A DRIVER MONITORING AND ALERTING SYSTEM (DMAS) ON DRIVER HABITS ON ROAD

YOONG KAH HAO

BACHELOR OF MECHATRONICS ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

A Driver Monitoring and Alerting System (DMAS) on driver habits on the roads

YOONG KAH HAO

A report submitted
in partial fulfilment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours

Faculty of Electrical Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I declare that this thesis entitled “A Driver Monitoring and Alerting System (DMAS) on driver habits on the roads” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

Signature :

Name : YOONG KAH HAO

Date : 18 Jun 2024

APPROVAL

I hereby declare that I have checked this report entitled " A Driver Monitoring and Alerting System (DMAS) on driver habits on the roads ", and in my opinion, this thesis fulfils the partial requirement to be awarded the degree of Bachelor of Mechatronics Engineering with Honours

Signature :

Supervisor Name : DR HAIROL NIZAM BIN MOHD SHAH

Date : 18 Jun 2024

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

In preparing this report, I was in contact with many people, including researchers, academicians, and practitioners. They have contributed to my understanding and thoughts. I wish to express my sincere appreciation to my main project supervisor, Dr. Hairol Nizam bin Mohd Shah, for his encouragement, advice, and constructive criticism. Without his continued support and interest, this project would not have been presented in the same way.

I extend my thanks to my university, Universiti Teknikal Malaysia Melaka (UTEM), for their special assistance in providing relevant literature and offering the space for me to study my bachelor's degree in mechatronic engineering. I would also like to express my gratitude to both panel members, Assoc. Prof. Ir. Dr. Ahmad Zaki bin Haji Shukor and Puan Fadilah binti Abdul Azis, for their critique and advice, which significantly contributed to the improvement of my project.

My sincere appreciation also goes to all my colleagues and other lecturers who have assisted me on various occasions. Their insights and tips have been truly valuable. Unfortunately, it is not possible to list all of them in this limited space. Lastly, I am grateful to all my family members as well. They have provided unwavering support both physically and mentally and have afforded me the opportunity to pursue further studies at this level.

ABSTRACT

The Driver Monitoring and Alert System (DMAS) was developed to address the increasing number of traffic accidents in 2024. The primary issue is the rising incidence of speed-related accidents on Malaysian roads, highlighting the urgent requirement for a comprehensive Driver Monitoring and Assistance System (DMAS). The objective of this project was to develop a system that involves real-time monitoring of driver speed, alerting drivers when they exceed speed limits, and collecting data for the analysis of driver speeding behavior using Fuzzy Logic Technique. The proposed DMAS aims to improve road safety in Malaysia and globally. The DMAS is implemented using the Sony Spresense board, which has Global Navigation Satellite System (GNSS) capabilities for accurate speed and location detection. The system is installed in the vehicle and powered with a USB 5V power supply, which easily found in the car's USB hub. When the DMAS is turned on, it establishes a connection with the GNSS system. After receiving coordinates from the GNSS, it shows real-time data of the actual speed and speed limit on a thin-film-transistor liquid-crystal display (TFT LCD). The real-time data is captured during the driver's path, including points, longitude, latitude, time, speed, and speed limit on an SD card. When the driver exceeds the speed limit, the TFT LCD and a buzzer act as an alerting system. The collected data can be analyzed and visualized using the DMAS app, which employs Fuzzy Logic Technique to interpret driver speed behavior. The app provides route maps, tables of exceeding speed incidents, and comparisons of speed versus time, along with interpretations of driver behavior. This feedback can educate drivers, promoting safer driving practices if a low safety score is obtained. The DMAS is believed to achieve its objectives of monitoring and alerting drivers while obtaining real-time data to enhance road safety globally.

ABSTRAK

Sistem Pemantauan dan Peringatan Pemandu (DMAS) telah dibangunkan untuk menangani peningkatan jumlah kemalangan jalan raya pada tahun 2024. Isu utama adalah peningkatan insiden kemalangan yang berkaitan dengan kelajuan di jalan raya Malaysia, yang menekankan keperluan mendesak untuk Sistem Pemantauan dan Bantuan Pemandu (DMAS) yang menyeluruh. Objektif projek ini adalah untuk membangunkan sistem yang melibatkan pemantauan kelajuan pemandu secara masa nyata, memberi amaran kepada pemandu apabila mereka melebihi had kelajuan, dan mengumpul data untuk analisis tingkah laku pemanduan laju menggunakan Teknik Logik Kabur. DMAS yang dicadangkan bertujuan untuk meningkatkan keselamatan jalan raya di Malaysia dan di seluruh dunia. DMAS dilaksanakan menggunakan Sony Spresense, yang mempunyai keupayaan Sistem Satelite Navigasi Global (GNSS) untuk pengesanan kelajuan dan lokasi yang tepat. Sistem ini dipasang di dalam kenderaan dan dikuasakan dengan bekalan kuasa USB 5V, yang mudah didapati di hub USB kereta. Apabila DMAS dihidupkan, ia akan membuat sambungan dengan sistem GNSS. Setelah menerima koordinat dari GNSS, ia akan menunjukkan data masa nyata mengenai kelajuan sebenar dan had kelajuan pada paparan kristal cecair transistor filem nipis (TFT LCD). Data masa nyata ini ditangkap sepanjang laluan pemandu, termasuk titik, longitud, latitud, masa, kelajuan, dan had kelajuan pada kad SD. Apabila pemandu melebihi had kelajuan, TFT LCD dan buzzer akan bertindak sebagai sistem amaran. Data yang dikumpul boleh dianalisis dan divisualisasikan menggunakan aplikasi DMAS, yang menggunakan Teknik Logik Kabur untuk mentafsir tingkah laku kelajuan pemandu. Aplikasi ini menyediakan peta laluan, jadual insiden melebihi kelajuan, dan perbandingan kelajuan berbanding masa, bersama dengan tafsiran tingkah laku pemandu. Maklum balas ini boleh mendidik pemandu, menggalakkan amalan pemanduan yang lebih selamat jika skor keselamatan yang rendah diperoleh. DMAS dipercayai dapat mencapai objektifnya untuk memantau dan memberi amaran kepada pemandu sambil mendapatkan data masa nyata untuk meningkatkan keselamatan jalan raya di seluruh dunia.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

DMAS	-	Driver Monitoring and Alerting System
LED	-	Light Emmiting Diode
GPS	-	Global Positioning System
LCD	-	Liquid Crystal Display
TFT LCD	-	Thin-Film-Transistor Liquid-Crystal Display
SD CARD	-	Secure Digital Card
IOT	-	Internet of Things
APP	-	Application
CSV	-	Comma-Separated Values
GNSS	-	Global Navigation Satelite System
ESD	-	Electric Static Damage
OBD	-	On-Board Diagnostics
GPRS	-	General packet radio service
GSM	-	Global System for Mobile Communications

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

The Driver Monitoring and Alert System (DMAS) or Driver Monitoring System (DMS) was originally designed to reduce the possibility of accidents brought on by human error. The traditional Driver Monitoring and Alert System (DMAS) was mainly focused on recognizing specific abnormal driving behaviors, such as fatigue, distracted driving, fainting, and furthermore. In the past, the Driver Monitoring and Alert System (DMAS) collected face data from drivers displaying unusual driving behaviors by using a combination of microcontrollers and cameras. [4] According to research conducted by Najib et al. utilizing the Analytic Hierarchy Process (AHP), there are more people using the roads, which leads to an increase in the frequency of traffic accidents. Additionally, the study reveals that "driving faster than the speed limit" was the accident's primary cause. [1]

Therefore, to reduce the frequency of accidents, the Driver Monitoring and Alerting System (DMAS) must devote greater attention to tracking the speed of the cars. Although governments across the globe have deployed various cameras at various roadside locations to track vehicle speeds, particularly on national highways, the number of accidents continues to rise year-round due to the limited number of tracked roads. The Driver Monitoring and Alert System (DMAS) is capable of detecting the speed of vehicles in various roadside environments, including cities, villages, and highways, and it can also link to other devices. [2]

In the context of vehicle operation, the DMAS continues to play a crucial role in observing driver errors and behaviors during speeding. Despite the introduction of autonomous driving technologies to the market, the high cost of sensors and cameras, coupled with their insufficient maturity for diverse road and weather conditions, remains a challenge. [3] With the used of the Global Navigation Satellite System

(GNSS) in the Driver Monitoring and Alert System (DMAS) the route and vehicle speed data can be recorded and analyzed. [4]

1.2 Motivation

In Malaysia, traffic accidents are a serious problem, and the number of accidents is rising annually, as shown by the indicator released by the appropriate government. Since traffic accidents and fatalities add to the nation's economic losses, immediate action on this issue is necessary. According to statistics from Malaysia's Road Transport Department, the proportion of Malaysians who own a car is almost one to one. Road accidents now account for more than 1.25 million deaths annually, therefore becoming a global problem. [5]

The Transport Minister, Anthony Loke, published a report stating that 6080 people died and 545588 traffic accidents were registered in 2022. In Malaysia, there are 1494 accidents on average each day, or one accident every minute.

Research conducted by the Malaysian Institute of Road Safety Research (MIROS) found that human behavior was the primary cause of most traffic accidents. Human behavior, which accounts for 80.6% of road accidents, is a more significant role than surrounding factors and road infrastructure (13.2%), with vehicles accounting for just 6.2%. [5]

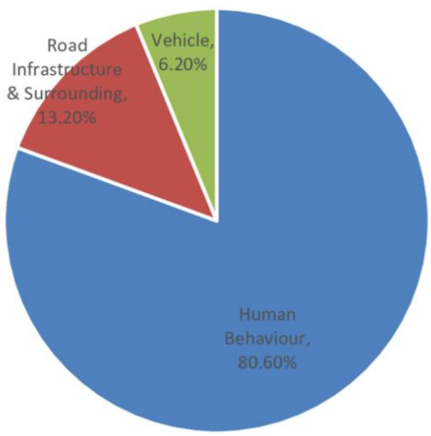


Figure 1-1 :Cause of Accident

Inappropriate speed for the traffic conditions has the largest weight when it comes to contributing to road traffic safety, according to research on the Analytic Hierarchy Process (AHP), which rates and evaluates the aspects that affect road traffic safety. (Google) When a driver is charged with a speed-related infraction or when racing, driving too fast for the conditions, or exceeding the posted speed limit is mentioned as a contributing factor in the collision, the crash is deemed speeding-related. A research conducted in 2021 found that 29% of all traffic fatalities were caused by speeding. The number of fatalities linked to speeding has increased for the third year in a row in 2021. Furthermore, the percentage of speed-related deaths (29%) is higher than the historic low of 26% set in 2019. The number of deaths linked to speeding increased 7.9% from 2020 to 2021, marking the highest number since 2007. [6]

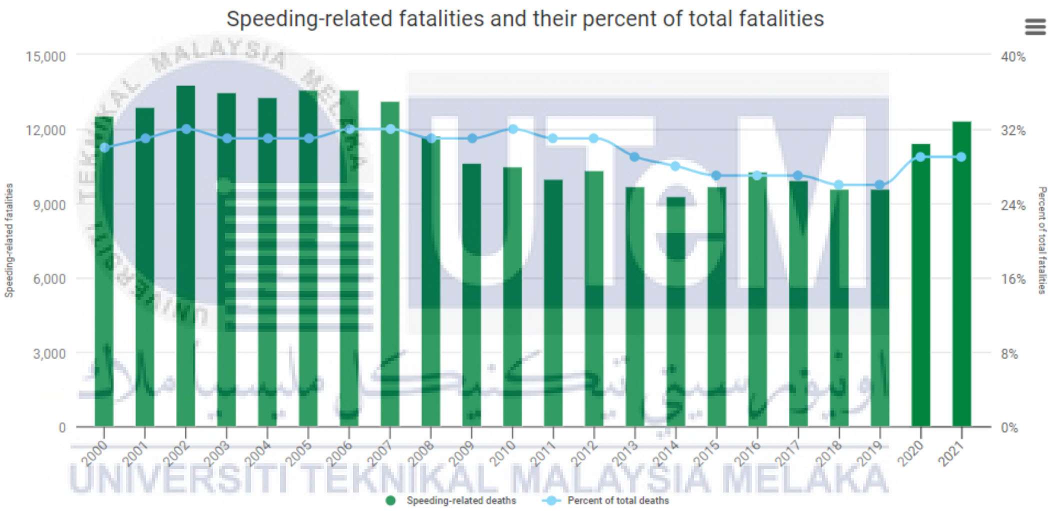


Figure 1-2 : Speed-related facilities and thier percentage of total facilities.

As a result, it was necessary to create the Driver Monitoring and Alert System (DMAS) to track the driver's speed behavior while the car was driving. It was thought that this would effectively lower the probability of an accident occurring. When a driver overspeed at any roadside in a rural or urban region, the Driver Monitoring and Alert System (DMAS) will sound an alert, and the data will be saved and used for analyzing.

1.3 Problem Statement

Speeding-related accidents are a major issue on Malaysian roadways, posing a substantial risk to public safety. The incidence of these accidents is not only becoming more frequent, but also becoming more serious, resulting in higher rates of deaths and serious injuries. The current speed limit detection systems, such as the Automated Enforcement System (AES), are being challenged for their lack of efficiency, ineffectiveness, and the risks they represent to traffic officer. [Real-time Monitoring and Detection of Drinkdriving and Vehicle Over-speeding] The AES is mostly utilised in cities and specific sections of highways with the objective of enforcing speed restrictions by automated methods. Nevertheless, the limited deployment of AES limits its ability to provide comprehensive coverage and optimal performance on Malaysia's varied road networks.

In addition, current solutions such as widely-used navigation applications (such as Google Maps and Waze) have limited capabilities when it comes to monitoring and enforcing speed restrictions. These apps mainly focus on providing navigation and traffic information, rather than offering thorough monitoring of drivers and implementing safety measures. These applications frequently have diminished GPS precision in urban settings due to tall structures obstructing satellite signals, resulting in delays or inaccuracies in speed limit notifications. Furthermore, commercial vehicle speed tracking systems (VSTS) that depend on GPS for monitoring speeds require subscriptions and experience similar difficulties due to GPS inaccuracy in urban regions. In order to successfully deal with these difficulties, there is an urgent need for a comprehensive Driver Monitoring and Alerting System (DMAS) specifically designed for the road conditions in Malaysia and globally.

1.4 Objective

i. To develop a Driver Monitoring and Alerting System (DMAS) that continuously monitors vehicle speed and provides real-time alerts to drivers when the speed limit is exceeded.

ii. To analyze the collected data to identify driver behavior patterns utilizing Fuzzy Logic techniques.

1.5 Scope and Limitation

The Driver Monitoring and Alert System (DMAS) was focusing of monitoring ,alerting and analyzing driver behavior during the journey. The monitoring and alerting utilises the capabilities of the Sony Spresense board, which is equipped with an integrated Global Navigation Satellite System (GNSS) receiver. The system detects vehicle speed and location by directly communicating with satellites, supporting GPS, GLONASS, Galileo, and QZSS. The Sony Spresense board is programmed using the Arduino IDE for the purpose to establish a connection with the GNSS. The 1.8-inch TFT LCD ST7735 display shows various data, such as speed and speed limitations, and a buzzer alerts the driver when the speed limit is exceeded. The system logs data onto an SD card connected via the Sony Spresense extension board for later analysis using a Python-based DMAS application.

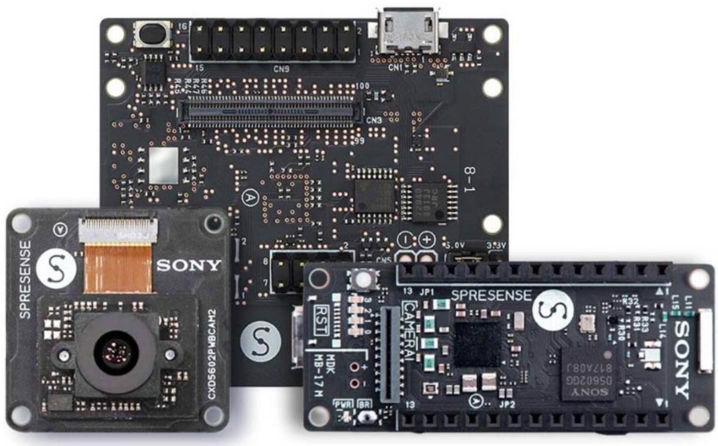


Figure 1-3 : Sony Spresense

The hardware platform's limitations include the performance of the Sony Spresense board and environmental factors that can affect GNSS accuracy, such as obstructions like tall buildings, trees, tunnels, atmospheric conditions, signal multipath effects, and satellite geometry. The system’s accuracy and performance should be assessed according to the hardware specifications provided in the datasheet, which indicate a horizontal positioning accuracy typically within 2.5 meters and a speed accuracy within 0.1 meters/second, with up to a 10 Hz update rate for GNSS data. This scope includes the system's monitoring and alerting capability, as well as the limitations caused by the hardware and environmental conditions.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will cover the theory and concept of the Driver Monitoring and Alerting System (DMAS) for tracking driver speed on the road. The details and information from the journal research that was located on Google Scholar will be covered in this chapter. This literature review will be focused within a few keywords, such as "Vehicle tracking" "Vehicle Speed" "Over speeding","Vehicle path." And “Driver Behavior”. The purpose of the overall summary was to provide an overview of this chapter and offer recommendations for improving the Driver Monitoring and Alerting System (DMAS).

2.2 Driver Monitoring and Alerting System (DMAS) for vehicle tracking

Pelaez C., G. A. et al., 2014 [3], mention that Driver Monitoring Systems (DMS) have become crucial in enhancing road safety and understanding driver behavior. With the increasing integration of technology in vehicles, the need to monitor drivers for factors like attention, fatigue, and distraction has grown. This is particularly important for developing intelligent transportation systems. Early Driver Monitoring System technologies were often costly and intrusive, involving biomedical sensors or complex onboard systems. The advent of computer vision, leveraging advancements in information technologies, has allowed for non-invasive and cost-effective driver monitoring system solutions.

According to M. Desai et al. (2017) [11], vehicle tracking and driver monitoring systems are becoming more crucial in the present day. Computing and Internet of Things technologies are more common in revolutionizing system design. The main purpose of installing driver monitoring systems is to make it easier for the automotive vehicle industry's research and development (R&D) teams to validate vehicle designs. Advanced driver monitoring solutions are required because manual records of test vehicles have proven to be a difficult and complex process for data analysis and comparison studies.

E. B. Panganiban et al., 2017 [15], state that transportation plays an important role in our society. The human population is growing along with the number of vehicles. Many problems are encountered when it comes to security, such as vehicle entries and exits from one place to another. The proposed driver monitoring system can assign and monitor vehicles entering the area from time to time with a record of vehicle data. This can resolve the difficulty of manually monitoring every vehicle entering an area from time to time, which is a too complex task and time-consuming.

S. Dukare et al., 2015 [7], address the importance of vehicle tracking and driver monitoring systems in issues with vehicle tracking, monitoring, and alerting. The main problem in public transportation is the uncertainty of waiting time due to traffic jams or any other issues like abnormal conditioning. The driver monitoring system can help to access real-time vehicle location data and predict time of arrival. In addition, the driver monitoring system can also guarantee the safety of passengers by monitoring the real-time vehicle location. Furthermore, the alerting system used in public transportation informs observers of the exact location of the vehicle to increase safety.

To follow the owner’s vehicle when others drive the owner's car to a different place, such as an urban area, Tummanapally, S. S et al., 2017 [10], presented the concept of a Smart Vehicle Tracking System, which is also a type of driver monitoring system that utilizes technologies to enhance vehicle security. It emphasizes the importance of security frameworks in protecting vehicles from loss or damage. The system's primary goal is to provide a cost-effective solution for real-time tracking and monitoring of vehicles, ensuring their safety and security. The driver monitoring and alerting system allows users to track their vehicles through a web page on their smartphones, enabling a quick response in case of theft or emergencies.

Driver Monitoring Systems (DMS) have evolved significantly, integrating computer vision and IoT technologies to become more cost-effective and non-invasive. Early systems were costly and complex, but advancements have facilitated better monitoring of driver behavior, vehicle tracking, and enhancing road safety. M. Desai et al., 2017 [11], noted the importance of these systems for automotive R&D, while E. B. Panganiban et al., 2017 [15], emphasized their role in improving vehicle security and monitoring. S. Dukare et al., 2015 [7], highlighted the benefits of DMAS for public transportation, including real-time location tracking and safety. Tummanapally, S. S et al., 2017 [10], discussed the integration of vehicle security frameworks, making tracking and monitoring more efficient and accessible.

In conclusion, the driver monitoring and alerting system was designed to monitor the safety of passengers in the vehicle with advanced security. The drivers' behavior, such as fatigue driving and vehicle theft activity, has been observed since the last decade P. A. Shinde et al., 2015 [4]. Nowadays, DMAS should focus on real-time monitoring and tracking of the vehicle while also storing and updating its database in certain situations, such as over speeding, path tracking and location tracking.

2.3 Vehicle location tracking

In this research, Tummanapally, S. S. et al., 2021 [10], propose a Smart Vehicle Tracking System using GPS and GSM Technologies to address growing concerns related to vehicle security. The system utilizes an Arduino UNO microcontroller board, GPS NEO-6M receiver, GSM 900A module, and a 16*2 LCD for real-time vehicle location tracking. The GPS module gathers location information such as longitude and latitude, which is transmitted via GSM to the Arduino UNO, enabling users to access the vehicle's location through SMS on their mobile devices. The system has the advantage of low-cost implementation, but it also has a limitation of the GSM module's compatibility only with 2G networks, which might affect real-time location tracking in rural areas.

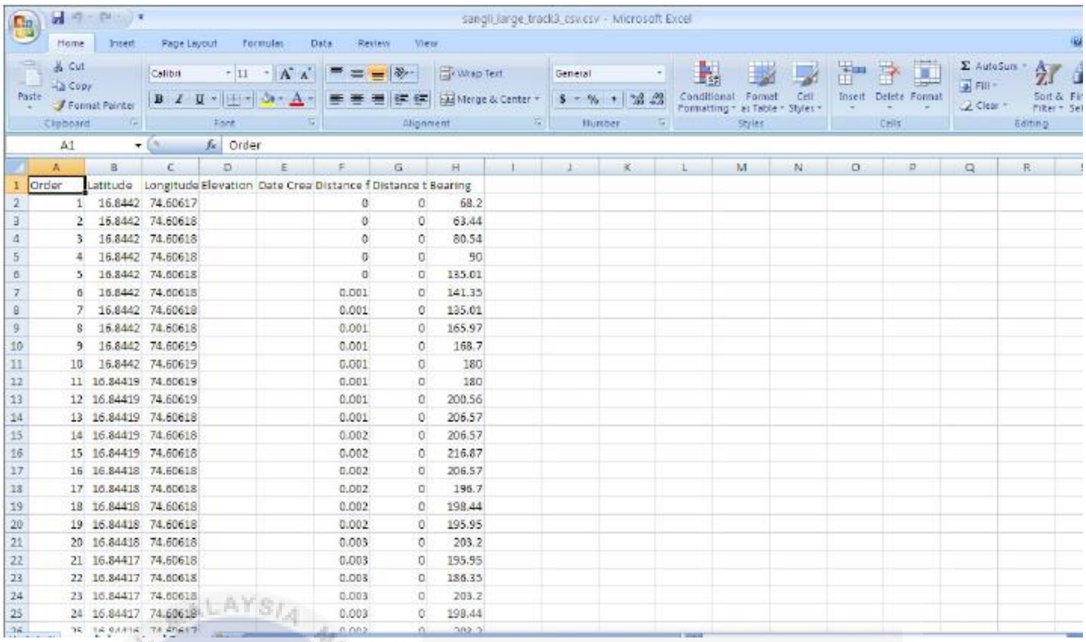
The presented research in the International Journal of Trend in Scientific Research and Development by Hlaing, N. N. S et al., 2019 [16] introduces a GPS and GSM-based vehicle tracking system using Arduino UNO. The method involves the integration of a GPS module to determine the vehicle's coordinates and a GSM module to transmit this information to a database and the user through SMS. The system also incorporates an optional LCD for real-time display and utilizes Google Maps for location visualization on a smartphone. The real-time result can be obtained from the serial monitor in the Arduino IDE, as shown in Figure 2-1. The results demonstrate the effective minute-by-minute tracking of a moving vehicle, providing latitude and longitude details. Limitations may arise in areas with poor GPS signal reception. The advantages of the system include cost-effectiveness, ease of implementation, and continuous monitoring capabilities. Future plans may involve improving GPS signal reception, exploring compatibility with higher GSM generations, and enhancing the system's features.

Figure 2-1 : Serial Monitor Vehicle tracking for Arduino IDE [16]

The research paper by M. S. Uddin et al., 2017 [17] introduces a Smart Anti-Theft Vehicle Tracking System designed for Bangladesh and rooted in the Internet of Things (IoT). The system relies on GPS, GSM/GPRS, and microcontrollers via the Arduino-based LinkIt ONE development board. The in-vehicle device integrates GPS for location data, GPRS/GSM for cloud connectivity, and an RFID-based driver authentication system to bolster security. The proposed mobile application facilitates real-time tracking and remote control of the vehicle's fuel line, constituting an effective anti-theft measure. The smartphone which configured with the developed application, successfully visualizes the GPS location from the in-vehicle device through IoT-cloud in real-time, as depicted in Figure 2-2. Future improvements could involve employing different IoT technologies to enhance the system's functionality.

Figure 2-2 : Smartphone visualise vehicle location [17]

P. A. Shinde et al., 2015 [4], proposed an advanced driver monitoring and alerting system, presenting a comprehensive solution for real-time monitoring of school vehicles using Embedded Linux technology, specifically the Raspberry Pi. Utilizing the Global Positioning System (GPS) module, longitudes and latitudes are stored in a particular file format inside the Raspberry Pi's database. The Raspberry Pi's file system stores longitudes and latitudes of the vehicle's owner-decided path in a specific file format which from point A to point B as figure 2-3. The longitudes and latitudes received from GPS are sent to the server with the help of General packet radio service (GPRS), enabling tracking of the vehicle's current location on the web page using a smartphone. This system achieves high accuracy in providing longitude and latitude at every second.



Order	Latitude	Longitude	Elevation	Date Crea	Distance f	Distance t	Bearing
1	16.8442	74.60617			0	0	68.2
2	16.8442	74.60618			0	0	63.44
3	16.8442	74.60618			0	0	80.54
4	16.8442	74.60618			0	0	90
5	16.8442	74.60618			0	0	135.01
6	16.8442	74.60618			0.001	0	141.35
7	16.8442	74.60618			0.001	0	135.01
8	16.8442	74.60618			0.001	0	165.97
9	16.8442	74.60619			0.001	0	168.7
10	16.8442	74.60619			0.001	0	180
11	16.84419	74.60619			0.001	0	180
12	16.84419	74.60619			0.001	0	200.56
13	16.84419	74.60618			0.001	0	206.57
14	16.84419	74.60618			0.002	0	206.57
15	16.84419	74.60618			0.002	0	216.87
16	16.84418	74.60618			0.002	0	206.57
17	16.84418	74.60618			0.002	0	196.7
18	16.84418	74.60618			0.002	0	198.44
19	16.84418	74.60618			0.002	0	195.95
20	16.84418	74.60618			0.003	0	203.2
21	16.84417	74.60618			0.003	0	195.95
22	16.84417	74.60618			0.003	0	186.35
23	16.84417	74.60618			0.003	0	203.2
24	16.84417	74.60618			0.003	0	198.44

Figure 2-3 : Data recorded from point A to point B [4]

The presented work describes the design and implementation of a Real-Time Vehicle Monitoring, Tracking, and Controlling System by P. Jyothi et al., 2016 [17]. Focusing on addressing issues of vehicle theft and driving insecurity, the system employs an LPC2148 Microcontroller, Liquid Crystal Display (LCD), GPS module, and GSM SIM900A module to monitor the location of vehicles in real-time. The integration of GPS provides accurate vehicle location, while GSM enables alert messages to be sent to the owner's mobile device. The experimental results successfully show latitude and longitude, which are the GPS coordinates, and display them on the LCD in real-time for monitoring as figure 2-4. The system offers real-time tracking and monitoring with security alerts but can be enhanced with more precise results and additional safety features if the microprocessor Raspberry Pi were selected.

Figure 2-4 : Liquid Crystal Display (LCD) print longitude and latitude [17]

The study presents the design and implementation of a Real-Time Google Map and Arduino-Based Vehicle Tracking System by Md. Marufi Rahman et al., 2016 [9]. The Arduino-based system integrates GPS and GSM technologies to provide continuous monitoring of a vehicle's location, as shown in Figure 2-5. GPS modules obtain geographic coordinates, and GSM modules transmit the location data to the owner's cell phone through SMS, as depicted in Figure 2-6. The location is also displayed on an LCD, and Google Maps are utilized to visualize the real-time position. The experimental results demonstrate successful tracking and reporting of the vehicle's location. Google Maps also used to test the accuracy of the coordinates obtained by the system, as shown in Figure 2-7. The advantages of the system lie in its low-cost, user-friendliness, and potential applications in fleet management and transportation systems. Enhancements such as distance calculation were suggested for better checking of coordinate results.

