I. J. Engineering and Manufacturing, **2025, 1, 81-92**
Published Online on February 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2025.01.05

Wireless Vehicular Communication

Saranya M.*
Department of Instrumentation & Control Systems Engineering, Tamilnadu, 641004, India
E-mail: msa.ice@psgtech.ac.in
ORCID iD: https://orcid.org/0000-0002-1470-5509
*Corresponding Author

Archana N.
Department of Electrical and Electronics Engineering, Tamilnadu, 641004, India
E-mail: naa.eee@psgtech.ac.in
ORCID iD: https://orcid.org/0000-0002-2976-3672

Received: 06 September, 2024; Revised: 25 October, 2024; Accepted: 15 November, 2024; Published: 08 February, 2025

Abstract: The proposal entitled as "Wireless Vehicular Communication" deals with communication between vehicles and infrastructural units. This proposal is designed to enable communication between vehicles and the surrounding infrastructure, such as traffic signals and road sensors. The aim of this project is to improve traffic flow, reduce congestion and enhance overall road safety. This project primarily aims at providing instantaneous communication between two vehicles or between vehicles and infrastructure to ensure safety of the passenger. This system is also known as Crucial Message Exchange System (CMES). CMES not only helps in message transmission but also in collection, storage and monitoring of the messages. This system processes data and has the ability to extract crucial information for future machine language-based processing. The CMES system uses the ESP8266 module as the on-board unit in a vehicle to implement Vehicle to Infrastructure Communication (V2I). This module interacts with the Raspberry Pi board using Message Queue Telemetry Transport (MQTT) protocol to relay information. The Raspberry Pi board is considered as the embedded unit in the infrastructure. Raspberry Pi not only facilitates the collection of data but also is responsible for storage and retrieval in cloud systems. All the said components working together ensures a reliable message passing system which will be of immense help to the drivers/passengers on road.

Index Terms: CMES, ESP8266, V2I, Raspberry Pi

1. Introduction

The chapter deals with the application study of project domain, software specifications and brief overview of tools and technologies which is used for "WIRELESS VEHICULAR COMMUNICATION."

1.1. Application study

Recent advances in wireless networks have led to the introduction of a new type of networks called Vehicular Networks. Vehicular Ad Hoc Network (VANET) is a form of Mobile Ad Hoc Network (MANET) technology which is distributed and self-organizing networks formed between moving vehicles equipped with wireless communication devices. This type of networks is developed as part of the Intelligent Transportation Systems (ITS) to bring significant improvement to the transportation system performance. The goal of ITS is to improve safety on the roads, reduce traffic congestion, waiting times and fuel consumptions[1]. The integrations of the embedded computers, sensing devices, navigation systems, digital maps and the wireless communication devices along with intelligent algorithms will help to develop numerous types of applications for the ITS to improve safety on roads. VANET has become an active area of research, standardization and development because it has tremendous potential to improve vehicle and road safety, traffic efficiency and convenience as well as comfort to both drivers and passengers. A lot of VANET research work has focused on specific areas including routing, broadcasting, Quality of Service (QoS), safety and security. It provides two types of communication in VANET namely Vehicle to Vehicle (V2V) and Vehicle to Infrastructure/Infrastructure to Vehicle (V2I/I2V). Fig. 1 represents the types of communication in VANET.

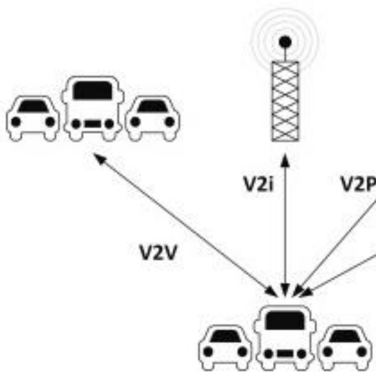

Fig. 1. Types of Communication

V2V communication uses multi-hop communication (multicasting/broadcasting) for transmission of data. Inter-Vehicular communication consists of two types of communication [2]. First is naïve broadcasting which produces beacons at regular intervals. Second is Intelligent Broadcasting which generates messages on demand V2I communication uses single-hop communication. Road Side Unit (RSU) broadcasts messages to the vehicles within the range. It requires high bandwidth link between the vehicles and the RSU.

We organize the remaining section of this research work as follows: In section two (2), we described the experimental setup in terms of hardware and software; in section three (3), we present the methodology and in section four (4), we discussed the results of the proposed model.

2. Experimental System Analysis

Analysis is the process of identifying the current and future roles of the proposed application. The system analyst examines the flow of documents, information and material to design the system that best meets the cost, performance and scheduling objectives.

