ISSN 2409-2665
Journal of Logistics, Informatics and Service Science
Vol. 9 (2022) No.3, pp.258-270
DOI:10.33168/LISS.2022.0318

2-Phase Crash Detection and Notification System

Siti Fatimah Abdul Razak[+], Faris Aiman Suhaimi, Sumendra Yogarayan and Mohd Fikri Azli Abdullah

Faculty of Information Science and Technology, Multimedia University, 75450 Malaysia

[+]*fatimah.razak@mmu.edu.my (corresponding author)*

Abstract. The automotive industry has grown at a fast phase and continues to grow day by day. One of the factors behind these improvements to the automotive business is the Internet of Things (IoT). This technology has exceptional possibilities and promise. The main factor encouraging is the convergence of several innovations and interaction methods. The 2-phase crash detection system can deliver messages in case of a crash or emergency to the first responders and user close contacts. The goal of this system is to create a device capable of improving the driver and passenger well-being. The device is made up of two parts, the hardware and the software which interacts for detection and notification purposes. The hardware is built on Arduino and four different sensors and modules, i.e., a sound detection sensor, a vibration sensor, a GPS module, and a gyroscope module. The data received from these modules and sensors are monitored at every second, and an alert will be sent out if the signal passed a certain threshold. The software part is made up of an android application which is used to store user information and a means to respond to the alert given out by the Arduino. The outcome of the project shows that the system is very straight forward to use and once the Arduino portion is built, pairing with the application will be seamless. The application can be left in the background while the user is driving and would not cause any interference if a navigation app is used at the same time. The project would significantly be part of active monitoring tool that would contribute towards intelligent transportation system.

Keywords: accidents, crash detection, internet-of-things, sensors.

1. Introduction

Expensive in-vehicle systems have the advantages of direct access to the vehicle which results in high accuracy of crash detection. However, the market of this type of systems are relatively limited compared to the majority of vehicle owners. The cost of these systems are not affordable to common people. Hence, it has become more important to propose an alternative solution with similar level of accuracy which can detect crashes and improve the response time for drivers to get help when a crash occurs. The response of an accident is vital in saving one's life, as every second goes by could mean life or death for the victims involved.

The use of crash detection systems is still lacking around the world, especially in Malaysia where when an accident occurs, victims still very much rely on the other bystanders to report the accident and call the emergency services. This type of response is very inefficient as it can take some time before someone decides to step up and call for rescue. Eventually, late responses could end up costing the victims their lives.

According to Rustagi et. al (2017), Haddon matrix has been utilised to understand interplay of human, vehicle and environmental factors such as pre-cash, crash and post-crash phases of accident scenarios. Hence, Crash Detection System is a system that is implemented in two phases as indicated in the title which includes detection (crash) and notification (post-crash). Ali and Alwan (2017) has also identified a similar approach, however, the development is primarily focused on smartphone based. Our proposed work was focused on addressing both hardware and software integration for crash detection. The system detects a crash involving the vehicle and relays the info to the e-call centers. Automated crash detection systems may help in reducing fatalities emanating from vehicular accidents by decreasing the emergency response time. Real-time crash detection and emergency alert can save lives. In this work, we propose a two-phase crash detection system which augments data from four different sensors and modules i.e., sound detection sensor, a vibration sensor, a GPS module, and a gyroscope module to reliably detect a vehicular crash. A delay timer is imposed to acquire confirmation of the crash from the driver and minimise any false alarm.

The following sections describe related works, methodology, system design and testing. Finally, the conclusion section concludes the work.

2. Related Works

There have been different attempts in proposing crash detection systems and they can be classified as pre-crash, crash and post-crash phases as described by the Haddon's matrix. The Haddon's matrix and counter measures have been utilized as effective injury diagnostic tools since 1970 (Rustagi et al., 2019).

Table 1: Crash phases according to Haddon Matrix.

