



Design and Implementation of a Car's Black Box System using Arduino

Rajendran Thanikachalam ^a, Rajendran Thavasimuthu ^b, J. Godwin John ^{c,*},
J. Maria Arockia Dass ^d, T. Nithya ^d, Anitha Thavasimuthu ^e

^a Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India

^b Department of Sustainable Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India.

^c Department of Mechanical Engineering, Rajalakshmi Institute of Technology (Autonomous), Chennai, Tamil Nadu, India.

^d Department of Computer Science and Business Systems, Rajalakshmi Institute of Technology (Autonomous), Chennai, Tamil Nadu, India.

^e Department of Artificial Intelligence and Data Science, SNS College of Engineering (Autonomous), Coimbatore, Tamil Nadu, India

* Corresponding Author Email: godwinjohn18@gmail.com

DOI: <https://doi.org/10.54392/irjmt24320>

Received: 22-11-2023; Revised: 09-04-2024; Accepted: 26-04-2024; Published: 22-05-2024



Abstract: A black box system (BBS) in a car is crucial for recording and analyzing critical data to enhance safety, investigate accidents, and improve vehicle performance. This research presents a BBS developed using Arduino for cars, aimed at using the power of modern technology for comprehensive data capture and analysis in vehicular contexts. The BBS, or Event Data Recorder (EDR), is an essential component for enhancing road safety, accident analysis, and overall vehicle performance evaluation. The proposed system uses Arduino, a versatile and cost-effective microcontroller platform, to create a robust and customizable solution. It integrates various sensors and data acquisition modules to collect critical data points, including speed, acceleration, GPS coordinates, engine performance, and vehicle diagnostics. The architecture of the system and its smooth integration into automobiles are described in this article through detailed hardware and software design. Data retrieval and analysis are made possible by the system's user-friendly interface, which helps with fleet management, driver behaviour analysis, and accident investigation. This paper addresses the importance of data privacy and security while highlighting technological improvements. It proposes measures to ensure that personal data is managed responsibly and in accordance with legal requirements. In conclusion, a major advancement in improving road safety and vehicle monitoring has been made with the integration of Arduino technology into the car's BBS. Considering data security and privacy, this system provides users with an extensive set of facts to enable them to make well-informed decisions.

Keywords: Black Box System, Arduino, Sensors, Microcontroller, Automobile, Safety.

1. Introduction

In today's rapidly evolving world, where technology intersects with our daily lives in countless ways, the automotive industry has witnessed a significant transformation. As a result, there occurs an accident that causes delays in traffic and assistance is delayed; road accidents are the primary cause of accident deaths worldwide. Due to the lack of quick access to safety facilities, this destroys the property and results in accidental deaths. A new era of unchallenged safety, efficiency, and data-driven insights has begun with the introduction of smart cars and networked systems. Car BBSs are at the core of this technology revolution; these amazing inventions have evolved beyond their initial use of recording accident data to become a necessary part of today's automobiles [1].

The people have conducted many public awareness efforts. However, because of riders' poor behaviours like speeding, riding without enough sleep, driving while drunk, etc., this issue is still becoming worse. Due to the delayed help provided to those injured in the accident, there are an excessive number of deaths and disabilities. For the individuals involved, these result in major social and economic difficulties [2]. Accidents are also rapidly rising because of the continual increase in car usage. The World Health Organization (WHO) reports that 1.3 million people die, and millions of people are injured in traffic accidents yearly. Drunk driving, inadequate sleep, and an incorrect gap between two vehicles are the main causes of accidents [3].

Road traffic accidents were the fourth most common cause of death worldwide in 2008. Globally, traffic-related accidents kill the lives of about 1.3 million

people years. Twenty to fifty million more individuals get non-fatal injuries, many of which lead to disabilities. The most common cause of death for young individuals between the ages of 15 and 29 is traffic-related injuries. It is estimated that by 2020, road traffic crashes will claim the lives of around 1.9 million people yearly if nothing is done. Those who survive have a significant risk of developing a disability because of the damage. 91% of traffic deaths take place in middle-class and lower-class nations [4].

This issue is still getting worse even though efforts to educate because of bad riding habits like speeding, intoxication, and riding without enough sleep, etc. Due to delayed aid to those who were involved in the accident, there are an unnatural number of deaths and disabilities. These place a significant financial and social burden on accident victims. Hence, a lot of research teams and significant vehicle producers have created safety equipment to protect riders from accidents and injury. Therefore, installing a reliable safety system in a car can be costly and challenging. Drivers often maintain a safe distance from one another when driving. However, a severe collision could happen because of another car's abrupt break, the driver's interruption, or tiredness from long driving. Despite being aware of the situation, the driver is unable to react quickly enough to control the vehicle. Unfavourable meteorological conditions, such as moisture, fog, and so on, can occasionally cause crashes [5]. BBSs are typically employed in aircraft to gather data for accident investigations. The proposed concept similarly employs black box technology for moving cars. Important data are locally stored and retrieved from all the data via the Internet of Things, including the engine temperature, vibrations, and the driver's alcohol intake. Information related to those accidents is needed to control the cause of an accident or identify the criminal. This type of data is typically obtained by investigators through rumour or questioning people who happen to be in the area at the time of the accident. Black boxes have been widely utilized in cars in recent years to identify the defective part that caused an accident [6].

1.1. The Evolution of Car's BBS

The EDR, commonly referred to as the Car's BBS, has experienced a significant transformation since its creation. Initially designed to record vital information in the event of a car accident, it has developed into a versatile tool that documents the minute details of a vehicle's travels. Conventional EDRs have faithfully captured data on airbag deployment, speed, and acceleration, which have been crucial resources for accident reconstruction and analysis. However, the potential applications of car BBSs have grown significantly in recent years. These days, modern versions of these systems not only keep track of crucial accident information, but they also cast a wider net, collecting GPS locations, audio and video recordings,

engine performance indicators, and even detailed analyses of driver behaviour. This extended scope has brought about a new era; these systems are now vital resources for fleet management, insurance investigations, and the objective of increased road safety, rather than only being limited to accident analysis.

In addition to automatically detecting accidents, car BBSs record the driver's activities and the vehicle's motion for a predetermined amount of time both before and after the impact. It consists of a CPU for unit control, a wireless modem for communication, and data collection devices for gathering information about the driver's activities and the status of the car. The development of a Black Box vehicle diagnostic model that can be put in any vehicle is the primary goal of the proposed work. Improved road conditions, safer automobile design, insurance companies' assistance in vehicle crash investigations, and the treatment of crash victims can all contribute to a reduction in the death rate.

Car black boxes record accident-related data. Car black boxes store driving information, image information, collision information, and location information prior to and following collisions. This information is used to easily assess the incident and settle a variety of automobile accident-related disputes. Along with its standard features, the car's black box containing a wireless communications system can also transmit accident location data to a central emergency. As a result, the number of deaths is reduced since drivers who need assistance can get it fast from police, hospital ambulances, and rack cars. Moreover, this initiative is dedicated to maintaining the highest standards of personal data protection at a time when data security and privacy are crucial. The need to uphold people rights to privacy while also utilizing technology to its maximum efficiency for safety is considered. The sensors used in this system uses Raspberry Pi and Arduino Controllers [7].