2.1. Existing System

V2I communication:

i. Dedicated Short Range Communication (DSRC): DSRC is a wireless communication technology that allows vehicles to communicate with roadside infrastructure such as traffic signals, toll booths and parking meters [3].

ii. Cellular Vehicle-to-Everything (C-V2X): C-V2X is a cellular-based communication technology that allows vehicles to communicate with other infrastructure and pedestrians using cellular networks

iii. Radio-Frequency Identification (RFID): RFID is a technology that uses radio waves to identify and track objects. In V2I, RFID tags can be placed on vehicles or roadside infrastructure to allow for communication between the two.

I2V communication:

i. Variable Message Signs (VMS): VMS are electronic signs that display information to drivers, such as traffic conditions, road closures and detours.

ii. Traffic Signal Control: Traffic signals can communicate with vehicles to provide real-time information on signal timing and to optimize traffic flow.

iii. Roadside Sensors: Roadside sensors can detect the presence of vehicles and provide information on traffic volume and speed to infrastructure systems for traffic management and planning purposes.

iv. Roadside Cameras: Roadside cameras can provide real-time video feed to transportation management centers for traffic monitoring and incident management.

V2V communication:

i. Dedicated Short-Range Communications (DSRC): This is a wireless communication technology designed specifically for V2V and V2I applications. DSRC uses the 5.9 GHz frequency band and provides low-latency, high-speed communication between vehicles and infrastructure [4].

ii. IEEE 802.11p: This is a wireless communication standard that is similar to Wi-Fi but designed for V2V and V2I applications. It operates in the 5.9 GHz frequency band and can provide low-latency, high-speed communication between vehicles.

iii. V2V Safety Applications: There are several V2V safety applications that have been developed to improve road safety, including Forward Collision Warning (FCW), Blind Spot Warning (BSW) and Intersection Movement Assist (IMA).

2.2. Proposed system

The proposed system is Crucial Message Exchange system (CMES), which is a type of communication system used in Vehicle-to-Infrastructure (V2I) communication to transmit critical information about traffic, road conditions and other important events. CMES is designed to provide real-time communication between vehicles and infrastructure, allowing drivers to make informed decisions and react to potential hazards in a timely manner. In V2I communication, CMES can be used to transmit information about traffic signals, road signs and other infrastructure-related information. For example, if a traffic signal is malfunctioning or has been turned off, the system can alert drivers to the change in traffic patterns, allowing them to adjust their driving accordingly [5]. Overall, VANET technology is an important component of the CMES communication system, as it allows for the transmission of critical information in real-time. This can help improve road safety, reduce traffic congestion and enhance the overall efficiency of the transportation system. CMES relies on a number of technologies to function effectively, including wireless communication systems, sensors and data processing algorithms. The system uses standardized communication protocols to ensure compatibility between different types of vehicles and infrastructure and can operate in both urban and rural environments. In addition to improving road safety, CMES can also be used to facilitate the development of intelligent transportation systems (ITS) and autonomous vehicles. By providing real-time information about traffic and road conditions, CMES can help autonomous vehicles make more informed decisions and navigate the roadways more safely and efficiently.

3. Methodology

3.1. Vehicular communication

I2V communication is implemented by using Raspberry Pi as infrastructure and NodeMCU ESP8266 as a vehicular unit that aims to improve transportation efficiency and safety by enabling real-time communication between vehicles and infrastructure. The Raspberry Pi can act as an infrastructure hub that collects data from various sensors such as traffic signals, weather sensors and road sensors. Using this system, vehicles can receive real time updates on road conditions, traffic flow and potential hazards, allowing drivers to make more informed decisions and avoid accidents. The use of low-cost microcontrollers and Wi-Fi network makes this system cost-effective and accessible to a wider audience [15]. Additionally, the system can be easily implemented in various transportation systems, making it a promising solution for improving transportation efficiency and safety. Fig. 2 illustrates the flow diagram of V2I.

In Vehicle to Infrastructure communication using Raspberry Pi and Node MCU ESP8266, Vehicle and infrastructure may communicate using MQTT Protocol [6]. Vehicle publishes the real time information to Raspberry Pi. Raspberry Pi extracts the information received from vehicles and processes the file with python and stores the information in ThingSpeak cloud. Any vehicle communicating with the infrastructure in the same network can extract the information from cloud and process the data.

3.2. Hardware and Software Components

RASPBIAN OS:

Raspbian is the official operating system (OS) for the Raspberry Pi series of single board computers [16]. It is a Debian-based Linux distribution that is optimized for the Raspberry Pi's hardware. Raspbian includes a user-friendly desktop environment based on the Lightweight X11Desktop Environment (LXDE) and a suite of pre-installed software, including the Chromium web browser, the VLC media player and the LibreOffice suite. It also includes programming tools and libraries such as Python, Scratch and Sonic Pi, making it a popular choice for educational use. Raspbian is available in two main versions: Raspbian with Desktop and Raspbian Lite. Raspbian with Desktop includes the full desktop environment, while Raspbian Lite is a minimal image that includes only the command-line interface, without the desktop environment.