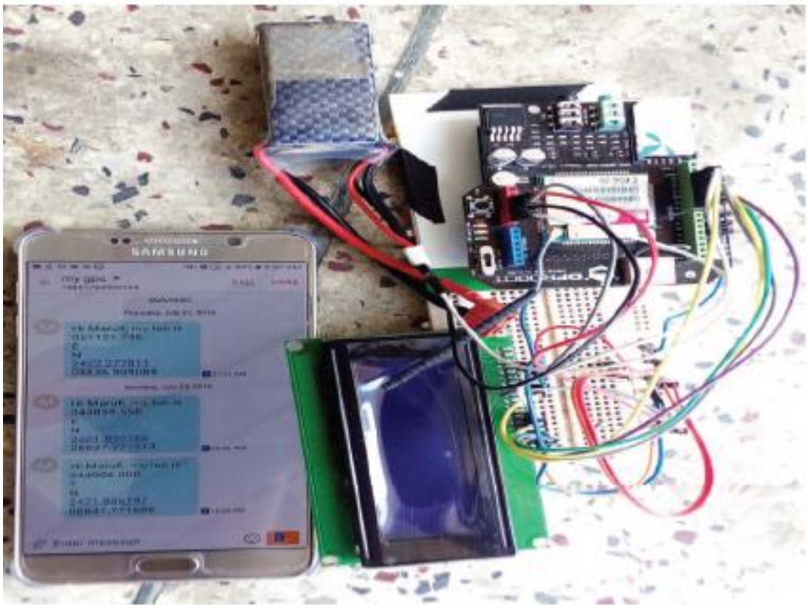


Figure 2-5 : System prototype [9]

Figure 2-6 : SMS vehicle location [9]

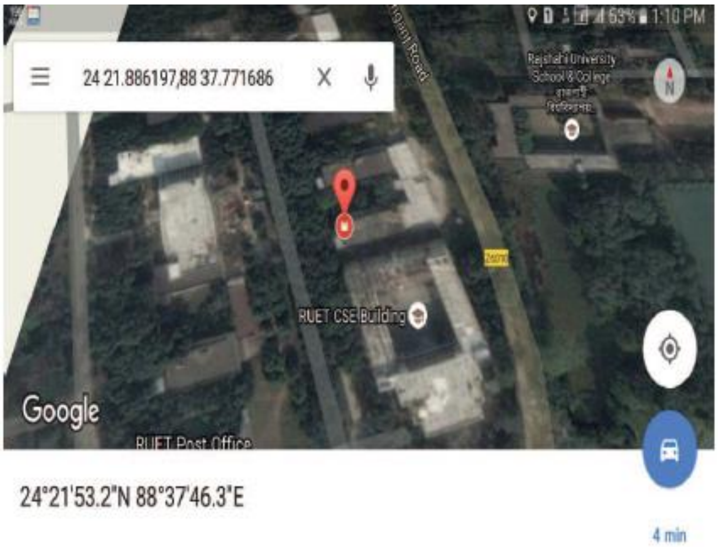


Figure 2-7 : Google Map location [9]

The existing vehicle tracking systems largely utilize GPS and GSM technologies integrated with microcontrollers such as Arduino UNO and Raspberry Pi for real-time tracking. Systems like the Smart Vehicle Tracking System proposed by Tummanapally, S. S. et al., 2021 [10], use GPS for location data and GSM for communication, allowing users to track vehicles via SMS. Hlaing, N. N. S. et al., 2019 [16], and M. S. Uddin et al., 2017 [17], further demonstrate the effectiveness of such systems in providing real-time monitoring and enhancing vehicle security. These systems are cost-effective, easy to implement, and offer continuous monitoring capabilities. They often include LCD screens for real-time data display, as discussed by P. Jyothi et al., 2016 [17]. However, limitations such as reliance on 2G networks for GSM and challenges in areas with poor GPS reception highlight the need for improvement. Systems employing microcontrollers with GPS modules, as discussed by P. A. Shinde et al., 2015 [4], add additional layers of security and real-time tracking through direct data storage on microcontrollers.

Despite their advancements, current vehicle tracking systems face challenges in real-time data accuracy, especially in urban environments with poor GPS signal reception. The reliance on 2G GSM networks limits their functionality in areas where these networks are obsolete. Future research should focus on enhancing the accuracy and reliability of GPS data and exploring higher-generation GSM networks only where necessary. A significant improvement would be developing systems that save coordinates from point A to point B directly onto an SD card in real-time. This method can reduce errors and delays caused by IoT and Wi-Fi connectivity issues. Additionally, improving database management for real-time storage and updates of vehicle coordinates is crucial. Developing more robust systems that can seamlessly integrate with existing vehicle infrastructure without relying on IoT will significantly enhance the effectiveness and applicability of vehicle tracking and monitoring systems.

Vehicle speed is a crucial parameter in Driver Monitoring and Alerting Systems (DMAS). Accurate speed tracking can be achieved using GPS modules programmed into microcontrollers, with data transmitted to a database for analysis, such as detecting overspeeding incidents. Future enhancements should include the use of SD cards for direct data storage to minimize IoT-related errors, and upgrading to better LCD displays for improved user experience. Ensuring the accuracy of GPS-based speed measurements by comparing them with actual speedometer readings and calculating the percentage of error will enhance the reliability of these systems. These improvements will make DMAS more effective, cost-efficient, and user-friendly.

2.4 Vehicle Speed Tracking

In this project, Nagar, R. et al., 2020 [2] presents a comprehensive IoT-based framework for vehicle speed tracking, leveraging technologies like Global Positioning System (GPS), Global System for Mobile Communications (GSM), and an open-source Arduino microcontroller. The methodology involves the integration of a microcontroller, GPS module, and GSM module, specifically the SIM808 module, to monitor and transmit real-time data on various vehicle parameters, including location, speed, engine compartment temperature, and fuel consumption. The system architecture or system flow chart is shown in Figure 2-8 to illustrate the connections between all the components and how data is transmitted. The GPS parameters with vehicle speed are obtained through Arduino programming. The results demonstrate successful tracking and monitoring of vehicles, with the system effectively displaying latitude and longitude coordinates on an LCD and Google Maps. The limitation of this system is the high cost of the GSM modem. The system can be enhanced by designing a more economical GSM modem or exploring additional wireless communication options to lower the cost and enhance efficiency.

Figure 2-8 : LCD display [2]

In this research by Prashant A. Shinde et al., 2015 [4] an advanced vehicle monitoring and tracking system based on Raspberry Pi is proposed, utilizing GPS and GSM technologies for real-time speed tracking. The system aims to enhance school vehicle safety by continuously monitoring location, speed, and various parameters. The methodology involves an Embedded Linux board, which is Raspberry Pi, interfaced with GPS/GPRS/GSM SIM908 Module to ensure more accurate speed tracking information. The system utilizes the LAMP system as an algorithm to store the longitude, latitude, speed, date, and time into the database of Raspberry Pi using Linux, Apache, MySQL, and PHP (LAMP) as shown in Figure 2-9. The results demonstrate the effectiveness of the system in real-time tracking location and speed. The limitation is that the accuracy of the speed tracking would not be known by the system. Thus, the speed tracking needs to be calculated for the percentage of error using the formula of the percentage of error as shown in Figure 2-10.

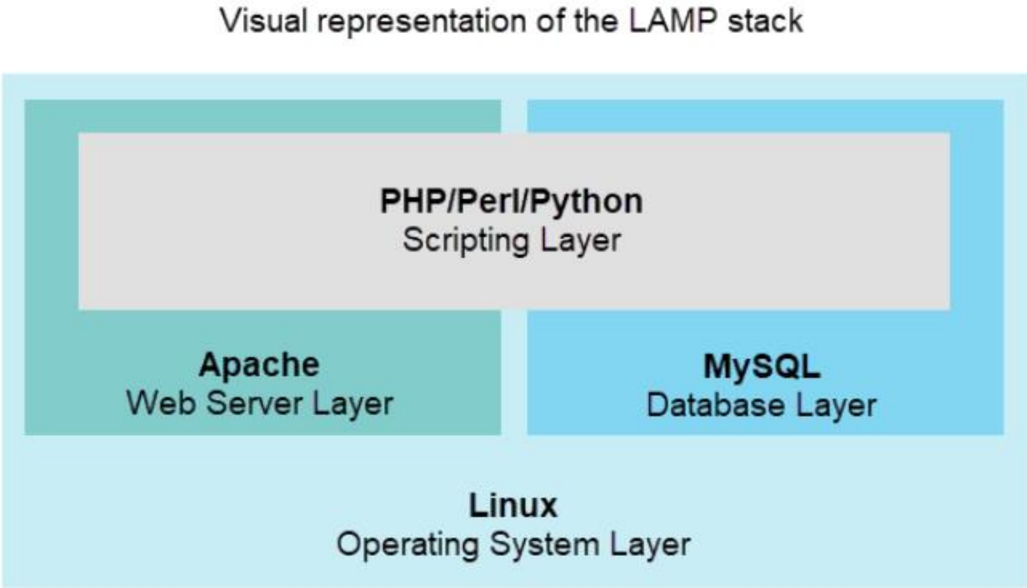


Figure 2-9 LAMP system [4]

$$\text{Percent Error} = \frac{|\,\text{measured} - \text{real}\,|}{\text{real}} \times 100\%$$

Figure 2-10 : Formula for percentage of error [4]

In this study by M. Desai et al., 2017 [11] present an Internet of Things (IoT) based vehicle monitoring system designed for research and development purposes in the automotive industry. Leveraging open-source hardware, the system utilizes an Arduino microcontroller and a SIM808 GPS/GSM/GPRS module to monitor various vehicle parameters, including location, speed, engine compartment temperature, and fuel consumption. Arduino is configured and interfaced with SIM808 to obtain GPS coordinates and vehicle speed, monitored in Arduino's serial monitor as shown in Figure 2-11. Arduino is programmed to transfer the collected data to a centralized server for further analysis and record-keeping on the webserver as illustrated in Figure 2-12. The system utilizes IoT and an open-source platform, making it efficient and cost-effective for monitoring vehicle parameters.

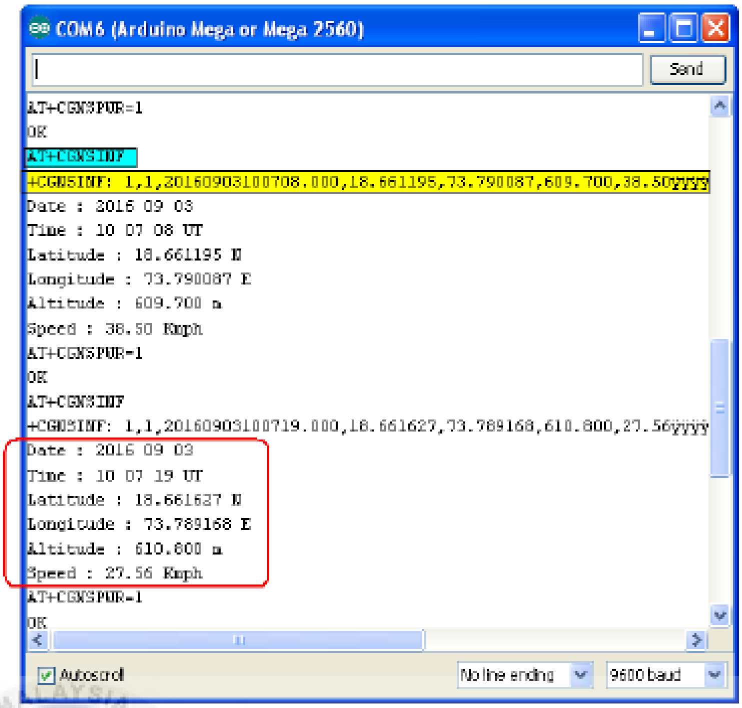
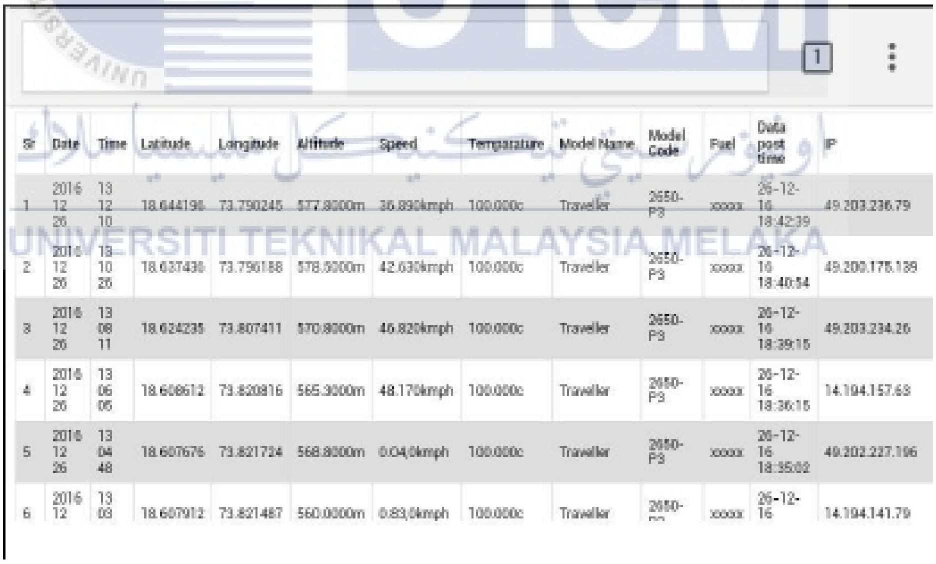


Figure 2-11 Arduino Serial monitor [11]



Sr	Date	Time	Latitude	Longitude	Altitude	Speed	Temparature	Model Name	Model Code	Fuel	Data post time	IP
1	2016 12 26	13 12 10	18.644196	73.790245	577.8000m	36.890kmph	100.000c	Traveller	2650-P3	xxxx	26-12-16 18:42:39	49.203.236.79
2	2016 12 26	13 10 26	18.637436	73.796188	578.5000m	42.630kmph	100.000c	Traveller	2650-P3	xxxx	26-12-16 18:40:54	49.200.175.139
3	2016 12 26	13 08 11	18.624235	73.807411	570.8000m	46.820kmph	100.000c	Traveller	2650-P3	xxxx	26-12-16 18:39:15	49.203.234.26
4	2016 12 26	13 06 05	18.608612	73.820816	565.3000m	48.170kmph	100.000c	Traveller	2650-P3	xxxx	26-12-16 18:36:15	14.194.157.63
5	2016 12 26	13 04 48	18.607676	73.821724	568.8000m	0.04,0kmph	100.000c	Traveller	2650-P3	xxxx	26-12-16 18:35:02	49.202.227.196
6	2016 12	13 03	18.607912	73.821487	560.0000m	0.83,0kmph	100.000c	Traveller	2650-	xxxx	26-12-16	14.194.141.79

Figure 2-12 Webserver [11]

Current vehicle speed tracking systems effectively integrate GPS and GSM technologies with microcontrollers such as Arduino and Raspberry Pi for real-time monitoring. Notable examples include Nagar, R. et al., 2020 [2], who presented an IoT-based framework, and Prashant A. Shinde et al., 2015 [4], who proposed a Raspberry Pi-based system for school vehicle safety. These systems leverage GPS for speed and location tracking, displaying data on LCD screens and utilizing Google Maps for visualization. While they demonstrate real-time monitoring and user-friendly interfaces, they face challenges such as the high cost of GSM modules and the need for accurate speed tracking verification. Moreover, reliance on IoT and wireless communication introduces potential delays and errors in data transmission. Future research should focus on reducing the cost of communication modules, improving data accuracy in areas with poor GPS reception, and verifying speed measurements through error percentage calculations. Additionally, incorporating web servers to create result tables can improve data accessibility and facilitate real-time analysis and reporting.

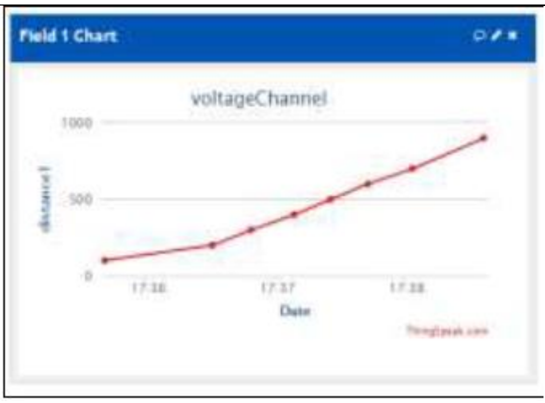
Vehicle speed is a crucial parameter in Driver Monitoring and Alerting Systems (DMAS). Accurate speed tracking can be achieved using GPS modules programmed into microcontrollers, with data transmitted to a database for analysis, such as detecting overspeeding incidents. Future enhancements should include the use of SD cards for direct data storage to minimize IoT-related errors and upgrading to better LCD displays for improved user experience. Ensuring the accuracy of GPS-based speed measurements by comparing them with actual speedometer readings and calculating the percentage of error will enhance the reliability of these systems. Additionally, integrating a web server to generate result tables can further improve the system's functionality by enabling comprehensive data analysis and reporting. These improvements will make DMAS more effective, cost-efficient, and user-friendly.

2.5 Vehicle Overspeed Detection

In this study M. A. Khan et al., 2018 [12] address the critical issue of over-speeding as a major contributor to road accidents, proposing a comprehensive system for real-time monitoring and enforcement. The system, implemented through Arduino Mega and the SIM908 module, utilizes GPS and GSM technologies to continuously calculate the speed and GPS coordinates of vehicles. The method involves preloading defined speed zones with corresponding maximum speed limits, employing a point-in-polygon algorithm to determine the vehicle's location. If an over-speeding violation is detected, an active buzzer alerts the driver, and an SMS containing vehicle details, GPS coordinates, and speed is automatically sent to traffic authorities. Speed tracking accuracy was predicted by a speed app using radar, showing around 40 to 80% accuracy based on the factor of internet speed as well as connectivity. The limitation of the system, such as signal dependency, needs continuous monitoring. The system has the benefit of low cost, remote monitoring using IoT, and high reliability. The integrated hardware solution should have more exploration for future plans to increase portability and reduce power supply complexity.

In this study by Nagar, R et al.,2020 [2] an Internet of Things (IoT) framework for intelligent vehicle monitoring on high-speed expressways is proposed. The system utilizes Radio Frequency Identification (RFID) sensors for real-time vehicle tracking, offering a low-cost and reliable solution compared to image processing-based systems. The architecture employs an array of RFID sensors placed at equal distances along the expressway, with vehicles equipped with passive RFID tags. The RFID reader nodes are connected to the internet via WiFi or Ethernet, enabling real-time data transmission to a cloud server. The study demonstrates the ability to generate distance versus time plots, allowing for the calculation of vehicle velocity. The use of Arduino and cloud computing facilitates remote monitoring, providing a potential solution for intelligent traffic management. The experimental results illustrate the successful implementation of the system, showcasing real-time distance versus time plots and vehicle velocity estimation in table 2-1. The system had advantages by reliability and is easy to complete by avoiding the complex image processing algorithm. The limitation of the system relies on consistent network connectivity and potential weather conditions. The system can be improved for traffic safety by adding an alerting system such as Light Emitting Diode (LED) or buzzer when the speed limit is exceeded to notify the driver or the observer.

Table 2-1 : Plot graph from the system [2]

Real-time distance versus time plot	Vehicle velocity estimation plot
	

The study conducted by Jeddi et al., 2013 [13] presents a study approach to addressing the issue of over-speeding and minimizing road accidents through the design and implementation of an embedded system. Unlike traditional radar systems with limited coverage, the system involves an in-vehicle monitoring and recording device equipped with GSM/GPS technology, the AT32UC3A3256 microcontroller, and SD cards as shown in figure 2-13. This recording device, made up of a microcontroller, GPS, and GPRS module, transmits real-time data, including the vehicle's speed and location, continuously to a central server. The server compares the actual speed with defined speed limits for each road, identifying speed violations and maintaining a comprehensive record in its database. The LED blink alert system was used to detect when the vehicle exceeds the speed limit. The testing of the complete system was done for every component to ensure functionality and effectiveness in detecting speed violations during real-time trials. This system has the advantage of recording detailed information on speed violations in an SD card and is cost-effective using an IoT server. However, the limitation of Wi-Fi reliance and network connectivity will need caution. Further optimization, scalability assessments, and integration with advanced accident prevention mechanisms are needed to enhance overall road safety.

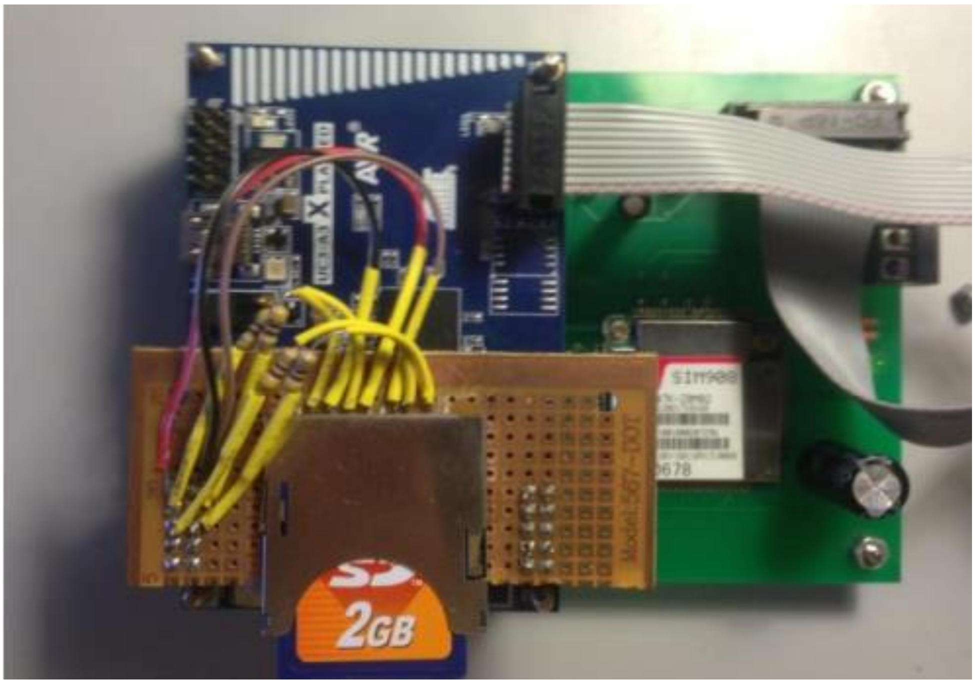


Figure 2-13 : System prototype with SD card [13]

The studies on vehicle overspeed detection present various innovative approaches to enhance road safety by leveraging GPS and GSM technologies. M. A. Khan et al., 2018 [12], utilized Arduino Mega and SIM908 modules to create a real-time monitoring system with speed zone enforcement, though it faced challenges in speed tracking accuracy and signal dependency. Nagar, R. et al., 2020 [2], proposed an IoT framework using RFID sensors for vehicle tracking on expressways, offering a low-cost alternative but requiring consistent network connectivity. Jeddi et al., 2013 [13], introduced an embedded system with GSM/GPS and SD cards for in-vehicle monitoring, emphasizing detailed speed violation records but also relying heavily on network connectivity. Common strengths across these studies include low-cost implementation, real-time monitoring, and potential for remote management. However, limitations such as signal dependency, accuracy verification, and network connectivity issues highlight areas needing further research and improvement.

Conclusion, vehicle speed limit tracking in Driver Monitoring and Alerting Systems (DMAS) is crucial for enhancing road safety. Utilizing GPS and GSM technologies, these systems can monitor and record real-time vehicle speeds, sending data to central servers for analysis and enforcement. Effective alert mechanisms, such as buzzers and

LCD displays, can notify drivers and observers of speed violations. Future enhancements should focus on integrating SD cards for reliable data logging, improving accuracy through error correction algorithms, and ensuring robust network connectivity. Additionally, implementing a system to generate plot graphs from the stored data on the SD card can provide visual representation and analysis of overspeed incidents. Addressing these challenges will enhance the effectiveness, reliability, and user-friendliness of DMAS for real-time speed monitoring and road safety enforcement.

2.6 Vehicle path Monitor

The Advanced Vehicle Monitoring and Tracking System, based on Raspberry Pi and developed by P. A. Shindeet al., 2015 [4], offers a comprehensive solution for real-time tracking and monitoring of vehicles, specifically designed for school buses. The system employs GPS, GPRS, and GSM technologies, utilizing the SIM908 Module, to provide accurate location, speed, and time information. Whenever the driver operates the vehicle, the longitudes and latitudes of the path from A to B are compared with the current data received from the GPS/GPRS/GSM SIM908 Module. If a wrong path detected the system sends an alert message to the vehicle owner's mobile to notifying the wrong path detectioin .The Expert GPS software interprets the stored longitude and latitude in Raspberry Pi's specific format, enabling the tracing of any path from location A to location B, as illustrated in Figure 2-14. The system offers tracking features on a webpage with different user logins:

(i) Super User Login for owners to access and modify data,

(ii) Primary User Login for students to view school vehicle tracking

(iii) Secondary User Login for registered parents to track the school vehicle through the online webpage.

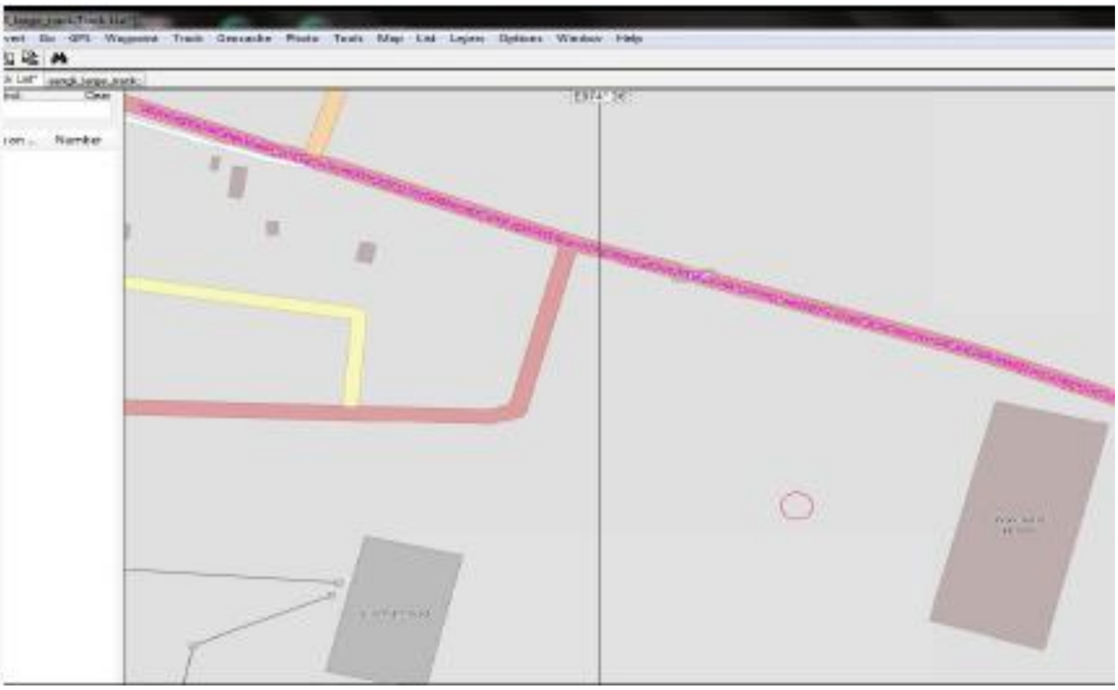


Figure 2-14 : Expert GPS software [4]

Jeddi et al., 2013 [13] proposed a paper introduces an Over-Speed Monitoring and Recording System designed to minimize road accidents caused by over-speeding. The system incorporates an embedded device in each vehicle, consisting of GPS/GSM modules, an impact detector, SD card, and a 32-bit microcontroller. This device continuously monitors and transmits real-time information, including vehicle speed and location, to a central server. The server loads GPS data, including vehicle longitude, latitude, and speed, onto OpenStreetMap.org (OSM.org). OpenStreetMap.org was a website providing free geographic data, offers advantages such as easy updates and edits based on uploaded data. Additional information can be added to the map using Java OpenStreetMap Editor (JOSM), a desktop application shown in Figure 2-15. The system provides a reliable, effective, and straightforward solution for updating maps on the Driver Monitoring and Alerting System (DMAS).

Figure 2-15 : Java OpenStreetMap Editor (JOSM) [13]

The presented paper focuses on developing a vehicle tracking system that leverages Internet of Things (IoT) technologies to monitor and analyze various parameters of test vehicles by Nagar, R et al.,2020 [2]. The system incorporates GPS, OBD, RFID, and an open-source controller, along with GPS, GSM, and GPRS modules for data transfer. The proposed system aims to track location, vehicle speed, engine compartment temperature, fuel consumption, and other parameters from a centralized place for research and development purposes. The parameters are stored in the database on a webserver, and a webpage is created to display vehicle parameter data. The vehicle location is simultaneously linked to Google Maps, displaying the vehicle's position on the map, as illustrated in Figure 2-16. The system has advantages of reliability and real-time tracking, allowing the position of the vehicles to be tracked using Google Maps and showing the running position of the vehicle in real-time. The system could be enhanced by incorporating IoT alert messages to notify observers through the webpage.

Figure 2-16 : Google Maps Plotting [2]

The presented smart vehicle tracking system, utilizing Arduino Uno interfaced with GPS and GSM modules, demonstrates an effective solution for real-time monitoring and location tracking by A. H. Alquhali et al., 2019 [8]. The system's method involves the GPS module fetching precise latitude and longitude, transmitted through the GSM module to ThingSpeak. The resulting data is then visualized on Freeboard, displaying the vehicle's exact location on Google Maps. Figure XXX shows the location of the vehicle on Freeboard in the form of latitude and longitude on Google Maps. The latitude of the vehicle can be found on the bottom left of the Freeboard, while the longitude is in the middle of the dashboard. The date and time can be seen in the bottom right corner of the dashboard as well. The users can also see the exact location on Google Maps, as shown by the red marker in Figure 2-17, and the path along the way was plotted.

