3	2
3	3	3	1
4	2	1	3
5	1	2	3

Justification

Mapping	Justification
P1 - PO1	The project involves designing and implementing an embedded system for smart transportation.
P1 - PO2	Develops a systematic approach for problem-solving using IoT and real-time monitoring.
P1 - PO3	Integrates hardware and software for efficient passenger tracking and alcohol detection.
P1 - PO4	It involves investigating sensor accuracy, GPS reliability, and system efficiency
P1 - PO5	The project involves using modern tools such as NodeMCU, IR sensors, GPS modules, and the MQ-3 alcohol sensor to develop a smart bus monitoring system.
P1 - PO6	Enhances public safety by implementing alcohol detection systems.
P1 - PO7	Promotes sustainability by optimizing transport efficiency and reducing traffic congestion.
P1 - PO12	Encourages ethical responsibility in transportation monitoring systems.
P2 - PO1	Applies fundamental technical skills to justify the selection of sensors and communication modules.
P2 - PO2	Evaluates the effectiveness of IR sensors, GPS modules, and alcohol sensors in real-time tracking.
P2 - PO3	The project requires designing an efficient system for passenger counting, GPS tracking, and alcohol detection. This involves selecting appropriate sensors, integrating hardware with software, and ensuring system reliability
P2 - PO4	Investigating various technical challenges and solving them inorder to increase the efficiency and reliability of the system
P2 - PO5	Uses IoT-based communication and cloud storage for better monitoring and analysis.
P2 - PO6	Adheres to safety regulations by ensuring accurate alcohol detection and alert mechanisms.
P2 - PO7	Integrates sustainable power sources like lithium-ion batteries to enhance system efficiency.
P2 - PO10	Documents technical aspects, making it easier for future enhancements.

Justification

Mapping	Justification
P2 - PO11	Managing hardware costs, power consumption, and scalability aligns with project management principles, ensuring an efficient and feasible implementation within budget constraints.
P2 - PO12	Follows ethical guidelines in data collection and privacy.
P3 - PO1	Enhances system efficiency by improving sensor accuracy and data processing.
P3 - PO2	Identifies potential hardware and software issues to refine system functionality.
P3 - PO3	Improves integration of different technologies, including GPS tracking and IoT connectivity.
P3 - PO4	The project requires continuous testing and refinement of system components, such as improving sensor accuracy, optimizing power consumption, and enhancing data transmission.
P3 - PO5	Refining and optimizing the use of components such as IR and alcohol sensors aligns with the application of modern engineering tools for project improvement
P3 - PO6	Implement security measures to prevent data breaches and unauthorized access
P3 - PO7	Optimizes battery life by reducing power consumption in real-time tracking.
P3 - PO10	Produces detailed reports on system improvements and optimizations.
P3 - PO11	Efficient resource management, cost-effective component selection, and system optimization to enhance performance while minimizing expenses plays an important role in project management.
P3 - PO12	Promotes ethical responsibility in handling real-time user data.
P4 - PO5	Team members must coordinate in hardware setup, software programming, and troubleshooting using available tools, demonstrating effective teamwork in modern technology application
P4 - PO8	Encourages teamwork in the development and testing of smart bus monitoring.
P4 - PO9	Encourages teamwork in the development and testing of smart bus monitoring.

Justification

Mapping	Justification
P4 - PO10	Enhances collaboration in designing and debugging system components.
P4 - PO11	Strengthens professional skills through teamwork and communication
P4 - PO12	Promotes responsibility in implementing public transportation innovations.
P5 - PO5	Uses software tools to analyze real-time bus tracking data effectively.
P5 - PO8	The project involves ethical responsibility in reporting accurate findings related to passenger monitoring, GPS tracking, and alcohol detection
P5 - PO9	Applies structured documentation techniques for system reporting.
P5 - PO10	Develops clear and concise user manuals and system documentation.
P5 - PO11	Enhances professional communication skills for presenting project findings.
P5 - PO12	Ensures ethical data collection and reporting practices.
P1 - PSO1	Design and implementation of an embedded system for smart transportation
P1 - PSO2	Uses EDA tools for testing and simulation.
P1 - PSO3	Ensures ethical and sustainable transport solutions.
P2 - PSO1	Selection and integration of microcontrollers, sensors, and GPS modules.
P2 - PSO2	Conducts experiments to validate IR sensors and alcohol detection.
P2 - PSO3	Implements security measures for responsible IoT use.
P3 - PSO1	Enhances system efficiency through refined circuit design.
P3 - PSO2	Develops software tools for data acquisition and communication.
P3 - PSO3	Ensures privacy and reliability in public transport monitoring.
P4 - PSO1	Works collaboratively on system development.
P4 - PSO2	Uses structured methodologies for troubleshooting errors.
P4 - PSO3	Ensures professional responsibility in system reliability.
P5 - PSO1	Documents technical aspects and improvements.
P5 - PSO2	Analyzes performance using EDA tools.
P5 - PSO3	Emphasizes sustainability and ethical deployment.