Raspbian can be downloaded for free from the Raspberry Pi Foundation's website and can be installed on a microSD card using the Raspberry Pi Imager software. Raspbian is regularly updated to include bug fixes and new features, ensuring that it remains a reliable and up-to-date OS for Raspberry Pi users.

Arduino IDE:

The Arduino Integrated Development Environment (IDE) is an open-source software application used to program and develop software for Arduino microcontrollers. It is written in Java and can be used on multiple operating systems, including Windows, macOS and Linux. The Arduino IDE includes a code editor, a compiler and a bootloader uploader, all in a single software package. It supports several programming languages, including C and C++, which are used to write Arduino sketches, or programs. These sketches are written using the Arduino programming language, which is a simplified version of C++. The Arduino IDE also provides a library manager, which makes it easy to install and manage additional libraries of pre-written code, making it easier to incorporate various sensors, actuators and other components into the project. Overall, the Arduino IDE is a user-friendly software development environment that simplifies the process of programming Arduino boards, making it accessible to beginners and advanced users alike. It is widely used in the maker community, education and prototyping applications.

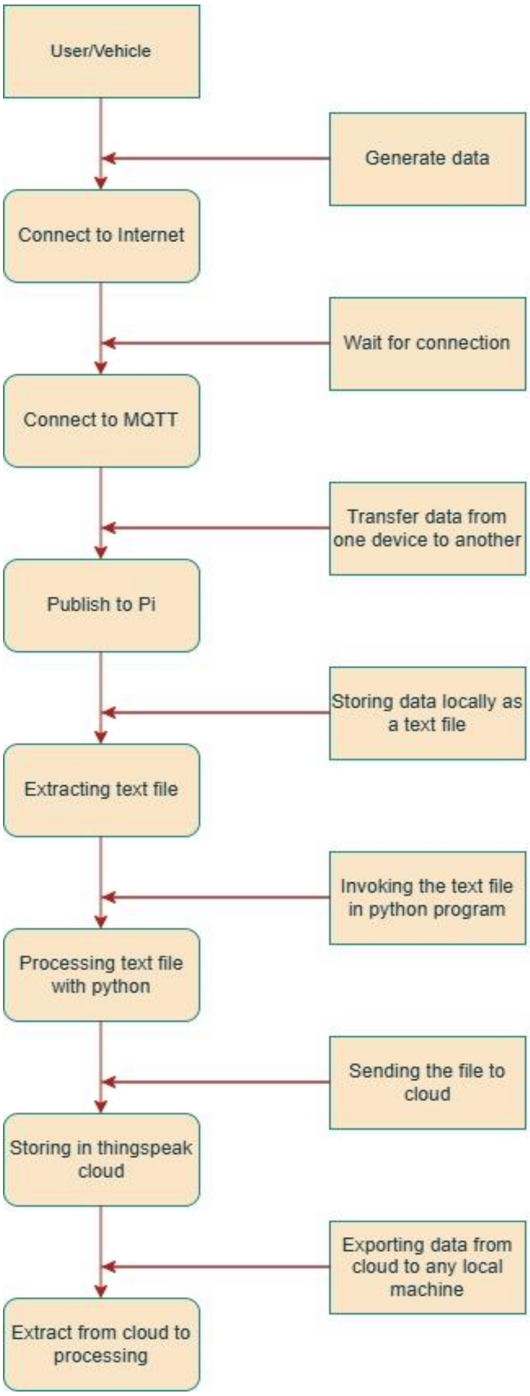

Fig. 2. Flow Diagram of V2I Communication

MQTT Explorer:

MQTT Explorer is an open-source Message Queue Telemetry Transport (MQTT) client that allows user to explore and monitor MQTT messages on their MQTT broker. A user friendly interface is available for connecting with an MQTT broker and subscribing to topics for viewing real-time messages published on those topics. MQTT Explorer supports multiple protocols such as MQTT and Web Sockets. With MQTT Explorer, user can view the payload of the messages, their QoS level and the timestamp when the message was published. It also allows one to publish messages to a specific topic, subscribe to multiple topics and filter messages by topic, payload and QoS level. Additionally, MQTT Explorer provides a powerful tool to manage MQTT broker by allowing user to view the broker's configuration and connected clients, publish retained messages and manage Access Control Lists [7]. It also supports plugins that provide additional functionality and can be used to extend its capabilities.