Figure 2-17 Freeboard dashboard [8]

The studies on vehicle path monitoring systems reveal various innovative approaches to enhance real-time tracking and road safety. P. A. Shinde et al., 2015 [4], developed a comprehensive vehicle monitoring system using Raspberry Pi, GPS, GPRS, and GSM technologies to track school buses. This system provides multiple user logins for different stakeholders and alerts vehicle owners of wrong path detections. Jeddi et al., 2013 [13], introduced an over-speed monitoring system that uses GPS/GSM modules and SD cards to upload real-time data to OpenStreetMap.org, allowing for easy updates and map edits. Nagar, R. et al., 2020 [2], proposed a vehicle tracking system integrating GPS, OBD, RFID, and open-source controllers, storing data on a webserver and displaying vehicle locations on Google Maps. A. H. Alquhali et al., 2019 [8], utilized Arduino Uno with GPS and GSM modules to transmit vehicle data to ThingSpeak, visualizing the information on Freeboard and Google Maps. These studies showcase strengths such as real-time monitoring, accurate path tracking, and user-friendly interfaces. However, they also face challenges like network dependency, signal accuracy, and the need for reliable data transmission and storage.

Understanding the vehicle path is crucial for visualizing and tracking the vehicle's journey in Driver Monitoring and Alerting Systems (DMAS). Effective path monitoring can be achieved through the integration of GPS technologies with microcontrollers, enabling real-time data collection and transmission in sd card. Utilizing platforms like OpenStreetMap.org enhances the user experience by providing clear visual representations of the vehicle's path. Future improvements should focus on integrating SD cards for reliable data logging, developing applications to analyze and upload SD card data to visualise all infomration in one application like freeboard which include the OpenStreetMap.org , table and graph, and implementing different colour of line to visualise normal and overspeed behavior dueing the route map. These enhancements will improve the reliability, accuracy, and user-friendliness of DMAS for real-time vehicle path monitoring and road safety enforcement.

2.7 Driver Behavior Analysis

In reviewing the paper by G. Castignani et al., [19] on driver behavior using fuzzy logic, previous research has generally focused on integrating various data sources such as GPS, accelerometers, and motion sensors to detect driving events. Methods typically involve the application of fuzzy logic for event detection, leveraging input from multiple sensors and external data like weather conditions, often sourced from APIs such as OpenWeatherMap. Studies, including those by SenseFleet, have shown that fuzzy logic can effectively classify driving behaviors into events like hard braking, acceleration, and over-speeding by defining fuzzy sets and rules tailored to sensor data. The process typically involves several steps: first, event detection, where motion sensor and GPS data are fused and analyzed using a fuzzy inference system to identify risky driving events; second, scoring, where detected events are evaluated based on their severity and context, such as weather conditions and time of day, to calculate an overall driver score.

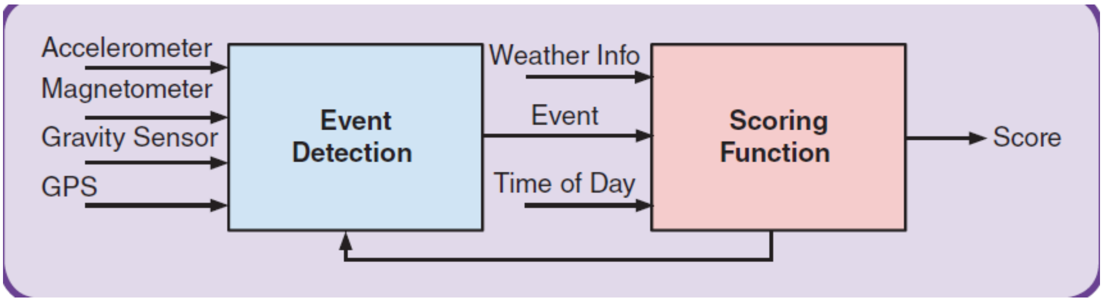


Figure 2-18 : Event detection and scoring [19]

The results from these studies indicate high accuracy in event detection and the ability to distinguish between different driving styles. However, limitations remain, such as the dependency on high-cost subscription services for weather data like the OpenWeatherMap API, which can increase operational costs. Additionally, the calibration phase necessary for adapting fuzzy set limits to different vehicles and devices can be non-representative if not conducted properly. Despite these challenges, the main advantage of using fuzzy logic lies in its flexibility and robustness in handling uncertain and imprecise data, leading to more reliable driver profiling. Furthermore, platforms like SenseFleet have been designed to be low-cost by utilizing a self-developed mobile application and an internal SQLite database for data storage, making the solution more accessible. Our proposed research aims to address these gaps by exploring cost-effective alternatives choice like time fo day others than the weather data and enhancing the calibration process to ensure representativeness across diverse driving conditions, thereby improving the overall reliability and scalability of fuzzy logic-based driver behavior analysis. By leveraging a low-cost platform and refining event detection and scoring mechanisms, we aim to develop a more robust and economically viable solution for driver monitoring and behavior profiling.

Previous studies by I. S. Feraud, M. M. Lara et al.,[20] on driver behavior using fuzzy logic have employed various methods such as fuzzy inference systems (FIS), adaptive neuro-fuzzy inference systems (ANFIS), and hybrid models integrating fuzzy logic with neural networks and genetic algorithms. These methods involve creating membership functions and rule-based systems to evaluate driving scenarios. Results indicate that fuzzy logic models effectively capture the uncertainty in driver behavior, accurately predicting decisions in scenarios like lane changing and speed regulation, with some models applied in real-time systems for road safety improvement.

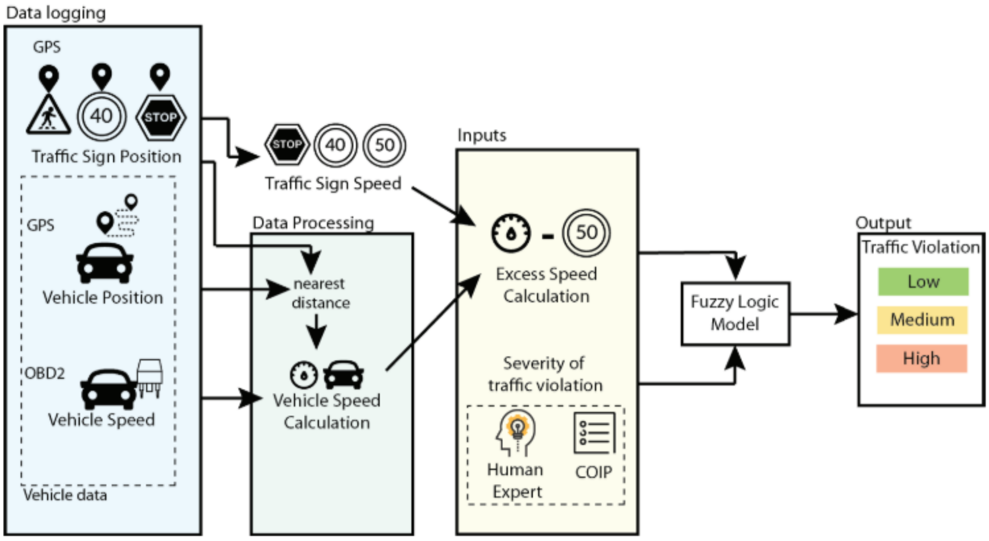


Figure 2-19 : Data logging and data processing for input variables calculation [20]

However, these studies face limitations, including the complexity of designing accurate membership functions, computational intensity, and challenges in integrating with existing traffic management infrastructure. Additionally, many studies rely on simulated or small datasets, limiting their real-world applicability. Fuzzy logic offers advantages in handling imprecise information and providing a flexible, interpretable framework. However, a significant research gap exists due to the lack of comprehensive real-world data and underexplored integration with connected vehicles and smart infrastructure. To address these gaps, future research should focus on extensive real-world data collection using advanced data logging and processing methods to calculate input variables, develop sophisticated models reflecting dynamic driving conditions, and integrate fuzzy logic with emerging technologies. This involves gathering real-time data, creating adaptive membership functions, and validating models in real-world scenarios to enhance traffic safety and efficiency.

2.8 Comparison Table

Table 2-2 : Comparison for Literature Review

Author	Hardware	Method	Result
M. Desai et al., 2017[11]	Arduino UNO (Microcontroller) SIM 808 (GPS module)	Vehicle location Vehicle speed Google maps	The vehicle parameters send by Arduino and SIM 808 to database and linked to Google maps for real time display.
P. A. Shinde et al.,2015 [4]	Rasberry Pi (Microcontroller) SIM 908 (GPS Module) DS18B20 (Temperature Sensor) MQ6 (Gas sensor)	Path detection Vehicle location Gas and temperature detection	The GPS module receives the location and compares it with the designated path. If there is an error in the path, or if a gas leakage is detected, or if the temperature exceeds the specified limit, an alert message will be sent using GSM.

A. H. Alquhali et al., 2019 [8]	Arduino UNO (Microcontroller) SIM 900A (GSM module) GYNE06MV2 (GPS module) LCD (Output Display)	Vehicle Location IOT	The GPS module receives the location and utilizes the GSM module to send the data to the Thing speak website. The information is then displayed on the Freeboard website.
M. M. Rahman et al., 2016 [9]	Arduino UNO (Microcontroller) SIM 908 (GPS module) LCD (Output Display)	Vehicle location Google map SMS	The vehicle coordinate send by Arduino and SIM 908 to phone through SMS and LCD display the latitude and longitude. The latitude and longitude can define vehicle location using google maps.
Nagar, R. et al., 2020 [2]	Arduino UNO (Microcontroller) GM 330D(GPS Module) SIM 808 (GSM module) LCD (Output Display)	Vehicle location Google map Mobile	The GPS module receives the signal and send data to Arduino. If the timer is up to date the data will transmit to mobile and show display in Google maps.

Tummanapally, S. S. et al., 2023 [10]	Arduino UNO (Microcontroller) SIM800L (GSM module) GYNE06MV2 (GPS module) KY-038 (Microphone) LCD (Alert Display) Vibration Sensor	Vehicle Location Vehicle Speed SMS Google map	The GPS module detects the vehicle's location while in motion and sends an alert via SMS to a mobile device in the event of a collision. The real time location also shows in google maps at the same time.
M. A. Khan et al., 2018 [12]	Arduino Mega SIM908 module GPS and GSM Module Buzzer	Overspeed detection Vehicle Location SMS Speed app using radar	If an over-speeding violation is detected, an active buzzer alerts the driver, and an SMS containing vehicle details, GPS coordinates, and speed is automatically sent to traffic authorities.

Jeddi et al., 2013 [13]	AT32UC3A3256 microcontroller SD cards GPS module GSM module LED	Data reording Vehicle Location Vehicle Speed SMS	The server compares the actual speed with defined speed limits for each road, identifying speed violations and maintaining a comprehensive record in its database.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will provide an overview of the methodology, including the techniques and methods used in this project. The chapter first using the flow chart to describe the overview of how to implement the Driver Monitoring and Alerting System. It will lead through the project step-by-step from start to finish. The system overview that follows will outline the process by which the driver monitoring system and driver alerting systems will collaborate in order to form a Driver Monitoring and Alerting System (DMAS). The next section of the chapter will present the block diagram of the Driver Monitoring and Alerting System (DMAS), which shows how the input, control, and output units are connected and how data is transmitted between them. This chapter will then cover the experiment design for the Driver Monitoring and Alerting System (DMAS). This project has two objectives, and multiple experiment designs have been developed to meet each objective. This section will cover over every component of software and hardware that is used, along with how to choose each one and consider its advantages and disadvantages as well.

3.2 Project Overview

An overview of the project is provided by the flow chart in Figure 3-1 below. To provide a basic understanding and project recommendations, the project started with preliminary research on previously Driver Monitoring and Alerting Systems (DMAS). The project proceeded on to identify key requirements, such as the problem statement, the objectives, the scope, and the limitations. Further research on Driver Monitoring and Alerting Systems (DMAS) was conducted, leading to the writing of the literature review. The project draft was then designed and developed to meet the specified objectives After confirmation, hardware purchasing was carried out, followed by experimentation on the project prototype. The testing results were recorded and analyzed to ensure they met the project requirements. Finally, in the last phase of the project flow, the report was prepared and submitted on time. The Gantt Chart was provided to visualize the planning of the project according to the time at Appendix A.

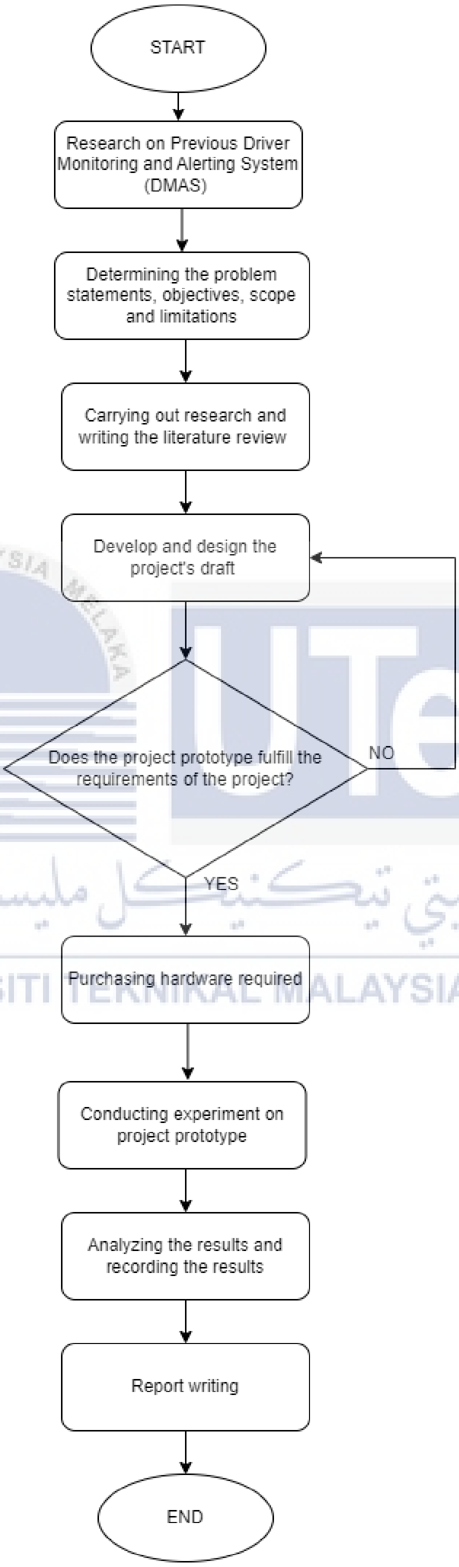


Figure 3-1 : Project flowchart

3.2.1 Project Design

3.2.1.1 Hardware Design

Hardware Selection

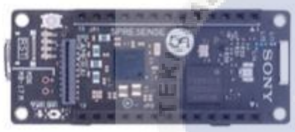
For DMAS project, the Raspberry Pi 5, Arduino Uno, and Sony Spresense microcontrollers are under consideration for microcontroller sleection. With careful consideration of their features, advantages, and disadvantages, one of the microcontroller will be selected for main baord for DMAS.

The Raspberry Pi 5 offers a high processing speed of 1.5 GHz, built-in Network Time Protocol (NTP), and integrated Bluetooth and Wi-Fi modules with providede AD card slot, making it suitable for computation-heavy tasks and seamless connectivity. However, its large size (85.6 x 56 mm) and highest power consumption (2.7 W) are significant drawbacks, particularly for projects with space constraints or limited power availability. Additionally, the need for an external GPS module adds to the complexity and cost.

The Arduino Uno is known for its medium size (68.6 x 53.4 mm) and lowest power consumption (0.25 W). However, its processing speed is the lowest (16 MHz), which limits its suitability for complex applications. It also requires additional modules for GPS, Bluetooth, Wifi and SD Card functionalities, complicating the hardware design and increasing costs.

The Sony Spresense stands out due to its compact size (50 x 20.6 mm), built-in Global Navigation Satellite System (GNSS) and Real-Time Clock (RTC), low power consumption (0.36 W), and the ability to use an SD card for data recording via the Sony Spresense extension board. These features make it particularly suitable for applications requiring precise positioning, timing, efficient power use, and data logging capabilities, despite its relatively lower processing speed (156 MHz).

Table 3-1 : Comparison Table for Microcontroller

Microcontroller	Advantages	Disadvatages
Raspberry Pi 5 	High processor speed (1.5GHz), built-in NTP, built-in Bluetooth ,Wifi and SD card slot.	Largest size (85.6 x 56mm), requires additional GPS module, highest power consumption (2.7W)
Arduino Uno 	Medium size (68.6 x 53.4mm), lowest power consumption (0.25W)	Requires additional GPS, Bluetooth, and WiFi modules, lowest processor speed (16MHz)
Sony Spresense 	Smallest size (50 x 20.6mm), built-in GNSS, RTC, low power consumption (0.36W), built-in WiFi (with extension board for SD card storage)	Lower processor speed (156MHz), requires additional Bluetooth module

The GNSS capabilities of the Sony Spresense provide a significant advantage over microcontrollers that rely on the GPS-only GEO-NEO-MV2 module for satellite connection. The Sony Spresense supports multiple satellite constellations, including GPS (USA), GLONASS (Russia), Galileo (Europe), and BeiDou (China), enhancing positioning accuracy, reliability, and global coverage. This multi-constellation approach ensures better performance in challenging environments, such as urban canyons or dense forests, by offering more satellite signals for precise positioning and reducing the risk of signal loss or degradation. In contrast, the GEO-NEO-MV2, while reliable for GPS-based positioning, lacks the multi-constellation benefits, which can affect its accuracy in environments with limited satellite visibility. Additionally, the Sony Spresense's built-in GNSS simplifies hardware design by integrating positioning functionality within the main microcontroller, whereas the GEO-NEO-MV2 requires an additional module, increasing overall complexity and power consumption. Notably, the Sony Spresense achieves a typical positioning accuracy of 1.5 meters CEP (Circular Error

Probable) in open sky conditions, compared to the 2.5 meters CEP accuracy of the GEO-NEO-MV2, making it a more precise solution for applications requiring accurate positioning.

Table 3-2 : Comparison Table for GNSS and GPS module

Feature	Sony Spresense GNSS	GEO-NEO-MV2 GPS Module
Satellite Constellations	GPS, GLONASS, Galileo, BeiDou	GPS only
Accuracy & Reliability	Higher, 1.5 meters CEP (typical)	Lower, 2.5 meters CEP (typical)
Environment Performance	Better in challenging environments (urban canyons, dense forests), effective area: <5 meters	Limited in challenging environments, effective area: <10 meters
Integration	Built-in GNSS	Requires additional GPS module
Power Consumption	Lower overall consumption, typically <100 mW	Higher due to the need for an additional module, typically >150 mW
Image		

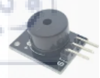
When selecting a display component for the DMAS output, the 2x16 LCD screen and the TFT LCD ST7735 1.8-inch display were compared. The 2x16 LCD screen is simple, easy to use, and consumes higher power, operating at 5V. It is larger in size at 82x35x18 mm but offers a lower resolution, suitable only for text display. However, its graphic capabilities are limited, and its night visibility, even with a backlight, may be suboptimal. On the other hand, the TFT LCD ST7735 1.8-inch display operates at a lower voltage of 3.3V, provides a higher resolution of 128x160, and is more compact at 28.03x35.04 mm. This display supports vibrant colors, faster refresh rates, and superior night visibility due to its integrated backlight. Although it consumes more power, its enhanced visual performance and flexibility make it the preferred choice for advanced applications requiring dynamic visual outputs.

Table 3-3 : Comparison for LCD Display Selction

Display Type	**2x16 LCD Screen**	**1.8-Inch TFT LCD ST7735**
Resolution	Lower, text only	Higher, 128x160
Operating Voltage	5V	3.3 V
Size	82x35x18 mm	28.03x35.04 mm
Advantages	Simple and easy to use , bigger display size	Higher resolution ,Vibrant colors and faster refresh rates, Superior night visibility with integrated backlight
Disadvantages	Lower resolution, Limited graphic capabilities, Night visibility can be limited	Compact display size
Images		

For DMAS application buzzer was choose as the output of the alerting system. The two-pin buzzers feature a simple design with positive and negative pins, requiring external oscillators or microcontrollers for sound generation, making them cost-effective and easy to integrate for basic sound alerts. In contrast, three-pin buzzers offer built-in oscillators and advanced features like volume and frequency control, allowing for more precise sound output but at a higher cost and complexity in integration.The two pin buzzer stand out as more suitable choice as its is simple and low cost to implement in DMAS.

Table 3-4 : Comparison of buzzers

Feature	Two-Pin Buzzer	Three-Pin Buzzer
Structure and Components	Positive (+) and negative (-) pins	Positive (+), negative (-), and signal (S) pins
Operation	Requires external oscillator or microcontroller to generate sound	Often includes built-in oscillator for self-sufficient sound generation
Circuit Design	Simple, minimal wiring	More complex, requires managing the signal pin
Advantages	Simple integration, Low cost, Suitable for basic applications	Built-in oscillator, Enhanced functionality, Precise sound control
Disadvantages	Requires external control, Limited functionality	More complex integration, Higher cost
Images		

In conclusion, the Sony Spresense, with its integrated GNSS, RTC, and SD card capabilities, combined with the TFT LCD ST7735 1.8-inch display and 2-pin buzzer, provides a robust and efficient solution for our project's hardware requirements. This combination offers superior positioning accuracy, efficient power usage, comprehensive data logging, and high-quality visual and audio outputs, making it the optimal choice for our research project.

Hardware Setup

The Figure 3-2 and Table 3-5 below illustrate the wiring diagram of the Driver Monitoring and Alerting System (DMAS). The code uploaded to the Sony Spresense, which includes the functions of the DMAS, is provided in Appendix B.

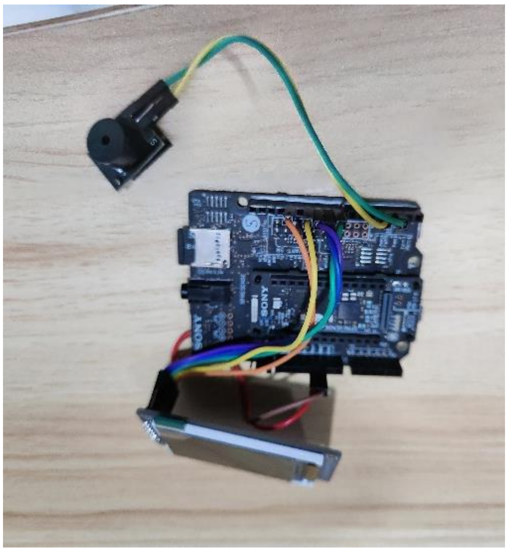


Figure 3-2 : Wiring Diagram of DMAS

Table 3-5 : Hardware Pin Connection

PIN (SONY SPRESENSE)	PIN (BUZZER& TFT LCD)
D3	Positive (Buzzer)
D8	RES (TFT LCD)
D9	DC (TFT LCD)
D10	CS (TFT LCD)
D11	SDA (TFT LCD)
D13	SCL (TFT LCD)
Gnd	Negative (Buzzer), GND (TFT LCD)
5V	VCC (TFT LCD)

After connecting the Sony Spresense with jumper wires according to the hardware pin connections in Table 3-5, a 3D-printed box was designed as figured 3-3 using Autodesk Fushion 360 and printed to cover the electronic components, including the Sony Spresense, TFT LCD display, and buzzer. This enclosure prevents electric shock damage (ESD) and ensures better connectivity.

Figure 3-3 : DMAS box design

Instead of using Polylactic Acid (PLA), Polyethylene Terephthalate Glycol (PETG) was chosen for the 3D printing material. This decision was made because the DMAS will be placed in a car in Malaysia's hot climate. PETG has a higher melting point (180°C) compared to PLA (160°C), making it a more suitable choice. Additionally, PETG offers better strength, wear resistance, and durability than PLA. The flecibility of PETG also provide more impat absorb and reducing risk of crack while PLA is more rigid and an be crack under stress. Thus PETG will be the preferable choice for DMAS box. The full CAD drawing with dimension provided at Appendix C.

3.2.1.2 Software Design

To better visualize the data collected from the Driver Monitoring and Alerting System (DMAS), Visual Studio Code (VS Code) was utilized to create the DMAS application using the Python programming language. Python is well-suited for developing applications that require data visualization and the implementation of Artificial Intelligence techniques. Specifically, fuzzy logic was used to monitor driver behavior effectively. The choice of VS Code as the Integrated Development Environment (IDE) was driven by its versatility, extensive library support, and robust debugging tools. These features make it an ideal platform for developing complex applications such as the DMAS.

Libraries of DMAS applications

The figure 3-4 below shows the necessary library of the DMAS application in VS code:

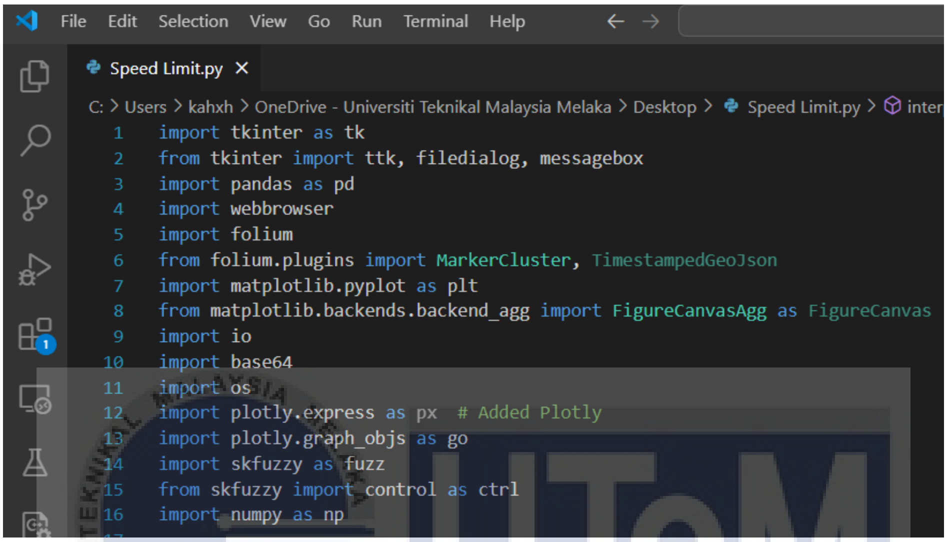


Figure 3-4 : Library of DMAS application

For data handling, Python's libraries such as ‘pandas’ and ‘numpy’ are instrumental. ‘Pandas’ is used for efficient data manipulation and analysis, allowing for easy reading, processing, and storage of coordinate data and speed limits. ‘Numpy’ supports numerical operations and can handle large datasets efficiently, making it a vital tool for managing and analyzing the collected data.

When it comes to data visualization, libraries like ‘matplotlib’ and ‘plotly’ show its ability. ‘matplotlib’ is a powerful plotting library that enables the creation of static, animated, and interactive visualizations in Python. It helps in visualizing driving behavior and identifying deviations from normal behavior. ‘Plotly’, on the other hand, is used for creating interactive plots, which are particularly useful for an engaging and dynamic representation of the data collected by the DMAS.

The implementation of Artificial Intelligence, specifically fuzzy logic, is achieved using Python's ‘scikit-fuzzy’ library. This library provides the tools needed to define fuzzy sets and rules, which are crucial for interpreting driver behavior. By using fuzzy logic, the DMAS can analyze driving patterns and provide insights into driver behavior, categorizing it into different levels of caution or aggressiveness and score of driver behavior.

The library download was done using Command Prompt (CMD) on the computer. The installation process is shown in Figure 3-5.

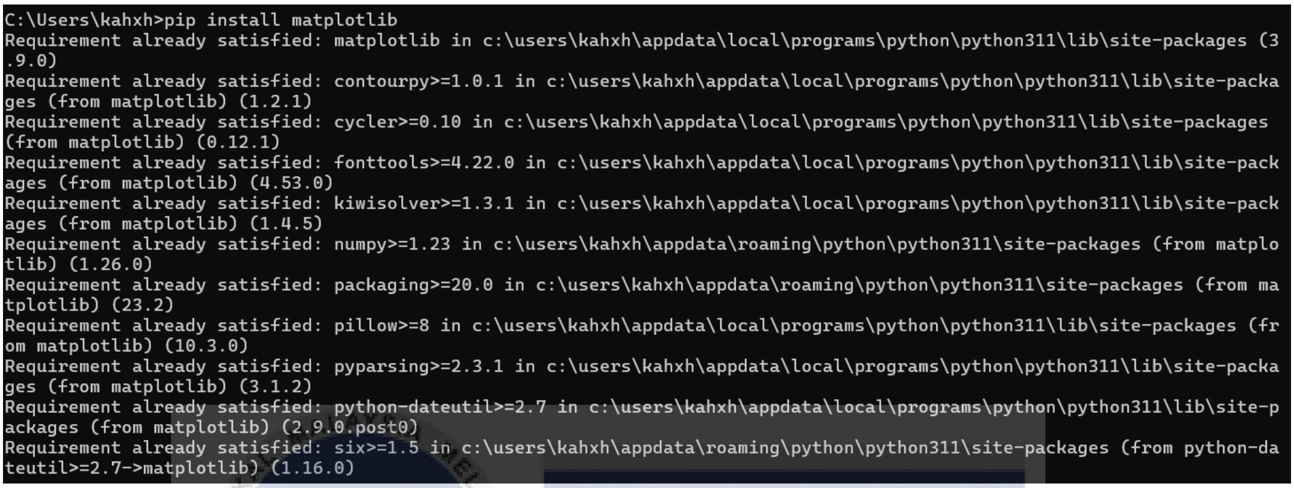


Figure 3-5 : Installation of DMAS library using CMD

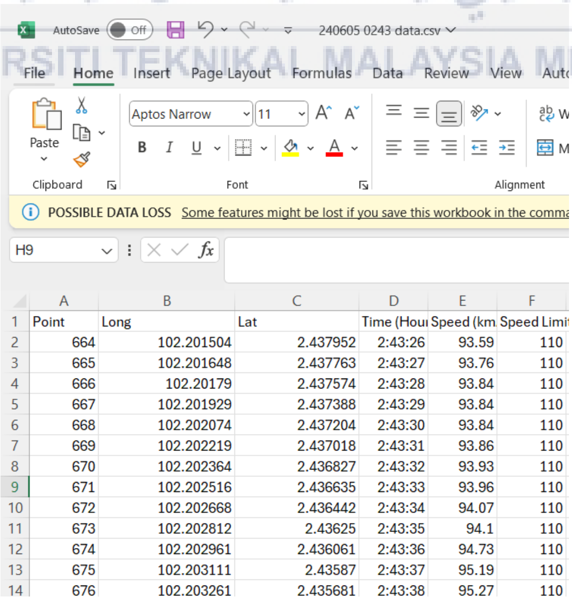
The full coding with the fucntion for DMAS application are provided at Appendix B.