Phases	Pre-crash	Crash	Post-Crash
Human	Crash prevention; Information; Attitudes; Impairment; Police enforcement	Injury prevention during crash; Use of restraints; Other safety devices; Crash protective design	Life sustaining; First aid skill; Access to medics
Vehicles and equipment	Road worthiness; Lighting; Braking; Handling; Speed management	Occupants restraints; Other safety devices; Crash protective design	Ease of access; Fire risk
Environment	Road design and layout; Speed limits; Pedestrian facilities	Crash protective roadside objects	Rescue facilities; Congestion

2.1. Pre-crash

Pre-crash detection systems are initialised whenever the vehicle is turned on. The system constantly evaluates a driver's position as well as any objects on the road, in order to prevent or minimize damage that may be caused by a crash. Data is augmented from sensors in the background during the driving experience. If there is a possibility of a crash ahead, the driver will be warned through a dashboard display and audible warning. Drivers will be able to take precautionary actions and avoid the crash. The most common sensors that will always be running are GPS sensors and accelerometer as these two can provide real-time location tracking as well as recording the velocity of the vehicle it is travelling. For example, Nissan introduced a Forward Collision Prevention System in its 2014 Infinity Q50. The system uses radar sensor to detect vehicle two cars ahead in order to help prevent pileup collisions. Additionally, Toyota implements millimeter wave radar to detect an obstruction (such as a leading vehicle) and determines a high possibility of collision. The driver will be notified with a warning buzzer.

2.2. Post-accident

Post-crash systems provide information about the crash location to allow first responders to locate the crash site and victim quickly. A more sophisticated system will automatically detect the crash and alerts emergency services and authorities. The system is also able to assess the type and severity of the crash, allowing medical assistance to be provided as soon as possible to the victims. Crash severity estimates maybe based on velocity during the crash, the principal direction of force, or whether the vehicle was on a rollover. On the other hand, estimates of probability of serious injury may be based on the vehicle data, presence of fire, air bag deployment and safety belt use. In one research, the system has been estimated to have the potential to reduce fatalities by up to 10.8% and could be especially useful in regional and remote areas. The emphasis is on life sustaining measures.

2.3. e-call false alarm

Post-crash systems are supposed to trigger the e-call module only in the event of serious crash, calling the emergency department first responders near the crash site. For instance, if an accident results in an attempt to deploy an airbag, excluding knee airbags and rear inflatable seatbelts, or to shut off the fuel pump, Ford e-Call system initiates a call to the emergency services. This call cannot be canceled. The system indicator status of Ford's e-call system is in Table 2.

Table 2: Ford's e-Call system indicator status

System Indicator Status	Description
Dimmed red	Normal operation
Rapidly flashing	Initiating an emergency call.
Moderately flashing	Transmitting vehicle data to the emergency services.
Slowly flashing	Connected to the emergency services and established communication.
Bright red	System malfunctioned.

Nevertheless, the e-call alert can be triggered under false assumptions i.e., when the vehicle's sensors made an erroneous interpretation that a crash has occurred and require medical attention. For instance, when the crash is just fender-bender type or while the car is being repaired or dismantled that may cause the first responders to be deployed to the location, wasting precious resources and time.

Table 3: Related work on post-crash detection system.

Source	Purpose	Limitations
Ali & Eid (2015)	Automated accident detection using accelerometer and gyroscope	A big device and can only send to one mobile number
Faiz, Imteaj & Chowdhury (2015)	Smartphone based accident detection and alarm system	Heavily rely on mobile internet connection
Nasr, Kfoury & Khoury (2016)	IoT based accident detection system using GPS and shock sensor	Require three alarm notifications and rely on IoT cellular
Chang, Chen & Su (2019)	Android based personal safety app	Limited accessibility and availability

Ali & Alwan (2017)	Smartphone based accident detection system	Easily vulnerable to false alarms situations, accuracy affected
Kattukkaran, George & Haridas (2017)	Bike fall detection using accelerometer and heartbeat sensor	Only reliable for bike riders
Mane & Rana (2019)	Vehicle collision detection using Arduino	System decides for itself whether true accident or false alarm
Nirbhavane & Prabha (2019)	Accident monitoring system using heartbeat sensor	Prone to inaccuracy as heartbeat is the only input
Shaik et al. (2018)	Accident detection system using raspberry pi	No mechanism to cancel false alarm
Win, Aung & Thin (2019)	Accident detection system using Arduino Uno	No false alarm cancellation and rely on IoT cellular
Rajkiran & Anusha (2019)	Multiple input accident detection system	No false alarm cancellation
Chaturvedi & Srivastava (2018)	Vehicle tracking and accident detection system	System only works when the vehicle is turned on
Amin, Jalil & Reaz (2021)	Accident detection and reporting system	Detection determines by GPS only
Desima et al. (2017)	Bus accident detection system	Rely on gas sensors and flame sensors only