Thonny IDE:

Thonny is a free and open-source Python Integrated Development Environment (IDE) that is designed to simplify the process of learning and teaching programming. It was developed by Aivar Annamaa, a computer science educator at the University of Tartu in Estonia and is available for Windows, macOS and Linux operating systems. Thonny provides a user-friendly interface that makes it easy for beginners to write and run Python code. It includes features such as code highlighting, autocomplete and error highlighting to help users write correct code. It also has a debugger that allows users to step through their code line by line to identify and fix errors. One of the unique features of Thonny is its simplified debugger. It allows users to easily inspect the values of variables and expressions at any point in their program execution, making it easier to understand what their code is doing and to find errors. Thonny also includes a code profiler that can help users identify performance bottlenecks in their code.

ThingSpeak:

ThingSpeak is an open-source Internet of Things (IoT) platform that allows users to collect, store, analyse and visualize data from connected devices or sensors. It was developed by MathWorks, a company that provides software solutions for technical computing and engineering. ThingSpeak provides a simple web-based interface that allows users to create channels for data collection and visualization. Channels are used to store data streams from devices or sensors, which can be displayed in real-time on a dashboard or visualized as graphs or charts. In addition to data visualization, ThingSpeak also provides tools for data analysis and decision-making. It includes MATLAB analytics, which allows users to perform complex data analysis and machine learning algorithms to extract insights from their data. It also provides integration with third-party services such as IFTTT (If This Then That), which allows users to create powerful automation workflows. ThingSpeak is flexible and scalable, making it suitable for a wide range of applications. It can be used to monitor environmental conditions, track the location of assets, or collect data from industrial machines, among other use cases. It supports a variety of communication protocols, including MQTT, HTTP and HTTPS, allowing it to be integrated with a wide range of devices and platforms.

Power cable:

A charging cable is a cable that is used to charge electronic devices such as smartphones, tablets, laptops and other battery-powered devices. A charging cable typically has two ends: one end that connects to the electronic device that needs to be charged and the other end that connects to a power source such as a USB charger, computer, or power bank. USB Type-B cable is less common and features a square-shaped connector that is used to connect to printers, scanners and other peripheral devices. Charging cables also vary in length, thickness and durability. It is important to choose a cable that is compatible with selected device and provides a safe and reliable charging experience.

HDMI to VGA converter:

An HDMI to VGA converter is a device that allows one to connect a device with an HDMI output (such as a laptop, Blu-ray player, or game console) to a display with a VGA input (such as an older monitor or projector). The converter takes the digital HDMI signal and converts it to an analog VGA signal, which can then be displayed on the VGA device. The converter typically has an HDMI input port on one end and a VGA output port on the other. It may also have an audio output port, which allows one to connect an external speaker or headphones to the converter if VGA display does not have built-in speakers. HDMI to VGA converters can be useful in situations where user need to connect a newer device with an HDMI output to an older display with a VGA input. However, it's important to note that VGA is an analog signal, whereas HDMI is a digital signal, so the quality of the output may be affected by the quality of the converter and the cables used.

Node MCU:

Node Microcontroller Unit (NodeMCU) is an open-source firmware and development board based on the ESP8266 Wi-Fi chip. The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capabilities. NodeMCU is designed to make it easy to create Internet of Things (IoT) projects with the ESP8266.The NodeMCU board has built-in Wi-Fi connectivity and runs the Lua programming language. It is also compatible with the Arduino IDE, which makes it easy to program and upload code to the board. The NodeMCU board has several GPIO pins that can be used for digital input/output, analog input and other functions [8]. 10 NodeMCU is designed to make it easy for makers and hobbyists to create IoT projects without having to deal with the complexities of low-level Wi-Fi programming. With NodeMCU, developers can quickly create Wi-Fi enabled projects such as home automation systems, weather stations and smart home devices.

Raspberry Pi3:

The Raspberry Pi 3 is a single-board computer designed for hobbyists, makers and educators. It is the third iteration of the Raspberry Pi series, released in 2016 by the Raspberry Pi Foundation. The Raspberry Pi 3 is a small, credit card-sized device that packs a lot of power and features in a tiny package. It is compatible with a wide range of operating systems, including Raspbian (a Linux-based operating system), Windows 10 IoT Core and various versions of Ubuntu, Debian and other Linux distributions. The Raspberry Pi 3 has a low power consumption, making it an energy-efficient choice for many projects. It can be powered using a 5V micro-USB power adapter, which is commonly used for mobile phones.

3.3. Implementation

I2V communication allows for real-time communication between the infrastructure and vehicles through networks, enabling drivers to access a range of services such as providing on road assistance, music streaming, navigation or vehicle diagnostics. V2I communication, on the other hand, enables vehicles to exchange information with the infrastructure, such as traffic signals, roadside units or toll booths [9]. This information can include traffic and road conditions, weather updates or road hazards.

