

Telematics

Declaration of Joint Authorship

We, the Galaxy Noise, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses that was used in this documents that came from any other authors were acknowledged where their works were used. A list of references are included in this document and the work breakdown can be find on Overall Description Section 2.3.

Galaxy Noise: Jerome Aganon, Jabneel Cruz, Jaewhoong Kang, Achille Mattapullut

Date: February 6, 2017

Proposal

Submission January 17, 2017

Prepared by Galaxy Noise (Jerome, Jabneel, Jaewoong)

Computer Engineering Technology Student

<https://semjerome.github.io>

Executive Summary

As students in the Computer Engineering Technology program, we will be integrating the knowledge and skills we have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with sensors and /or actuators for detecting car collision and save the captured image/video in a black box. The database will store the timestamp and the folder location of the video that were saved in the blackbox. The mobile device functionality will include notifying the car owner of an accident happening and if possible, an image or video capture of the accident and will be further detailed in the mobile application proposal. We will be collaborating with the following company/department Engineering Club. The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 2 or 3 student group.

Background

The problem solved by project is to not prevent any accident but to get evidence if an accident happened. Having a dash cam does not mean that if an accident happened there would be footage of what happened. Sometimes, accident happens on the side or back of the car. In the event that such thing happened, the people will rely on the story of a witness or the victims. Sometimes the people will try to change their story in order for them to get away with it. If such thing happens, the police will have to look around and see if a CCTV has captured the accident.

I have searched for prior art via Humber's IEEE subscription selecting "My Subscribed Content" and have found and read which provides insight into similar efforts.

The first journal that we found presents " the approach to use smartphones as a multi sensor platform in a field operational test respectively a naturalistic driving study". (Pfriem & Gauterin, 2014)

The second journal talks about optical camera used for vehicles. (Nguyen, Islam, & Jang, 2016)

For the last journal, it then talks about inevitable accidents that can happen with motorcycles and automobiles. (Savino, Giovannini, Fitzharris, & Pierini, 2016)

In the Computer Engineering Technology program we have learned about the following topics from the respective relevant courses:

- Java Docs from CENG 212 Programming Techniques In Java
- Construction of circuits from CENG 215 Digital And Interfacing Systems,
- Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
- Micro computing from CENG 252 Embedded Systems,
- SQL from CENG 254 Database With Java,
- Web access of databases from CENG 256 Internet Scripting; and,
- Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project: Phase 1 Hardware build. Phase 2 System integration. Phase 3 Demonstration to future employers.

Phase 1 Hardware build

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

Phase 2 System integration

The system integration will be completed in the fall term.

Phase 3 Demonstration to future employers

This project will showcase the knowledge and skills that I have learned to potential employers.

The tables below provide rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

Labour Estimates	Hrs	Notes
Phase 1		
Writing proposal.	9	Tech identification quiz.
Creating project schedule. Initial project team meeting.	9	Proposal due.
Creating budget. Status Meeting.	9	Project Schedule due.
Acquiring components and writing progress report.	9	Budget due.
Mechanical assembly and writing progress report. Status Meeting.	9	Progress Report due (components acquired milestone).
PCB fabrication.	9	Progress Report due (Mechanical Assembly milestone).
Interface wiring, Placard design, Status Meeting.	9	PCB Due (power up milestone).
Preparing for demonstration.	9	Placard due.
Writing progress report and demonstrating project.	9	Progress Report due (Demonstrations at Open House Saturday, November 12th, 2016 from 10 a.m. - 2 p.m.).
Editing build video.	9	Peer grading of demonstrations due.
Incorporation of feedback from demonstration and writing progress report. Status Meeting.	9	30 second build video due.
Practice presentations	9	Progress Report due.
1st round of Presentations, Collaborators present.	9	Presentation PowerPoint file due.
2nd round of Presentations	9	Build instructions up due.
Project videos, Status Meeting.	9	30 second script due.
Phase 1 Total	135	
Phase 2		
Meet with collaborators	9	Status Meeting
Initial integration.	9	Progress Report
Meet with collaborators	9	Status Meeting
Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Meet with collaborators	9	Status Meeting
Incorporation of feedback.	9	Progress Report
Meet with collaborators	9	Status Meeting

