Course Project Documentation

CS308 Project

Accident Auto Dialer

Team 7

Karan Ganju | 120050021
Sanket Kanjalkar | 120050011
Roshan Raghupathy | 120050082
Deepanjan Kundu | 120050009

Table of Contents

Introduction

In car crash relief, there is a golden hour rule. If medical aid is not provided to the victim within an hour of a serious accident, there is some irreparable damage to his/her body. Often, accidents involve people searching for nearest hospitals, unable to relay their own locations effectively or in particular cases, unable to contact for help at all.

Our project is a step towards improving this situation and aims to reduce the time between the occurrence of an accident and the relief provided to the corresponding victims. Our device has to be secured firmly within the car, after which it monitors for crashes, segregating minor brushes from dangerous crashes. In case of car crashes, it immediately alerts people about the accident along with providing the location. Hence, this accident auto dialer plays a big role in providing relief for car crash victims.

An additional feature which is very closely tied to the nature of hardware we are using and readings we are taking is to provide a measurement of how well driven a car ride has been. In essence, a well driven ride is one without many jerks and our system, apart from providing accident alerts, also rates the drive on a total score of 5. The drive rating feature can be integrated into taxi applications and perhaps even an indicator for drunk driving.

Problem Statement

The aim of the project is to create an accident autodialler. The accident auto dialer should consist of three parts which are each explained as follows

- Accident and driver rating detection - The embedded system should, via detection of changes in linear and angular acceleration, detect appropriately whether an accident has occurred. The frequency and severity of jerks or momentum changes also provide a basis to judge the drive rating.

- Location detection - At the time of the accident, the location of the accident, in terms of GPS coordinates has to be relayed to either an emergency contact or to the hospital. Hence, as soon as an accident is detected the GPS should be powered on and coordinates should be obtained.

- Communication - This relay of information should take place via cellular network as WiFi networks cannot be expected in the car. Hence, communication of this data has to take place over 2G/3G networks with the occasional use of SMS messages to relay messages for emergency contacts.

Requirements

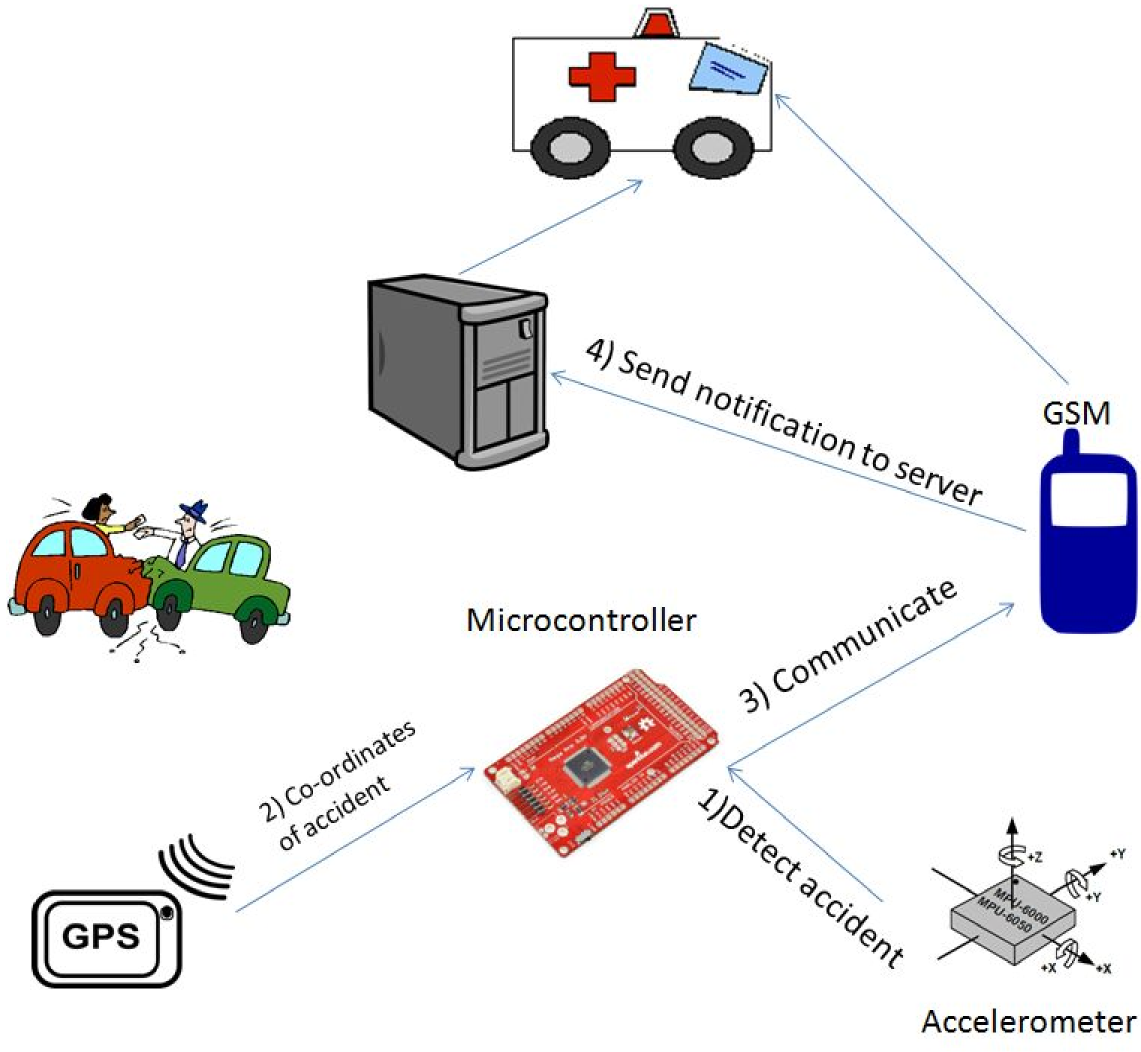
- Functional Requirements
    - Detects car crashes and provides alerts via both GPRS and SMS
    - Obtains drive rating and communicates to a selected number