Raspberry Pi3:

The Raspberry Pi 3 board is the most suitable component to be embedded in the infrastructure because of its versatile features. For the initial configuration, the Raspberry Pi setup requires peripherals like monitor, keyboard and mouse. But once all the network specifications are configured the Raspberry Pi board can function in a headless manner. The ability of the board to be remotely connected and controlled by Microsoft Remote Desktop protocol is the main reason why one is able to operate in headless mode [10]. The key advantage of using this board is that its functionality can be extended whenever needed. Currently, this board is only being used for simple data transmission and basic cloud computing. But in the future this board is fully capable of handling additional complex VANET operations. This is the reason for using Raspberry pi board as the embedded unit in this project

NODEMCU ESP8266:

NodeMCU ESP8266 acts as the on-board unit in vehicles. This board may not be as versatile as the Raspberry Pi board but it is more than enough to satisfy the basic requirement of establishing a MQTT connection with the infrastructure and transmitting data. There is no requirement for data processing or data handling in the vehicle itself, all data handling processes are done in the infrastructure. So, any microcontroller board with the ability to connect to the internet would suffice to implement this project. The NodeMCU ESP8266 is chosen because it can be easily configured using Arduino IDE [11]. Unlike the Raspberry Pi board the ESP8266 cannot operate on headless mode as it requires user to type in the input or it has to display the information received from the infrastructure. This project will use desktops to display the output of ESP8266 units, but in an ideal environment ESP8266 board will be interfaced with the vehicle’s media player or any other inbuilt display device in the vehicle.

THINGSPEAK:

The project data is recorded in ThingSpeak cloud. All the data is sent and received through the Raspberry Pi terminal. So, in order to collect the sent and received data, “script” command is used to store all terminal data into a text file. This text file is then uploaded into the ThingSpeak cloud through python coding. Before one upload the files, a channel with necessary settings has to be created. The channel Id and keys are of utmost importance because they help the Raspberry Pi board to connect to the cloud through python [12]. Once the file is uploaded it can be accessed from anywhere by anyone with the cloud account credentials. The added advantage of cloud computing is that it extends the life of the technology by helping it evolve with the technology around it. i.e., the Raspberry Pi board may not have the ability to process the data that will be generated in the future, but once the data is in the cloud any super computer can access and process it, hence helping the system to evolve and preventing it from becoming obsolete. The flow diagram of the publisher and subscriber node are depicted in fig. 3 and fig.4 respectively.