Testing.	9	Progress Report
Meet with collaborators	9	Status Meeting
Prepare for demonstration.	9	Progress Report
Complete presentation.	9	Demonstration at Open House Saturday, April 8th, 2017 10 a.m. to 2 p.m.
Complete final report. 1st round of Presentations.	9	Presentation PowerPoint file due.
Write video script. 2nd round of Presentations, delivery of project.	9	Final written report including final budget and record of expenditures, covering both this semester and the previous semester.
Project videos.	9	Video script due
Phase 2 Total	135	
Phase 3		
Interviews	TBD	
Phase 3 Total	TBD	
Material Estimates	Cost	Notes
Phase 1		
RaspBerry Pi 3 Starter Kit	\$89.99	https://www.amazon.com/Vilros-Raspberry-Ultimate-Starter-Kit-Clear/dp/B01CYWE20U
Pi Camera Module with Case	\$38.79	https://www.amazon.com/Raspberry-Pi-Camera-Module-Megapixel/dp/B01ER2SKFS/ref=sr_1_2?s=pc&ie=UTF8&q2&keywords=raspberry+pi+camera
Piezo Buzzer Element (Vibration Sensor)	\$5.19	Canada Robotix
LED	\$0.50	Canada Robotix
USB GPS Dongle.	\$50	Amazon
Phase 1 Total	>\$200.00	
Phase 2		
Materials to improve functionality, fit, and finish of project.		
Phase 2 Total	TBD	
Phase 3		
Off campus colocation	<\$100.00	An example: .
Shipping	TBD	
Tax	TBD	
Duty	TBD	
Phase 3 Total	TBD	

Concluding remarks

This proposal presents a plan for providing an IoT solution for mini computer and black box for car. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects such as the initiative described by our professor. I request approval of this project.

Abstract (Executive Summary)

In today's society, car has been part of our daily life. Based on National Safety Council, around 27,000 accidents occur everyday in United States alone but only 0.3 percent of it result in a fatality. The problem solved by our project is not to prevent any accident, but to get evidence if an accident happened. Having a dash cam does not mean that if an accident happened there would be footage of what happened. Sometimes, accident happens on the side or back of the car. In the event that such thing happened, the people will rely on the story of a witness or the victims. Sometimes the people will try to change their story in order for them to get away with it. If such thing happens, the police will have to look around and see if a CCTV has captured the accident. With our project, it will help the police or the Insurance Company to determine who is at fault if an accident occurs without any hassle. And at the same time, it will encourage the driver to abide the road rules since this project can be used against them if they do not.

Table of Contents

Declaration of sole Authorship

Proposal

Abstract

1. Introduction

1.1 Purpose

1.2 Scope

2. Software Requirements Specification (SRS)

2.1 Technology Introduction

2.1.1 Purpose

2.1.2 Product Overview

2.1.3 Targeted Audience Group

2.2 Product Information

2.2.1 Main Functionality

2.2.2 Extra Requirements

2.2.3 Best Performance

2.3 Overall Description

2.3.1 System Interface

2.3.2 Database

2.3.3 Hardware

2.3.4 Mobile Application

2.3.5 Web Application

2.4 Future Consideration

2.4.1 Operating Environment

2.4.2 Safety consideration

2.4.3 Future Additions

2.5 Build Instructions

2.5.1 Introduction

2.5.2 System Diagram

2.5.3 Bill of Material/Budget

2.5.4 Time Commitment

2.5.5 Mechanical Assembly

2.5.6 PCB/Soldering

2.5.7 Unit Testing

2.5.8 Production Testing

Conclusion

Recommendation

Reference

1. Introduction

1.1 Purpose

The purpose of this documentation is to give a detailed description of the requirements for the "Telematics" hardware and all of the software that comes with it. It will explain system constraints, system interface and interactions with other external applications. This documentation is primarily intended for our professor and his approval as well as reference for developing the system.

1.2 Scope

The "Telematics" is a hardware that helps the police, insurance companies and drivers to determine the cause of a car incident. The hardware will have 3 cameras (one on each side and one on the rear). Although in this prototype, it will only have one camera because the raspberry pi3 can only support 1 raspberry pi camera at a time. It will also have an Android application where the user can review any previous incident that has happened and add more information if needed.

Upon installing the hardware, the user has to register through the official website and enter the appropriate information and product key. From then on, the hardware will be able to send any information to the database once it is connected to the internet. The hardware does not need to maintain a network connection in order to work. Once the car starts, it will start recording and will replace the old file with a new file until it detects an accident. Once the hardware detects an incident, it will lock the file and will not be able to be overwritten until it is successfully uploaded to the server. Once the hardware is connected to the network, it will upload the video to the server and any information related to incident such as date and the location of the incident.

