REAL TIME GPS/GSM BASED VEHICLE MONITORING SYSTEM

FINAL YEAR PROJECT UG 2020

By
Capt. Farooq Azam

Capt. Ghazanfar Ghafar

Capt. Musaddiq Hussain

Capt. Ahsan Nadeem Asif

PROJECT SUPERVISOR

Assistant Professor Dr. Mehmood Alam

Submitted to the Faculty of Telecommunication Engineering (EE Department), Military College of Signals, National University of Sciences and Technology, Rawalpindi in preferential fulfillment for the requirements of a B.E. Degree in Telecom Engineering

June 2020

ABSTRACT

This report presents a method for real time GPS/GSM based vehicle monitoring system. The project is implemented to locate a vehicle under real time scenario. The project is implemented by using distinct methodologies like GPS and other ground based navigational systems. Using Trilateration or Triangulation methods the monitoring system enabled to determine easy and accurate exact location of the vehicle. Automobile information like exact details of location, speed, Driver's image, Crew members etc. can be viewed on android based application. This project is very important hardware kit for monitoring each vehicle at specified time interval or period. Our vehicle monitoring system is consisting mainly three parts which includes vehicle unit (Hardware part), database/Server and software android application. In true sense our project is an integrated part of vehicle security protection approach called "layered approach". This layered approach recommends four different layers of security depending upon the risk involved to some specific vehicle during move. Vehicle Monitoring system is one of such layers. Thus, real time vehicle monitoring system is important and essential part for any vehicle against theft or misuse.

ENDORSEMENT OF CORRECTNESS AND APPROVAL

It is affirmed that data presented in this thesis "**REAL TIME GPS/GSM BASED VEHICLE MONITORING SYSTEM**" carried out by 1) Capt. FAROOQ AZAM 2.Capt. GAZANFAR GHAFAR 3) Capt. MUSADDIQ HUSSAIN4) Capt. AHSAN NADEEM ASIF under the direction of Asst. Prof. Dr. MEHMOOD ALAM is in complete satisfaction of our level of Bachelor of Telecommunication Engineering, is right and endorsed. Percentage of plagiarism found in document as per software Turnitin available on LMS NUST comes out to be________________

Affirmed By

____

Asst. Prof. Dr. Mehmood Alam

Project Supervisor

DECLARATION

We hereby affirm that no content and variety of toil bestowed during this Thesis has been submitted in support of another award of qualification or degree either during this course or anyplace else.

This proposition is devoted in thanks to **ALLAH ALMIGHTY,** our Creator, who has blessed us with wisdom, knowledge and understanding, then to our parents for their direction and their endless support. I would also like to thank our faculty for their guidance and supervision. Without their help and supervision this project would not have been made possible.

ACKNOWLEDGEMENT

The accomplishment of this project would not have been possible without the contribution of different people the names of which cannot be reckoned with. Their assistances are sincerely respected and thankfully recognized. We would like to thank following:

- To Allah Almighty who is the most Powerful and Merciful and everything is in His obedience.
- Asst Prof Dr Mehmood Alam for his support and guidance during the entire project.
- We as a team acknowledge our sincere gratitude to our Parents for their love, desire and sacrifice throughout our life. We are grateful for their patience, and understanding that were certain to make this work possible. Their sacrifice had inspired us from the day we started learning how to read and write until what we have become now. We cannot find the suitable words that could properly explain our appreciation for their affection, support and confidence in our ability to achieve our ambitions.
- Lastly, we as team would like to thank any person who participated to our final year project directly or indirectly. We would like to admit their comments and recommendations, which were important for the successful completion of this project.

Table of Contents

LIST OF FIGURES

ABBREVIATIONS

1.	GPS	GLOBAL POSITIONING SYSTEM
2.	GSM	GLOBAL SYSTEM FOR MOBILE COMMUNICATION
3.	VTS	VEHICLE TRACKING SYSTEM
4.	LAT	LATTITUDES
5.	LONG	LONGITUDES
6.	VMS	VEHICLE MONITORING SYSTEM
7.	NMEA	NATIONAL MARINE ELECTRONICS ASSOCIATION

CHAPTER 1

REAL TIME GPS/GSM BASED VEHICLE MONITORING SYSTEM

1.1 Introduction

1.1.1. Background

Vehicle Monitoring System (VMS) is the is a technology which is used to extract the exact location of an automobile like car or truck or any other vehicle using different methodologies like GPS and satellite guided navigational systems and some ground navigational stations. The methods like triangulation or trilateration enables determination of accurate and exact location of the vehicle. Automobile concerned information like exact location, speed at specific time, driver image, crew members informations etc. can be seen on an android application via Internet. This project is an essential toolkit for monitoring/tracking each vehicle at some desired time and now this technology is becoming popular for people having expensive cars and hence giving the the sense of security against theft prevention and theft vehicle retrieval device.

1.1.2. Problem statement

"Design and implement a real time GPS/GSM based vehicle monitoring system." The Project is designed to be used for convey move, car antitheft measures etc.