Overview of DMAS application

The design of the Driver Monitoring and Alerting System (DMAS) application integrates various web technologies to provide a robust and user-friendly interface for monitoring and improving driver behavior. The application's web pages are structured using HTML, organizing essential components such as headers, buttons, links, and sections like the sidebar, map view, and results area. CSS is utilized to enhance visual appeal and usability, ensuring the application is aesthetically pleasing, responsive, and accessible across devices. Detailed styling of elements like the sidebar, buttons, maps, and graphs ensures an intuitive interface. JavaScript adds interactivity and dynamic behavior, enabling real-time updates, handling user interactions with the map, and integrating data visualization libraries. The DMAS app can be launched by clicking on the 'RUN' button in VS Code, which will display the page as shown in Figure 3-6.

Figure 3-6 : DMAS application

Users can upload a CSV file saved on the SD card in the DMAS, which automatically logs vehicle parameters during the trip. The CSV file is displayed as shown in Figure 3-7.

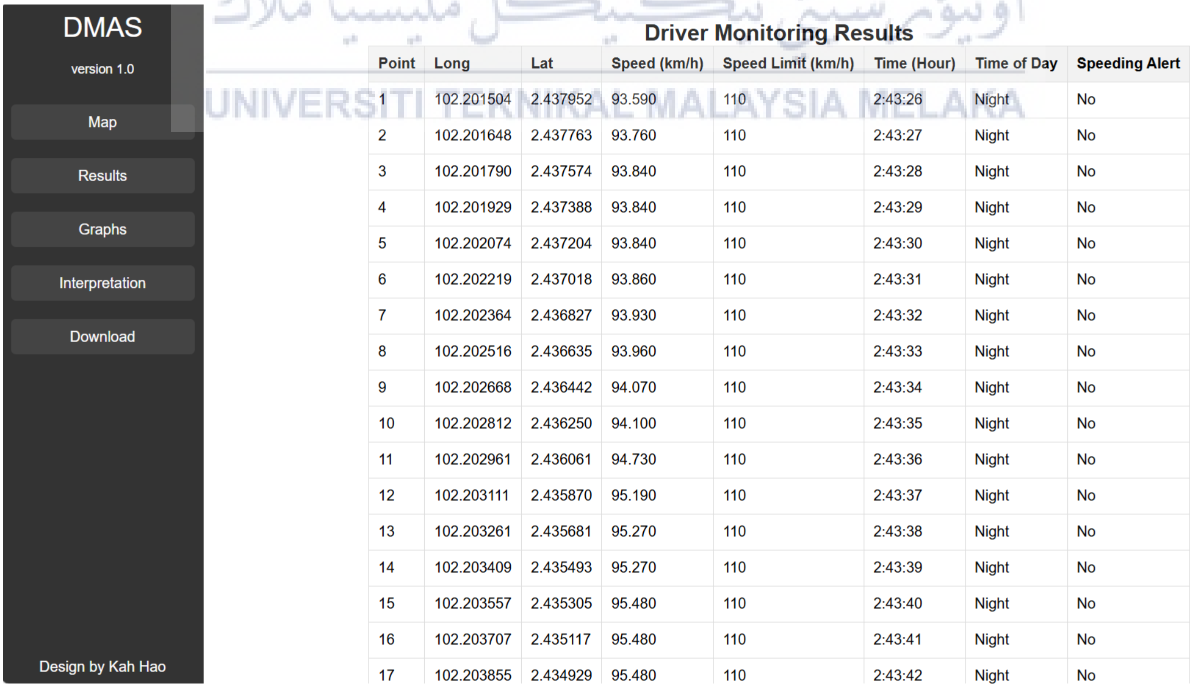


Point	Long	Lat	Time (Hou	Speed (km	Speed Limi
664	102.201504	2.437952	2:43:26	93.59	110
665	102.201648	2.437763	2:43:27	93.76	110
666	102.20179	2.437574	2:43:28	93.84	110
667	102.201929	2.437388	2:43:29	93.84	110
668	102.202074	2.437204	2:43:30	93.84	110
669	102.202219	2.437018	2:43:31	93.86	110
670	102.202364	2.436827	2:43:32	93.93	110
671	102.202516	2.436635	2:43:33	93.96	110
672	102.202668	2.436442	2:43:34	94.07	110
673	102.202812	2.43625	2:43:35	94.1	110
674	102.202961	2.436061	2:43:36	94.73	110
675	102.203111	2.43587	2:43:37	95.19	110
676	102.203261	2.435681	2:43:38	95.27	110

Figure 3-7 : Logging CSV file

Key features of the DMAS app include a map view displaying routes with markers for speed violations (Figure 3-8), a results section summarizing collected data in table format (Figure 3-9), and speeding violation graphs generated by Plotly to visualize driving trends (Figure 3-10). The interpretation feature employs fuzzy logic to analyze driver behavior, providing insights and scores to help improve driving habits (Figure 3-11). Additionally, the application allows users to download collected data in HTML format for further analysis.

Figure 3-8 : Route Map History in DMAS app



Point	Long	Lat	Speed (km/h)	Speed Limit (km/h)	Time (Hour)	Time of Day	Speeding Alert
1	102.201504	2.437952	93.590	110	2:43:26	Night	No
2	102.201648	2.437763	93.760	110	2:43:27	Night	No
3	102.201790	2.437574	93.840	110	2:43:28	Night	No
4	102.201929	2.437388	93.840	110	2:43:29	Night	No
5	102.202074	2.437204	93.840	110	2:43:30	Night	No
6	102.202219	2.437018	93.860	110	2:43:31	Night	No
7	102.202364	2.436827	93.930	110	2:43:32	Night	No
8	102.202516	2.436635	93.960	110	2:43:33	Night	No
9	102.202668	2.436442	94.070	110	2:43:34	Night	No
10	102.202812	2.436250	94.100	110	2:43:35	Night	No
11	102.202961	2.436061	94.730	110	2:43:36	Night	No
12	102.203111	2.435870	95.190	110	2:43:37	Night	No
13	102.203261	2.435681	95.270	110	2:43:38	Night	No
14	102.203409	2.435493	95.270	110	2:43:39	Night	No
15	102.203557	2.435305	95.480	110	2:43:40	Night	No
16	102.203707	2.435117	95.480	110	2:43:41	Night	No
17	102.203855	2.434929	95.480	110	2:43:42	Night	No

Figure 3-9 : Table data in DMAS app

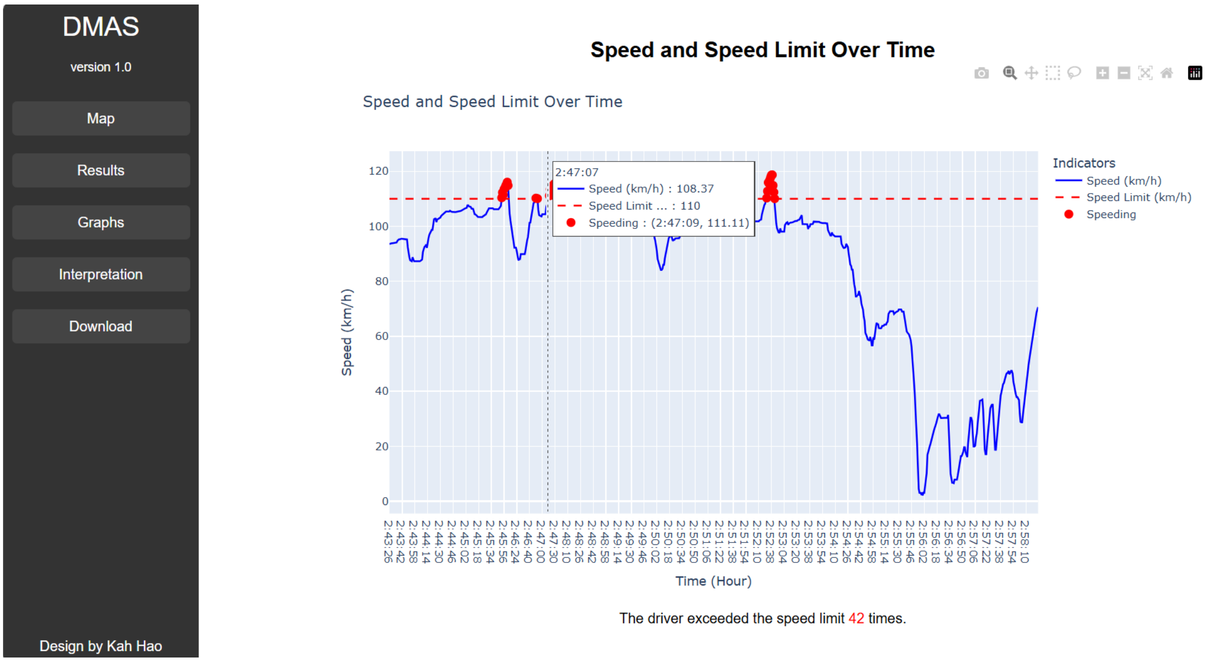


Figure 3-10 : Speeding Violation graph in DMAS app

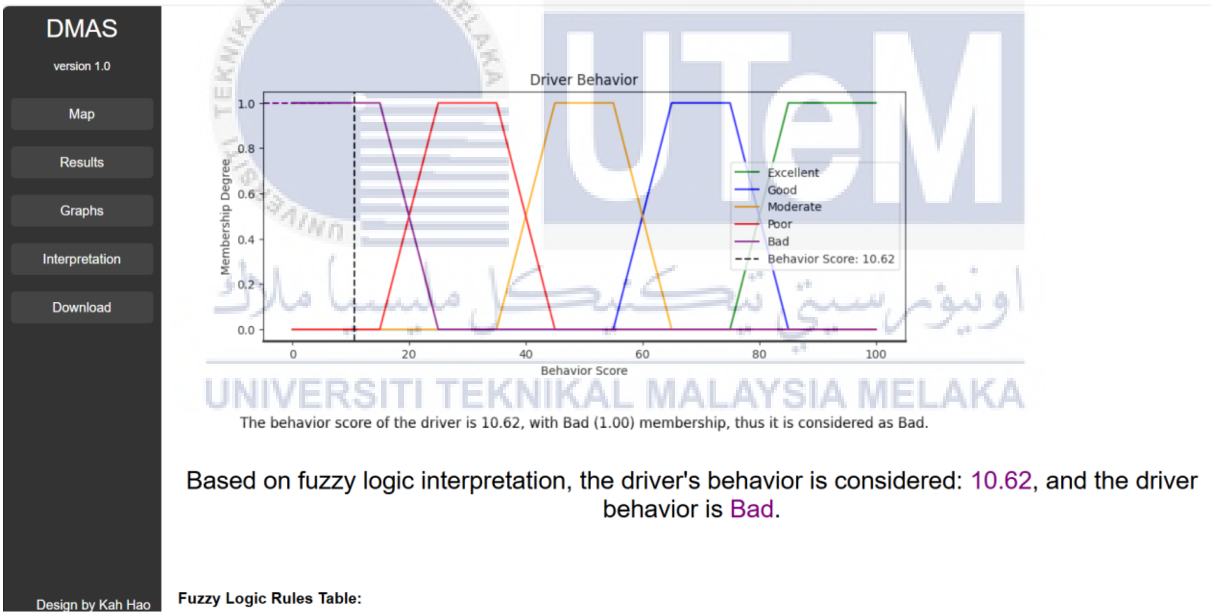


Figure 3-11 : Interpretation driver behavior using Fuzzy Logic in DMAS app

In conlusion, the combination of HTML, CSS, JavaScript, and Python libraries for backend processing and visualisation allows the development of a complex and user-centered DMAS application through a complete methodology. The integration of these technologies provides an effective and user-friendly interface that efficiently tracks and enhances driver behaviour, resulting in a highly functioning and easily accessible DMAS solution.

3.3 System Overview

The Driver Monitoring and Alerting System (DMAS) was designed with a flowchart to provide an overview of the system's testing and operation, as shown in Figure 3-12. Initially, the Sony Spresense requires the installation of drivers and the uploading of code to function as the controller for the DMAS. The Sony Spresense also needs a power supply to power the microcontroller and establish a connection with the Global Navigation Satellite System (GNSS) Upon powering on, the system automatically initializes the Sony Spresense, GNSS, and SD card to ensure proper operation, and it starts showing status information on the TFT display. The SD card is preloaded with latitude, longitude, and speed limit data. Initially, the TFT LCD display shows the number of satellites until the GNSS establishes a connection with the Sony Spresense. Once the satellite connection is established, the latitude and longitude are printed and compared with the values stored on the SD card to determine the speed limit of the road. If the current coordinates match those on the SD card, the system prints the new speed limit on the TFT LCD display. If there is no match, it prints the default or last recorded speed limit. The speed limit and actual speed of the vehicle are displayed on the TFT screen every second. When the system detects that the vehicle exceeds the speed limit, it prints "Overspeed" on the TFT LCD display and sounds a buzzer to alert the driver. The system records parameters such as latitude, longitude, speed, speed limit, and time to the Sony Spresense SD card. These parameters are logged every second, in sync with the values displayed on the TFT LCD screen. Logging stops and the data file is closed when the Sony Spresense is powered off. The recorded data can be transferred to the DMAS application from the SD card to view historical route maps, tabulated results, speeding graphs, and driver behavior scores using fuzzy logic..

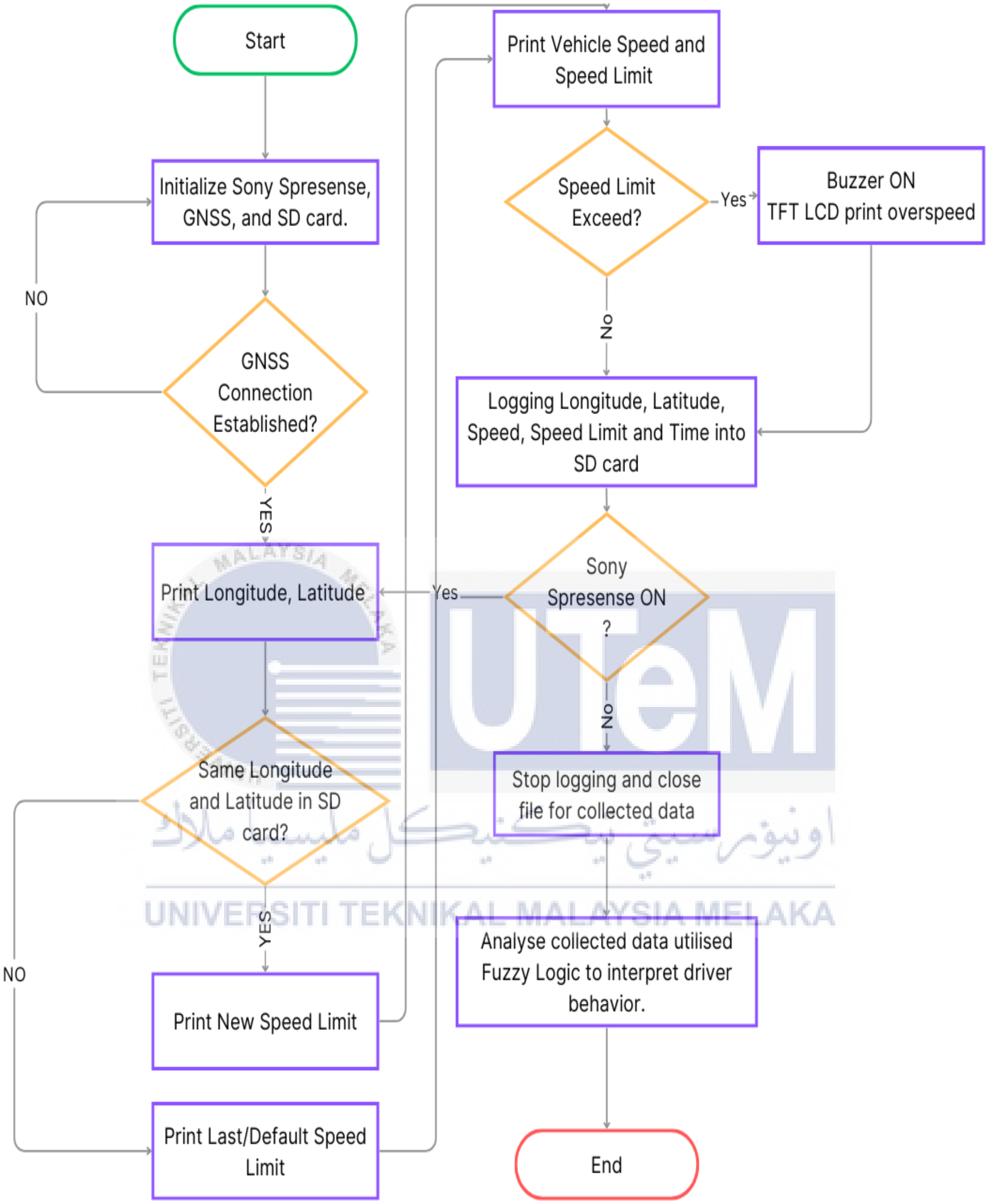


Figure 3-12 : System Flowchart

3.4 System Block Diagram

The system block diagram, shown in Figure 3-13, illustrates three major units: the input unit, control processing unit, and output unit. Since the Sony Spresense has a built-in Global Navigation Satellite System (GNSS) antenna, the input unit only involves positioning data, specifically latitude and longitude from satellites such as GPS (United States), GLONASS (Russian Federation), Beidou (China), and Galileo (European Union). The GPS coordinates are transmitted from satellites and compared with the latitude and longitude stored on the SD card, and this process is handled by the control processing unit, which is the Sony Spresense microcontroller. The SD card returns the speed limit value for the corresponding coordinates. In the output unit, vehicle parameters are recorded on the SD card and later uploaded to the DMAS application to visualize and analyze driver behavior. The vehicle's speed and the speed limit are displayed on the TFT LCD every second. If the vehicle exceeds the speed limit, the system alerts the driver using the TFT LCD display and a buzzer to ensure the driver notices their over speeding behavior.

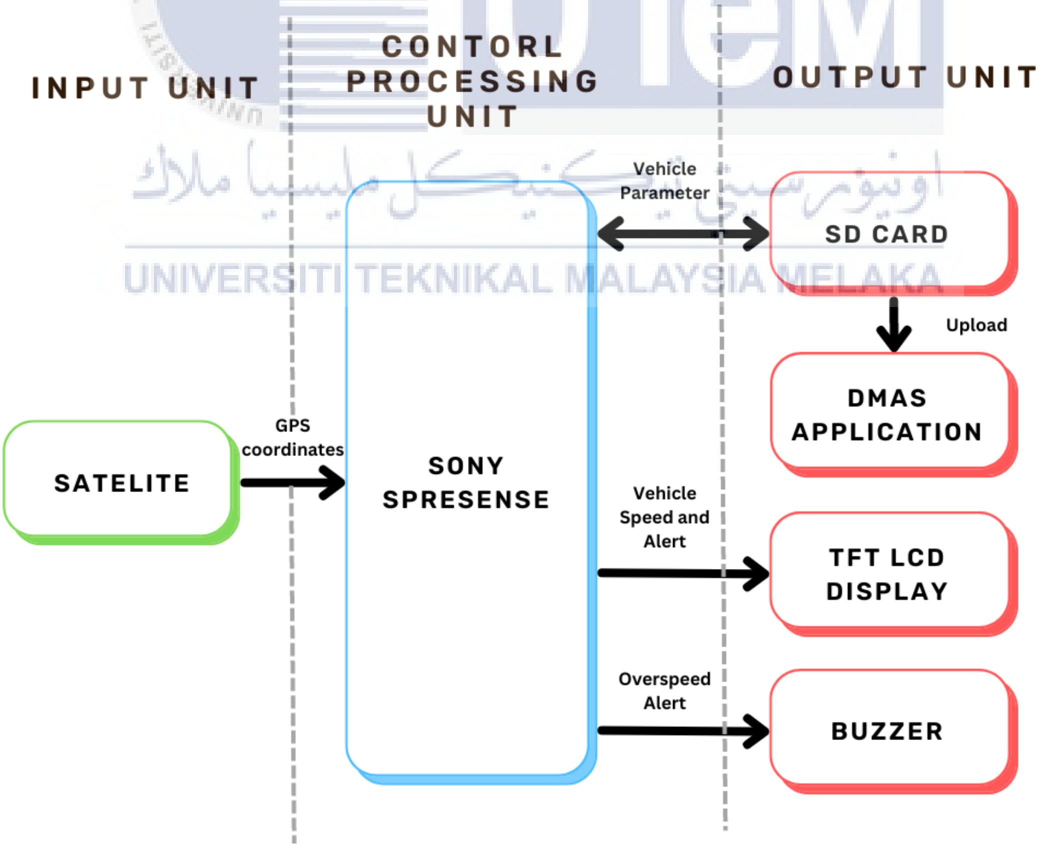


Figure 3-13 : Block Diagram

3.5 Experiment

Table 3-6 shows the experiments that will be conducted and their mapping to each of the objectives stated earlier. The experiments are designed to test the accuracy of GNSS coordinates, speed monitoring, alert mechanisms for fulfilled objective 1, and the overall reliability and functionality of the DMAS application, including the fuzzy logic behavior scoring system for fulfilled objective 2.

Table 3-6 : Experiment map with objective

Experiment	Objective 1	Objective 2
Testing the accuracy of GNSS coordinates received by Sony Spresense and mapping the driver's route in.	✓	
Testing the accuracy of speed monitoring by comparing the speedometer readings with the actual speed limits in the CSV file.	✓	
Testing the accuracy of alerting mechanisms when the vehicle exceeds the speed limit.	✓	
Testing the overall reliability and performance of the DMAS app.		✓
Testing the fuzzy logic behavior scoring system in the DMAS app.		✓

3.5.1 Experimental setup

The setup of the Sony Spresense is required before the experiment is able to start and the steps are as follows:

1. The Arduino IDE was download and install on the operating system.
2. The USB-to-serial drivers was download and install for Window 11.
3. The 64-bit Linux Ubuntu was download and install to enroll it in the dialout group.
4. Microsoft PowerShell was launch and add the user to the dialout group, as illustrated in Figure 3-14.

```
$ sudo usermod -a -G dialout $USER
```

Figure 3-14 : Coding Microsoft Powershell

5. The Sony Spresense board was connect to the PC using a USB type-B cable as figure 3-15.

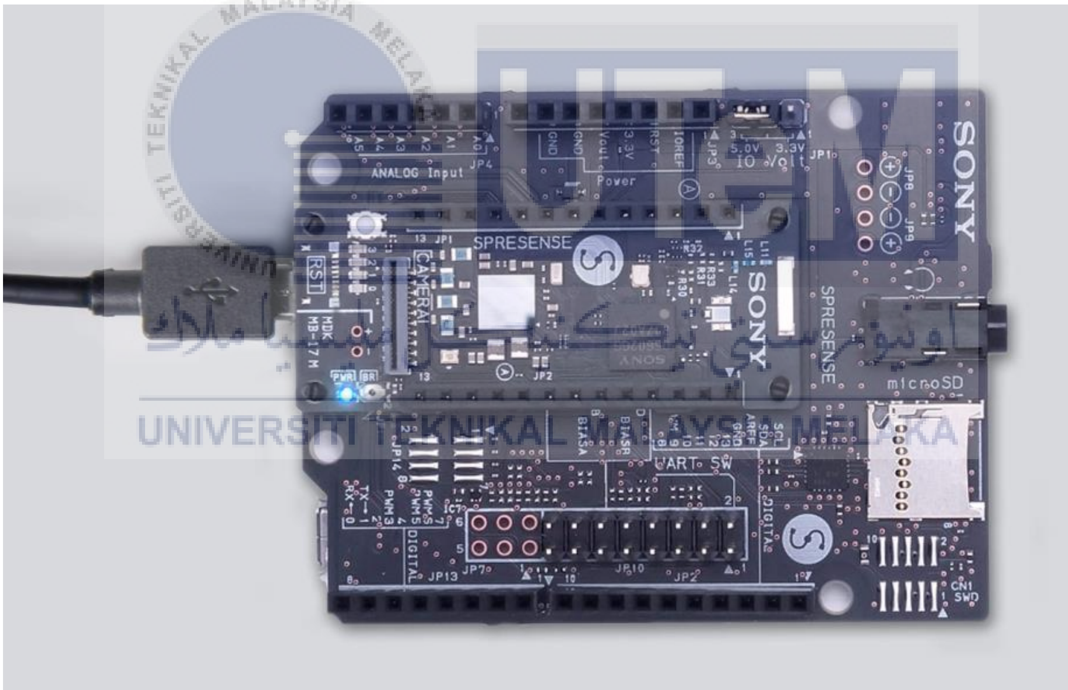


Figure 3-15 : Sony Spresense Board

6. The connection of sony spresense was confirm by checking in device manager as figure 3-16.

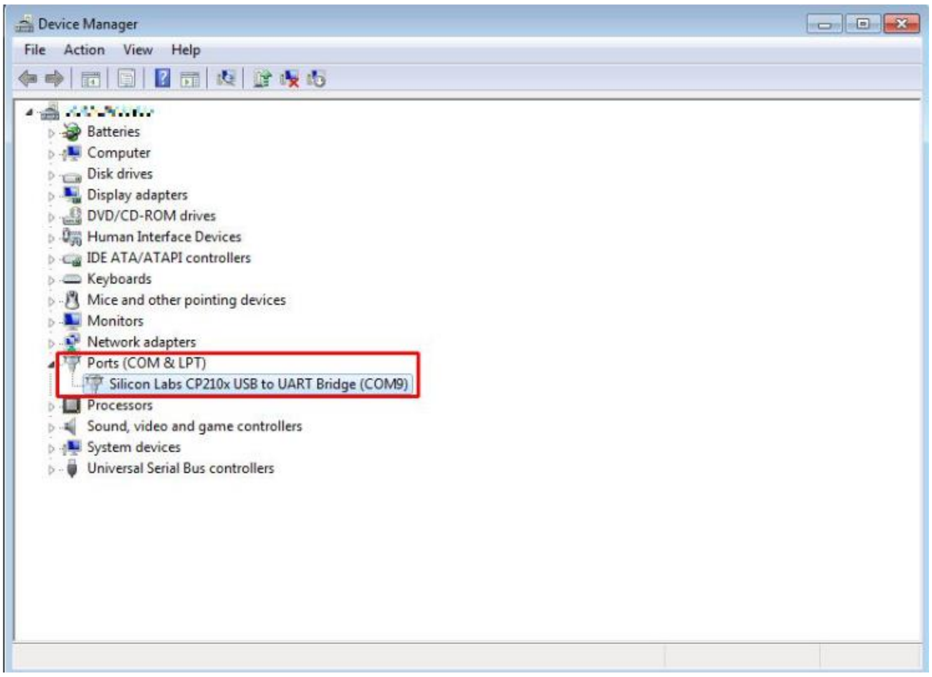


Figure 3-16 : Device Manager

7. The Additional Board Manager URL was copy and paste into the Arduino IDE, as show in Figure 3-17.

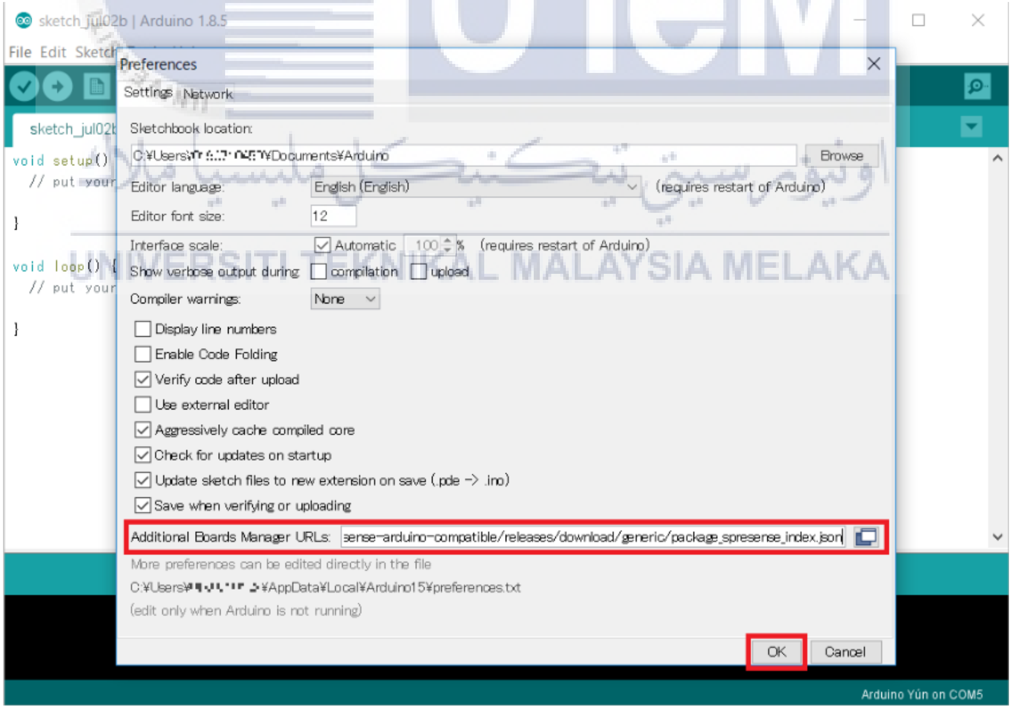


Figure 3-17 : Additional Board Manager URL

8. The "Spresense Reference Board by Spresense Community" was download and install in the Arduino Spresense library, following the steps in Figure 3-18.