3. Methodology

Embedded system design is used to determine how to integrate both hardware and software together as in Fig. 1. This involves proper planning and execution with the intention of trying to get the best output of both systems. The first stage is to collect and determine the product requirements and translate them into specifications as follows.

i. A user must be registered to use the system. Registration is only done for first time users as the app requires them to store their personal information and emergency contacts. This data is then stored in the online database and local storage in the Shared Preferences.

ii. Users login to the app by entering email and password and will be able to access their data. This can be done once, and users can keep using the app without having to log in every time. The user however can set in the setting for the app to log out after every use.

iii. Users will be required to connect to their Arduino device for the first time, during setting up of the device. This allows the application to seamlessly and quickly connects to the device every time the device is turned on.

iv. The smartphone should be in range of the user to respond to the warning notification if an accident is detected. Preferably in the area where it can be easily reached by the driver.

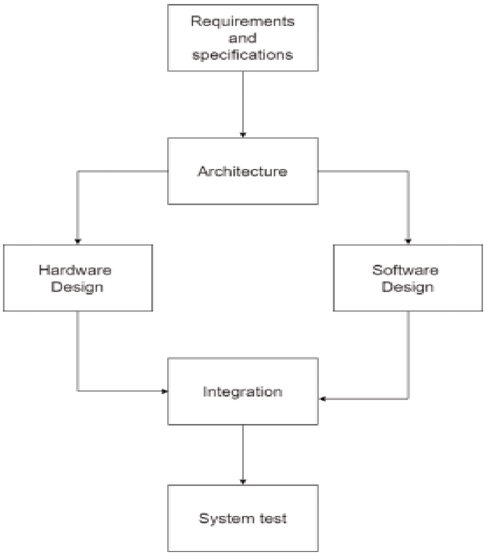

Fig. 1: Embedded system design methodology

The aim of the project is to create a device that is able to detect the crash incident and provide alert. The development tackles issues of Chang, Chen & Su (2019), Rajkiran & Anusha (2019) and Win, Aung & Thin (2019). In our work, we both hardware and software integration for crash detection was addressed. The architecture is planned out to analyse how both parts can interact with each other, which components and tools are needed and how much resources would be sufficient in developing them. Once that has been determined, the software and hardware are designed and later integrated. Testing was done based on the threshold (Kattukkaran, George & Haridas, 2017) as in Table 4.

Table 4: Testing threshold values.

Angle of X-Axis (degree)	Vibration (digital output)	Changes of speed in the last 3 seconds (km/h)	Changes in sound sensor in last 3 seconds (digital output)	Detection
90	0 - 500	0 – 35	0-19	Normal
0 - 40	501 - 1023	> 35	>19	Accident
140 - 180	501 - 1023	> 35	>19	Accident
90	501 - 1023	> 35	>19	Accident

4. System Implementation and Testing

The device is designed as a portable system with the Arduino Mega 2560 R3 microcontroller placed inside a small compact box that is 15cm in length, 11cm in width, and 5cm in height. This box can be placed anywhere inside the car while connected to a USB power supply. Fig. 2 shows the architecture of the proposed work where it illustrates the component and module used and how it interacts with one another.