1.2. Research Objectives

The primary objective of this project is to design, develop, and implement a sophisticated Car's BBS utilizing Arduino technology. This system will provide an in-depth solution for data acquisition, analysis, and real-time monitoring within the vehicular context. This project intends to build a flexible and easily accessible tool for car owners, accident investigators, and fleet managers by combining many sensors and data-gathering modules and making sure the interface is easy to use.

The project is outlined to define the boundaries, objectives, and output of the project. It includes a range of elements and parts that will be produced and considered while the project is being carried out: The project involves selecting and integrating sensors and data-collecting modules into the car's BBS. Examples of these include accelerometers, GPS modules,

microphones, cameras, and engine diagnostic interfaces. For hardware development, Arduino boards and equivalent parts will be utilized, guaranteeing affordability and simplicity of integration. The BBS of the Car is designed to collect a variety of vehicle data, such as GPS coordinates, speed, acceleration, engine performance metrics, audio, and video recordings, and more. One important component of the system will be real-time monitoring of various data points, which will enable users to obtain information as needed. The system will have features for effectively storing and retrieving data. There will be consideration of a range of data storage options, including SD cards and external drives. The development of an interface that is easy to use will facilitate the retrieval and analysis of stored data.

The research will involve creating an interface that is easy to use so that consumers can communicate with the car's BBS. Alerting methods will be incorporated into the system to inform users of important events, like accidents or anomalous vehicle behaviour. Notifications from mobile applications may be a part of these alerts. The BBS for Cars will be made to be easily installed and integrated into a variety of vehicles, from private automobiles to business fleets.

Detailed testing and validation processes will be carried out to guarantee the correctness, dependability, and resilience of the system in a variety of driving scenarios. Simulated crashes, severe weather, and different driving styles are examples of testing scenarios. In-depth instructions and a user manual will be created to help users understand the features, installation procedure, and troubleshooting techniques of the system. The project scope may include future development priorities, such as more sensors, data analytics tools, remote monitoring, and networking choices. The system will comply with applicable automobile safety and compliance norms and laws, making it appropriate for deployment in a variety of locations. The project intends to produce a fully functional car BBS that utilizes Arduino technology and will be a useful tool for fleet managers, car owners, accident investigators, and those who are concerned about safety by adhering to this extensive scope.

2. Related Works

Automotive Monitoring using the Black-Box system was proposed in [1]. The structure and functionality of an advanced alerting and monitoring system for car features are covered in this study. It periodically monitors the car's data, such as temperature, humidity, speed, exact position, gas detection and accident detection. Through the website, users can view the data that is kept in a cloud database and black box. In the event of an accident, quick medical assistance can also be instantly accessed from the nearby hospital by detecting an alert message that includes the exact spot of the car. Based on testing

results, the suggested method outperforms RFID, SVM, CNN, and RNN.

The research in [2] developed of a GSM module black box model to track the location of car accidents and vehicle speed using an innovative approach. When this black box model is installed in any vehicle, it will monitor the accident by automatically sending a text message to the rescue squad and the police control room, along with direct links to a Google Map showing the position of the accident, depending on the closest latitude and longitude. The suggested model will additionally record an accident's video and send it over MMS in addition to this feature. Additionally, the black box model will continuously monitor speed. If the speed goes beyond the predetermined limit, it will warn the driver, and if it does so again, it will send fine information to both the driver and the police. By monitoring irresponsible driving at high speeds, the black box model will help accident victims receive assistance efficiently and contribute to the reduction of accident rates.

The main objective of research in [3] was to provide safety guidelines for cars together with a method that will alert drivers to drive cautiously on their own. Both the driver's behaviour and the performance of the vehicle are continuously monitored using sensors and IoT technology. Several sensors, including the Raspberry Pi microcomputer and IoT module, send data to the car's black box. GPS is used to monitor the location of the car in the case of an accident, and a message with the necessary details is delivered to the local police and hospital. The database provides accident data. As soon as the accident happens, a location-based alert message is sent to a nearby police station, hospital, and contact person. The accident that occurs is the message delivered over the network. The plan was tested in real-life scenarios using cars. There will be no false alert messages regarding the test findings. In this research project, a unique framework for auto-shading recognition and tag number identification is presented, with the findings being sent to the Waveshare SIM7600G-H 4G module after that. Additional features are also added to the standard auto black box, such as the ability to retrieve data requests and transfer stored data, as well as tag number and shading recognition of nearby cars. It also illustrates the suggested framework's reenactment and usage points.

In the research [6], a car BBS with false traffic data, fine recording techniques and safety features was developed utilizing a 433MHz long-range wireless modulation technology and vehicular sensor network under the Internet of Vehicles idea. Sensors for gas and flame are employed in security approaches. Moreover, a GPS was utilized to pinpoint the exact position of the car, as shown by longitude and latitude values, time, and date; this information guarantees that the car was present at the specified place when the traffic penalty is being recorded, protecting drivers from false recordings

of traffic fines. When comparing this research to the previously observed related works, LoRa technology is applied rather than Bluetooth or GSM technology, which reduces development costs and provides a wide coverage area for the system.

The study in describes the car's BBS utilizing an Internet of Things module and a Raspberry Pi microprocessor. A Raspberry Pi microcontroller, many sensors, alcohol sensor modules, audio, GPS camera and the electronic control unit were used to build this system. With the help of the ThingSpeak platform and the Waveshare SIM7600G-H 4G module, the data were stored in the cloud as well as on a secure digital card. The outcomes demonstrate the embedded system's ability to gather and process data on GPS, audio, alcohol content, speed, temperature, and other variables. A functional BBS model with a message alert for use in road cars was developed in [7] for the purpose of detecting and reporting vehicle accidents. This system was created for automobile accident reporting and employed an Arduino UNO, GPS tracking device, and GSM module. Emergency responders receive vital information from it as soon as possible. The system utilizes the GPS module to locate the accident site, while the GSM module is employed to transmit a message notification to the registered number.

The work in [8] mainly focuses on a classifier for fundamental driving behaviors, based on a similar methodology that may be used to recognize fundamental daily activities: a one-dimensional convolutional neural network (1D CNN) combined with electrooculographic (EOG) inputs. Notification system is utilized in [9] with the intention of giving details to driver. It makes use of an CNN and EoG signal platform designed for automobile reporting. Emergency responders receive vital information from it as soon as possible. There is typically a critical window of time between an accident and the victim receiving medical assistance that determines whether the victim survives or not. This system provides better safety than none. By establishing a postal relationship and building a Black box to communicate data over the IoT and locate the vehicle using GPS coordinates, they established an automobile-based BBS in [11]. In the event of an unintentional collision, every sensor provides input data, and the location coordinates are transmitted to the registered mobile phone number through GSM, therefore saving lives. The data will also be kept on an SD card for later retrieval.