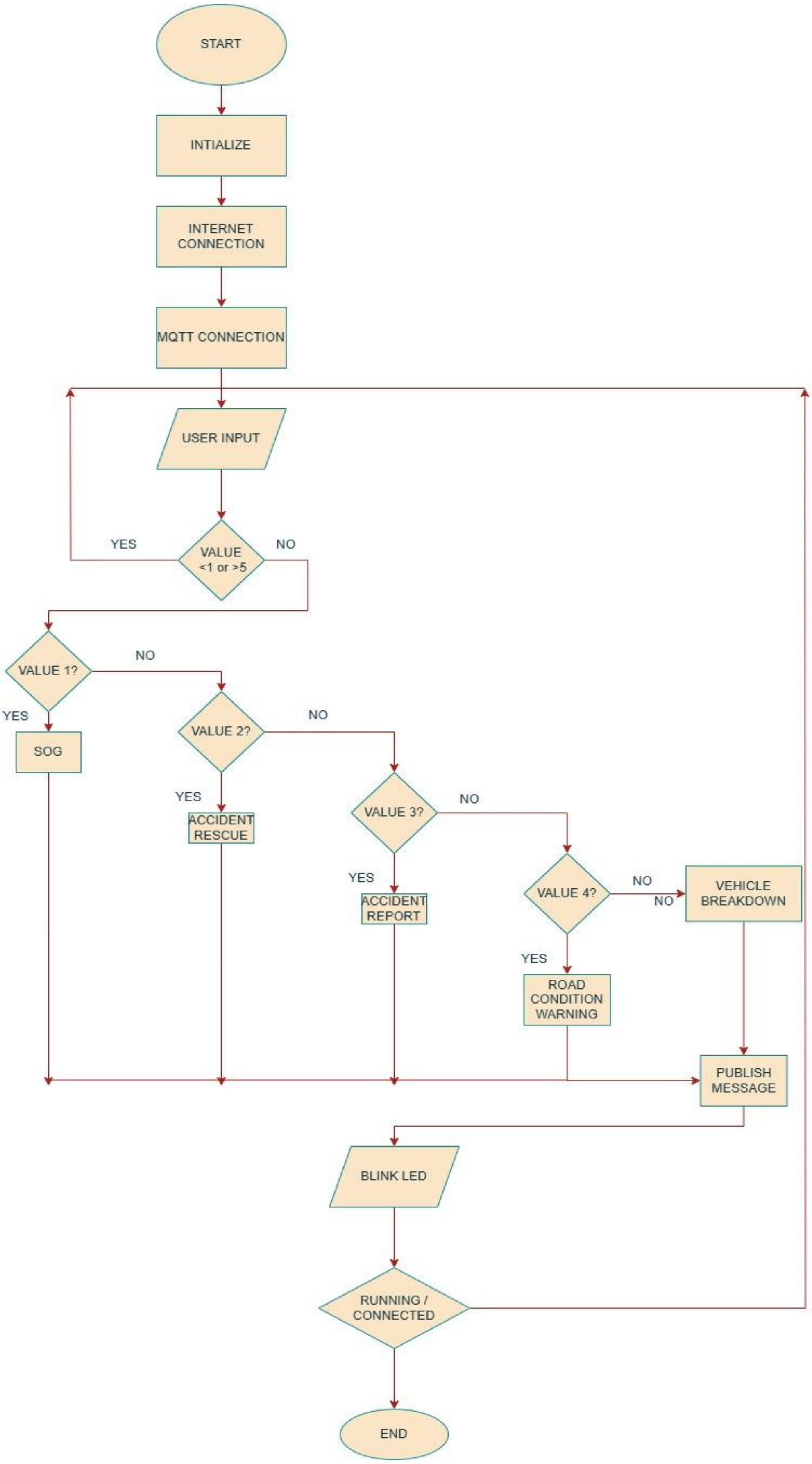

Fig. 3. Flow diagram of Publisher node

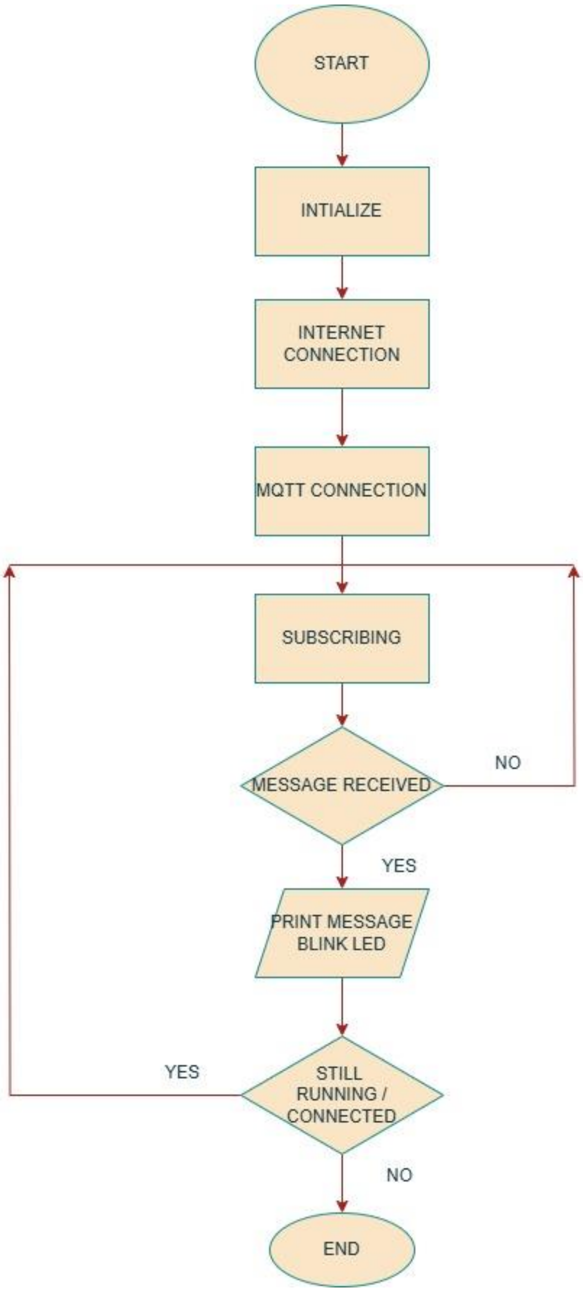

Fig. 4. Flow diagram of subscriber node

4. Results and Discussion

This chapter gives an introduction to the various testing process involved in the project. The testing process focuses on the logical internals of the program, ensuring that all statements have been tested and also the functional externals that is conducting the test to uncover errors and ensuring that the designed input will produce actual results that are accurate and reliable

4.1. Testing Process

The testing process is carried out by means of manual testing. While testing, the project is run against the test cases. Each test case has to be simulated and the observed outcome is noted. If the observed outcome does not match the expected outcome, then the particular case is marked as failed. Once the project has been tested, the test results are discussed and the issues if any, are fixed and the failed cases are repeated.

4.2. Unit Testing

Unit testing is a level of software testing that involves individually testing unit of code to ensure that it works on its own correctly, independent of the other units. The key purpose is to validate that every single unit of the software performs as it is intended to perform. Table 1 shows the unit testing which has been done to ensure whether the unit such as Raspberry Pi, NodeMCU ESP8266 works precisely.