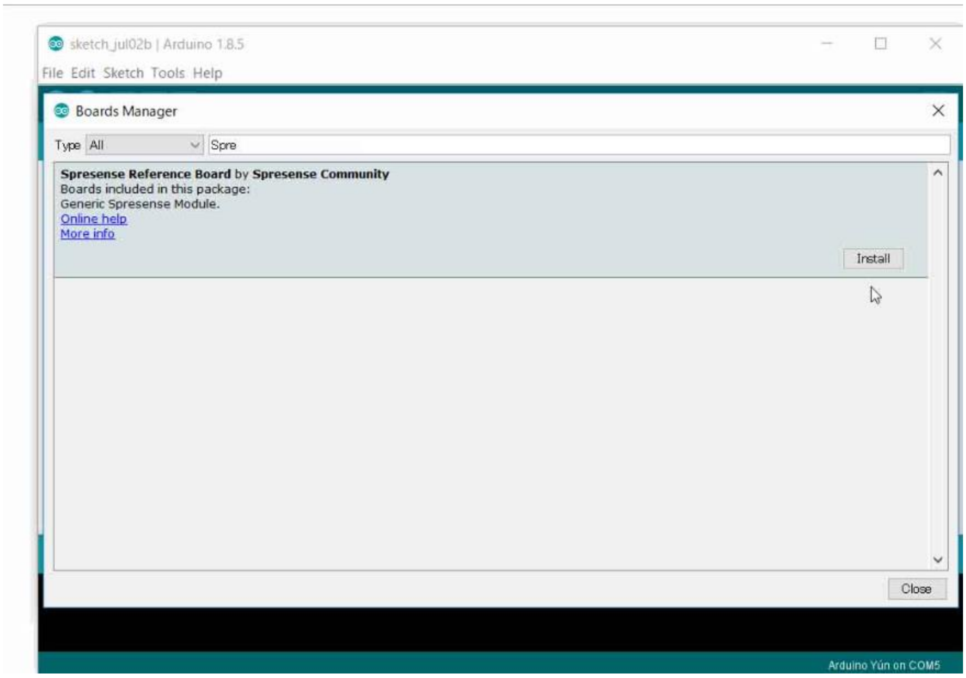


Figure 3-18 : Board Manager

9. The USB connector was connect of the main Sony Spresense board to the PC for driver installation.
10. The Spresense board was select as the functioning board in the Arduino IDE, as shown in Figure 3-19.

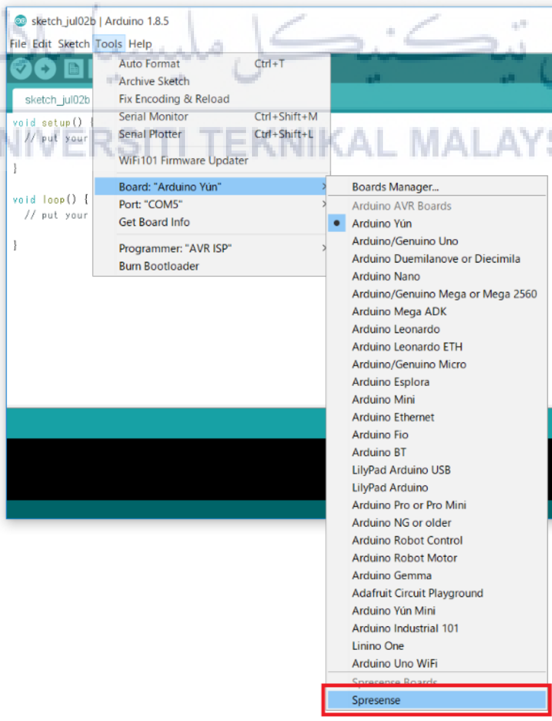


Figure 3-19 : Spresense Function Board

11. The serial port of sony Spresense was choose from the menu of Arduino IDE as Figure 3-20.

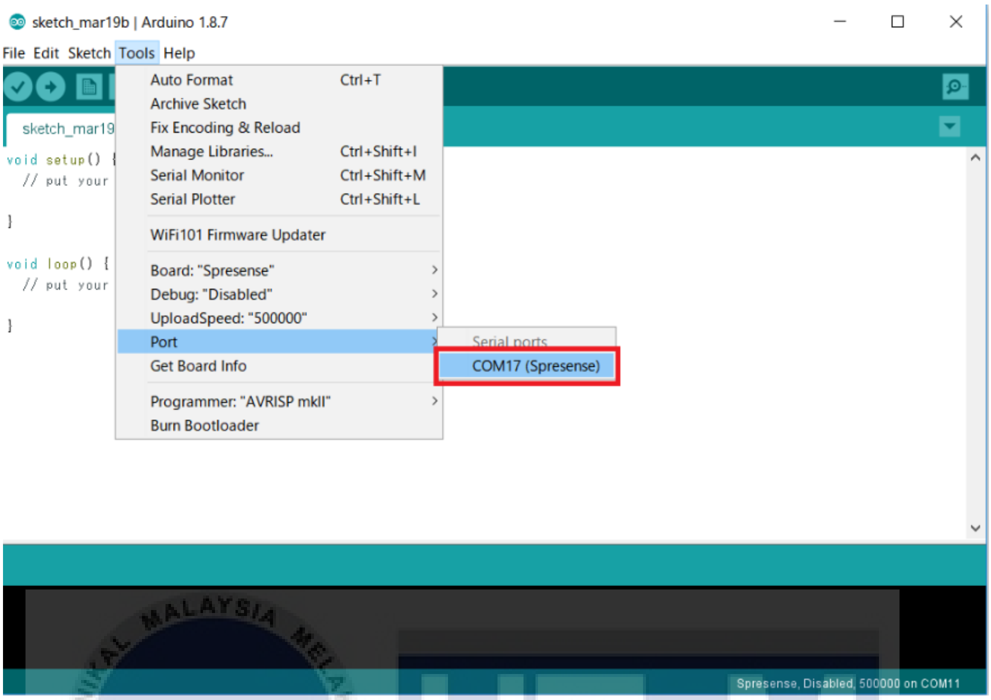


Figure 3-20 : Seleted Port

12. The programmer "Spresense firmware updater" was choose to be able to flash the boot loader or the firmware as figure 3-21.

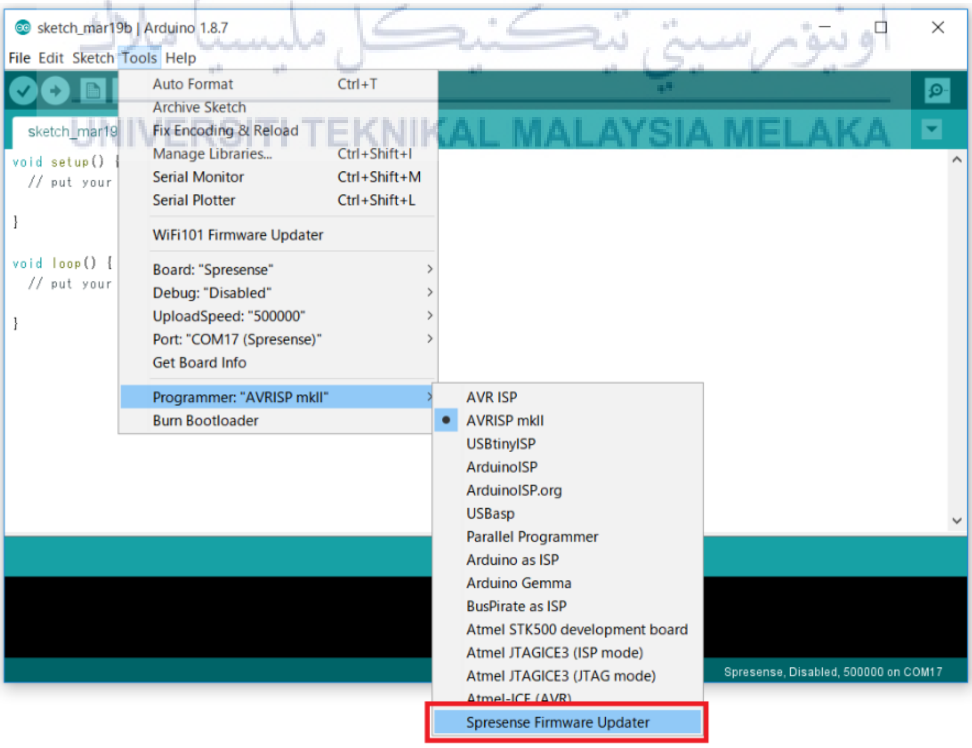


Figure 3-21 : Spresense firmware updater

13. The boot loader is burn into the Sony Spresense as figure 3-22.

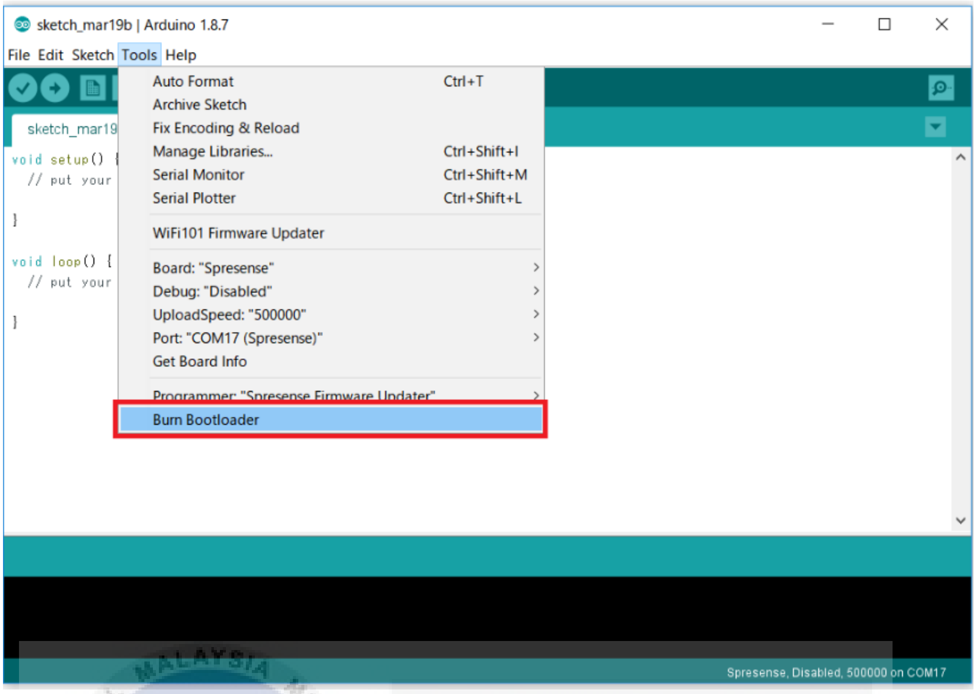


Figure 3-22 : Burn bootloader

14. The boot loader was sucesfullly burn into the Sony Spresense as figure 3-23 and ready for coding.

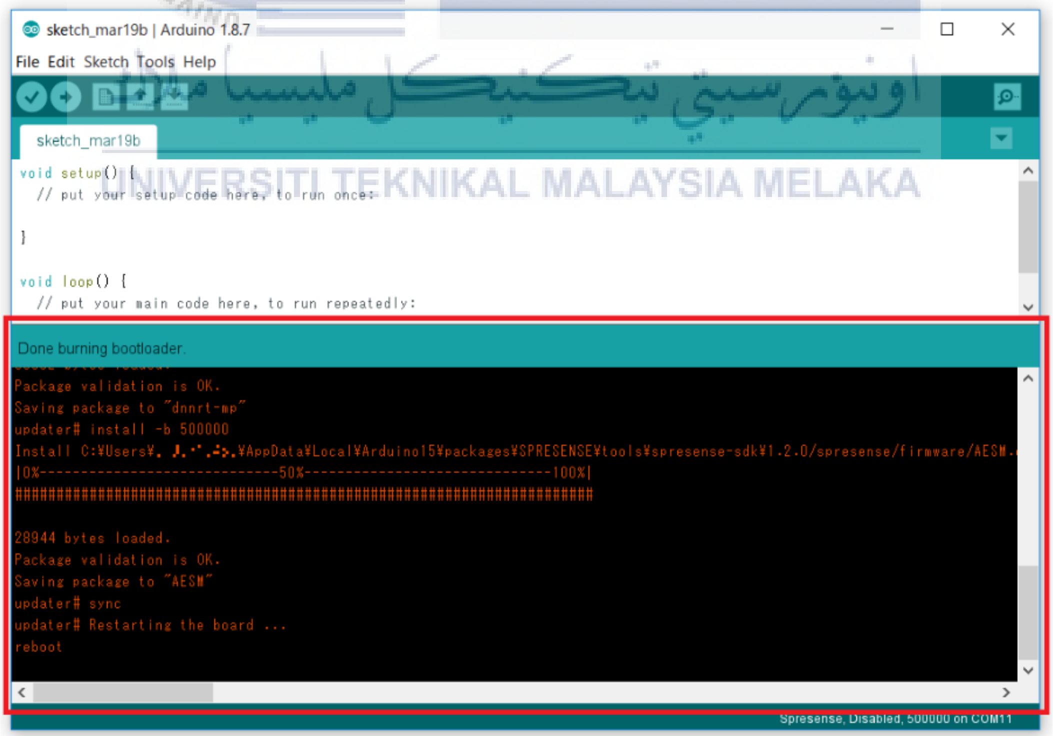


Figure 3-23 : Sucessfully burn bootloader

3.5.2 Experiment 1 : Testing the accuracy of GNSS coordinates received by Sony Spresense and mapping the driver's route.

Objective

1. To develop a Driver Monitoring and Alerting System (DMAS) that continuously monitors vehicle speed and provides real-time alerts to drivers when the speed limit is exceeded.

Components

1. USB type-B cable
2. Sony Spresense Main Board
3. Sony Spresense Extension Board
4. SD card
5. Jumper Wire

Procedure

1. The USB connector was connected of the main Sony Spresense board to the computer using USB type-B cable.
2. The connection of the sony spresense and computer was verified as figure 3-24.

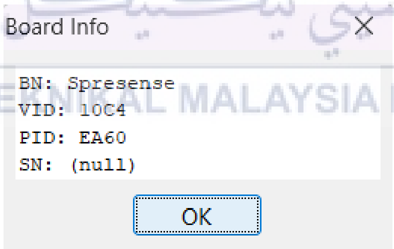


Figure 3-24 : Verified connection

3. The coding was written to connect the Sony Spresense to Global Navigation Satellite System (GNSS) in Arduino IDE as figure 3-25.

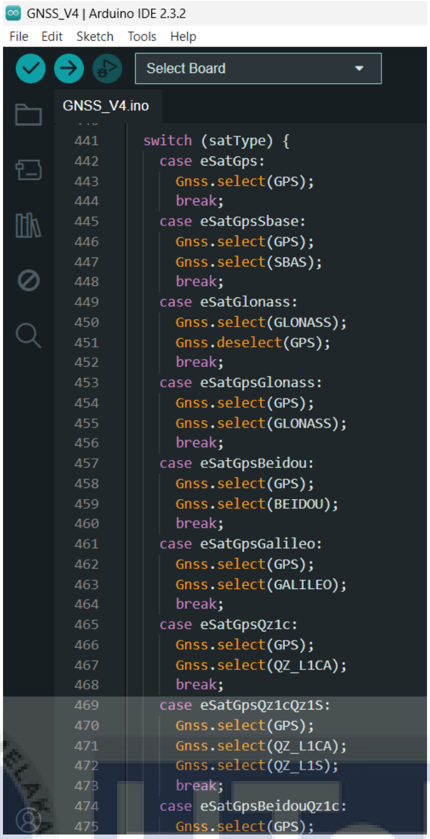


Figure 3-25 : Coding GNSS

4. The coding was then upload to the Sony Spresense board.
5. The GNSS connection was test and the result was print in the serial monitor as figure 3 -26.

```
16:06:30.420 -> SpGnss : stop in
16:06:30.495 -> SpGnss : stop out
16:06:30.495 -> SpGnss : end in
16:06:30.495 -> SpGnss : end out
16:06:30.495 -> Gnss stop OK.
16:06:30.495 -> SpGnss : begin in
16:06:31.390 -> SpGnss : begin out
16:06:31.390 -> SpGnss : start in
16:06:31.390 ->   mode = HOT_START
16:06:31.794 -> SpGnss : start out
16:06:31.794 -> Gnss restart OK.
16:06:32.530 -> 2023/12/23 08:06:33.107628, numSat: 8, Fix, Lat=2.698768, Lon=101.955523
16:06:33.413 -> 2023/12/23 08:06:34.000690, numSat: 8, Fix, Lat=2.698768, Lon=101.955523
16:06:34.433 -> 2023/12/23 08:06:35.000678, numSat: 8, Fix, Lat=2.698781, Lon=101.955904
16:06:35.412 -> 2023/12/23 08:06:36.000698, numSat: 8, Fix, Lat=2.698697, Lon=101.955822
16:06:36.418 -> 2023/12/23 08:06:37.000688, numSat: 8, Fix, Lat=2.698700, Lon=101.955772
16:06:37.413 -> 2023/12/23 08:06:38.000708, numSat: 8, Fix, Lat=2.698720, Lon=101.955721
```

Figure 3-26 : Result GNSS

6. The latitude and longitude were paste in the Google Maps to check whether the GNSS giving a precise location.

7. To ensure that the coordinates provided sufficient accuracy, the experiment was repeated in several different locations and analysis using Equation 1 and Equation 2.
8. The coding was then added function of CSV and upload to the Sony Spresense board as figure 3- 27.

GNSS_V4.ino

```
if (Gnss.waitUpdate(-1)) {
  SpNavData NavData;
  Gnss.getNavData(&NavData);

  if (getNavData(&NavData)) {
    pNavData = &NavData;

    // Log data to CSV if SD card is available
    if (sdCardOK) {
      if (switchOn) {
        if (fileClosed) {
          newFile(pNavData);
          fileClosed = false;
        }

        if (!fileClosed) {
          writeCSVData(&NavData);
        }

        // Close file after 15 minutes (900000 ms)
        if (millis() - startTime >= 900000) {
          mygnssFile.close();
          fileClosed = true;
          delay(60000); // Delay to give a subsequent opportunity to record
          switchOn = true;
          startTime = millis(); // Reset the timer for the next recording
        }
      }
    }
```

Figure 3-27 : Coding logging CSV file

9. The experiment continues with testing and compare the coordinates recorded in CSV file in DMAS.
10. To ensure the recorded coordinates data during the driver's journey provide sufficient accuracy, the experiment was repeated in same road for several times and analysis using Equation 1 and Equation 2.
11. The accuracy of the coordinates was recorded and tabulated using Microsoft Excel.

$$Percentage\ of\ error\ \% \ = \left|\frac{Yn - Xn}{Yn}\right| \times 100$$

Equation 1 : Equation of Percentage of Error

$$Accuracy\ (\%) = 100\% - Percentage\ of\ error\%$$

Equation 2 : Equation of Accuracy

3.5.3 Experiment 2 : Testing the accuracy of speed monitoring by comparing the speedometer readings with the actual speed limits in the CSV file.

Objective

1. To develop a Driver Monitoring and Alerting System (DMAS) that continuously monitors vehicle speed and provides real-time alerts to drivers when the speed limit is exceeded.

Components

1. USB type-B cable
2. Sony Spresense Main Board
3. Sony Spresense Extension Board
4. SD card
5. Jumper Wire
6. 3D printing casing
7. 1.8-inch TFT LCD Display ST7735

Procedure

1. The speed limit of road coordinates was save in CSV file as figure 3-28 and upload to SD card in Sony Spresense.

A	B	C
101.922	2.696992	60
101.922	2.697004	60
101.9222	2.696994	60
101.9222	2.697002	60
101.9223	2.697002	60
101.9224	2.696996	60
101.9225	2.705359	60
101.9225	2.697	60
101.9225	2.696996	60
101.9226	2.705223	60
101.9226	2.70509	60
101.9227	2.697	60
101.9227	2.704959	60
101.9227	2.696997	60
101.9228	2.704827	60
101.9228	2.696998	60
101.9228	2.704698	60
101.9228	2.696997	60
101.9229	2.714806	80
101.9229	2.714974	80
101.9229	2.715144	80
101.9229	2.714639	80
101.9229	2.715312	80
101.9229	2.704571	80
101.9229	2.714471	80

Figure 3-28 : Speed Limit CSV file

2. The coding was modified as figure 3-29 to obtain the speed and speed limit of vehicle in the Arduino IDE and print the value of velocity at TFT LCD display.
3. The coding was then upload to the Sony Spresense board.

```
tft.drawBitmap(5, 70, speed, 50, 50, ST7735_CYAN);
tft.fillRect(60, 85, 60, 50, ST7735_BLACK);
tft.setFont();
tft.setTextSize(0);
tft.setCursor(60,80);
tft.print("Actual Speed:");
tft.setFont(&FreeSans9pt7b);
tft.setTextSize(1);
tft.setCursor(60,110);
tft.print(pNavData->velocity * KMH);
tft.setCursor(118,110);S
tft.setFont(&FreeSans9pt7b);
tft.setTextSize(0);
tft.print("km/h");
```

Figure 3-29 : Coding of veloctiy at TFT LCD display

4. The USB connector was connected of the main Sony Spresense board to the USB hub on car using USB type-B cable as figure 3-30.

Figure 3-30 : Connect to USB hub on car

5. The connection of the sony spresense and sd card is inistialised and show the welcome interface at TFT LCD display as figure 3-31.

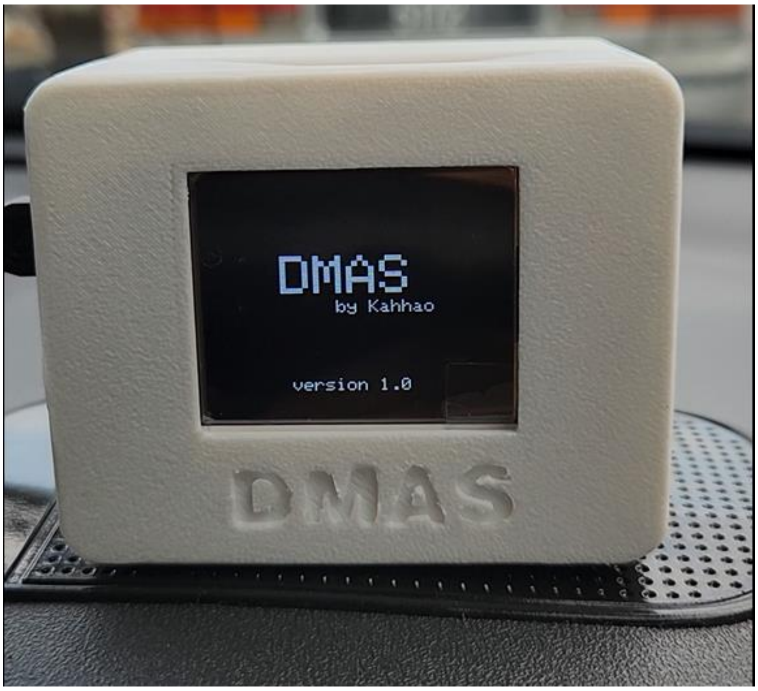


Figure 3-31 : Welcome interface of DMAS

6. The obtained velocity of vehicle and speed limit results were view the in the TFT LCD display as figure 3-32.

Figure 3-32 : TFT LCD shows speed limit and actual speed

7. The speed values were compared with the speedometer and analyse to ensure the GNSS accurately provides real-time speed information using Equation 1 and Equation 2.
8. The speed limit data also compared and analyse with the road sign to ensure the DMAS change according to the road sign speed limit.

3.5.4 Experiment 3 : Testing the accuracy of alerting mechanisms when the vehicle exceeds the speed limit.

Objective

1. To develop a Driver Monitoring and Alerting System (DMAS) that continuously monitors vehicle speed and provides real-time alerts to drivers when the speed limit is exceeded.

Components

1. USB type-B cable
2. Sony Spresense Main Board
3. Sony Spresense Extension Board
4. SD card
5. Jumper Wire
6. 3D printing casing
7. 1.8-inch TFT LCD Display ST7735
8. Buzzer

Procedure

1. The wiring of Sony Spresense was connected as figure 3-3.
2. The coding was modified to add as figure 3-33 to print “Overspeed” at TFT LCD display and sound buzzer in Arduino IDE when exceeding speed limit.

```
if (pNavData->velocity > speedlimit) {
  tft.fillScreen(ST7735_BLACK);
  tft.drawBitmap(50, 50, Overspeeds, 64, 64, ST7735_FERRARI);
  tft.setTextColor(ST7735_WHITE);
  tft.setFont(&FreeSans9pt7b);
  tft.setTextSize(1);
  tft.setCursor(30,20);
  tft.print(" Overspeed! ");
  tone(buzzerPin, 2000); // Set the frequency of the tone to 2000 Hz (adjust as needed)
  delay(1000);// Wait for 1 second
  tone(buzzerPin, 1500);
  delay(500);
  noTone(buzzerPin); // Stop generating the tone
  delay(500); // Wait for 1 second
  tft.fillScreen(ST7735_BLACK);
```

Figure 3-33 : Coding of buzzer and TFT LCD

3. The coding was then upload to the Sony Spresense board.

4. The USB connector was connected of the main Sony Spresense board to the USB hub on car using USB type-B cable as figure 3-29.
5. The connection of the sony spresense and sd card is inistialised and show the welcome interface at TFT LCD display as figure 3-30.
6. The obtained velocity of vehicle and speed limit results were view the in the TFT LCD display as figure 3-31.
7. The vehicle was speeding over speed limit and check the functionality of the buzzer and TFT LCD to alert the driver.
8. The data is recorded and analyse.

3.5.5 Experiment 4 : Testing the overall reliability and performance of the DMAS app.

Objective

1. To analyze the collected data to identify driver behavior patterns utilizing Fuzzy Logic techniques.

Components

1. SD card
2. DMAS application

Procedure

1. The SD card was removed from sony spresense and upload the collected data to the computer using SD card reader as figure 3-34.

Figure 3-34 : SD card Reader

2. The CSV file with collected data as figure 3-7 was uploaded to the DMAS application.
3. The result history route of DMAS app was compared with the result of history route of Mygeodata Cloud.
4. The result table of DMAS app was compared with the uploaded CSV file to ensure the result was correct.
5. The speeding graph of DMAS app was compared with the graph generate by Microsoft Excel.
6. The step 3,4 and 5 was repeated for three times to ensure the data analyse using DMAS applications was reliable.

3.5.6 Experiment 5 : Testing the fuzzy logic behavior scoring system in the DMAS app.

Objective

1. To analyze the collected data to identify driver behavior patterns utilizing Fuzzy Logic techniques.

Components

1. SD card
2. DMAS application

Procedure

1. The SD card was removed from sony spresense and upload the collected data to the computer using SD card reader as figure 3-34.
2. The driver behavior utilise fuzzy logic was first fuzzification using Equation 3.

$$\mu(x) = \begin{cases} \text{degree of membership} & \text{if } x \text{ belongs to the fuzzy set} \\ 0 & \text{otherwise} \end{cases}$$

Equation 3 : Fuzzification

3. The rule evaluation with IF-Then rules was done and generate a fuzzy logic rules table as figure 3-

Time of Day	Average Speed	Times		
		Low	Moderate	High
Morning	Slow	Excellent	Good	Poor
	Moderate	Good	Moderate	Bad
	Fast	Moderate	Poor	Bad
Afternoon	Slow	Excellent	Good	Poor
	Moderate	Good	Moderate	Bad
	Fast	Moderate	Poor	Bad
Night	Slow	Excellent	Good	Poor
	Moderate	Good	Poor	Bad
	Fast	Moderate	Poor	Bad

Figure 3-35 : Fuzzy Table Rule

4. The aggregation of rule output was done.
5. The defuzzification was done using centroid method as Equation 4 to calculate the driver behavior score and category utilise the rule.

$$y_{\text{centroid}} = \frac{\int x \cdot \mu_{\text{aggregated}}(x)\,dx}{\int \mu_{\text{aggregated}}(x)\,dx}$$

Equation 4 : Defuzification Centroid Method

6. The DMAS application was updated by adding the coding of fuzzy logic for centroid as figure.
7. The CSV file with collected data was uploaded to the DMAS application.
8. The calculation driver behavior score and the DMAS application was compared and analyse the accuracy using Equation 1 and Equation 2.

3.6 Ethics and Safety Method

In this project, a Driver Monitoring and Alerting System (DMAS) was developed and tested using the Sony Spresense board. Various experiments were conducted to ensure the accuracy and reliability of the system. To ensure the safety and ethics of the methods used, specific risk assessments were carried out, particularly focusing on the electronic board's Electrostatic Discharge (ESD) protection using a 3D casing. The hazards and risk assessments are summarized below.

Table 3-7: Hazard and Risk Assessment

Hazards/Risk Assessment	**Explanation**
ESD Protection	Ensuring proper ESD protection is crucial for the safe operation of the Sony Spresense board. Using a 3D printed casing helps to shield the electronic components from static electricity, which can cause damage or malfunctions. The casing must be made from materials that do not generate static charge and should be properly grounded.
Model Validation	The validation of the DMAS system involved conducting accuracy tests in experiments to ensure correct setup and data reliability. This included verifying the Sony Spresense's connection, GNSS accuracy, and speed monitoring. Any errors in setup could lead to inaccurate data collection or system failure.

Hazards/Risk Assessment	Explanation
Numerical Stability	The system is sensitive to numerical instability, which can lead to incomplete or erroneous results. To mitigate this, appropriate data processing methods and error-checking mechanisms were implemented and tested to ensure stability in the data analysis phases.
External Interfaces	Variables and settings in the interfacing software, such as Arduino IDE or data logging tools, may interfere with the DMAS operation, causing errors. Ensuring proper configuration and clearing unnecessary variables or settings before running the experiments is essential to prevent such issues.