In order to improve vehicle safety standards, this study proposes a prototype automobile backbox system that is modeled after an airline backbox. It does this by tracking driving habits and vehicle performance, which can be utilized for data analysis in the event of an accident. Microcontrollers, sensors, and other hardware components are combined with MQTT and Node-Red software in the suggested system. To increase trust in

the accident's severity, data must be shared between the MQTT server and the blackbox. The Internet of Things (IoT) makes it possible to prototype intelligent solutions at the lowest possible cost. The shortcomings of the current car safety systems are emphasized in this research, along with the missing parameters needed for a precise analysis [12].

The research stating that a black box was like a plane black box in that it is used for accident investigation. Their analysis presents a prototype of a discovery framework that can be installed in automobiles. The system records separate driving parameters by using five different sensors. An ARM 7 microprocessor controls these sensors. It may also have a global positioning system for obtaining vehicle ground coordinates in the unlikely event that the Black Box malfunctions and an SMS message is sent to a preset number. The suggested structure also plans to use an external camera and amplifier to record audio and video. On a memory card that may be retrieved, the data gathered from various sensors and modules is stored [13].

In order to improve vehicle safety standards, this study proposes a prototype automobile backbox system that is modeled after an airline backbox. It does this by tracking driving habits and vehicle performance, which can be utilized for data analysis in the event of an accident. Microcontrollers, sensors, and other hardware components are combined with MQTT and Node-Red software in the suggested system. To increase trust in the accident's severity, data must be shared between the MQTT server and the blackbox. The Internet of Things (IoT) makes it possible to prototype intelligent solutions at the lowest possible cost. The shortcomings of the current car safety systems are emphasized in this research, along with the missing parameters needed for a precise analysis. [14].

3. Proposed BBS Model

This project is designed with microcontroller Arduino Mega 2560, sensors, GPS receiver and SD module as shown in Figure 1. The vehicle's information and driving behaviours are recorded via the black box. It records many parameters, including location, speed, and engine temperature, mileage, and fuel level. It contains the capacity to gather statistically significant crash data in order to raise the standard of safety for automobiles and trucks, establish the effectiveness of traffic regulations, and enable emergency personnel to be alerted of accidents immediately. When an automobile crashes, the system uses a Wi-Fi module to send an accident alarm and the vehicle's current position to a pre-programmed mobile device. The purpose of the alcohol sensor is to stop drunk drivers from driving a vehicle. Once more, the potentiometer is adjusted to set the reference voltage.

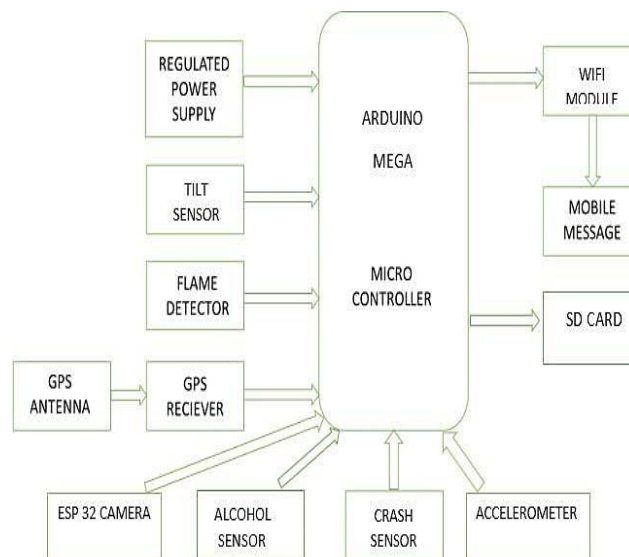


Figure 1. Block Diagram of the Proposed BBS Model

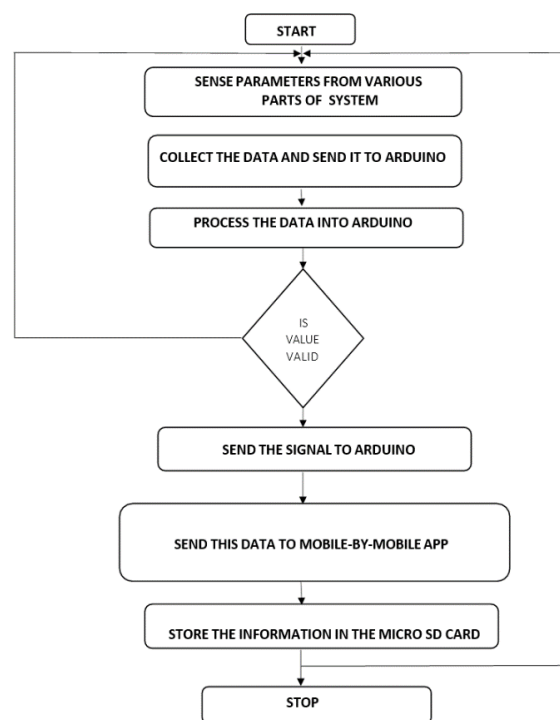


Figure 2. Flowchart of the Proposed Model

The motor stops, and the sensor goes high when alcohol is detected. The vibration sensor's task is to detect any vibrations that go above the threshold. The threshold can be adjusted using the on-board potentiometer. The output logic is low when there is no vibration and high when there is vibration, critical notifications are sent. Figure 2 represents the flowchart of the model.

Black box technology, like flight recorders on professional airplanes, can now have a significant impact on accident investigations [15]. A considerable majority of automobiles now on the road lack electronic systems that retain information in the case of an accident. Hence, it is crucial to ensure that robots remain

impartial and effectively monitor the events preceding, during, and following a crash, in addition to the primary information often gathered from sufferers, witnesses, and police reports. This project provides an overview of the setup and operations of the vehicle's 'Black box' system. The system will prioritize real-time traffic monitoring and retain all data for future analysis in case of an accident. This approach will aid accident investigators and insurance companies in determining the root cause of the failure. The device also offers other capabilities such as speed control, tilt adjustment, and alcohol detection. The objective of the project is to enhance the user's sense of security when on the car's roof and aid in determining the exact reason of the accident [16, 17].

The hardware module consists of crucial components, such as the Arduino Mega Board, Flame Sensor (HW 201), Accelerometer (HW 123), MEMS, WIFI Module ESP (8266), Crash Sensor (YL 99), Wi-Fi Module (ESP 32 Camera), GPS Module (GY-GPS6MV2), SD Card Module (HW 125), and MQ3 Alcohol Sensor. The software module utilizes the Arduino IDE for programming Arduino boards and Flutter for developing a user-friendly interface and applications for mobile devices. Together, these modules create a full Car's Black Box System (BBS) employing Arduino technology, intended at increasing road safety, accident investigation, and overall vehicle performance evaluation.

The novelty of this Car's Black Box System (BBS) stems from its use of Arduino technology, which offers a flexible and cost-efficient framework for extensive data collection and examination within the automotive domain. The system integrates a wide array of sensors and modules, including accelerometers, GPS, and cameras, to gather crucial data points such as speed, acceleration, and engine performance. The interface, designed to be easily understood and used by anybody, together with the ability to monitor in real-time, improves accessibility. This feature is particularly beneficial for fleet managers, car owners, and accident investigators, as it adds value to their work. The emphasis on data privacy and security, adherence to car safety regulations, and the possibility for future development goals make this BBS distinct, providing a sophisticated solution for improving road safety and vehicle performance.