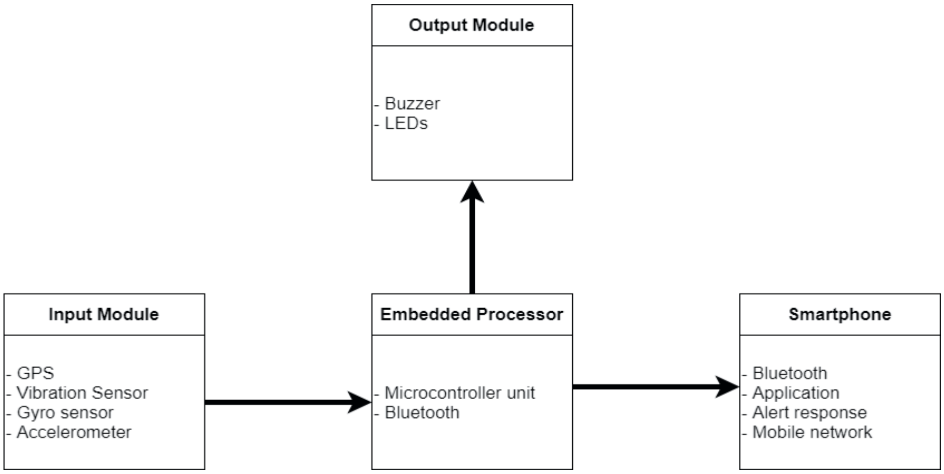

Fig. 2: System architecture.

The accuracy of the readings may definitely vary if it is placed in the front or back due to heavy vibration or road structure impacts. Hence, the device was placed inside the vehicle (center). The placement in center was much considerable where the sense of rotational motion and changes in orientation can be detected. Moreover, placing the device on the floor mat enables the vibration sensor to get a stable reading, while also avoiding any false positives as the box would not be able to move around due to the movement of the vehicle. Another option was to place the device under the vehicle (center). However, we do not implement this since it was challenging to monitor the device during the experiments if it was placed under the vehicle.

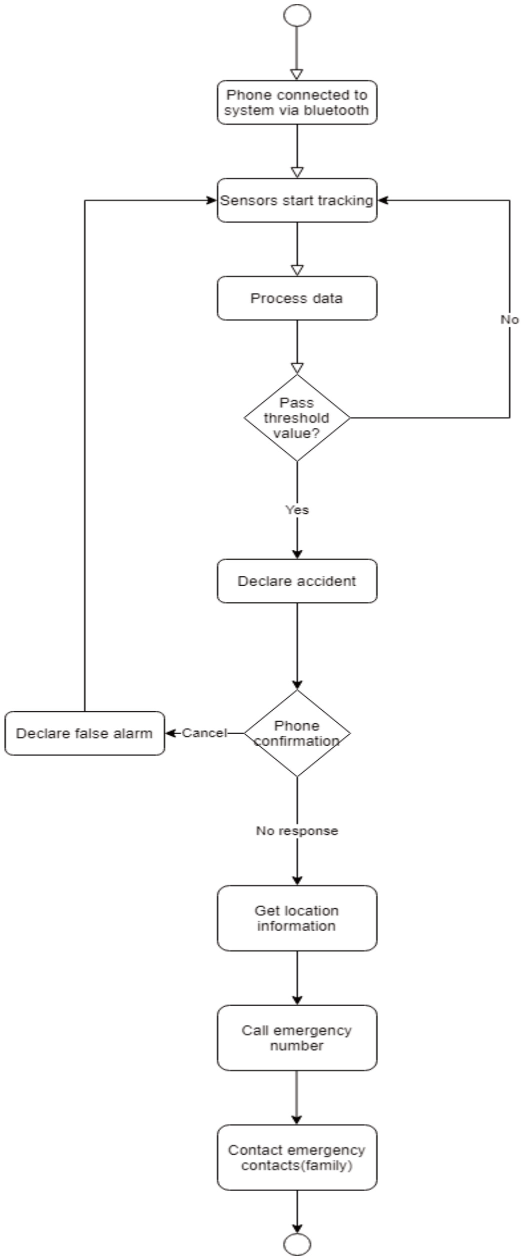

Fig. 3: System flowchart.

The HC-05 Bluetooth Serial module enable the microcontroller to exchange data with other Bluetooth devices, or Android smartphones whereas the Ublox NEO-6M GPS module allows the microcontroller to connect to GPS satellites to acquire the vehicle geo location while also providing current time and date of the crash. The MPU6050 Accelerometer and Gyroscope module, Piezo vibration sensor and KY-037 sound detection sensor are important components to detect any crash occurrence.