It is important to note that the specific hazards and risk assessment considerations can vary depending on the nature of the system being simulated and the context in which Sony Spresense is used. Therefore, it is advisable to perform a thorough analysis of the simulation system and consult relevant safety standards or guidelines to ensure a comprehensive hazard and risk assessment.

By adhering to these precautions and performing detailed risk assessments, the project ensures the ethical and safe use of technology in developing the DMAS. This approach helps in maintaining the integrity and reliability of the system while protecting the electronic components from potential hazards.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter will discuss the analysis results and the discussion of the experiments conducted in line with the objectives of the project. The analysis results were conducted for Experiments 1, 2, and 3, which are part of Objective 1, while the analysis results for Objective 2 were conducted in Experiments 4 and 5. This chapter will present the five experiments designed to fulfill the objectives of the project, ensuring the functionality and reliability of the Driver Monitoring and Alerting System (DMAS).

4.2 Testing the Accuracy and Reliability of GNSS coordinates of DMAS

Experiment 1: Testing the accuracy of GNSS coordinates received by Sony Spresense and mapping the driver's route in.

The first experiment will start by testing the GNSS connectivity and accuracy by displaying the location on Google Maps. The code was uploaded to the Sony Spresense and connected to a laptop to show the output using the serial monitor. The serial monitor will display the date, time, number of satellites connected, position, latitude, and longitude. The test was conducted in two different places to check the accuracy of the coordinates provided by the Sony Spresense. The coordinates will be checked using Google Maps by typing in the latitude and longitude.

Address 1: 2, Jalan D6,75450, Ayer Keroh, Melaka (Indoor)

Address 2: 49, Jalan Seri Binjai 3, Taman Seri Binjai,70100, Seremban ,Negeri Sembilan (Outdoor)

The figure below shows the real location and measured location using Google Maps. All the latitude and longitude coordinates were obtained from the serial print of the Arduino IDE from the Sony Spresense board and compared with the addresses keyed in on Google Maps.

Address 1: 2, Jalan D6,75450, Ayer Keroh, Melaka (Indoor)

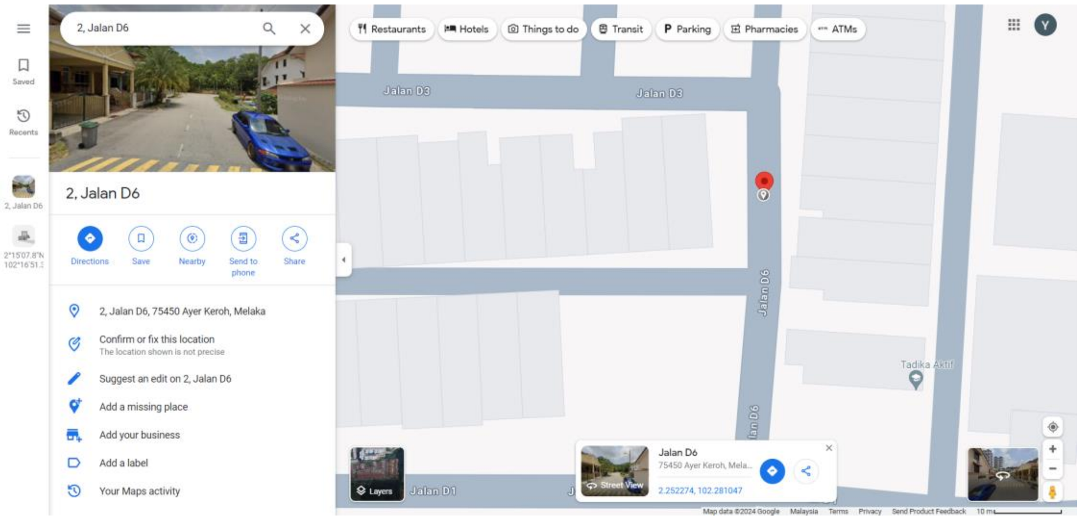


Figure 4-1 : Real location for address 1 coordinates via Google Maps

```
2024/01/17 07:20:00.000672, numSat: 8, Fix, Lat=2.252163, Lon=102.281112
2024/01/17 07:20:01.000692, numSat: 9, Fix, Lat=2.252158, Lon=102.281109
2024/01/17 07:20:02.000682, numSat: 9, Fix, Lat=2.252155, Lon=102.281108
2024/01/17 07:20:03.000671, numSat: 8, Fix, Lat=2.252153, Lon=102.281106
2024/01/17 07:20:04.000691, numSat: 8, Fix, Lat=2.252141, Lon=102.281103
2024/01/17 07:20:05.000682, numSat: 8, Fix, Lat=2.252137, Lon=102.281102
2024/01/17 07:20:06.000670, numSat: 8, Fix, Lat=2.252137, Lon=102.281103
2024/01/17 07:20:07.000690, numSat: 8, Fix, Lat=2.252136, Lon=102.281107
2024/01/17 07:20:08.000681, numSat: 8, Fix, Lat=2.252127, Lon=102.281112
2024/01/17 07:20:09.000669, numSat: 8, Fix, Lat=2.252129, Lon=102.281119
```


Figure 4-2 : Serial print result latitude and longitude obtain through Sony Spresense for address 1

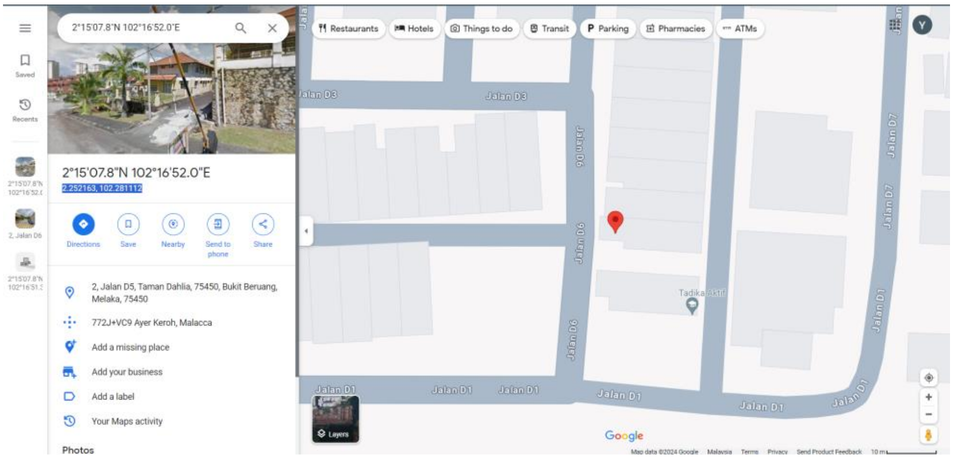


Figure 4-3 : Measured location for address 1 coordinates via Google Maps

Address 2 : 49, Jalan Seri Binjai 3, Taman Seri Binjai,70100, Seremban ,Negeri Sembilan (Outdoor)

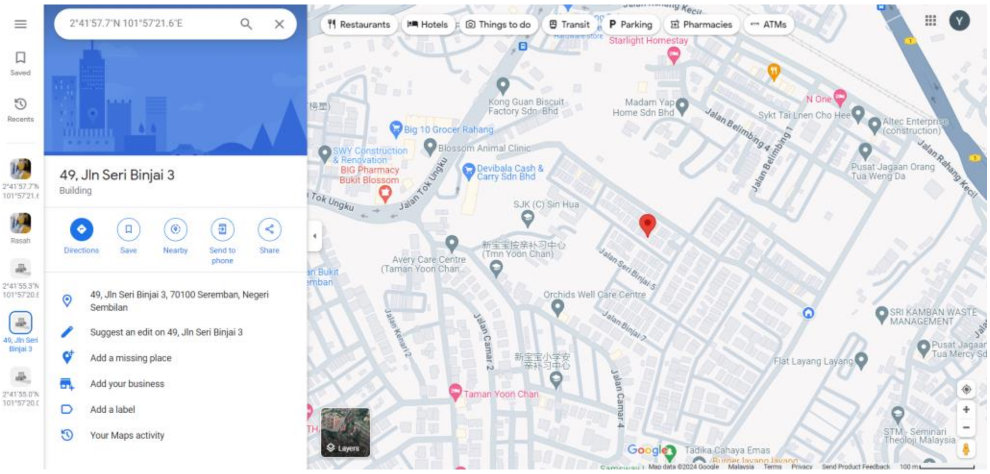


Figure 4-4 : Real location for address 2 coordinates via Google Maps

```
16:06:30.420 -> SpGnss : stop in
16:06:30.495 -> SpGnss : stop out
16:06:30.495 -> SpGnss : end in
16:06:30.495 -> SpGnss : end out
16:06:30.495 -> Gnss stop OK.
16:06:30.495 -> SpGnss : begin in
16:06:31.390 -> SpGnss : begin out
16:06:31.390 -> SpGnss : start in
16:06:31.390 ->   mode = HOT_START
16:06:31.794 -> SpGnss : start out
16:06:31.794 -> Gnss restart OK.
16:06:32.530 -> 2023/12/23 08:06:33.107628, numSat: 8, Fix, Lat=2.698768, Lon=101.955523
16:06:33.413 -> 2023/12/23 08:06:34.000690, numSat: 8, Fix, Lat=2.698768, Lon=101.955523
16:06:34.433 -> 2023/12/23 08:06:35.000678, numSat: 8, Fix, Lat=2.698781, Lon=101.955904
16:06:35.412 -> 2023/12/23 08:06:36.000698, numSat: 8, Fix, Lat=2.698697, Lon=101.955822
16:06:36.418 -> 2023/12/23 08:06:37.000688, numSat: 8, Fix, Lat=2.698700, Lon=101.955772
16:06:37.413 -> 2023/12/23 08:06:38.000708, numSat: 8, Fix, Lat=2.698720, Lon=101.955721
```

Figure 4-5 : Serial print result latitude and longitude obtain through Sony Spresense for address 2

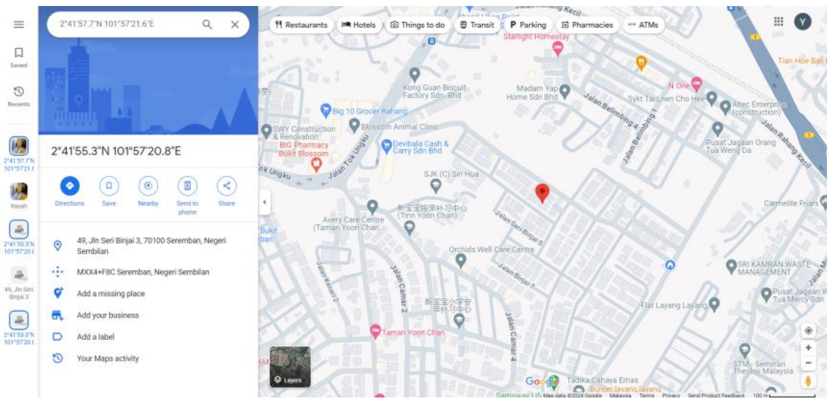


Figure 4-6 : Measured location for address 2 coordinates via Google Maps

Table 4-1 : Result of real location and measured location

Address	Real location		Measured location	
	Latitude	Longitude	Latitude	Longitude
1	2.252272	102.281048	2.252163	102.281112
2	2.698700	101.955772	2.698720	101.955720

Table 4-2 : Percentage of error and accuracy of real location and meaured location

Address	Percentage of error		Accuracy	
	Latitude (%)	Longitude (%)	Latitude (%)	Longitude (%)
1	0.01	0.007	99.99	99.993
2	0.002	0.005	99.998	99.995

Calculation Formula:

$$Percentage\ of\ error\ \%\ = \left|\frac{Yn - Xn}{Yn}\right| \times 100$$

$$Yn = Latitude\ or\ Longitude\ (Real\ Location)$$

$$Xn =: Latitude\ or\ Longitude\ (Measured\ Location)$$

$$Accuracy\ (\%) = 100\% - Percentage\ of\ error\%$$

From the Google Maps results for Address 1 and Address 2, both locations were accurately determined as shown in Figures 4-5, 4-7, 4-4, and 4-3. To ensure the reliability of the coordinates received by the Sony Spresense, calculations were performed using percentage of error and accuracy metrics. The latitude for Address 1 achieved an accuracy of 99.99%, and the longitude for Address 1 had an accuracy of 99.993%. Similarly, the latitude for Address 2 had an accuracy of 99.998%, while the longitude for Address 2 had an accuracy of 99.995%. These results demonstrate that the coordinates received by the Sony Spresense through GNSS are highly accurate and reliable.

The experiment can proceed with comparing the coordinates logged in the CSV file during the journey. The DMAS box was set up in the car to collect data on the SD card. Logging will start when the DMAS successfully connects with GNSS, and the latitude and longitude will be logged into a CSV file on the SD card. To test the accuracy of obtaining coordinates on the same road, the experiment was conducted by passing through the same road three times to log data into the CSV file. The three CSV file logs were converted to visualizations using the MYGEODATA online converter and shown in Figure 4-7. The data was then combined into one Excel file for analysis of accuracy, as shown in Figure 4-8, with the first route having coordinates Longitude 1 and Latitude 1, the second route having coordinates Longitude 2 and Latitude 2, and the third route having coordinates Longitude 3 and Latitude 3. The full data table is provided in the Appendix B.

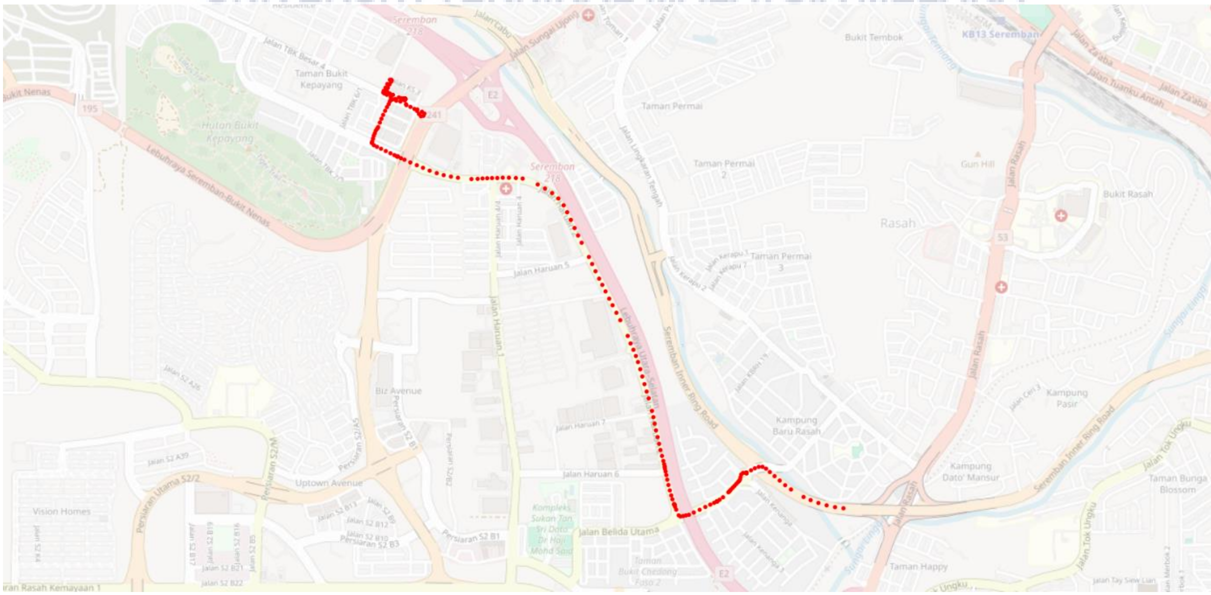


Figure 4-7 : CSV converted to map using MYGEODATA converter

Point	Long 1	Lat 1	Long 2	Lat 2	Long 3	Lat3
1	101.912856	2.7147	101.912857	2.714737	101.912856	2.714739
2	101.91285	2.7147	101.912853	2.714661	101.912856	2.714664
3	101.91286	2.7147	101.912862	2.714689	101.912864	2.714693
4	101.912874	2.7147	101.912878	2.714705	101.912882	2.714711
5	101.912881	2.7147	101.912886	2.714716	101.912891	2.714723
6	101.912882	2.7147	101.91289	2.714718	101.912898	2.714726
7	101.912879	2.7147	101.912885	2.714715	101.912891	2.714717
8	101.912881	2.7147	101.912882	2.714715	101.912883	2.714718
9	101.912885	2.7147	101.912886	2.714723	101.912887	2.714727
10	101.912884	2.7147	101.912887	2.714729	101.91289	2.714735
11	101.912874	2.7147	101.912876	2.71474	101.912878	2.714747
12	101.91287	2.7147	101.912874	2.714749	101.912878	2.714757
13	101.912872	2.7147	101.912877	2.71474	101.912882	2.714742
14	101.912872	2.7147	101.91288	2.714741	101.912888	2.714744
15	101.912872	2.7147	101.912878	2.714742	101.912884	2.714746
16	101.912871	2.7147	101.912872	2.714744	101.912873	2.71475
17	101.91287	2.7147	101.912871	2.714743	101.912872	2.71475
18	101.912866	2.7147	101.912869	2.714742	101.912872	2.71475
19	101.912859	2.7147	101.912861	2.714735	101.912863	2.714737
20	101.912852	2.7147	101.912856	2.714735	101.91286	2.714738
21	101.912843	2.7147	101.912848	2.714735	101.912853	2.714739
22	101.912806	2.7147	101.912814	2.714738	101.912822	2.714744
23	101.912793	2.7147	101.912799	2.714731	101.912805	2.714738

Figure 4-8 : CSV file of Longitude and Latitude

The accuracy of the coordinates (Longitude and Latitude) was calculated using Equation 1 for percentage of error and Equation 2 for accuracy percentage. The data was collected and tabulated in an Excel file as shown in Figure 4-9. The graph of the accuracy of coordinates for Route 1 and Route 2 was formed using a line graph, as shown in Figure 4-10. The accuracy of coordinates for Route 2 and Route 3 was formed using a line graph, as shown in Figure 4-11. The graph of the accuracy of coordinates for Route 1 and Route 3 was formed using a line graph, as shown in Figure 4-12.

	Accuracy of Route 1 and Route 2		Accuracy of Route 2 and Route 3		Accuracy of Route 1 and Route 3	
Point	Accuracy Longitude	Accuracy Latitude	Accuracy Longitude	Accuracy Latitude	Accuracy Longitude	Accuracy Latitude
1	99.99999902	99.99992633	100	99.99985266	99.99999902	99.99992633
2	99.99999706	99.99988949	99.99999411	99.99977898	99.99999706	99.99988949
3	99.99999804	99.99985265	99.99999608	99.99970531	99.99999804	99.99985265
4	99.99999608	99.99977898	99.99999215	99.99955796	99.99999608	99.99977898
5	99.99999509	99.99974215	99.99999019	99.99948429	99.99999509	99.99974215
6	99.99999215	99.99970531	99.9999843	99.99941062	99.99999215	99.99970531
7	99.99999411	99.99992633	99.99998823	99.99985265	99.99999411	99.99992633
8	99.99999902	99.99988949	99.99999804	99.99977898	99.99999902	99.99988949
9	99.99999902	99.99985266	99.99999804	99.99970531	99.99999902	99.99985266
10	99.99999706	99.99977898	99.99999411	99.99955797	99.99999706	99.99977898
11	99.99999804	99.99974215	99.99999608	99.9994843	99.99999804	99.99974215
12	99.99999608	99.99970531	99.99999215	99.99941063	99.99999608	99.99970531

Figure 4-9 : Excel File of Accuracy of Longitude and Latitude on different routes

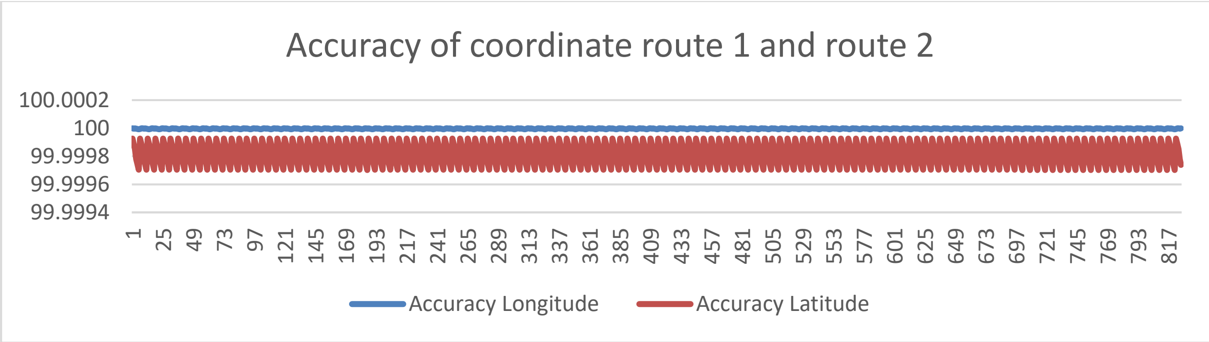


Figure 4-10 : Line graph of accuracy of coordinates for Route 1 and Route 2

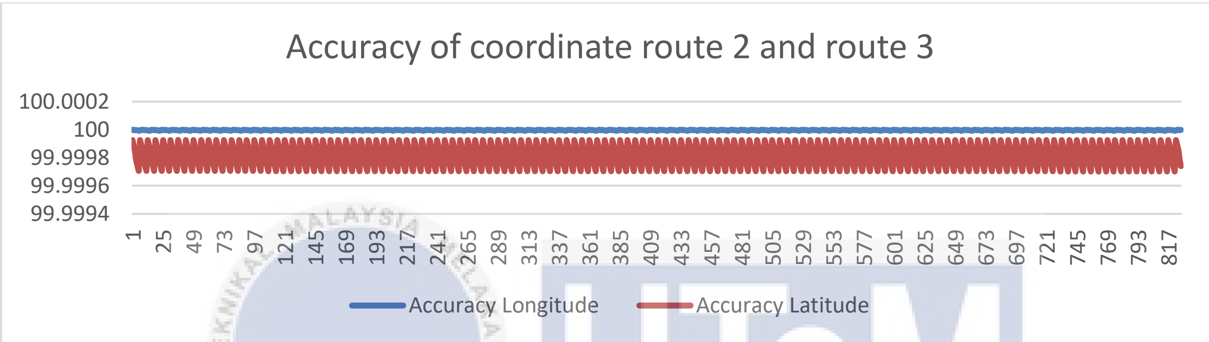


Figure 4-11 : Line graph of accuracy of coordinates for Route 2 and Route 3

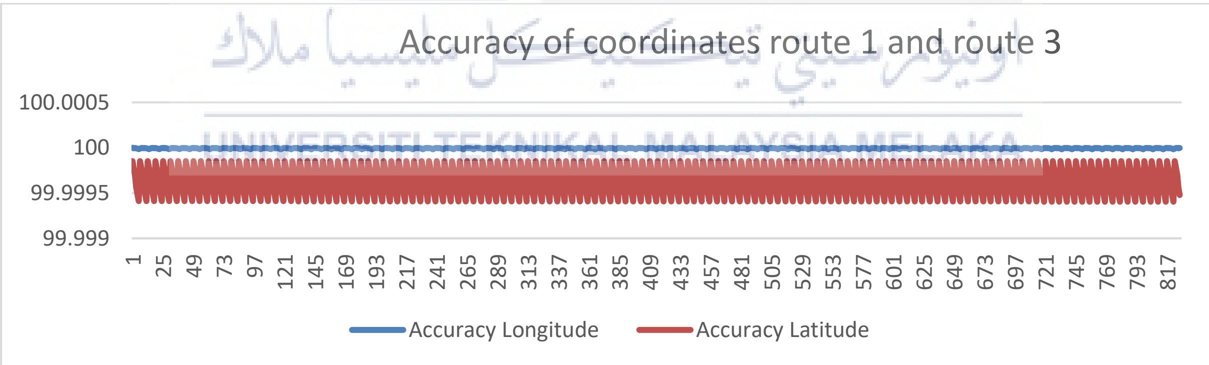


Figure 4-12 : Line graph of accuracy of coordinates for Route 1 and Route 3

The average of accuracy for coordiantes was calculate using Equation 5.

$$Average = \frac{X_n}{n}$$

Equation 5 : Average formula

Which n is number of value and Xn is dataset of value.

The average accuracy for latitude and longitude for Route 1 and Route 2 can reach 99.99999633% and 99.99981495%, respectively. The accuracy for Route 2 and Route 3 is 99.99999266% and 99.99962991%, respectively, while the accuracy for Route 1 and Route 3 is 99.99999633% and 99.99981496%. Most of the coordinates achieved an average accuracy of 99.99%, demonstrating that the recorded data coordinates are reliable and correct for data analysis. This shows that Experiment 1 successfully meets part of Objective 1.

However, the accuracy of the latitude and longitude might be affected when passing through covered areas such as tunnels, underground passages, or inside buildings. Figure 4-13 shows the visualization of the CSV file from the MYGEODATA converter. The points deviate from the exact road. The sampling method was used to check the accuracy of the disturbed points on the map and calculated using Equation 1 and Equation 2 to determine the accuracy of the actual coordinates and the measured coordinates, as shown in Table 4-3.

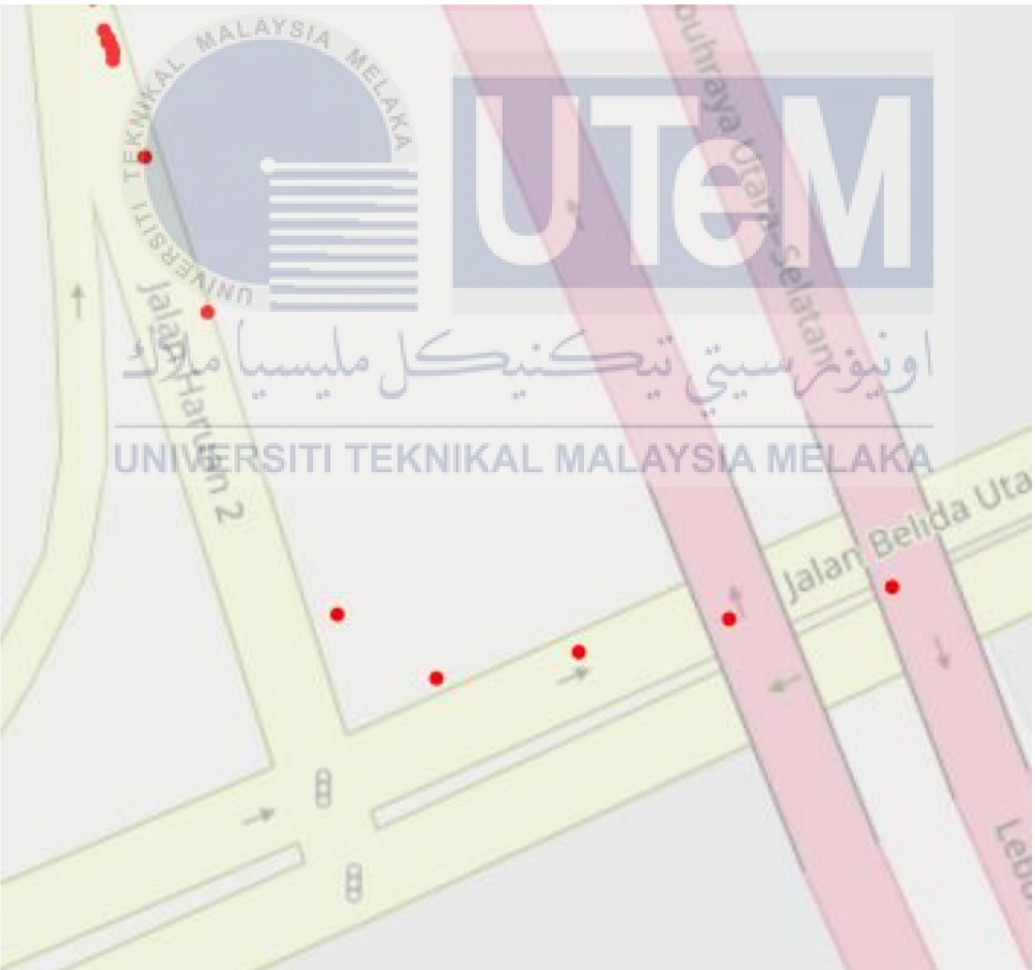


Figure 4-13 : Affected coordinates points due to covered area

Table 4-3 : Table of measured coordinates and actual coordinates passing through a covered area.