3.1. Hardware Description

3.1.1 Arduino Mega Board

Arduino creates the Arduino Mega microcontroller board, an open-source electronics platform primarily built around the AVR Mega2560 microprocessor. The most recent Arduino Mega model has a USB interface, sixteen analogue input pins, and fifty-four I/O digital ports for connecting to external electronic circuits. It contains all the components needed to support the microcontroller, which is basically all it takes to link it to the PC via USB connection or initiating it using a battery or an air conditioner to start. By positioning a base plate, this board can be protected from the shocking electrical release. The AREF pin is connected to the Mega board's SDA and SCL pins. Moreover, the two pins are located near the RST pin. One pin makes up the IOREF, which provides safety against changing the voltage provided by the Arduino. Another pin was kept for the ultimate purpose of impending and is unrelated. Even though these sheets can comply with the most recent safeguards, which require these extra pins, they are compatible with all present safeguards [17].

3.1.2 Flame Sensor (HW 201)

The Flame Sensor HW-201 is primarily used in automotive BBSs to identify any fires or flames within the vehicle. The sensor keeps an eye out for flames or extremely hot conditions inside the car. The sensor can initiate rapid safety procedures, such as turning on fire suppression systems or warning the driver and passengers if it detects a fire. Although there are other ways to create a flame sensor, this project uses an infrared radiation-sensitive sensor module. The YG1006 NPN Photo Transistor is the base of this flame sensor. This specific kind of flame sensor has a 60-degree detection angle and can detect infrared light up to 100 cm away. With the YG1006 Photo Transistor, flame sensors can be implemented in two different ways: one with a digital output only and the other with both analog and digital outputs. The components needed for each of these implementations are the same; the only distinction is that one module—the one with the analog output—supplies the sensor output in analog form [18, 19].

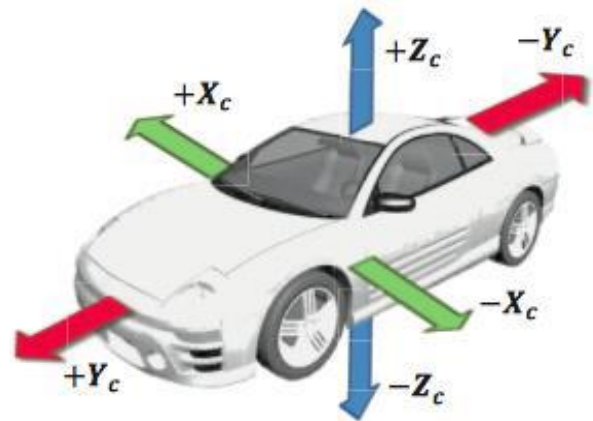


Figure 3. Accelerometer Axis in Vehicle

3.1.3 Accelerometer (HW 123)

An accelerometer is an electromechanical instrument that divides the change in velocity or speed by the passage of time to determine an object's acceleration [19]. In this application, the accelerometer is connected to a small device that is fixed to the car at the bottom position. The accelerometer detects an impact if the car tilts off its axis. The vehicle's 3-axis x, y, and z are continuously detected by this sensor, as seen in Figure 3 [7].

3.1.4 Micro Electromechanical Systems (MEMS)

MPU6050 module contains two internal components: a 3-axis accelerometer and a 3-axis gyroscope. The acceleration, velocity, orientation, displacement, and many other motion-related properties of an object or system can now be measured more easily as a result.

- A 6-axis motion tracking device is integrated inside the MPU6050 sensor module.

- It contains a temperature sensor, a digital motion processor, a 3-axis accelerometer, and a 3-axis gyroscope, all in a single integrated circuit.
- By using its Auxiliary I2C bus, it may receive inputs from various sensors, such as a pressure sensor or a 3-axis magnetometer.
- A full 9-axis Motion Fusion Output can be obtained by connecting an external 3-axis magnetometer.
- This module can communicate with a microcontroller through the I2C communication protocol. Several parameters can be found by using the I2C connection to read values from addresses of registers.
- The two complement forms of the accelerometer and gyroscope readings along the X, Y, and Z axes are available.
- Accelerometer values are given in g units; gyroscope readings are given in degrees per second (dps) [20].

3.1.5 WIFI Module ESP (8266)

The separate SOC ESP8266 Wi-Fi Module enables any microcontroller to establish a connection to your Wi-Fi network due to its built-in TCP/IP protocol stack. The ESP8266 can run a program on its own or designate another application processor to handle all Wi-Fi networking tasks. Almost the same amount of Wi-Fi functionality as a Wi-Fi shield can be obtained right out of the box by connecting any ESP8266 module to your Arduino device, as each module comes pre-configured with an AT command set firmware! The ESP8266 module has a large and developing community and is a relatively affordable board. Due to the GPIOs on this module, it may be easily connected to sensors and other app-specific devices, which minimizes setup time and loading when the module is in use. It also has enough computing power and storage capacity on board. The front-end module's high level of on-chip integration allows it to have the least amount of external circuitry, saving PCB space. As an all-in-one device that can function in any operating environment and does not require any extra RF components, the ESP8266 supports Bluetooth coexistence interfaces and APSD for VoIP applications.

3.1.6 Crash Sensor (YL 99)

The crash sensor detects the collision and converts it to usable signals within milliseconds. Following a collision, strong acceleration forces are pressed on the sensors. All bodies or things that are not securely fastened to the car, for instance, will continue to move at the impact speed when the car suddenly

comes to a halt. This acceleration is measured by the sensors, which then provide useful data to the control unit [21]. The features of Crash Collision Sensor include low output module, No high-output class, No lightning bumps. The module output is low, there is no collision, and the output is high when it is installed in front of a collision. M3 mounting holes on the module are retained for simple installation in a compact car. When a light switch is hit, lights come on, lights go out, and there is a collision.

3.1.7 Wi-Fi Module (ESP 32 Camera)

The OV2640 Camera Module 2MP For Face Recognition, which is housed in the ESP32 CAM Wi-Fi Module Bluetooth, has an extremely attractive small-size camera module that can function independently as a limited system and a deep sleep current of up to 6mA. Its footprint is only 40 x 27 mm. It is also extensively utilized in several IoT applications. It works well with smart home appliances, wireless control in industries, remote monitor, and various applications of IoT. This component comes with a DIP package to enable quick product production, and it may be installed straight into the backplane. It offers clients a high-reliability connection mode, making it suitable for use in a range of IoT hardware terminals. ESP combines a 7-stage pipeline architecture, two powerful 32-bit LX6 CPUs, Wi-Fi, conventional Bluetooth, and BLE Beacon. This device's main frequency adjustment range is 80MHz to 240MHz. It also features temperature, Hall, and on-chip sensors.

3.1.8 GPS Module (GY-GPS6MV2)

The Global Positioning System, or GPS, is a tool used for location tracking that is frequently used in real-time applications. The GPS module continuously receives data from the satellite and sends the relevant information. The GPS antenna continuously gathers the car's location, and the module continuously transmits the car's latitude and longitude to the Arduino. This module's power-saving mode is one of its most interesting characteristics. This makes it possible to reduce the system's power usage. The module's current usage decreases to just 11mA when power-saving mode is activated [22].