- Non-functional Requirements
    - Availability - There should be no obstacle at any time which prevents relay of information between the device and the server
    - Efficiency - In terms of both memory and power
    - Secure - The server should be secure because any attack on the integrity of the server can result directly in loss or damage to life.
    - Interoperability - so that other medical services can be integrated with this
    - Fault-tolerant
    - Robustness - It should stay securely fixed in the car so as to provide accurate readings.
    - Minimum response time

- Hardware Requirements
    - TIVA Microcontroller
    - MPU9250 Accelerometer and Gyrometer
    - SIM908 GPS+GSM Module
    - RC Car (for testing purposes)
    - Sim Card

- Software Requirements
    - AVR Studio
    - CCS Editor

System Design



Implementation

Accident Detection

The MPU9250 module is used to obtain the linear and angular acceleration acting on it. These values are not by themselves as important as is the difference between two consecutive readings. The difference itself is indicative of an additional force that has been applied in between. This force can be a sudden brake, a slope, a pothole or even a life-threatening accident.

We currently employ a simplistic algorithm to identify a fatal car crash. This algorithm can be further improved to eliminate false positives and false negatives but that was not within the scope for our model of the prototype.

We maintain a threshold level to detect fatal crashes and if the squared value of the change in acceleration exceeds this, we classify the incident as an accident and take appropriate action. So, for a threshold value T and acceleration a, with $\Delta_x a$ representing the change in acceleration along the x axis., we say that an accident has occurred if

$$(\Delta_x a)^2 + (\Delta_y a)^2 + (\Delta_z a)^2 \ = \ \delta \ \geq T$$

Side Detection

We also detect which side of the car has been damaged in case of an accident as this can give information of how badly the driver is injured. We have only considered the front, back, left and right sides of the car for this. To derive this, as soon as an accident is detected we detect the summation of acceleration values for a pre-defined x iterations after this and see the direction along which acceleration is maximum. Then, according to the alignment of the MPU9250 module within the car, which we require to be fixed with +y axis along the car's length towards the driver, we predict the direction of the accident using this net value .