Furthermore, the user will be able to access the information through their Android phone. From the mobile application, they will need to log in in order to retrieve the information related to their account. The user can then review the incident and add information about the incident such as, the car and driver's information. If the user does not have an android phone, they can do everything that was mentioned above through the official website.

1.3 Definitions, acronyms, and abbreviations

Term	Definition
Telematics	Name of the hardware.
User	Someone who interacts with the web or software application
GPS	Global Positioning System
Official Website	http://autaccreport.com/

2. Software Requirements Specification (SRS)

2.1 Technology Introduction

2.1.1 Purpose The purpose of our software is to provide an automated service for drivers that notifies them of any accidents or damages that may have occurred to their car. With the use of mobile application, the owner will have the data of the location and video feed of the accident. It will also serve as a data storage regarding the accident for the owner. They will be able to input additional information about the driver, the car, as well as information needed for the insurance company. A website will be available for the user to register an account for them to use the mobile application.

2.1.2 Product Overview The product will have piezo vibration sensors that will detect if there are any collisions detected. Raspberry Pi camera module will also be used to record video.

2.1.3 Targeted Audience Group: Our target audience is any person that has a car and insurance company for they will be able to use the video feed as a proof of an incident.

2.2 Product Information

2.2.1 Main Functionality The main functionality of our system is to provide an incident report on user's mobile device in case of an accident. Information that the user saves on the application can then be used by them for their insurance company. The hardware will provide the video feed as well as the coordinates of the report.

2.2.2 Extra Requirements Internet connection is needed for the hardware device for it to be able to send the report, video and location, to the database. Implementation of battery powered raspberry pi will also be needed for it to be placed inside the car. (Jerome please confirm)

2.2.3 Best Performance To achieve best performance of the product, we recommend sharing a stable wifi internet connection with the device.

2.3 Overall Description

2.3.1 System Interface System interface for our project includes: collision detector hardware, database server, and mobile application. If an accident is detected from the hardware, it sends an accident report to the server. The user through the use of mobile application in which he/she can retrieve the video file and location of the accident can then access this report. The user will also be able to delete the report (in case of false report), as well as add information to the report.

2.3.2 Database We are going to use the GoDaddy's database to store the information about the user, the accident, the driver they had an accident with and the car that the other driver was using. For the user information, we are going to store the username password and its user's id. For the password, we are going to encrypt the user's password using MD5 to protect the password just in case someone tries to hack our system. In accident table, we are going to store the reportId, incident date, video name, longitude and latitude. In the driver table, we are going to store the other driver's license, first name, last name, gender and insurance number. In the car table, we are going to store the other driver's car information such as plate number, brand, model and the year it was manufactured. (Jerome)

A separate database will be created for car makes and models, which is then going to be used for the mobile application. It will be used for drop-down spinners of both car makes and models. (Jabneel)

2.3.3 Hardware Our hardware is a collision detector. Once the car has start up, the hardware will start running immediately. It will start recording as soon as it is start up. It will have loop and lock functionality. With the loop functionality, the hardware will replace the old video with a new video, which prevents the hardware from using too much memory. The lock functionality will lock up the video file if the hardware detects any collision. It will lock up the file until the hard is connected to the Internet.(Jerome)

Once it is connected, it will upload the file to the server and store the date that the video was captured, the location and the name of the video. It will also have a GUI(Graphic USer Interface so that it will be easy to use for new user.(Jaewoong)

2.3.4 Mobile Application Our mobile application is called AutAcc Report and is exclusive to Android. With the mobile app, the user is able to retrieve any incident related to the user. Once it is able to retrieve an incident report, the user will be able to add more information about the driver that he/she had an accident with and the car he/she was using. The application has playback functionality where it can watch the video that was captured during the incident. It also has map functionality where it will pinpoint the exact location of the incident by using Google map. (Jabneel)

The mobile app also has functionality where the user can contact his/her insurance agent where he/she can report the incident. User accounts and having access to different reports is also a requirement. (Jerome)

2.3.5 Web Application For our project, we will also provide a website. The front-end design and user interface of the website will be worked on by Jabneel and Achille. Usability and design of the website will require. Developed by (Jabneel) and user testing done by (Achille).

In the website, the user will be able to register. If the user is already registered, he/she will be able to review the information related to his/her account. He/she will also be able to delete an incident or modify any information related to that incident. The user can also change their password. This will be worked on by (Jerome).