3.1.9 SD Card Module (HW 125)

For projects that need data logging, the SD card module is extremely helpful. The Arduino may write and store data in a file on an SD card by using the SD library. Various models from various suppliers are available, but they all use the SPI communication standard and function similarly. A 74LVC125A logic level shifter chip is also included in the module, which enables a simple and safe connection with your preferred 3.3V or 5V microcontroller without damaging the SD card.

3.1.10 MQ3 Alcohol Sensor

One module for the detection of alcohol is the MQ3 alcohol gas sensor. Alcohol gases can be detected by the alcohol sensor, a fully analog instrument, with concentrations ranging from 0.05 to 10 mg/L. The MQ3 alcohol gas sensor has a high sensitivity to fuel, smoke, and alcohol and operates extremely quickly. As the concentration of gases increases, so does the conductivity. The best low-power module for detecting alcohol cups is Mq3, which also has an easy-to-use interface for microcontrollers and Arduino. In the proposed system, this sensor is generally referred to as a breath analyzer.

3.2. Software Description

3.2.1. Arduino IDE

The Arduino Software or Integrated Development Environment (IDE), is a multi-menu text editing tool, message box, text console, toolbar with buttons for commonly performed functions, and text editor for creating code. It links to actual hardware (Arduino) so that programs may be uploaded and used. Cross-platform software developed in functions from C and C++ is the Arduino Integrated Development Environment. Programs are written with it and uploaded to boards that are compatible with Arduino.

Sketches are programs created with Arduino Software (IDE). The sketches are created in a text editor and stored as files with a certain extension number. The editor offers tools for searching through and replacing text as well as cutting and pasting. In addition to displaying faults, the message box provides feedback during exporting and saving. Complete error messages and other information are displayed in the console together with text output from the Arduino Software (IDE).

Library inclusion and variable declaration: The code includes several libraries required for different functionalities, such as Wire (for I2C communication), SD (for SD card operations), SPI (for SPI communication), TinyGPS++ (for GPS data parsing), and SoftwareSerial (for software-based serial communication). The code also declares variables for GPSBaud (the GPS module's baud rate), lat and lon (strings to store latitude and longitude), dateG and timeG (strings to store date and time), AccX, AccY, and AccZ (floats to store accelerometer readings), AccErrorX and AccErrorY (floats to store accelerometer error values), and c (an integer used in the calculate_IMU_error function).

Setup function: The setup function is called once at the start of the program. The pin modes are set for the input pins connected to various sensors (T1P, T2P, CP, FP, AP) using the pinMode function. Serial communication is initiated for different interfaces: Serial

communication is started at a baud rate of 19200 using Serial. Begin (19200). Software Serial is initialized with pins 8 and 9 at a baud rate of 19200 using my Serial. Begin (19200). Serial1 is initialized at the GPS Baud rate (9600) using Serial1.begin (GPSBaud). The Wire library is used to initiate communication with the MPU6050 sensor (accelerometer) using Wire. Begin Transmission and Wire. Write functions. The MPU6050 is reset by writing a value of 0x00 to the register 0x6B using Wire. Write. SD card initialization is performed using SD. begin (CSpin), where CSpin is the pin connected to the SD card's chip select pin (CS). If SD card initialization is successful, a file named "data.csv" is opened in write mode (FILE_WRITE) and then closed using SD.open and sensor Data. Close functions [23].

Loop function: The loop function is executed repeatedly. Get Acc Data function is called to read accelerometer data from the MPU6050 sensor and store the readings in AccX, AccY, and Acc Z variables. Get GPS Data function is called to read GPS data from the GPS module using the Tiny GPS++ library? The latitude, longitude, date, and time values are stored in the corresponding variables (lat, lon, date G, time G). The process and Save function is called to check the status of sensor input pins and save the corresponding data to the SD card. The display Info function (commented out) can be used for debugging purposes to print the state of sensor pins and accelerometer readings. A delay (currently commented out) can be added if needed [24].

Process and Save function: This function checks the status of different sensor pins (T1P, T2P, CP, FP, AP) using digital Read functions. If a sensor pin is activated (reads LOW), it enters the corresponding if statement. Inside each if statement, a data string is created by concatenating the sensor type ("Crash," "Flame," "Alcohol," "Left Tilt," "Right Tilt"), date G, time G, lat, lon, AccX, AccY, and Acc Z values using String concatenation. The data String is then sent via Software Serial (my Serial. write) after being enclosed within "<" and ">" characters. The save Data function is called to append the data string to the "data".

3.2.2. Flutter

Google developed the open-source Flutter UI (User Interface) software development kit (SDK). Using the same codebase, developers may construct natively compiled desktop, web, and mobile applications. Flutter saves time and effort by enabling you to create code once and have it deployed on various platforms. Google created the open-source Flutter UI software development kit. It is used to create cross-platform apps for Windows, Linux, macOS, Android, Google Fuchsia, and the web with a single codebase. Flutter was released in May 2017 after being first described in 2015.

Library Inclusion: The required libraries for Wi-Fi communication, web server functionality, serial

communication, and UDP communication are included using the #include directive.

Global Variable Declaration: Various global variables are declared, including network addresses, buffer sizes, and objects for communication and networking [25].

Function Definitions:

- **setIP ():** This function handles an HTTP request sent to the "/setClientIP" endpoint of the web server. It extracts the client's IP address and port from the request and stores them in variables. It also sends a response with the status code 200 ("success").
- **SendHTTPMsg ():** This function sends an HTTP GET request to a client-server. It constructs the URL using the client's IP address, port, and data stored in the tempChars array. It sends the request, receives the response, and prints it to the serial monitor.

Setup Function: Serial communication is initialized for debugging purposes. The ESP8266 is set to Access Point (AP) mode, and a soft AP network is created with the SSID "CLASS" and an empty password. The web server is started on port 80, and the "/setClientIP" endpoint is registered to the setIP() function. The UDP receiver is initialized to listen on the local IP address and multicast address 224.10.10.10 on port 4200.

Loop Function: The web server handles incoming client requests. The recv With Start End Markers () function is called to read data from the MySerial serial connection. If new data is received, it is copied to the temp Chars array, the show Parsed Data () function is called to print the data, and the send HTTP Msg () function is called to send an HTTP GET request to the client-server.

- **Recv with Start End Makers () Function:** This function reads data from the MySerial serial connection until it receives a start marker ("<") and an end marker (">"). The received characters are stored in the receivedChars array.
- **Show Parsed Data () function:** This function prints the contents of the tempChars array.

3.3. Implementation of the Model

The integration of the Car's Black Box System (BBS) with Arduino requires combining several hardware components, including as accelerometers, GPS, cameras, and microphones, with the versatile and cost-effective Arduino platform. The system is specifically built to gather a wide range of vehicle data, including GPS coordinates, speed, acceleration, engine

performance metrics, and other related information. The Arduino Mega Board functions as the main processing unit, while the software module utilizes the Arduino IDE for programming and Flutter for developing a user-friendly interface and mobile apps. The resultant BBS offers real-time surveillance, data retention, and retrieval functionalities, rendering it a resilient instrument for fleet managers, vehicle owners, and accident investigators, hence enhancing road safety and vehicle supervision.