Fig. 3 illustrates the system flowchart. The system requires a one-time registration from the vehicle user or driver to store vehicle data (car model, engine capacity, etc.) and personal details (residence address, contact information, emergency contact details, etc.). In addition, one-time login is also required to retrieve the data before starting a journey. It also requires the driver's mobile phone to be connected to the device via Bluetooth and encouraged the driver to place the mobile phone within his reach. Data is augmented from sensors in the background during the driving experience. The app will record journeys based on the data received from the sensors, therefore the app can show the location history of the user when connected to the device.

The main screen of the app (Fig. 4) will be launched if the user is a registered user. The screen consists of four main menu. The first one is the Profile, where users can change their details such as name, phone number, and email. The second one is Device, where they would have to connect the Arduino through the app itself. The third one is Emergency Contact, where they can change the details of their contacts and lastly, Logout, where they can log out of the application. Once logged out, all the data stored by the previous user will be erased. It is advisable for the users not to log out as they might need to key in the details every time they sign in again.

Fig. 4: Main Screen

If any of the data exceed the predefined threshold values, the system will detect that a crash has occur. This represents Phase 1 of the system which is the detection. A confirmation screen will appear. The user can confirm whether it is a false alarm, a minor crash or require medical attention as in Fig. 5.

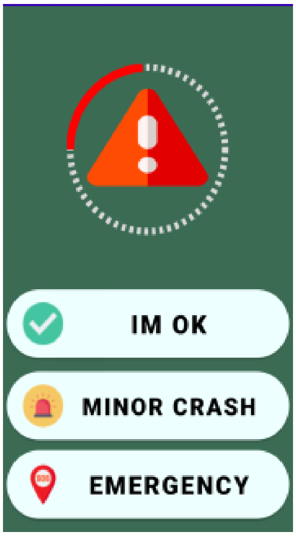

Fig. 5: Confirmation messages.

If an emergency is declared, Phase 2 will be triggered where an SMS notification with the location of the vehicle will be sent to all emergency contact numbers specified by the user. Similar action will be triggered if no response received within 60 seconds. During the 60 seconds, the Arduino would keep reading the data from the GPS module and check whether the vehicle is still moving or has stopped. If it is still moving, then the app would cancel out the alarm by itself, assuming that the car was driven over a pothole or bumpy/uneven road. Otherwise the app would send out an alarm for the user to respond. This minimises the possibilities of false alarm and avoiding unnecessary e-Call to be sent to the first emergency responders. The driver can also notify a crash if the system fails to detect the crash detected by the system, or turn off the notification if it is just a minor accident and no medical attention is required i.e., declare as false alarm.

In the event of the driver is unconscious and unable to provide confirmations to the app, the app will transfer the vehicle location and notify the emergency department. The app will also send SMS notification to the emergency contact numbers provided by the user during registration/login. In addition, a by-stander or anyone who is around can provide confirmation or call emergency number using the app as well.

The system was tested in a few scenarios as in Table 3. We considered influence of road conditions including straight and uneven road and running into potholes as well as driving up and down a hill at average normal speed between 20-80 km/h. Furthermore, the system was also tested to consider in-vehicle environment like loud music. The integration between both software and hardware components was able to fulfil the system functional requirements with no false alarms.

Table 3: Testing scenarios.

Scenario	Results
Car moving at 40km/h - 80km/h on straight road.	No crash detected or false positive.
Car moving at 20-40km/h on uneven road.	No crash detected or false positive.
Car moving at 40km/h-80km/h up and down a hill.	No crash detected or false positive.
Car running over potholes.	No accident detected or false positive.
Car driven with loud music in the background.	No accident detected or false positive.
Car moving at 20-40km/h and hits empty boxes.	Crash detected, confirmation required from user.
Car hits an object and assumed minor crash.	SMS sent and received by emergency contact.
Car hits an object and assumed require medical attention.	Auto-dial emergency line; SMS sent and received by emergency contact.