2.4 Future Consideration

2.4.1 Operating Environment To fully utilize this product you will need an Android device (Tablet or phone) on the OS android 4.4 KitKat and above (API 19 and above). This product does not require constant internet connection. An Internet connection is only required when you need to synchronize the hardware with the database.

2.4.2 Safety Consideration Safety precaution that has to be considered when using the hardware:

- When using the hardware, make sure that the hardware is properly plugged in before turning on the car.
- Avoid sunlight exposure

2.4.3 Future Additions We are planning on creating a new case for the hardware since we are planning on adding a touchscreen capability on our hardware.

2.5 Build Instructions

2.5.1 Introduction The project will be able to detect car collision and save the recorded video into a file server to be used by the android application. The goal of this project is to notify the car owner of an accident happening. The owner should be able to access the video of the recorded accident using their mobile application. We are using a vibration sensor (buzzer) to detect an impact signal that will be sent to the raspberry pi which then would run the camera to record a 20 second video that will be sent to the

Figure 1: alt text

Figure 2: alt text

server. This video file can then be accessed by the owner by using an android application. Take note that for now, our current production build is only sending the video file to the owner's email address.

2.5.2 System Diagram

Equipment	Quantity	Cost With tax
Raspberry Pi 3 Starter Kit	1	\$89.99
Pi Camera Module	1	\$38.79
Piezo Buzzer Element (Vibration Sensor)	2	\$5.19
Camera Module Case	1	\$25.28
1 Mega Ohm Resistor	1	\$0.35
330 Ohm Resistor	1	\$0.25
Full Sized Breadboard	1	\$8.81
USB GPS Dongle	1	\$50.00
LED	1	\$0.25
Total		\$193.24

2.5.3 Bills of Material / Budget

Todo	time Required
Raspberry Pi 3 power up	(30 mins) I will take around 30 min to set up the raspberry Pi. You have to install the OS and connect it to the internet.
PCB Soldering	1~2 hours for schematics and soldering. Printing the PCB could take 1 hour. And 30 min to solder it.
Project Wiring	15~30 mins
Importing Codes	1 hour, depends if you have all the required packages for using the camera and accessing email.
Running Code	5-10 mins, depends on the length of the video. The longer it is, the more time it needs to perform

2.5.4 Time Commitment

2.5.5 Mechanical Assembly This shows the circuit you have to build for the sensor to detect a vibration coming from tapping it. The LED will blink if the video file was sent to your email.

JP1 --- 2 is for the inner side of the buzzer. Positive. --- 1 outer part of the buzzer. Negative.

Following the diagram and placing each components in a breadboard would ensure a working vibration sensor program

Connect the particular camera module that I listed above to your Raspberry Pi 3.

2.5.6 PCB/Soldering We tried implementing our working circuit to a PCB but it didn't work.

2.5.7 Unit Testing You can use these programs for testing your Camera and Vibration Sensor if they are working properly.

Figure 3: alt text

Camera

```
"
from picamera import PiCamera
from time import sleep

camera = PiCamera()

camera.start_preview()
camera.start_recording('/home/pi/Desktop/Camera/video1.h264')
sleep(5)
camera.stop_recording()
camera.stop_preview()
"
```

Vibration Sensor

```
"
from time import sleep
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup(5, GPIO.IN, pull_up_down=GPIO.PUD_DOWN )
sleep(1)

while True:
    result = GPIO.input(5)
    if result == 1:
        print("Vibrated")
        sleep(1)
"
```

2.5.8 Production Testing We wanted the video file recorded to be sent to a file server, but as of now , it is being sent to your email.

Main program

testDrive.py

```
"
from time import sleep
import RPi.GPIO as GPIO
import time
from picamera import PiCamera
from alertnoise import sendAlert
from cameraCode import initRecvideo, stopVideo, recVideo
import os

#initRecvideo()

# initialize sesnor and leds
GPIO.setmode(GPIO.BCM)
GPIO.setup(5, GPIO.IN, pull_up_down=GPIO.PUD_DOWN )
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
```

```

sleep(1)
isTrig = False

def trig():
    print "LED on"
    GPIO.output(18,GPIO.HIGH)
    time.sleep(1)
    print "LED off"
    GPIO.output(18,GPIO.LOW)
    time.sleep(1)

print("Telematics~ running ...");
while True:
    GPIO.output(18,GPIO.HIGH)

    if isTrig == True:
        trig();
        #stopVideo()

    result = GPIO.input(5)
    if result == 1:
        print("Vibrated\n*****\nRecording Video")
        isTrig =True
        vidName = time.strftime("%d%m%Y")+time.strftime("%I%M%S")
        recVideo(vidName)
        sleep(1)
        hname = vidName+"video.h264"
        mname = vidName+"video.mp4"
        print("Converting video file h264 to MP4");
        commands = "sudo MP4Box -add" + " " +hname + " " +mname
        os.system(commands)
        print("Sending video file to email");
        sendAlert(vidName)
        print("File sent");
        "