Any application's primary priority is security. Since our prototype contains sensitive data, data transmission security is crucial. This prevents data leaks from attacks, reducing data misuse. This system was designed and developed for roughly 6000 rupees, making it cost-effective. In terms of applications, this technology may be utilized not only for assisting in traffic accidents, but also for the benefit of insurance organizations. The presence of such data can assist an insurance firm in more accurately assessing the risks associated with insuring a customer. Current insurance processes entail assessing the customer based on factors such as their residence, age, car type, make, and model. The data from the Black box may be utilized to assess driving behaviour, including determining if the driver is mistreating the vehicle or driving in a cautious manner. This information can facilitate the development of an improved pricing plan that is mutually advantageous for both insurance firms and drivers. It can provide real-time risk assessments. A single device may be utilized to oversee a fleet of automobiles, such as those in a rental car firm. The system may offer data on the precise location of the car, its speed, and any vibrations it may experience. This information can be used to assess if the vehicle is being used in a manner that is considered abusive or excessive. This information may also lead to cost reductions and lower prices for both the firms and their customers.

4. Results and Analysis

The black box system developed for car, utilizing an Arduino Mega 2560 microcontroller, sensors, GPS receiver, and SD module, plays a pivotal role in enhancing safety and data-driven decision-making. It records a comprehensive set of parameters, including location, speed, engine temperature, mileage, and fuel level, enabling the collection of statistically significant crash data to improve safety standards, assess traffic regulations' effectiveness, and facilitate immediate accident alerts to emergency personnel via a Wi-Fi module. Furthermore, it includes features such as an alcohol sensor to prevent drunk driving and a vibration sensor for detecting critical events, making it a versatile tool for monitoring, and enhancing vehicle safety.

The appropriate sensors are being used to measure the sensing portion of all car's black boxes. Using both SD cards and the Black Box Mobile Application software, the detected data is stored in the

database. Additionally, a comparative examination of the parameters using sensor data is carried out. The hardware and Experimental results of the Car's BBS Using an Arduino are shown in the following Figures. The top view model of the car and front view of the BBS is shown in figures 4 and 5. Figure 4 represents a top view of the BBS installed within the vehicle. It showcases the arrangement and positioning of the components, such as the Arduino Mega 2560, sensors, GPS receiver, SD module, and other associated hardware. The top view allows us to observe the spatial layout of these components and understand their integration within the vehicle's architecture.

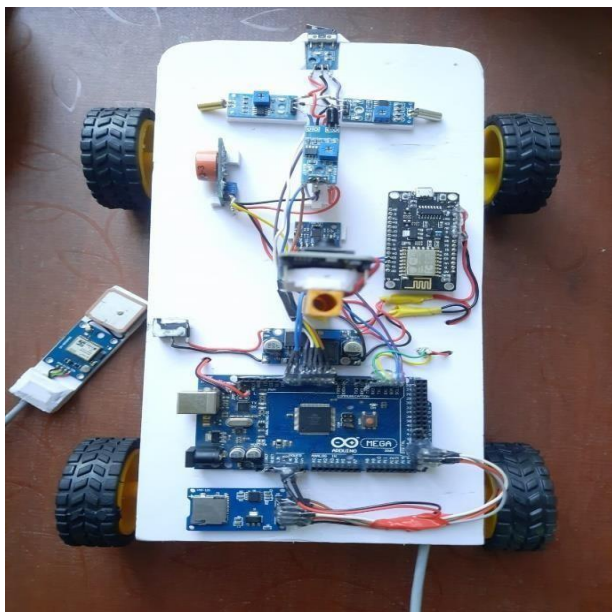


Figure 4. Top View of the Model

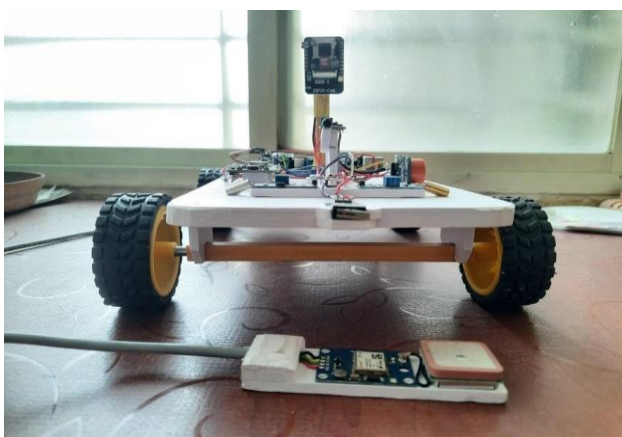


Figure 5. Front View of the Model

The front view image provides a detailed look at the front-facing aspect of the BBS. This perspective highlights how the system is placed within the vehicle's interior. It may reveal how the system is secured, connected, or integrated with the vehicle's existing structure, and it can be useful for assessing accessibility and potential impact on the vehicle's design. All the sensors in the car keep functioning when the battery is turned on.

Attendance	Reconnect
21. CRASH DATE: 25/4/2023 TIME: 21:6:4 LAT: 13.320821 LON: 77.133125 ACC X: -0.02 Y: -0.06 Z: 1.20	
20. RIGHT CRASH DATE: 25/4/2023 TIME: 21:6:1 LAT: 13.320874 LON: 77.133102 ACC X: 0.01 Y: 0.36 Z: 1.11	
19. LEFT CRASH DATE: 25/4/2023 TIME: 21:5:49 LAT: 13.320768 LON: 77.133133 ACC X: 0.01 Y: -0.69 Z: 0.89	
18. LEFT CRASH DATE: 25/4/2023 TIME: 21:5:48 LAT: 13.320739 LON: 77.133133 ACC X: 0.02 Y: -0.60 Z: 0.98	

Figure 6. Data Shown in Mobile App

Initially, the SD card stores all the sensor readings as well as the place where the date and time are recorded. This system gives information on an accident, including the date, time, latitude, and longitude, as shown in Figure. 6. It represents the screenshot of the mobile app interface used to view and analyze the data recorded by the BBS. It includes visual representations or textual information, showcasing real-time or historical vehicle data. This data includes parameters like location, time, and crash information. The interpretation here is about the user-friendly interface design and the capability to access critical vehicle information conveniently. The data will be stored in an easily understandable CSV format. Anytime a verification of information is required, the user can conduct one. Data stored on an SD card and data stored in the cloud are shown, respectively.

Figure 7 presents a representation of the data stored in the BBS in CSV (Comma-Separated Values) format on the SD card. This tabular data format is commonly used for easy storage and sharing. The interpretation of this image involves understanding the structure and content of the CSV file, including the arrangement of columns and rows, which might contain time-stamped data points for the recorded parameters. It is crucial for data analysis and report generation. Compared to the existing black box models, the proposed car's BBS, which utilizes Arduino technology, outperforms current versions due to many significant benefits.