4.3. Integration Testing

It is a level of software testing where individual units are combined and tested as a group to expose faults in the interaction between integrated units. Table 2 shows integration testing of wireless vehicular emission and their results.

Table 1. Unit Testing for Wireless Vehicular Communication

TEST ID	TESTING COMPONENT	TEST CASE	EXPECTED OUTCOME	OBSERVED OUTCOME	STATUS
01	Raspberry Pi	Power supply is given	Blinking of LED	Blinking of LED	Success
02	Remote Desktop Connection	xrdp packages installed	Remote access is established	Remote access not established	Failure
03	Raspberry Pi	Remotely controlling it from another system wirelessly	Blinking of LED	Blinking of LED	Success
04	NodeMCU ESP8266	Power supply is given	Blinking of LED	Blinking of LED	Success
05	MQTT Broker	Verifying protocol with MQTT Explorer	Connection established	Connection established	Success

Table 2. Integration Testing for Wireless Vehicular Communication

TEST ID	TESTING COMPONENT	TEST CASE	EXPECTED OUTCOME	OBSERVED OUTCOME	STATUS
01	Complete System	Remotely controlling LED	LED blinking while transmission of messages	LED blinking while transmission of messages	Success
02	Integrating Raspberry Pi and NodeMCU	Communication from Raspberry Pi to NodeMCU	Message sent and received	Message sent and received	Success
03	Integrating Raspberry Pi and NodeMCU	Communication from NodeMCU to Raspberry Pi	Message sent and received	Message sent and received	Success
04	Accessing and processing cloud data in remote system	Downloading and processing Data from cloud Using python	All required lines are extracted	Certain amount of data is lost during processing	Failure

5. Conclusion

Vehicle-to-infrastructure (V2I) communication is a type of communication technology that enables vehicles share information with each other, such as their speed, location and direction [13]. Direct V2I communication on the other hand, enables vehicles to communicate with infrastructure elements such as traffic lights, road signs and toll booths. The primary goal of VANET is to improve road safety, reduce traffic congestion and enhance the overall driving experience as vehicles can better anticipate and respond to potential hazards and avoid accidents. V2I communication can also provide drivers with real-time information about road conditions, such as traffic congestion, construction zone and weather conditions [14]. One of the main challenges is the need for a reliable and secure communication network. The communication network must be able to handle large amounts of data in real-time and must be able to operate in a variety of weather and road conditions. In future, the network must be secure to prevent unauthorized access and malicious attacks. Despite the challenges, VANET communication is rapidly gaining momentum and is expected to become an integral part of future transportation systems. The development and deployment of these technologies are being driven by advances in wireless communication, sensor technology and data analytics. With the widespread adoption of V2V and V2I communication, this project can expect significant improvements in road safety, traffic flow and overall driving experience.

6. List of Abbreviations

1. CMES –Crucial Message Exchange System
2. V2I – Vehicle to Infrastructure
3. I2V-Infrastructure to Vehicle
4. V2P-Vehicle to Personal devices
5. V2S-Vehicle to Sensors
6. V2R-Vehicle to Road side units
7. MQTT-Message Queue Telemetry Transport

8. VANET-Vehicle Ad Hoc Network
9. QoS-Quality of Service
10. MANET-Mobile Ad Hoc Network
11. ITS-Intelligent Transportation System
12. DSRC-Dedicated Short Range Communication
13. C-V2X Cellular- Vehicle to Everything
14. RFID-Radio Frequency Identification
15. VMS- Variable Message Signs
16. V2V-Vehicle to Vehicle
17. IEEE-Institute of Electrical and Electronics Engineering
18. BSW-Blind Spot Warning
19. FCW- Forward Collision Warning
20. IMA- Intersection Movement Assist
21. MCU-Microcontroller Controller Unit
22. OS-Operating System
23. LXDE- Lightweight X11 Desktop Environment
24. VLC-Video LAN Client
25. IDE-Integrated Development Environment
26. IoT-Internet of Things
27. IFTTT-If This Then That
28. HTTP-Hyper Text Transfer Protocol
29. HDMI-High Definition Multimedia Interface
30. VGA-Video Graphics Array
31. USB-Universal Serial Bus
32. WiFi – Wireless Fidelity

7. Declarations

• Funding

No funding was obtained for this project study.

• Authors' Contributions

All the authors has contributed towards the work published in this paper.

• Data Availability

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

• Conflict Of Interests

Authors have no conflict of interest in publishing this paper.

Figure Title & Legend

• Figure 1. Types of Communication

This figure represents the types of communication in VANET, such as V2I, I2V and V2V.

• Figure 2. Flow Diagram of V2I Communication

This figure gives out a clear picture of the steps to be followed in implementing the V2I communication.