Acceleration Change	Location where car is hit (from Driver's POV)
+y	Back
-y	Front
+x	Left
-x	Right

Drive Rating

For drive rating, we take a summation of δ values calculated above across a fixed time period. Then, we compare this to the same value for the worst drive that we have obtained experimentally. Again, more complex algorithms can be used which take more factors into account such as angular accelerations to determine how rashly turns are taken, but these were not the focus for the prototype. Hence, the driver rating out of 10 is obtained as R given the value k for the worst drive and measurements for time t.

$$R = floor(max(1 - \frac{(\sum^{t} \delta)}{k}, 0) * 6)$$

Location Detection

The location is detected using GPS module which uses AT commands to obtain an NMEA string. We obtain the RMC sentence which basically give the minimum GPS information required for acquiring location.

Communication

To set up GPRS communication between the victim and the emergency contact, we needed to create a server hosted on openStack. For any communication, the embedded device sends the data to the server. The android app possessed by the emergency contact regularly polls the server for request for any new information. If there is new information, it is sent to the emergency contact. Hence, any data, either the occurrence of an accident or the rating of the drive, is shared via this server. The occurrence of this server adds possibility for further processing on such data. This data can be sent to custom-set people or to hospitals for immediate action.

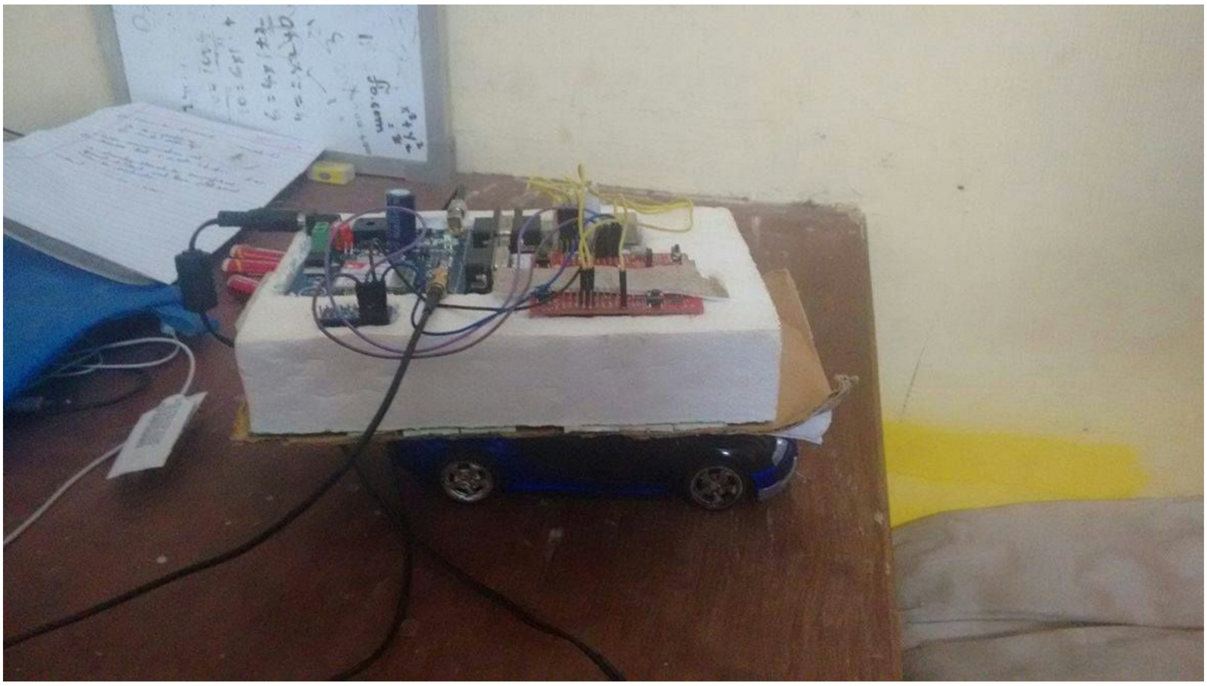
However, we felt notification of the occurrence of accidents should not be conditional on whether the sim card is connected to the internet and hence, we also send an SMS in case there is an accident.