Point	Measure Longitude	Measure Latitude	Actual Longitude	Actual Latitude	Accuracy Latitude (%)	Accuracy Longitude (%)
1	101.925662	2.695392	101.928662	2.698392	99.99705668	99.88869894
2	101.925717	2.695256	101.945717	2.715256	99.98037787	99.25795546
3	101.925831	2.694991	101.926031	2.695191	99.99980378	99.99257882
4	101.925918	2.694935	101.929918	2.698935	99.99607558	99.85157341
5	101.926043	2.694958	101.931043	2.699958	99.99509448	99.81446835
6	101.926175	2.694987	101.928175	2.696987	99.9980378	99.92578814
7	101.926318	2.695015	101.927318	2.696015	99.9990189	99.96289446
8	101.92655	2.695123	101.92755	2.696123	99.9990189	99.96289594
9	101.926826	2.695252	101.928826	2.697252	99.99803781	99.92579544
10	101.927093	2.695388	101.930093	2.698388	99.99705672	99.88869877
11	101.927328	2.695496	101.930328	2.698496	99.99705673	99.88870323

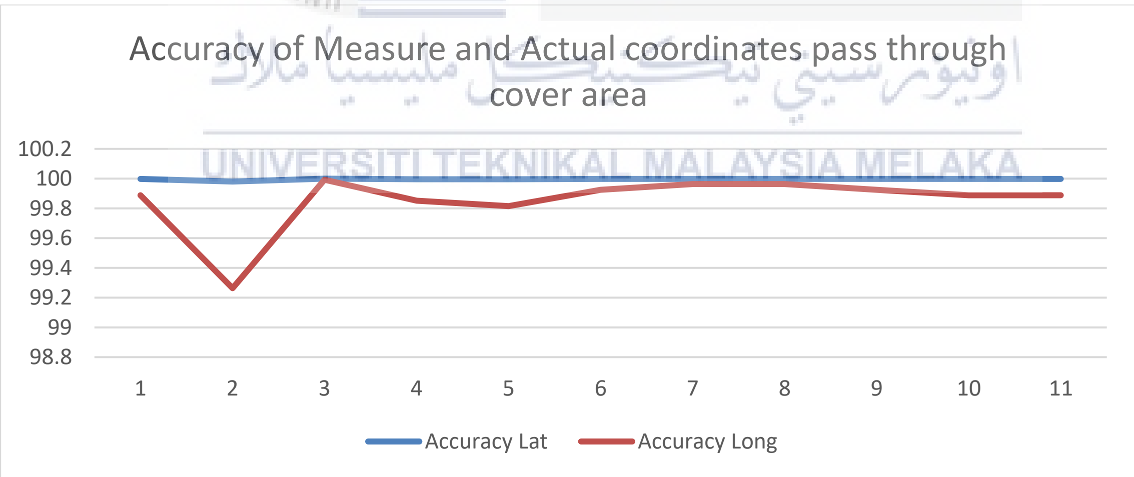


Figure 4-14 : Graph of accuracy of measured and actual coordinates passing through a covered area

From Table 4-3 and Figure 4-14, the average accuracy was calculated using Equation 5. Although there is visualization affected during mapping, it still has an average accuracy of 99.99% for latitude and 99.85% for longitude when passing through covered areas. Thus, it still offers precise and correct coordinates even when passing through cover area.

4.3 Testing the Accuracy and Realibility of Speed Monitoring of DMAS

Experiment 2 : Testing the accuracy of speed monitoring by comparing the speedometer readings with the actual speed limits in the CSV file.

The second experiment involved testing the GNSS connectivity and accuracy in displaying a moving vehicle's speed. The coding was adjusted and uploaded to the Sony Spresense, which was connected to a USB hub in the car to power on the device. A TFT LCD display was connected to the Sony Spresense to view the speed limit and actual speed in real-time. The SD card was also uploaded with the Speed Limit CSV, as shown in Figure 3-30. The test involved recording the speed from the car's speedometer and the speed from the DMAS, as shown in Figure 4-15. The accuracy of the speed provided by the DMAS was checked every second by tabulating the data, as shown in Table 4-4.

Figure 4-15 : Recording of speed from DMAS and speedometer

Table 4-4 : Table of Measured speed and actual speed

Point	Measured Speed (KM/H)	Actual Speed (KM/H)	Percentage of Error (%)	Accuracy (%)
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	3.23	5	35.4	64.6
10	15.51	15	-3.4	103.4
11	16.7	17	1.764705882	98.23529412
12	19.32	20	3.4	96.6
13	23.9	25	4.4	95.6
14	23.9	25	4.4	95.6
15	36.47	39	6.487179487	93.51282051
16	39.82	43	7.395348837	92.60465116
17	39.44	43	8.279069767	91.72093023
18	41.75	43	2.906976744	97.09302326
19	45.61	49	6.918367347	93.08163265
20	44.93	47	4.404255319	95.59574468
21	45.46	47	3.276595745	96.72340426
22	44.93	45	0.155555556	99.84444444
23	43.27	44	1.659090909	98.34090909
24	38.21	40	4.475	95.525
25	29.84	38	21.47368421	78.52631579
26	29.05	35	17	83
27	29.79	30	0.7	99.3
28	31.31	33	5.121212121	94.87878788
29	35.95	39	7.820512821	92.17948718
30	39.43	42	6.119047619	93.88095238
31	43.97	45	2.288888889	97.71111111
32	43.44	48	9.5	90.5
33	47.68	50	4.64	95.36
34	48.78	50	2.44	97.56
35	50.9	54	5.740740741	94.25925926

36	51.6	54	4.444444444	95.55555556
37	53.17	54	1.537037037	98.46296296
38	53.17	56	5.053571429	94.94642857
39	53.17	55	3.327272727	96.67272727
40	52.55	54	2.685185185	97.31481481
41	52.07	54	3.574074074	96.42592593
42	53.85	56	3.839285714	96.16071429
43	54.28	55	1.309090909	98.69090909
44	54.71	55	0.527272727	99.47272727
45	55.14	57	3.263157895	96.73684211
46	55.57	57	2.50877193	97.49122807
47	56	56	0	100
48	56.43	57	1	99
49	56.85	57	0.263157895	99.73684211
50	57.28	60	4.533333333	95.46666667
51	57.71	59	2.186440678	97.81355932
52	58.14	59	1.457627119	98.54237288
53	51.6	53	2.641509434	97.35849057
54	47.68	49	2.693877551	97.30612245
55	39.43	42	6.119047619	93.88095238
56	38.503	37	-4.062162162	104.0621622
57	34.898	33	-5.751515152	105.7515152
58	31.293	30	-4.31	104.31
59	27.688	28	1.114285714	98.88571429
60	24.083	26	7.373076923	92.62692308
61	23.9	24	0.416666667	99.58333333
62	23.9	24	0.416666667	99.58333333
63	36.47	38	4.026315789	95.97368421
64	39.82	42	5.19047619	94.80952381
65	39.44	44	10.36363636	89.63636364
66	41.75	45	7.222222222	92.77777778
67	45.61	47	2.957446809	97.04255319

68	44.93	48	6.395833333	93.60416667
69	45.46	50	9.08	90.92

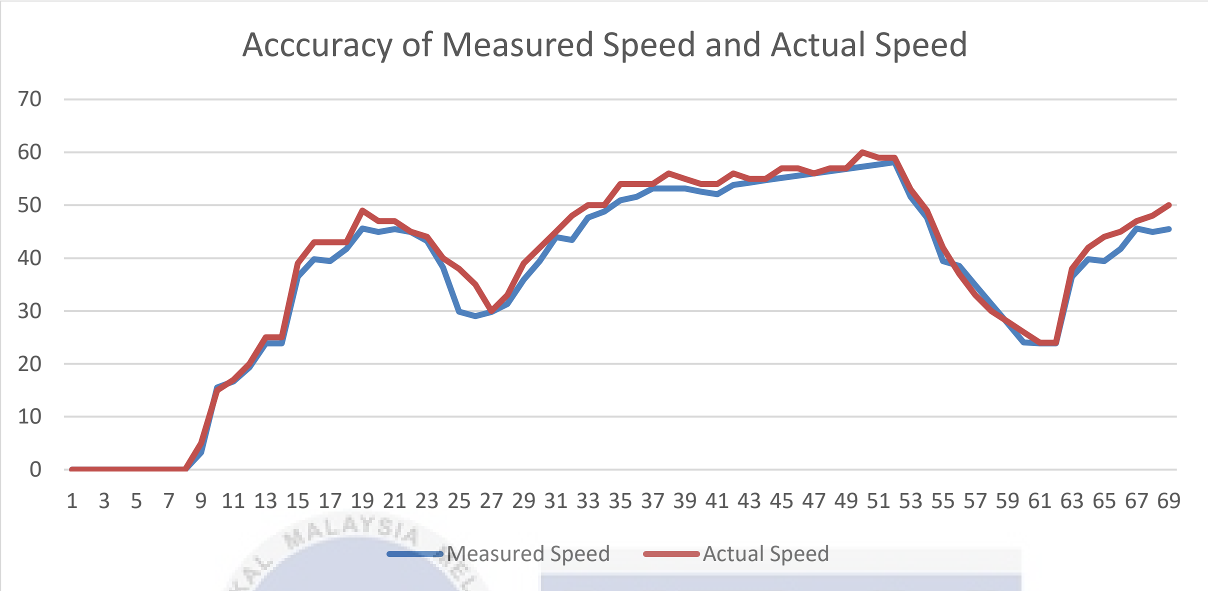


Figure 4-16 : Graph of accuracy of measured speed and actual speed

From Table 4-4 and Figure 4-16, the average accuracy of the speed was calculated using Equation 5. The average accuracy for the speed was 96.73648787%, or 96.73%, which is acceptable. However, since the DMAS works in real-time, an accuracy of 99% should be reached for more accurate results and to alert the driver on time. This accuracy might be affected by the delay in connection between the Sony Spresense and satellites. Nevertheless, the result is still considered acceptable since it exceeds 90%, making this speed result usable for the DMAS system. The speed limit was saved as shown in Figure 3-30. The speed limit test was conducted by checking if the DMAS would change the TFT LCD to the exact speed limit when reaching certain roads with specific coordinates. The speed limit was compared, as shown in Figures 4-17 and 4-18. The recorded result was tabulated in Table 4-5.

Figure 4-17 : Speed limit showing 80 km/h on TFT LCD display of DMAS

Figure 4-18 : Speed limit showing 80 km/h on the road sign

Table 4-5 : Table of speed limit at TFT LCD and Actual speed limit at specific coordinates

Road	Actual Longitude	Actual Latitude	Show Speed Limit (KM/H)	Actual Speed Limit (KM/H)	Accuracy (%)
1	102.128138	2.465526	110	110	100
2	101.936	2.695065	110	110	100
3	101.9361	2.6951	110	110	100
4	101.951143	2.702762	80	80	100
5	101.9315	2.695753	60	60	100
6	101.9327	2.695493	60	60	100
7	101.930732	2.69656	30	30	100
8	101.912	2.693536	30	30	100
9	101.9153	2.693907	30	30	100
10	101.9168	2.694077	30	30	100

The actual coordinates were recorded when a speed limit sign was found and compared with the speed limit shown on the DMAS TFT LCD display. From Table 4-5, all the displayed speed limits and actual speed limits were the same, thus providing an accuracy of 100%. This means the DMAS will show the correct speed limit when it detects the coordinates from the satellites, making the DMAS reliable for speed limit monitoring. The speed and speed limit tests confirm that they are acceptable and reliable, achieving Objective 1 as required by this project.

4.4 Testing the Reliability of the Alerting System of DMAS

Experiment 3 : Testing the accuracy of alerting mechanisms when the vehicle exceeds the speed limit.

The third experiment involved obtaining the speed and the speed limit in real-time and comparing them to determine if the vehicle was over speeding. If the vehicle exceeded the speed limit, the system would alert the driver using a buzzer and display "OVERSPEED" on the TFT LCD screen. The coding for the buzzer and the "OVERSPEED" display was uploaded to the Sony Spresense board. The DMAS box was placed on the car's dashboard to monitor the speed in real-time. The speed limit of the road was intentionally exceeded several times to ensure that the alerting system could alert the driver in real-time. The collected data was tabulated, where “1” in the table means YES or ON, and “0” means NO or OFF.

Table 4-6 : Result of alerting using buzzer and TFT LCD when exceed speed limit

Point	Speed Limit (km/h)	Measured Speed (km/h)	Difference	Overspeed	Buzzer	TFT LCD Display print “Overspeed”
1	30	23.9	-6.1	0	0	0
2	30	30.02	0.02	1	1	1
3	30	35.98	5.98	1	1	1
4	60	35.98	-24.02	0	0	0
5	60	60.19	0.19	1	1	1
6	60	67.93	7.93	1	1	1

7	80	67.93	-12.07	0	0	0
8	80	81.39	1.39	1	1	1
9	80	100.23	20.23	1	1	1
10	110	101.23	-8.77	0	0	0

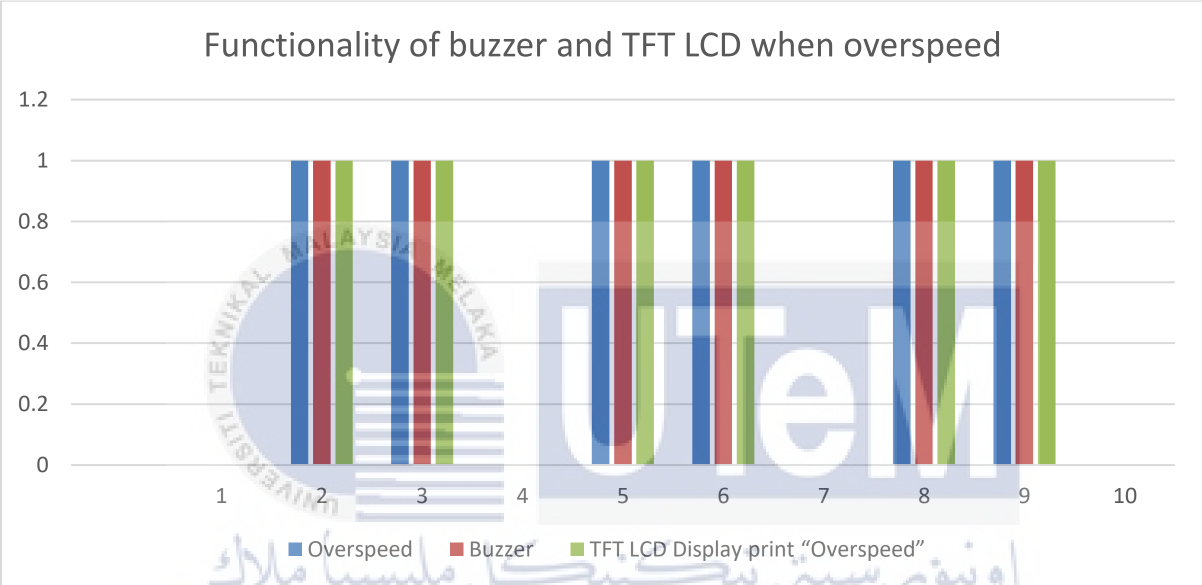


Figure 4-19 : Graph of functionality of buzzer and TFT LCD when overspeeding

Based on the recorded data, even if the driver exceeded the speed limit by just 0.01 km/h, the system would sound the buzzer and display "OVERSPEED" on the TFT LCD to alert the driver, as shown in Figure 4-20.

Figure 4-20 : Overspeeding shown on the TFT LCD of DMAS

The graph of the functionality of the buzzer and TFT LCD shows that when there is no overspeeding, the buzzer and TFT LCD do not alert the driver. However, when the driver exceeds the speed limit, the buzzer and TFT LCD alert the driver. This DMAS adjusts according to the printed speed limit and alerts the driver on different roads if the driver engages in speeding behavior. The functionality of the alerting system and the monitoring of speed and speed limits by DMAS with precise coordinates demonstrate that the Driver Monitoring and Alerting System (DMAS) successfully achieves Objective 1which to develop a Driver Monitoring and Alerting System (DMAS) that continuously monitors vehicle speed and provides real-time alerts to drivers when the speed limit is exceeded.

4.5 Testing the Reliability of DMAS application for map route, table result and Speeding Graph

Experiment 4 : Testing the overall reliability and performance of the DMAS application.

The fourth experiment tested the function and reliability of the Driver Monitoring and Alerting System (DMAS) application. The DMAS automatically logs the points, latitude, longitude, time, speed, and speed limit during the driver’s journey. When the driver stops, it automatically stops and closes the file. The SD card was removed from the DMAS box and uploaded to the DMAS application through a computer. The DMAS application allows viewing the map route, table results, and speeding graph by simply uploading the CSV file. The test was conducted by comparing the results shown in the DMAS application with the results from outside sources such as MYGEODATA converter and Microsoft Excel. The test utilized one of the recorded data sets which is data sets at 05-06-2024 at 1:52PM as figure 4-24.

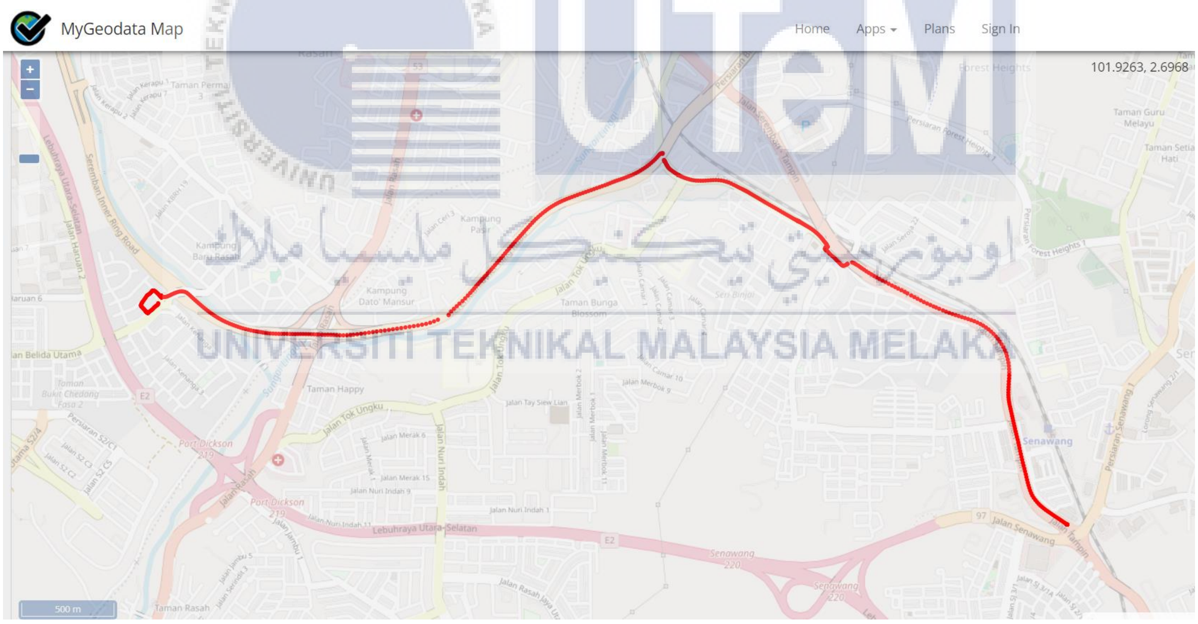


Figure 4-21 : Map route of MYGEODATA converter

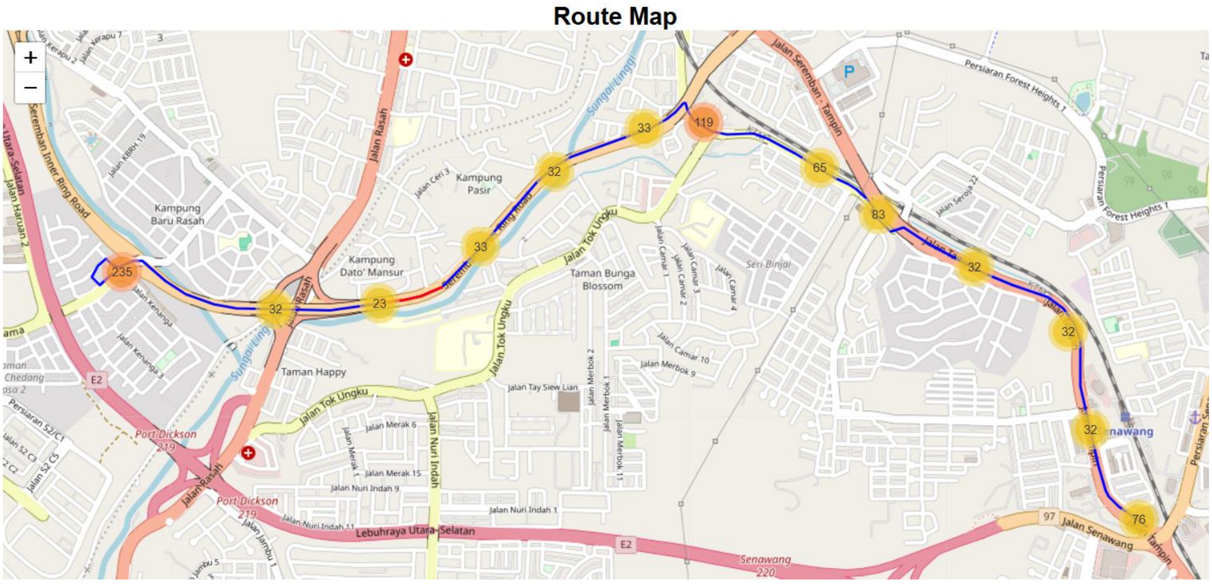
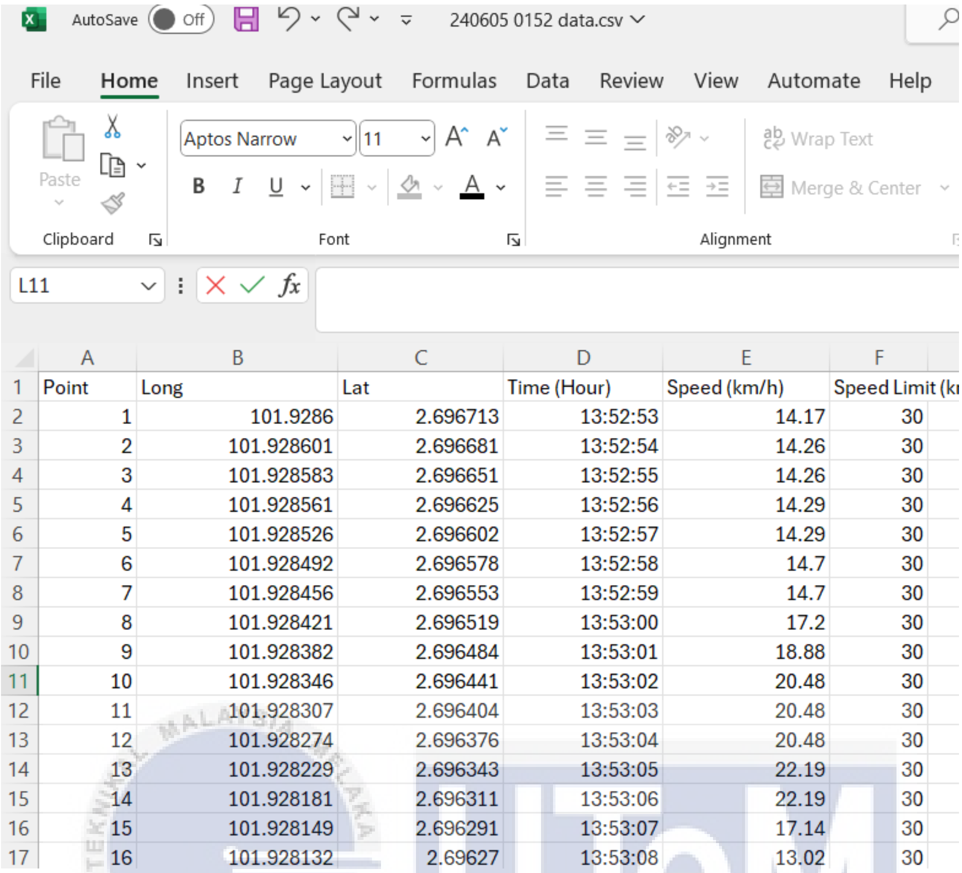


Figure 4-22 : Map route of DMAS application

Figure 4-23 : Point of map with information viewer



Point	Long	Lat	Time (Hour)	Speed (km/h)	Speed Limit (k
1	101.9286	2.696713	13:52:53	14.17	30
2	101.928601	2.696681	13:52:54	14.26	30
3	101.928583	2.696651	13:52:55	14.26	30
4	101.928561	2.696625	13:52:56	14.29	30
5	101.928526	2.696602	13:52:57	14.29	30
6	101.928492	2.696578	13:52:58	14.7	30
7	101.928456	2.696553	13:52:59	14.7	30
8	101.928421	2.696519	13:53:00	17.2	30
9	101.928382	2.696484	13:53:01	18.88	30
10	101.928346	2.696441	13:53:02	20.48	30
11	101.928307	2.696404	13:53:03	20.48	30
12	101.928274	2.696376	13:53:04	20.48	30
13	101.928229	2.696343	13:53:05	22.19	30
14	101.928181	2.696311	13:53:06	22.19	30
15	101.928149	2.696291	13:53:07	17.14	30
16	101.928132	2.69627	13:53:08	13.02	30

Figure 4-24: Logging CSV file for 05-06-2024 at 1:52 PM viewed in Microsoft Excel

Driver Monitoring Results

Point	Long	Lat	Speed (km/h)	Speed Limit (km/h)	Time (Hour)	Time of Day	Speeding Alert
1	101.928600	2.696713	14.17	30	13:52:53	Afternoon	No
2	101.928601	2.696681	14.26	30	13:52:54	Afternoon	No
3	101.928583	2.696651	14.26	30	13:52:55	Afternoon	No
4	101.928561	2.696625	14.29	30	13:52:56	Afternoon	No
5	101.928526	2.696602	14.29	30	13:52:57	Afternoon	No
6	101.928492	2.696578	14.70	30	13:52:58	Afternoon	No
7	101.928456	2.696553	14.70	30	13:52:59	Afternoon	No
8	101.928421	2.696519	17.20	30	13:53:00	Afternoon	No
9	101.928382	2.696484	18.88	30	13:53:01	Afternoon	No
10	101.928346	2.696441	20.48	30	13:53:02	Afternoon	No
11	101.928307	2.696404	20.48	30	13:53:03	Afternoon	No
12	101.928274	2.696376	20.48	30	13:53:04	Afternoon	No
13	101.928229	2.696343	22.19	30	13:53:05	Afternoon	No
14	101.928181	2.696311	22.19	30	13:53:06	Afternoon	No
15	101.928149	2.696291	17.14	30	13:53:07	Afternoon	No
16	101.928132	2.696270	13.02	30	13:53:08	Afternoon	No

Figure 4-25 : Table result of collected data of Logging CSV file for 05-06-2024 at 1:52 PM

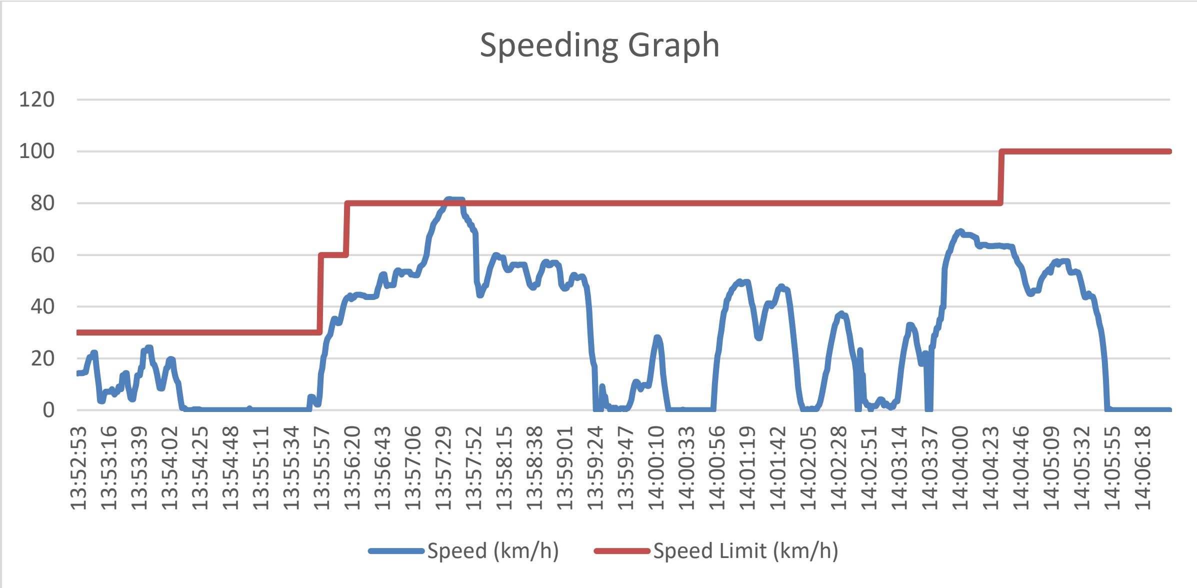


Figure 4-26: Speeding Graph by Microsoft Excel

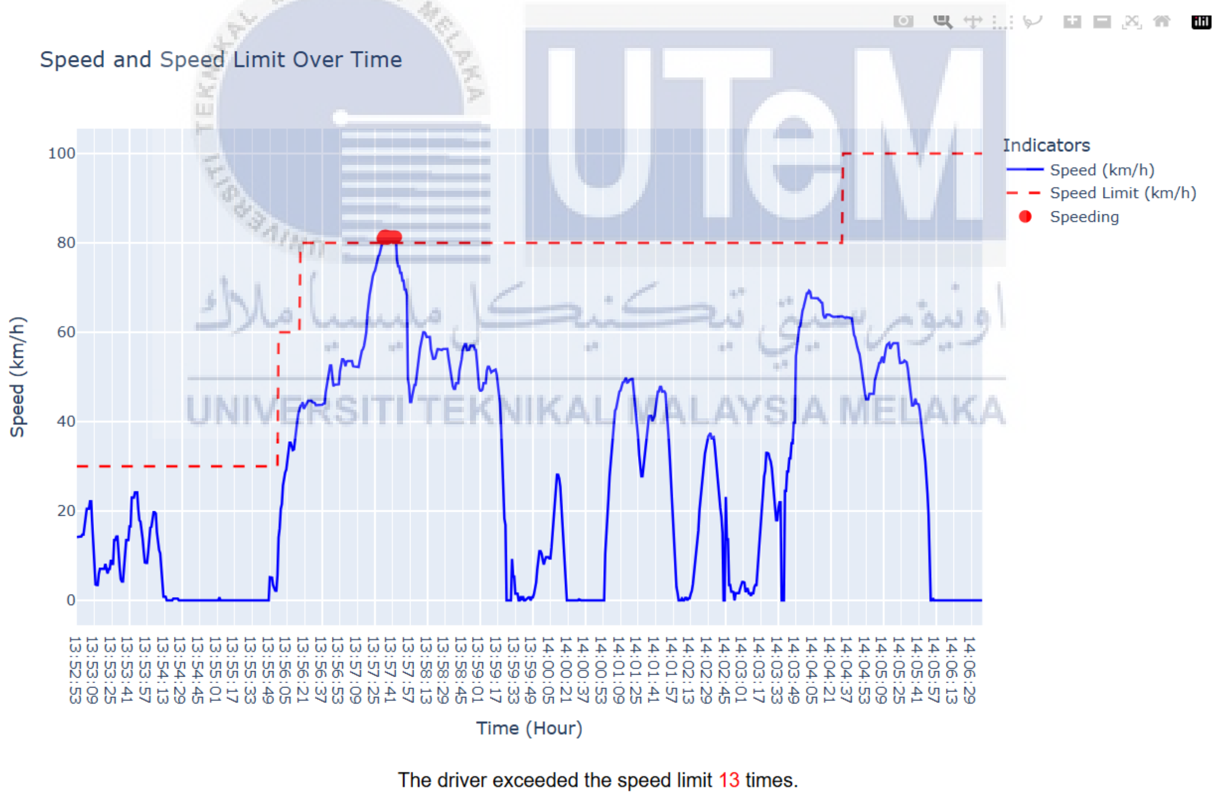


Figure 4-27 : Speeding Graph by DMAS application

Based on Figures 4-21 and 4-22, the map routes shown by the MYGEODATA converter and DMAS application are exactly the same, indicating that the map results are reliable. The DMAS application also includes a function to connect points with lines and shows a red line if overspeeding and a blue line when not overspeeding. The DMAS route map also has a function to click on the points to view information such as time, longitude, latitude, speed, speed limit, and overspeed status, as shown in Figure 4-23.