5. Conclusion

The 2-phase crash detection system is an alternative for existing vehicles which are not equipped with intelligent tools as high-end vehicle models. Since the device is portable, the pool of potential users are bigger. The system was able to detect a crash occurrence based on data from the vibration sensor, sound sensor, gyroscope and accelerator modules. Notification is made possible using the GPS and Bluetooth module. The app can be run in the background during a driving experience where data from sensors will be acquired in real-time. False alarms are minimised by allowing user to respond within 60 seconds after the detection is triggered and checking whether the car is still moving or have come to a halt based on the gyroscope and accelerator module. Nevertheless, more testing need to be conducted to increase the reliability and durability of the system. In our future work, we plan to benchmark the performance of our proposed system with existing system in various situational based experiments.

References

Ali, A. & Eid, M. (2015). An automated system for accident detection. *In 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, (pp. 1608-1612). DOI:10.1109/I2MTC.2015.7151519.

Ali, H. M. & Alwan. Z. S. (2017). Car accident detection and notification system using smartphone. *International Journal of Computer Science and Mobile Computing (IJCSMC)*, 4(4), pg. 620-635.

Amin, M. S., Jalil, J. & Reaz, M. B. I. (2021). Accident detection and reporting system using GPS, GPRS and GSM technology. *In 2021 International Conference on Informatics, Electronics & Vision (ICIEV)*, IEEE , 640-643). DOI:10.1109/ICIEV.2012.6317382.

Chang, W. J., Chen, L. B. & Su, K. Y. (2019). DeepCrash: A deep learning-based internet of vehicles system for head-on and single-vehicle accident detection with emergency notification. *IEEE Access*. 7, 148163-148175. DOI:10.1109/ACCESS.2019.2946468.

Chaturvedi, N. & Srivastava, P. (2018). Automatic vehicle accident detection and messaging system using GSM and GPS modem. *Int. Res. J. Eng. Technol. (IRJET),* 5(3), 252-254. DOI:10.14445/22315381/IJETT-V67I8P211.

Desima, M. A., Ramli, P., Ramdani, D. F. & Rahman, S. (2017). Alarm system to detect the location of IOT-based public vehicle accidents. *In 2017 International Conference on Computing, Engineering, and Design (ICCED),* IEEE, 1-5. DOI:10.1109/CED.2017.8308118.

Faiz, A. B., Imteaj, A. & Chowdhury, M. (2015). Smart vehicle accident detection and alarming system using a smartphone. *In 2015 International Conference on Computer and Information Engineering (ICCIE)*, (pp. 66-69). IEEE. DOI:10.1109/CCIE.2015.7399319.

Kattukkaran, N., George, A. & Haridas, T. M. (2017). Intelligent accident detection and alert system for emergency medical assistance. *In 2017 International Conference on Computer Communication and Informatics (ICCCI),* IEEE, 1-6. DOI:10.1109/ICCCI.2017.8117791.

Mane, A. & Rana, J. (2019). Vehicle collision detection and remote alarm device using Arduino. *International Journal of Current Engineering and Technology*, 4(3), 401-404. DOI:10.11591/ijres.v1.i2.pp49-54.

Nasr, E., Kfoury, E. & Khoury, D. (2016). An IoT approach to vehicle accident detection, reporting, and navigation. *In 2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, IEEE, 231-236. DOI:10.1109/IMCET.2016.7777457.

Nirbhavane, M. & Prabha, S. (2019). Accident monitoring system using wireless application. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1532-1535.

Rajkiran, A., & Anusha, M. (2019). Intelligent automatic vehicle accident detection system using wireless communication. *Int. J. Res. Stud. Sci. Eng. Technol*, 1(8), 98-101.

Rustagi, N., Kumar, A., Norbu, L. & Vyas, D. (2019). Applying Haddon matrix for evaluation of road crash victims in Delhi, India. *Indian Journal of Surgery*. 80(5), 479-487. DOI:10.1007/s12262-017-1632-0.

Shaik, A., Bowen, N., Bole, J., Kunzi, G., Bruce, D., Abdelgawad, A. & Yelamarthi, K. (2018). Smart car: An IoT based accident detection system. *In 2018 IEEE Global Conference on Internet of Things (GCIoT)*, 1-5. IEEE. DOI:10.1109/GCIoT.2018.8620131.

Win, Y. Y. M., Aung, M. M. & Thin, T. (2019). Accident detection system using Arduino Uno. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, 3(5), 2090-2093.