Testing Strategy and Data

Various features had to be tested for the functioning of this device but we mention only the main tests which occur after the whole device was composed.

Accident and Side detection

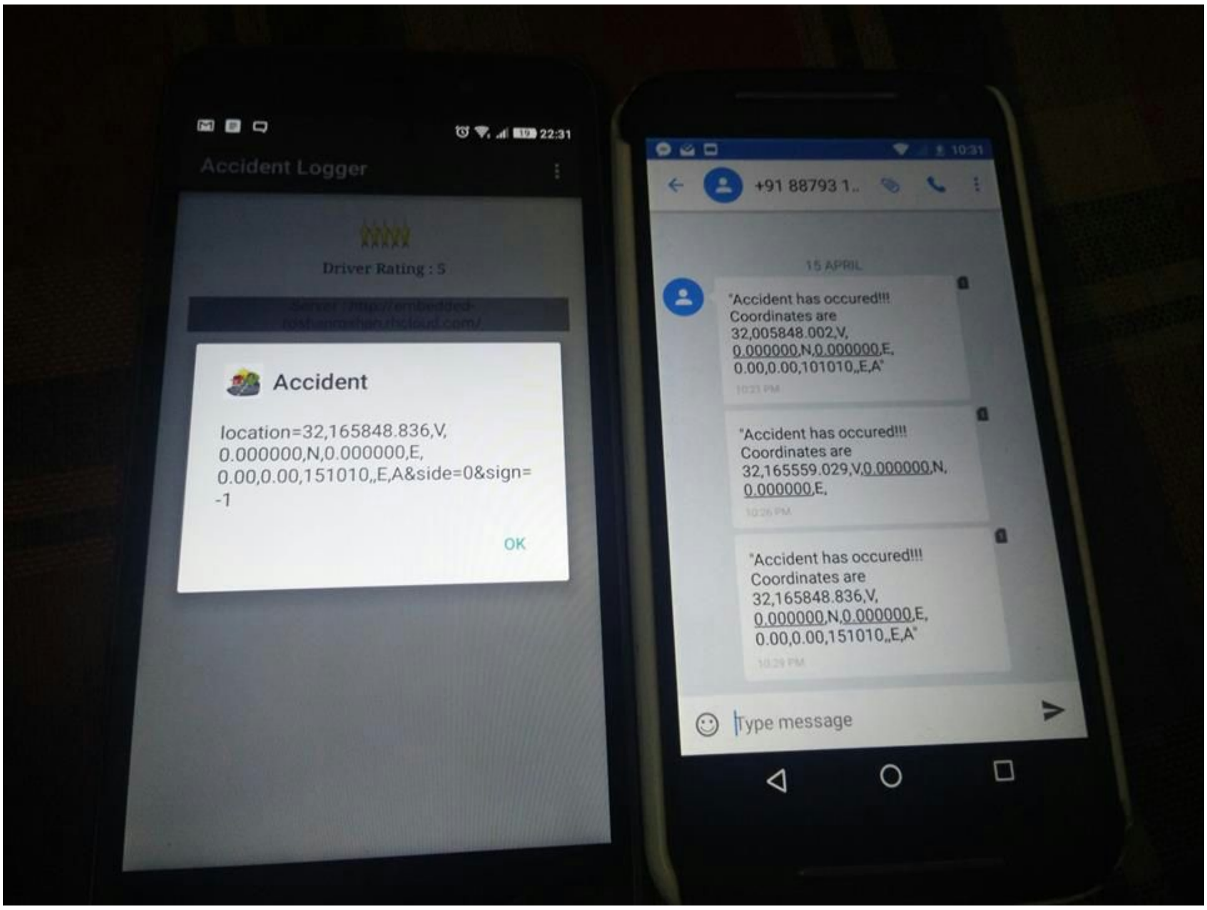
We had attached out device to the RC car as shown below.