Figures 4-24 and 4-25 show that the table results of the DMAS application are the same as those viewed in Microsoft Excel for the Logging CSV file for 05-06-2024 at 1:52 PM. The DMAS application also includes additional columns such as the Time of Day, which states morning, afternoon, or night, depending on the time, and the time is shown in a 24-hour format. This shows that the table results in the DMAS application are reliable.

Figures 4-26 and 4-27 show the speeding graphs using the DMAS application and Microsoft Excel, and both of them are the same, as they generate the graph from the table result. The DMAS application has a function to click to view the value of the point and concludes the number of times the driver was overspeeding. The graphs of both applications are the same, indicating that the speeding graph of the DMAS application is reliable. All the map routes, table results, and speeding graphs were compared, and all achieved reliable results. In summary, Experiment 4 successfully fulfilled Objective 2 which is to analyze the collected data to identify driver behavior patterns utilizing Fuzzy Logic techniques .

4.6 Testing the accuracy and realibility of driver behavior using fuzzy logic technique in DMAS application

Experimetn 5 : Testing the fuzzy logic behavior scoring system in the DMAS app.

The fifth experiment focuses on assessing the accuracy and reliability of evaluating driver behavior using artificial intelligence, specifically the fuzzy logic technique. Fuzzy logic was chosen because driver behavior involves uncertainty and variability, which cannot be adequately represented by crisp values alone. Crisp values such as time, speed, and instances of exceeding the speed limit were collected, but these need to be interpreted within the context of different conditions and types of vehicles. For instance, the speed limits for different types of vehicles vary: heavy vehicles might have a speed limit of 90 km/h on a highway, while light vehicles might have a limit of 110 km/h. Additionally, factors like the time of day and the season, which affect daylight hours, introduce further variability. The fuzzy logic technique is essential for handling these uncertainties and providing a more nuanced assessment of driver behavior.

The crisp values are inputs to the fuzzy logic system, which then interprets these values to generate a driver behavior score or category. This method allows for a more flexible and context-sensitive evaluation, accommodating the variability in driving conditions and vehicle types. The fuzzy logic approach starts with defining the fuzzy sets.

Speed:

- Slow: 0,0,40,60
- Moderate: 40,60,80,100
- Fast: 90,100,180,180

Times Exceeding Speed Limit:

- Low: 0,0,10,15
- Moderate: 10,15,20,25
- High: 20,30,80,80

Time of Day:

- Morning: 5,6,12,13
- Afternoon: 12,13,18,19
- Early Night: 0,0,5,6
- Midnight: 18,19,24,24

Behavior Score:

- Excellent: 75,85,100,100
- Good: 55,65,75,85
- Moderate: 35,45,55,65
- Poor: 15,25,35,45
- Bad: 0,0,15,25

Based on the CSV file at 05-06-2024, the average speed was calculated using Equation 5:

$$\boldsymbol{Average\ speed} = \frac{\sum \boldsymbol{Speed_{827}}}{\boldsymbol{827}} = \mathbf{28.34\ km/h}$$

The times exceeding the speed limit are given as 13 times, and the time was around 13:52 until 14:02, thus is given as 14:00. Next, we proceed with the fuzzification to calculate the input membership levels. The graphs were drawn according to the fuzzy set with all graphs being trapezoidal in shape.

Given that average speed = 28.34km/h

Times Exceeding Speed Limit = 13

Average Times of Days = 14

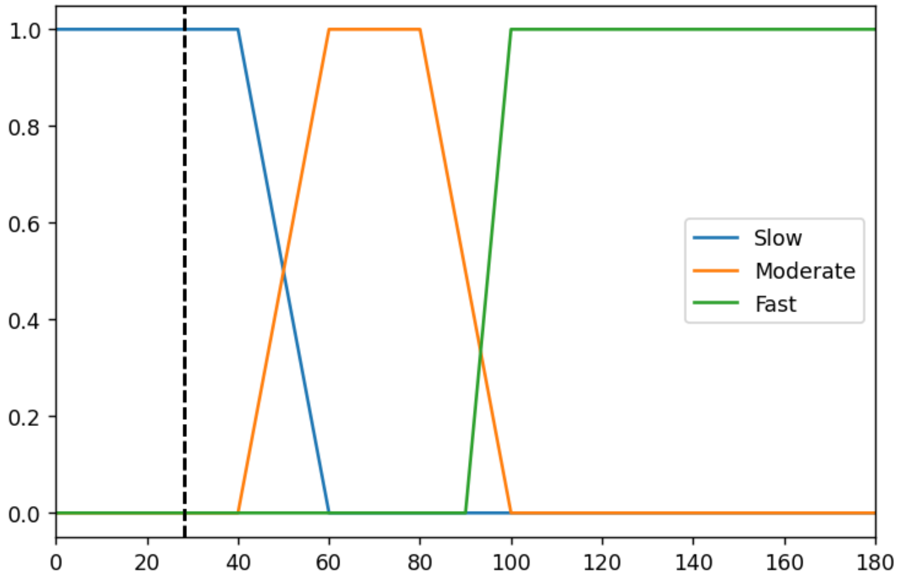


Figure 4-28 : Map the membership funciton of average speed

From the graph, the line points from the x-axis, which represents speed with 28.34 km/h, and the line connects with the y-axis membership function of 1.0 for the slow category. Thus:

Speed Membership Values:

$$\mu\text{Slow}(28.43) = 1$$

$$\mu\text{Moderate}(28.43) = 0$$

$$\mu\text{Fast}(28.43) = 0$$

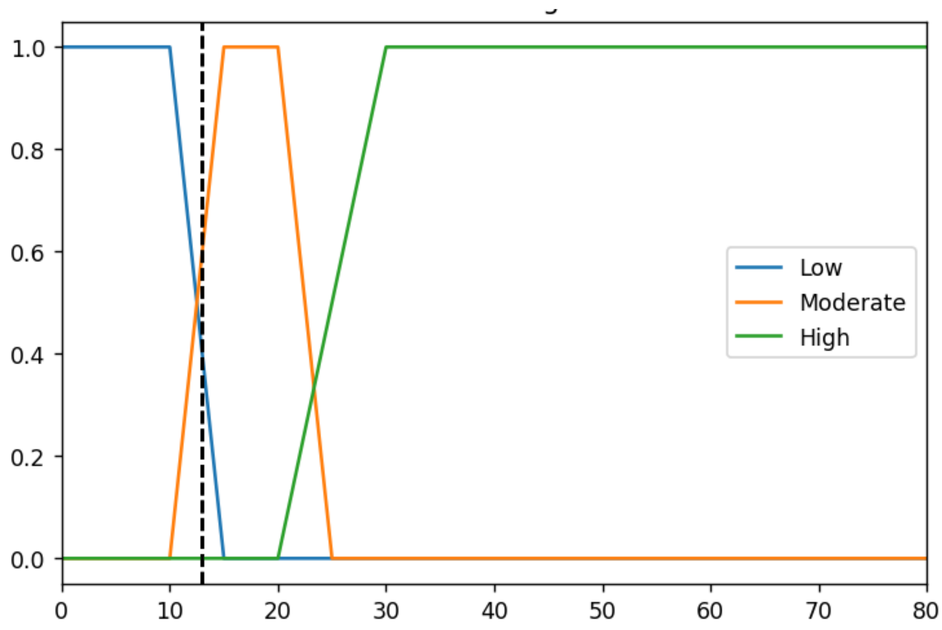


Figure 4-29 : Map the membership function for times of speed limit

From the graph, the line points from the x-axis, which represents times exceeding the speed limit with 13 times, and the line connects with the y-axis membership function of 0.4 for the low category and 0.6 for the moderate category.

Times Exceeding Speed Limit Membership Values:

It can also be calculated using the formulas below to get accurate value:

$$\mu \text{Low}(13) = \begin{cases} 1, if\ x \leq 0 \\ \frac{15 - x}{15 - 10}, if\ 10 < x \leq 15 \\ 0, if\ x > 15 \end{cases}$$

$$\mu \text{Moderate}(13) = \begin{cases} 1, if\ x \leq 10 \\ \frac{x - 10}{15 - 10}, if\ 10 < x \leq 15 \\ \frac{x - 20}{25 - 20}, if\ 20 < x \leq 25 \\ 0, if\ x > 25 \end{cases}$$

Since the given x = 13 times, the calculations are:

$$\mu \text{Low}(13) = \frac{15 - 13}{15 - 10} = \frac{2}{5} = 0.4$$

$$\mu\text{Moderate}(13) = \frac{13 - 10}{15 - 10} = \frac{3}{5} = 0.6$$

$$\mu\text{High}(13) = 0$$

This yields the same answer: a membership function of 0.4 for the low category and 0.6 for the moderate category, for either calculation or graph.

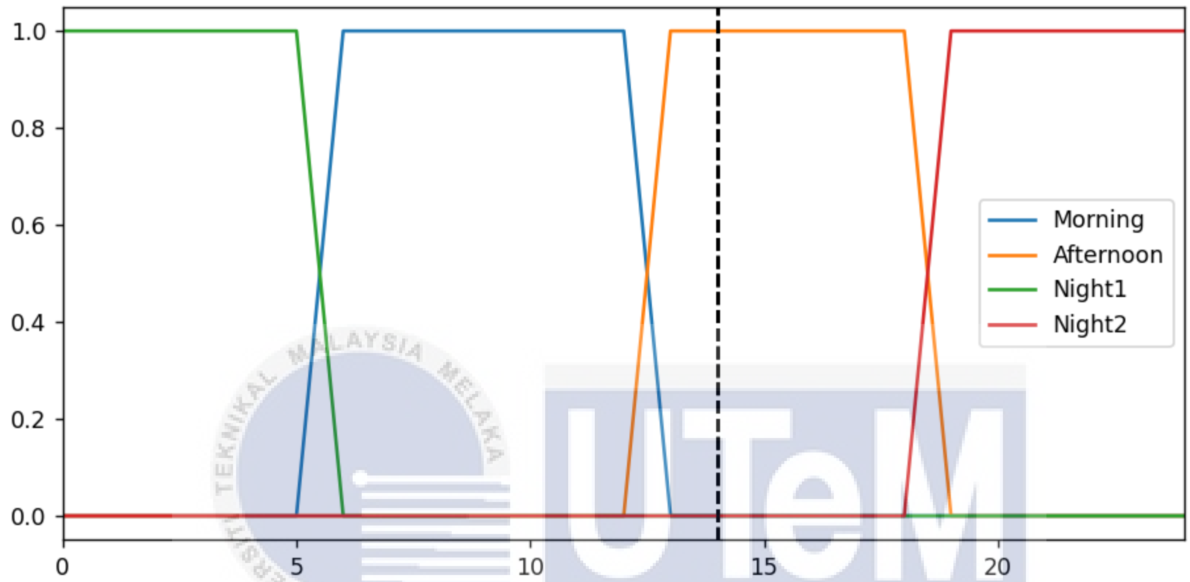


Figure 4-30: Map the membership fucntion for times of day

From the graph, the line points from the x-axis, which represents times around the 14th hour, and the line connects with the y-axis membership function of 1.0 for the afternoon category.

$$\boldsymbol{\mu\text{Morning}(13.87) = 0}$$

$$\boldsymbol{\mu\text{Afternoon}(13.87) = 1.0}$$

$$\boldsymbol{\mu\text{Night1}(13.87) = 0}$$

$$\boldsymbol{\mu\text{Night2}(13.87) = 0}$$

Next, the fuzzy rules were applied utilizing the fuzzy table as illustrated in Figure 4.31, and the max-min method was used to calculate the behavior score.

Fuzzy Logic Rules Table:

Time of Day	Average Speed	Times		
		Low	Moderate	High
Morning	Slow	Excellent	Good	Poor
	Moderate	Good	Moderate	Bad
	Fast	Moderate	Poor	Bad
Afternoon	Slow	Excellent	Good	Poor
	Moderate	Good	Moderate	Bad
	Fast	Moderate	Poor	Bad
Night	Slow	Excellent	Good	Poor
	Moderate	Good	Poor	Bad
	Fast	Moderate	Poor	Bad

Figure 4-31: Fuzzy logic Rules Table

Based on the fuzzy table, the relevant rules for the "Afternoon" period are:

If it is Afternoon and the speed is Slow and the exceeding limit is Low, then the behavior is Excellent:

$$Min\ (1,1,0.4) = 0.4$$

$$Excellent = 0.4$$

If it is Afternoon and the speed is Moderate and the exceeding limit is Low, then the behavior is Good:

Good.

$$Min\ (1,1,0.6) = 0.6$$

$$Good = 0.6$$

Aggregating the Results:

$$Excellent = 0.4\ , Good = 0.6$$

The last step is defuzzification using the centroid method:

$$\textbf{Centroid Excellent} = \frac{75 + 85 + 100 + 100}{4} = \frac{360}{4} = 90$$

$$\textbf{Centroid Good} = \frac{55 + 65 + 75 + 85}{4} = \frac{280}{4} = 70$$

Thus, the behavior score is

$$\textbf{behaviour\_score} = \frac{(0.4 \times 90) + (0.6 \times 70)}{0.4 + 0.6} = \frac{36 + 42}{1} = 78$$

Based on fuzzy logic, the driver's behavior score is considered as 78, and the driver behavior is categorized as Good.

The fuzzy logic interpretation based on the DMAS application is shown in the following figures:

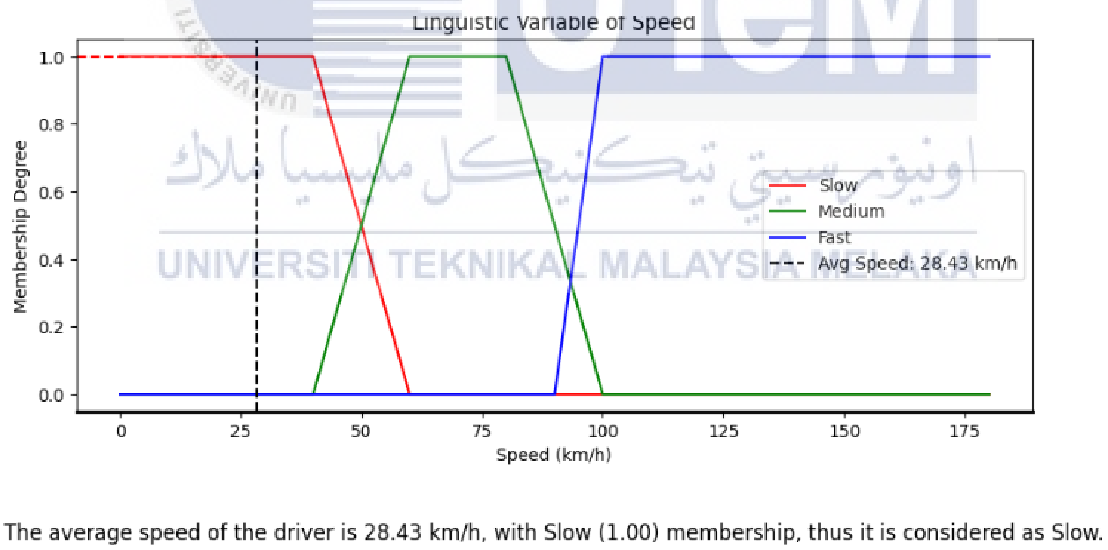
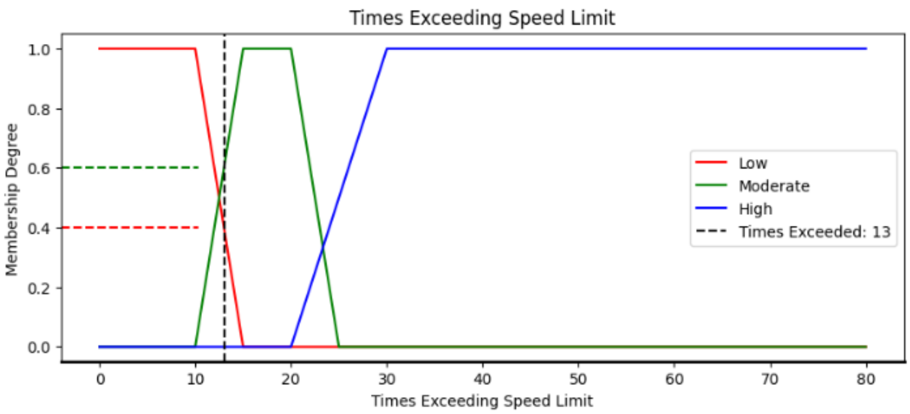
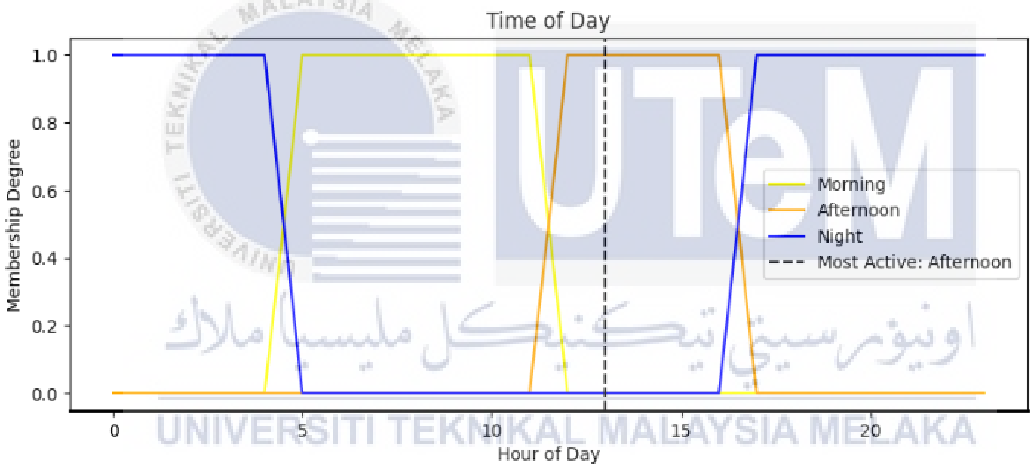


Figure 4-32 : DMAS app map of membership function speed



The driver exceeded the speed limit 13 times, with Low (0.40), Moderate (0.60) membership, thus it is considered as Moderate.

Figure 4-33 : DMAS app map on memerbship fucntion of times exceeding speed limit



The driver drove mostly during the afternoon, thus it is considered as Low Risk.

Based on fuzzy logic interpretation, the driver's behavior is considered: 78, and the driver behavior is Good.

Figure 4-34 : DMAS application map of membership function time of day and conclusion of driver behavior score and category.

Based on the comparison of calculations and the DMAS application, the fuzzy logic in the DMAS application was correct and reliable. The driver behavior score and category comply with the rules of the fuzzy logic table. This experiment successfully fulfills Objective 2 of the project, which is to analyze the collected data to identify driver behavior patterns utilizing fuzzy logic techniques.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the objectives of this project was to develop a Driver Monitoring and Alert System (DMAS) to continuously monitor vehicle speed and provide real-time alerts when speed limits are exceeded and to analyze driver behavior patterns using Fuzzy Logic techniques. The system comprised several key components and experiments, all of which have been successfully executed. The first experiment assessed the accuracy of the GNSS coordinates received by the Sony Spresense, which was used to map the driver's route. This experiment demonstrated exceptional accuracy, achieving a 99.99% match between real and measured coordinates, confirming the system's reliability. However, the accuracy of coordinates dropped slightly to 99.85% in covered areas. The second experiment evaluated the accuracy and reliability of the DMAS in monitoring vehicle speed. By comparing the actual speed from the car's speedometer with the measured speed from the DMAS and the speed limits indicated by road signs, the system achieved an accuracy of approximately 96.73%, which is acceptable. Real-time speed monitoring was predicted to achieve around 99.99% for more reliable result. The third experiment tested the effectiveness of the buzzer and TFT LCD display in alerting the driver when the speed limit is exceeded, even by a margin of 0.01 km/h. The DMAS application was validated by comparing route maps, tabulated results, and speed graphs with other sources, consistently obtaining accurate and reliable results. The system's fuzzy logic was tested through calculations, accurately determining driver behavior scores and categories as outlined in the fuzzy logic table. The results indicated that the DMAS was able to accurately categorize driver behavior and provide meaningful insights. Overall, the project has successfully met its objectives, demonstrating the feasibility and effectiveness of the DMAS. However, further refinements are recommended to improve the real-time speed monitoring accuracy, which, despite being high, did not fully meet the expected standards. The comprehensive coverage in the introduction, literature review, methodology, results, and discussion sections underscore the project's success and potential for future improvements system.

5.2 Future Work

To improve the accuracy of speed monitoring in the Driver Monitoring and Alert System (DMAS), future work should focus on integrating advanced GPS RTK modules instead of GNSS modules. GPS RTK modules offer centimeter-level differential positioning and orientation. With Real-Time Kinematic (RTK) positioning technology, these modules maintain signal quality even in urban or heavily wooded areas. This advancement will exceed current expectations, enhancing road safety and driver behavior monitoring. By incorporating these high-precision GPS RTK modules, future iterations of the DMAS can achieve greater accuracy in speed monitoring. This leads to more reliable real-time alerts and better overall system performance, making the DMAS a more effective tool for improving road safety and driver behavior monitoring.

REFERENCES

[1] Mahat, Jamil, & Sarah Raseli. (2020, August). ANALYSING ROAD ACCIDENT TRIGGERS IN MALAYSIA BY USING ANALYTICAL HIERARCHY PROCESS. GADING Journal of Science and Technology Vol 3 No (2) (2020), eISSN: 2637-0018, 118–125. https://ir.uitm.edu.my/id/eprint/46019/1/46019.pdf

[2] Nagar, R. (2020). Speed Tracking Vehicle System. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3608672

[3] Pelaez C., G. A., Garcia, F., de la Escalera, A., & Armingol, J. M. (2014, August). Driver Monitoring Based on Low-Cost 3-D Sensors. IEEE Transactions on Intelligent Transportation Systems, 15(4), 1855–1860. https://doi.org/10.1109/tits.2014.2332613

[4] P. A. Shinde and Y. B. Mane, "Advanced vehicle monitoring and tracking system based on Raspberry Pi," 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 2015, pp. 1-6, doi: 10.1109/ISCO.2015.7282250.

[5] Idris, A., Hamid, H., & Teik Hua, L. (2019, November 1). Factors contributing to motorcycle accidents in Malaysia. IOP Conference Series: Earth and Environmental Science, 357(1), 012039. https://doi.org/10.1088/1755-1315/357/1/012039

[6] Speeding - Injury Facts. (2023, May 4). Injury Facts. https://injuryfacts.nsc.org/motor-vehicle/motor-vehicle-safety-issues/speeding/

[7] S.Dukare, S., A. Patil, D., & P. Rane, K. (2015, June 18). Vehicle Tracking, Monitoring and Alerting System: A Review. International Journal of Computer Applications, 119(10), 39–44. https://doi.org/10.5120/21107-3835

[8] A. H. Alquhali, M. Roslee, M. Y. Alias and K. S. Mohamed, "IOT Based Real-Time Vehicle Tracking System," 2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET), Penang, Malaysia, 2019, pp. 265-270, doi: 10.1109/CSUDET47057.2019.9214633. https://ieeexplore.ieee.org/document/9214633

[9] M. M. Rahman, J. R. Mou, K. Tara and M. I. Sarkar, "Real time Google map and Arduino based vehicle tracking system," 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), Rajshahi, Bangladesh, 2016, pp. 1-4, doi: 10.1109/ICECTE.2016.7879577.

[10] Tummanapally, S. S., & Sunkari, S. (2021). Smart Vehicle Tracking System using GPS and GSM Technologies. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3884903

[11] M. Desai and A. Phadke, "Internet of Things based vehicle monitoring system," 2017 Fourteenth International Conference on Wireless and Optical Communications Networks (WOCN), Mumbai, India, 2017, pp. 1-3, doi: 10.1109/WOCN.2017.8065840.

[12] M. A. Khan and S. F. Khan, "IoT based framework for Vehicle Over-speed detection," 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2018, pp. 1-4, doi: 10.1109/CAIS.2018.8441951.

[13] Jeddi, Sayed & Hassouna, Ahmed & Shahin, Mahmood & Mir, Mustahsan. (2013). Design and Implementation of an Over-Speed Monitoring and Recording System. i Zhao, Jegatheesan, & Chee Loon. (2015, February). Exploring IOT Application Using Raspberry Pi. Exploring IOT Application Using Raspberry Pi, 2(1), 27–34. https://ijcna.org/Manuscripts/Volume-2/Issue-1/Vol-2-issue-1-M-04.pdf

[14] and G. Harish, "Design and implementation of real time vehicle monitoring, tracking and controlling system," 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2016, pp. 1-4, doi: 10.1109/CESYS.2016.7889814.

[15] E. B. Panganiban and J. C. Dela Cruz, "RFID-based vehicle monitoring system," 2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Manila, Philippines, 2017, pp. 1-6, doi: 10.1109/HNICEM.2017.8269489.

[16] Hlaing, N. N. S., Naing, M., & Naing, S. S. (2019, June 30). GPS and GSM Based Vehicle Tracking System. International Journal of Trend in Scientific Research and Development, Volume-3(Issue-4), 271–275. https://doi.org/10.31142/ijtsrd23718

[17] M. S. Uddin, M. M. Ahmed, J. B. Alam and M. Islam, "Smart anti-theft vehicle tracking system for Bangladesh based on Internet of Things," 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh, 2017, pp. 624-628, doi: 10.1109/ICAEE.2017.8255432. https://ieeexplore.ieee.org/document/8255432

[18] Isong, B., Khutsoane, O., & Dladlu, N. (2017). Real-time Monitoring and Detection of Drink-driving and Vehicle Over-speeding. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, *9*(11), 1-9.

[19] G. Castignani, T. Derrmann, R. Frank and T. Engel, "Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring," in IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 1, pp. 91-102, Spring 2015, doi: 10.1109/MITS.2014.2328673

[20] I. S. Feraud, M. M. Lara and J. E. Naranjo, "A fuzzy logic model to estimate safe driving behavior based on traffic violation," 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), Salinas, Ecuador, 2017, pp. 1-6, doi: 10.1109/ETCM.2017.8247536.

APPENDICES

APPENDIX A GANTT CHART

	FINAL YEAR PROJECT 1														FINAL YEAR PROJECT 2													
Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Project Briefing																												
Literature Review																												
1) Research and review academic journal																												
2) Summarize and conclude academic journal																												
Introduction																												
1) Define problem statement , objective and scopes																												
Methodology																												
1) Coding and programming to connect microcontroller with satelite																												
2) Collect and analysis data obtain to visualise on map																												
3) Develop interface for alert driver and observer																												
4) Design cover for protection of microcontroller																												
Seminar FYP 1																												
Prepare and Finalize report FYP 1																												
Seminar FYP 2																												
Prepare and Finalize report FYP 2																												

APPENDIX B DMAS CODING AND DATA ANALYSIS

Coding for DMAS app and System

Coding for DMAS application(python)

```python
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
import pandas as pd
import webbrowser
import folium
from folium.plugins import MarkerCluster, TimestampedGeoJson
import matplotlib.pyplot as plt
from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas
import io
import base64
import os
```

Coding for DMAS on sony spresense(Arduino IDE)

```cpp
#include <GNSS.h>
#include <GNSSPositionData.h>
#include <GNSS.h>
#define STRING_BUFFER_SIZE  128
#define RESTART_CYCLE       (60 * 5)
#include <Adafruit_GFX.h>    // Core graphics library
#include <Adafruit_ST7735.h> // Hardware-specific library for ST7735
#include <SPI.h>
#define TFT_CS         10
```


DMAS data analysis

10	9	101.9284	2.696484	13:53:01	18.88	30
11	10	101.9283	2.696441	13:53:02	20.48	30
12	11	101.9283	2.696404	13:53:03	20.48	30
13	12	101.9283	2.696376	13:53:04	20.48	30
14	13	101.9282	2.696343	13:53:05	22.19	30
15	14	101.9282	2.696311	13:53:06	22.19	30
16	15	101.9281	2.696291	13:53:07	17.14	30
17	16	101.9281	2.69627	13:53:08	13.02	30
18	17	101.9281	2.696259	13:53:09	9.19	30
19	18	101.9281	2.696248	13:53:10	3.53	30
20	19	101.9281	2.696236	13:53:11	3.47	30
21	20	101.9281	2.696236	13:53:12	3.47	30
22	21	101.9281	2.696239	13:53:13	5.9	30
23	22	101.9281	2.696247	13:53:14	7.08	30
24	23	101.9281	2.696259	13:53:15	7.08	30

Coordinates | Accuracy of route | Accuracy of coordinates on cove | Speed | data240605 1352

APPENDIX C CAD DRAWING