DATA.CSV

Crash	25/4/2023	20:21:57	13.321079	77.133095	-0.01	-0.05	1.20
Right Tilt	25/4/2023	20:22:9	13.321065	77.133102	0.08	0.47	1.07
Left Tilt	25/4/2023	20:22:12	13.321055	77.133095	0.05	-0.80	0.81
Crash	25/4/2023	20:22:16	13.321048	77.133102	0.02	-0.04	1.20
Crash	25/4/2023	20:22:17	13.321051	77.133095	-0.02	-0.05	1.20
Flame	25/4/2023	20:22:22	13.321068	77.133087	0.00	-0.06	1.20
Alchoal	25/4/2023	20:22:28	13.321074	77.133095	0.01	-0.35	1.16

Figure 7. Data Stored in SD in CSV Format

Table 1. Comparison of Proposed Model with Existing Models

Model	Features	Technology Used	Main Objective	Limitations Overcome in Proposed Model
Proposed Model	Comprehensive BBS for data acquisition, analysis, and real-time monitoring. Integration of multiple sensors and data modules. User-friendly interface. Easy installation and integration into various vehicles. Compliance with safety norms and regulations.	Arduino technology	To develop a sophisticated BBS for car safety and monitoring.	
[2]	Accident tracking with automatic text message alerts. Google Maps link for accident location. Video recording of accidents. Speed monitoring and warning system.	Arduino technology	To improve accident response and reduce accident rates.	Limited sensor integration. Lack of user-friendly interface. Data privacy and security concerns.
[3]	Continuous monitoring of driver behavior and vehicle performance. Location-based alert system. Additional features for auto shading recognition and tag number identification.	IoT technology	To provide safety guidelines and alert drivers effectively.	Limited real-time monitoring capabilities. Dependency on specific technologies.
[6]	False traffic data detection. Fine recording techniques. Safety features including gas and flame sensors. GPS-based accident location tracking.	Internet of Vehicle	To enhance car safety and protect drivers from false fines.	Dependency on specific technologies. Limited real-time monitoring capabilities.
[7]	Use of Raspberry Pi microprocessor and various sensors. Data storage in the cloud and on SD card. Monitoring of GPS, audio, alcohol content, speed, temperature, etc.	Raspberry Pi, ThingSpeak	To gather and process vehicle data effectively.	Lack of user-friendly interface. Data privacy and security concerns.
[8]	Accident reporting with GPS tracking and GSM notification. Utilizes Arduino UNO, GPS tracking device, RPi and GSM module.	Arduino UNO, GPS, GSM	To facilitate prompt notification and assistance in accidents.	Lack of sensor integration. Limited real-time monitoring capabilities. Data

				privacy and security concerns.
[9]	Collision alert system with cloud computing. Low-power microcontroller for automation. Dependable and power-efficient IoT-based black boxes. Utilizes CNN model.	IoT, cloud computing	To enhance vehicle safety with real-time collision alerts.	Limited sensor integration. Dependency on specific technologies.
[10]	Accident identification and notification using IoT platform. Sensors for ultrasonic, alcohol, speed, and AXDL. Prompt emergency response system.	IoT platform, Arduino UNO	To improve accident response time and overall safety.	Lack of user-friendly interface. Limited real-time monitoring capabilities.
[12]	Vehicle location tracking with GPS coordinates. GSM notification in case of collision. Data storage on SD card for retrieval.	GSM, GPS	To enable efficient accident response and location tracking.	Limited sensor integration. Lack of user-friendly interface. Data privacy and security concerns.
[13]	Evidence collection system for post-accident analysis. Remote data transmission to database server. Investigation of misconduct, salvage, and insurance claims. Used MQTT and Node - Red.	Cloud & IoT	To enhance accident investigation and claims processing.	Dependency on specific technologies. Lack of user-friendly interface. Limited real-time monitoring capabilities.
[14]	Black box system for accident investigation. Sensor-based data recording. SMS alerts and external camera for additional data capture.	ARM 7 microprocessor	To provide comprehensive accident data for analysis.	Dependency on specific technologies. Lack of user-friendly interface. Limited real-time monitoring capabilities.
[15]	Wireless black box with GPS tracking and MEMS accelerometer. Alert messages for collision prevention. Tilt operation detection. It also used API cloud service.	Arduino, Raspberry Pi	To prevent collisions and provide real-time accident alerts.	Lack of user-friendly interface. Limited real-time monitoring capabilities.

By employing Arduino technology, the cost-effectiveness, adaptability, and seamless integration of the system are guaranteed, hence expanding its accessibility to a broader spectrum of consumers. The system's incorporation of diverse sensors, such as accelerometers, GPS, and cameras, yields a more extensive array of data points for analysis, facilitating a more profound comprehension of vehicle performance and events. The interface, designed to be easily understood and used by users, together with the ability to monitor in real-time, improves the ease of use and enables prompt and well-informed decision-making. Furthermore, the emphasis on data privacy and security guarantees the conscientious handling of personal information. The BBS's adherence to automotive safety standards and its potential for future development

emphasize its effectiveness and potential for extensive implementation, making it a superior solution for improving road safety, investigating accidents, and evaluating overall vehicle performance.

5. Conclusion

A working model of a Car's BBS using an Arduino for vehicles was developed for accident detection and reporting. This model tracks and reports on car accidents using an Arduino, a GPS module, a Wi-Fi module, and several sensors. In the case of an accidental crash, all the sensors broadcast data to Arduino, and the smartphone app uses GPS to give the victim's location to save their life. Emergency responders receive essential data from it as soon as possible. The

critical period following an injury before the victim receives medical assistance can frequently mean the difference between life and death. Compared to no safety, this approach offers greater safety. In addition, an SD card will be used to save the data for later retrieval. By reducing the probability of accidents, the suggested method will lead to the development of safer modes of transportation. If an accident does occur, alerts will be sent out to maximize victim assistance and rescue. Overall, the system is economical and successful in every way. Both post-accident investigations and search and rescue operations will benefit from the suggested system. Webcams installed inside cars may be used in the future to help drivers. It is possible to prevent accidental injuries by using long-range infrared sensors. With many more functions than microcontrollers, CPLD Chips can be used in place of microcontrollers. Much more information about the car can be obtained by using additional sensors, such as seatbelt sensors and pressure sensors, to determine the state of the brakes.