Three test scenarios were held to distinguish between car crashes and sudden jerks due to brakes and to distinguish between the direction of the car accident. For all these test scenarios, the TIVA processor was attached directly to the PC with debug statements printing the value of summation of squares of acceleration along the three axises. The three test scenario are mentioned below

- Running the RC car into a wall at max speed

In this case, an accident is detected for the following summation value and a threshold value of 1000 calibrated for the RC car. The following SMS is also sent to the emergency contact as shown below (coordinates are 0,0 as this experiment was conducted indoors).



- Tapping the MPU9250 attached to the RC car from the side

Similarly, here as well, the accident is detected with the correct value of direction.

- Running the RC car at full speed and then braking it to a stop

Here, the value of summation does not cross the threshold as shown and so no accident is detected.

Drive Rating Detection

Three tests were held again to differentiate between good, average and rash drives. The scenarios are mentioned below

- Crashing the car every now and then (worst drive)

```
Driver stats : 19
Driver stats : 9
Driver stats : 6
Driver stats : 4
AT+HTTPPARA="URL","embedded-roshanroshan.rhcloud.com/add/Driver_rating=1"
```

In this case, the device identifies these crashes and the application reflects this as shown below. Extreme rash drive gave summation average values 19,9,6,4. The last value 4 denotes the average of summation, hence the drive is given a rating of 5-4 = 1.

- Sudden brakes and starts (average drive)

Similarly, here the drive is given a rating of 3.

- Smooth drive (best drive)

Here, the drive is given the best rating of 5.

Discussion of System

- Successfully able to
  - detect momentum using MPU9250
  - send SMS and GPRS data using GSM module
  - obtain location from GPS
  - create and algorithm and calibrate it for RC car to correctly judge accidents

- Added features (as per investor's pitch)
  - create an Android app which displays relevant data from the system
  - provide a drive rating based on how many jerks there were or how smooth the ride was
  - obtain an estimate of which side the accident has taken place (sometimes not accurate)

- Changes made in plan
  - Had to create a server hosted on openStack to relay information between the embedded system and the android app because we did not take into account that the smartphone on the 3G network lies behind a NAT box

Future Work

- Accident Detection Algorithm - The algorithm for accident detection can be made more fault proof. Machine learning concepts can be introduced to reduce the number of false positives and false negatives.

- Fully Automated - The current prototype requires the user to press some buttons to individually switch on the microprocessor, and then the GPS/GSM module. It would be best to connect this to the car's battery and have switched on immediately when the key is inserted.

- Power conservation - We can make the system more power-conservative by switching the GPS and GSM module on only after the accident.

- Better App - Google maps can be integrated with the Android application to provide more visual and intuitive location details.

- Integration with Emergency Hotlines - A very impressive thing would be to generate a voice message using the location and contact the emergency hotline numbers and relay this message to them. This automated system would really drive forward road safety.

- Button and Beeper for False Positives - A button and beeper can be integrated within the system which beeps if an accident has been detected. If there is no accident, the user can press the button within some defined time period to prevent the notification so as to prevent false positives.

Conclusion

We sincerely feel that such a system, if installed within vehicles universally and integrated well with the nation's medical facilities, can go a long way in improving the road safety conditions and mortality rates of the country. There are already Android apps for the same but they limited in use due to limited or varying hardware and battery and their casual nature of usage. Some high-end cars have an auto dialer but we feel that this system has to be made cheaply available.

Secondly, the drive rating feature can be used for licence approval, by taxi companies to ensure its drivers are good, even perhaps to detect drunk driving. We end with the note that we are moving towards a smarter world, where cars, along with lots of other devices, are becoming smarter and so the ability to report damage and potential fatality was long overdue. On a more amusing note however, don't be a big jerk or else we'll report you.

References

AT command manual for both GPS and GSM
http://www.dfrobot.com/image/data/TEL0051/3.0/SIM908_AT%20Command%20Manual_V1.01.pdf

MPU9250 Datasheet
https://store.invensense.com/datasheets/invensense/MPU9250REV1.0.pdf

Tutorial for GPS/GSM module
https://www.cooking-hacks.com/documentation/tutorials/geolocation-tracker-gprs-gps-geoposition-sim908-arduino-raspberry-pi/