```

NOTE: make sure to fill the values for raspaddress and useraddress. I removed it for obvious privacy reason.

Program that sends the recorded video file to your email

alertnoise.py

```

"
#Libraries necessary to send email using python
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEText import MIMEText
from email.MIMEBase import MIMEBase
from email import encoders

#Email of the Raspberry Pi
raspaddress = ""

#User's Email address
useraddress = ""

```

```

#Structure of the email
msg = MIMEMultipart()
msg['From'] = raspaddress
msg['To'] = useraddress
msg['Subject'] = "Aegis Alert Message"

def alert1(vidName):
    body1 = "Movement detected in the room \n"
    filename = vidName+"video.mp4"
    openAtt = "/home/pi/Desktop/Final Presentation/"+filename
    attachment = open(openAtt, "rb")
    part = MIMEBase('application', 'octet-stream')
    part.set_payload((attachment).read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition', "attachment; filename= %s" % filename)
    msg.attach(part)
    return body1

def alert2():
    body2 = "Termperature Change Alert \n"
    return body2

def alert3():
    body3 = "Smoke Detected \n"
    return body3

def sendAlert(vidName):
    body = alert1(vidName)
    msg.attach(MIMEText(body, 'plain'))
    #Parameters for GMail Server
    server = smtplib.SMTP('smtp.gmail.com', 587)
    # Security function needed to connect to the Gmail server to protect the password.
    server.starttls()
    #Password of Raspberry Pi's Email
    server.login(raspaddress, "P@$WORD")
    #Sending the email
    text = msg.as_string()
    server.sendmail(raspaddress, useraddress, text)
    server.quit()
    "

```

Records the video for 20 seconds. Can be adjusted by changing the value of sleep

cameraCode.py

```

"
from picamera import PiCamera
from time import sleep

camera = PiCamera()

def initRecvideo():
    # initialize camera
    camera = PiCamera()
    camera.start_preview()

```

```
def stopVideo():
    camera.stop_preview()

def recVideo( sNum ):
    dest = '/home/pi/Desktop/Final Presentation/'+str(sNum)+'video.h264'
    camera.start_preview()
    camera.start_recording(dest)
    sleep(20)
    camera.stop_recording()
    camera.stop_preview()
    return;
"
```

3. Conclusion

In this report, we wrote the purpose of this telematics and how to reproduce it. By following the instructions, you should be able to reproduce it and should run on how it is supposed to. On the other hand, it is not possible to create a working PCB as there are some connection issues.

4.Recommendation

When reproducing this hardware, it is recommended to get a Raspberry Pi 3 or later version. It is not recommended to use an earlier version because it does not have any built-in wi-fi or bluetooth capability.

To increase the reliability of the hardware, it is advisable to use Gravity sensor instead of vibration sensor. By using Gravity sensor, the video will only be lock when there is a disruption on the movememnt of the car. On the other hand, vibration sensor will lock the video everytime it sense a vibration.

Improvements that can be made on Android Mobile Application are the following: Have each fragments for video, location, driver and car in separate page fragments.

5.Reference

References (Generated in pdf)

Nguyen, T., Islam, A., & Jang, Y. M. (2016). Region-of-interest signaling vehicular system using optical camera communications. *IEEE Photonics Journal*, PP(99), 1–1. <https://doi.org/10.1109/JPHOT.2016.2644960>

Pfriem, M., & Gauterin, F. (2014). Employing smartphones as a low-cost multi sensor platform in a field operational test with electric vehicles. In *2014 47th hawaii international conference on system sciences* (pp. 1143–1152). <https://doi.org/10.1109/HICSS.2014.148>

Savino, G., Giovannini, F., Fitzharris, M., & Pierini, M. (2016). Inevitable collision states for motorcycle-to-car collision scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 17(9), 2563–2573. <https://doi.org/10.1109/TITS.2016.2520084>