References

- [1] P. Josephinshermila, S. Sharon Priya, K. Malarvizhi, R. Hegde, S. Gokul Pran, B. Veerasamy, Accident detection using Automotive Smart Black-Box based Monitoring system. *Measurement: Sensors*, 27, (2023) 100721. <https://doi.org/10.1016/j.measen.2023.100721>
- [2] M. Karthik, L. Sreevidya, K. Vinodha, M. Thangaraj, G. Hemalatha, T. Viswak Sena, Automatic messaging system by detecting the road accidents for vehicle applications. *Materials Today: Proceedings*, 80(3), (2023) 3124-3128. <https://doi.org/10.1016/j.matpr.2021.07.177>
- [3] A. Garcia-Barrientos, D. Torres-Uresti, F.R. Castillo-Soria, U. Pineda-Rico, J.A. Hoyo-Montaño, O. Perez-Cortes, P. Ordaz-Oliver, Design and Implementation of a Car's Black Box System Using a Raspberry Pi and a 4G Module. *Applied Sciences*, 12(11), (2022) 5730. <https://doi.org/10.3390/app12115730>
- [4] C. Kang, S.W. Heo, (2017) intelligent safety information gathering system using a smart blackbox. 2017 IEEE International Conference on Consumer Electronics (ICCE), IEEE, USA. <https://doi.org/10.1109/ICCE.2017.7889294>
- [5] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5), (2018) 1-42. <https://doi.org/10.1145/3236009>
- [6] G.A. Aramice, A.H. Miry, T.M. Salman, Vehicles Black Box Implementations for Internet of Vehicle Based Long Ranges Technology. *Journal of Engineering and Sustainable Development*, 27(2), (2023) 245-255. <https://doi.org/10.31272/jeasd.27.2.8>
- [7] M.J. Prasad, S. Arundathi, N. Anil, B.S. Kariyappa, (2014) Automobile black box system for accident analysis. *International Conference on Advances in Electronics Computers and Communications*, IEEE, India. <https://doi.org/10.1109/ICAEECC.2014.7002430>
- [8] S. Uma, R. Eswari, Accident prevention and safety assistance using IOT and machine learning. *Journal of Reliable Intelligent Environments*, 8(2), (2022) 79-103. <https://doi.org/10.1007/s40860-021-00136-3>
- [9] S. Sethuraman, S. Santhanalakshmi, (2020) Implementing Vehicle Black Box System by IoT based approach. 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), IEEE, India. <https://doi.org/10.1109/ICOEI48184.2020.9142906>
- [10] G. Falco, J. Siegel, A Distributed "Black Box" Audit Trail Design Specification for Connected and Automated Vehicle Data and Software Assurance. *SAE International Journal of Transportation Cybersecurity and Privacy*, 3(2), (2020) 97-111. <https://doi.org/10.4271/11-03-02-0006>
- [11] M. Vanitha, K. Arunkumar, A. Hemamalini, A. Yaswanth, A Smart IoT Based Black-Box System for Automobiles, *Journal of Physics: Conference Series*, IOP Publishing, 2484, (2023) 012052. <https://doi.org/10.1088/1742-6596/2484/1/012052>
- [12] A.A. Alsahlawi, M.A. Mangoud, (2022) IoT based vehicle blackbox for enhanced safety standards. 6th Smart Cities Symposium (SCS 2022), Hybrid Conference, Bahrain. <https://doi.org/10.1049/icp.2023.0664>
- [13] M. Karrouchi, I. Nasri, H. Snoussi, I. Atmane, A. Messaoudi, K. Kassmi, (2021) Black box system for car/driver monitoring to decrease the reasons of road crashes. 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT), IEEE, Saudi Arabia. <https://doi.org/10.1109/ISAECT53699.2021.9668545>
- [14] S. Jawad, H. Munsif, A. Azam, A.H. Ilahi, S. Zafar, (2021) Internet of things-based vehicle tracking and monitoring system. 15th International Conference on Open Source Systems and Technologies (ICOSST), IEEE, Pakistan. <https://doi.org/10.1109/ICOSST53930.2021.9683883>
- [15] S. Nanda, H. Joshi, S. Khairnar, (2018) An IoT Based Smart Systems for Accidents Preventions and Detections, *International Conferences on Computing Communications Controls and Automations*, IEEE, India. <https://doi.org/10.1109/ICCUBE.2018.8697663>
- [16] G. Franzè, W. Lucia, A. Venturino, A Distributed

- Model Predictive Control Strategy for Constrained Multi-Vehicle Systems Moving in Unknown Environments. *IEEE Transactions on Intelligent Vehicles*, 6(2), (2021) 343-352. <https://doi.org/10.1109/TIV.2020.3029746>
- [17] V. Fors, B. Olofsson, L. Nielsen, Autonomous Wary Collision Avoidance. In *IEEE Transactions on Intelligent Vehicles*, 6(2), (2021) 353-365. <https://doi.org/10.1109/TIV.2020.3029853>
- [18] L.S. Prakash, M.Z. Bellary, M.A. Ali Baig, T.S. Kumar, S. Merugu, An economical Black-box system for vehicles, *AIP Conference Proceedings*, 2477(1), (2023) 030080. <https://doi.org/10.1063/5.0129950>
- [19] A. Levering, M. Tomko, D. Tuia, K. Khoshelham, Detecting Unsigned Physical Road Incidents From Driver-View Images. in *IEEE Transactions on Intelligent Vehicles*, 6(1), (2021) 24-33. <https://doi.org/10.1109/TIV.2020.2991963>
- [20] L. Jiang, C. Yu, (2010) Design and Implementation of Car Black Box Based on Embedded System. *International Conference on Electrical and Control Engineering*, IEEE, China. <https://doi.org/10.1109/iCECE.2010.860>
- [21] M.M. Rahman, A.Z.M.T. Kabir, S.Z. Khan, N. Akhtar, A. Al Mamun, S.M.M. Hossain, (2021) Smart Vehicle Management System for Accident Reduction by Using Sensors and An IoT Based Black Box. *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, IEEE, Indonesia. <https://doi.org/10.23919/EECSI53397.2021.9624240>
- [22] Z.F. Li, J.T. Li, X.F. Li, Y.J. Yang, J. Xiao, B.W. Xu, Intelligent tracking obstacle avoidance wheel robot based on arduino. *Procedia Computer Science*, 166, (2020) 274-278. <https://doi.org/10.1016/j.procs.2020.02.100>
- [23] S. Saha, (2022) Crash Recovery and Accident Prediction Using a IoT Based Blackbox System. *IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, Vijaypur, India. <https://doi.org/10.1109/NKCon56289.2022.10127044>
- [24] A. Ponmalar, B. Chandra, S. Aarthi, G. Bhavana, A.A. Jose, S. Gomathi, (2022) IoT Based Automotive Drive Recorder as Black Box. *International Conference on Computer, Power, and Communications (ICCP)*, Chennai, India. <https://doi.org/10.1109/ICCP55978.2022.10072081>
- [25] A. Annapurna, S. Monika, K. Aruna Manjusha, Smart wireless black box with facial recognition and accidental monitoring of vehicles using IoT. *AIP Conference Proceedings*, 2492, (2023) 030088. <https://doi.org/10.1063/5.0119182>

Acknowledgement

The authors would like to thank their parent organisations for the continued support.

Authors Contribution Statement

Rajendran Thanikachalam: Investigation, Writing-Original Draft preparation, Reviewing; Rajendran Thavasimuthu: Investigation, Related Work, Flow diagram, Writing-Original Draft preparation, Writing - Reviewing and Editing; Godwin John J: Conceptualization, Literature Analysis, Methodology, Supervision, Data Validation-Reviewing and Editing. J Maria Arockia Dass: Conceptualization, Algorithm, Review, Editing. T. Nithya: Data Analysis, Concept Review, Editing. Anitha Thavasimuthu: Algorithm, Literature Analysis, Concept Review, Editing. All the authors read and approved the final version of the manuscript.

Funding

This research was conducted without the aid of any financial grants

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes

About the License

© The Author(s) 2024. The text of this article is open access and licensed under a Creative Commons Attribution 4.0 International License.