• Figure 3. Flow Diagram for Publisher code

This figure gives out a clear picture of the steps to be followed in implementing the publisher code in MQTT protocol.

• Figure 4. Flow Diagram for Subscriber code

This figure gives out a clear picture of the steps to be followed in implementing the subscriber code in MQTT protocol.

Acknowledgment

The authors deeply appreciate the management of PSG College of Technology, Peelamedu, Coimbatore, Tamilnadu, India for providing enabling environment to carry out this research.

References

[1] Eric J. Braude, "Software Engineering: Modern Approaches", Wiley Publications, 2022.
[2] Haibo Zhang and Wenbo YE, "V2X-Enabled Cooperative Intelligent Transport Systems: Towards the Next Generation of Intelligent Vehicles", Springer Publications, 2019.
[3] Hannes Hartenstein and Kennneth P Laberteaux, "Vehicular applications and Inter networking technologies", Willey Publications, 2015.
[4] Latha Parameswaran, Venkata Krishna Puppala, and Nalini Venkatasubramanian, "Communication Technologies for Vehicles", Springer Publications, 2020.
[5] Luca Delgrossi and Tao Zhang, "V2V/V2I Communications: A Comprehensive guide", Academic Press, 2017.
[6] Sherali Zeadally, Naveen Chilamkurti, and Hakima Chaouchi, "Connected Vehicles: Intelligent Transportation Systems and Smart Technologies", CRC P ress, 2018.
[7] Z. A. Abood, H. B. Taher, and R. F. Ghani, "Detection of Road Traffic Congestion Using V2V Communication Based on IoT," Iraqi Journal of Science, pp. 335–345, Jan. 2021, doi: https://doi.org/10.24996/ijs.2021.62.1.32.
[8] Y. Mali, S. Deore, P. Rakholiya, P. Taskar, and R. Scholar, "V2V & V2I Communication for Prevention of Road Accidents, Review Paper," International Journal of Scientific Development and Research, vol. 3, no. 10, pp. 2455–2631, 2018, Accessed: Apr. 11, 2024
[9] "An affordable solution to V2V Communication using X Bee," International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 12, pp. 3931–3935, Oct. 2019, doi: https://doi.org/10.35940/ijitee.l3442.1081219.
[10] P. Keeratiwintakorn, "Ubiquitous Communication for V2V and V2I for Thailand Intelligent Transportation System," www.academia.edu, Jan. 2009.
[11] "Map-based navigation for autonomous vehicles," ADAS & Autonomous Vehicle International, Sep. 11, 2023. https://www.autonomousvehicleinternational.com/features/feature-map-based-navigation-for- autonomous-vehicles.html
[12] I. Journal, "ACCIDENT DETECTION AND AVOIDANCE USING VEHICLE TO VEHICLE COMMUNICATION (V2V)," IRJET, Jan. 2023, Accessed: Apr. 11, 2024. [Online].
[13] R. Wang, Z. Xu, X. Zhao, and J. Hu, "V2V - based method for the detection of road traffic congestion," IET Intelligent Transport Systems, vol. 13, no. 5, pp. 880–885, Jan. 2019, doi: https://doi.org/10.1049/iet-its.2018.5177.
[14] A. M. Vegni and T. D. C. Little, "Hybrid vehicular communications based on V2V-V2I protocol switching," International Journal of Vehicle Information and Communication Systems, vol. 2, no. 3/4, p. 213, 2011, doi: https://doi.org/10.1504/ijvics.2011.044263.
[15] I. J. of A. R. in C. S. and M. S. [IJARCSMS] Ijarcsms.com, "A Review on Vehicle-To-Vehicle Communication Protocols for Traffic Safety Using CoOperative Collision Warning," www.academia.edu, Accessed: Apr. 11, 2024. [Online].
[16] "Raspberry Pi Documentation - Raspberry Pi Hardware," www.raspberrypi.com. https://www.raspberrypi.com/documentation/computers/raspberry-pi.html

Authors' Profiles

Saranya M. is currently working as Assistant Professor in the dept. of Instrumentation & Control Systems Engineering at PSG College of Technology, Peelamedu, Coimbatore. She serves in this position since June 2017. She has experience in Academic, Research and Industry. Her main research areas include Edge Computing and Internet of Things.

Archana N. is currently working as Assistant Professor in the dept. of Electrical & Electronics Engineering at PSG College of Technology, Peelamedu, Coimbatore. She serves in this position since June 2011. Her main research areas include Soft Computing and Internet of Things.

How to cite this paper: Saranya M., Archana N., "Wireless Vehicular Communication", International Journal of Engineering and Manufacturing (IJEM), Vol.15, No.1, pp. 81-92, 2025. DOI:10.5815/ijem.2025.01.05