1.2. Project description

1. The project consists of modern hardware components, server and software application facilitating one to monitor their car online anytime. Every automobile monitoring system is consisted of only hardware with an alert generation mechanism. But our project consists of following three parts for smooth vehicle monitoring and better control
2. **Automobile Unit:** It consists of integrated hardware components duly fitted inside vehicle. Its main part is a GPS Receiver. The hardware part is integrated about a primary microcontroller called Resberry Pi with camera and GPS receiver. It functions with the monitoring application by receiving signals from GPS satellites. The Resberry Pi converts the GPS Lat/Longs and drivers image into a burst of data and sends the data packet to the server/firebase.
3. **Server:** It is remotely accessed server. It accepts the data and make it in readable form for the android application. Base stations are equipped with tracking software and geographic map useful for determining the vehicle location.
4. **Android Application:** The location information in terms of Lat/Longs of each visiting place are sent and stored in the database or server. These are later viewed on google maps through the application. Although User of application needs to login first into application. For a limitation, the user must be connected to internet as online monitoring is possible. This part consists of live tracking, crew members information, and driver image at application end.

1.3. Prospective Application Areas

2 Single User to enhance sense of security for vehicles

3 Police/Army and paramilitary forces, to enable precise and quick response for their vehicles on road.

4 Logistic companies to track their vehicles

5 Adopt antitheft measure in case of stolen car

1.4 Scope, Objectives, Specifications and Deliverables

1.4.1. Scope and Objectives

The scope of this project was to achieve the following objectives,

1. Planning an active and online vehicle Tracking/Monitoring system
2. Understanding the usage of GPS, Microcontrollers and other sensors working in favor of our project
3. Designing the comprehensive hardware , database/server and application
4. Implementing the algorithm on android application.

Following are the goals we achieved with the project

1. Efficiently determining the location of vehicle in 2D plane
2. Modular design, we can enhance the design by interfacing different sensors by minute change in existing design
3. Different vehicles can be monitored and data can be used for antitheft measures of stolen vehicle.

1.4.2. Specifications

1. GPS for real time location of vehicle
2. 1x Camera for Drivers image
3. Python IDLE.
4. Android application
5. 5V USB power module
6. 3.7v 1000mAh Lipo Battery

1.4.3. Deliverables

1. GPS interface with Resberry pi zero wireless
2. Camera interface with Resberry pi zero wireless
3. Server/Database (3g/4g Cloud)
4. Android application
5. Integration of Application with server/database

CHAPTER 2
ANALYSIS OF PRESENT LITERATURE

2.1. Literature review

2.1.1. Overview of existing literature

1. **Advanced Vehicle Tracking System on Google Earth Using GPS and GSM**

the paper represents GPS based vehicle implementing tracking/navigation system. Vehicle Informations like speed , distance etc will be fetched by executing by using GSM and GPS. Given Characteristics showed the desired results .User define some kind of time period to update the statistics like speed or position. Then Tracker system will receive data after specified time period. The Google maps are used to show site on the display unit or any other displaying unit [1]

2. **Automobile monitoring and Emergency warning structure implemented on FPGA**

GPS(Global Positiong Systtem) is used to determine the position of automobile. With the help of GSM , an SMS containing Lat/Longs will be sent to the user. Through GSM Receiver module a response type message is sent by the possessor of vehicle. To check the emergency happened to the vehicle , a sensor named accelerometer is used in conjunction with the existing hardware.It also transmit the some kind of response signals. Spartan processors are used as microcontrollers. It will manage all parts according to the code installed in the system. [2]

3**. Tracker Structure Using GSM And GPS**

Management system and remote monitoring of commercial automobiles as busses and taxis are done by using the subject trackers. Under discussion paper Position is exactly calculated by the automobile hardware part of the project. The car module is interfaced by integrating GSM receiver and ARM processor. The Text message being sent by GPS is received and calculated by the bus terminal part. Lat/Longs of the vehicle position are then determined by the satellites . This location is then made a form of message and using GSM is sent to the owner of car. It is the Forwarded to the server through GSM network. [3]

4. **Google Maps Monitoring Using GSM/GPS and microcontrollers**

Satellite generated Lat/Longs are received by GPS and then coordinated are calculated or determined during these important data. In this era of technology remotely tracking is very critical. To locate solders, for antitheft measures, several tracker systems are available

Main components of this system consists of GPS/GSM and any micro-controllers. Two way communication is possible using the above mentioned components. Here GSM is enabled by a SIM card. It has a regular communication as we are using phone for this communication.[4]

5. **Location determination using GPS/GSM Tracker**

Main part of this structure is RF transmitter. This component has its own identification. Micro—controller is connected with RF module. The information’s will be endlessly being transmitted to the said module. From satellites the GPS will admire the data and further push to the connected microcontroller. In case of theft of vehicle, the RF receiver module stop receiving the info. In this case the microcontroller will generate the message of car theft..

Automatically generated message including information of current position is sent to the possessor of the vehicle. This is done using GSM modem. In caseof vehicle automatically stopping is also highlighted through message. [5]

2.2. Problem formulation

The real time vehicle monitoring system is a new surveillance and remotely vehicle monitoring system. To achieve the monitoring effectively , we select minimum number of sensors and effectively monitoring the vehicle and its speed. To achieve antitheft measures we placed a drivers seat camera to check who is driving the vehicle. All information as output of hardware part are sent to the server/database as a message burst. Server stores and then application manipulate the data to show the desired results. The results are then shown on android application.

CHAPTER 3
DETAILED DESIGN AND FLOW METHODOLOGY

3.1. Detailed Design

The project is divided into three major parts, the hardware, Server and the android application. The hardware consists of Resberry pi (microcontroller) GPS and camera. The application consists of the algorithm for calculating the speed of vehicle in km/hr as well as the code for the GUI made in Python.

The components would send the data as their output , to the server through 3G/4G cloud and the Python IDLE algorithm in application would determine the exact location of vehicle using latitudes and longitudes and other parameters like speed, crew member information etc.

Thus results then displayed on android application.

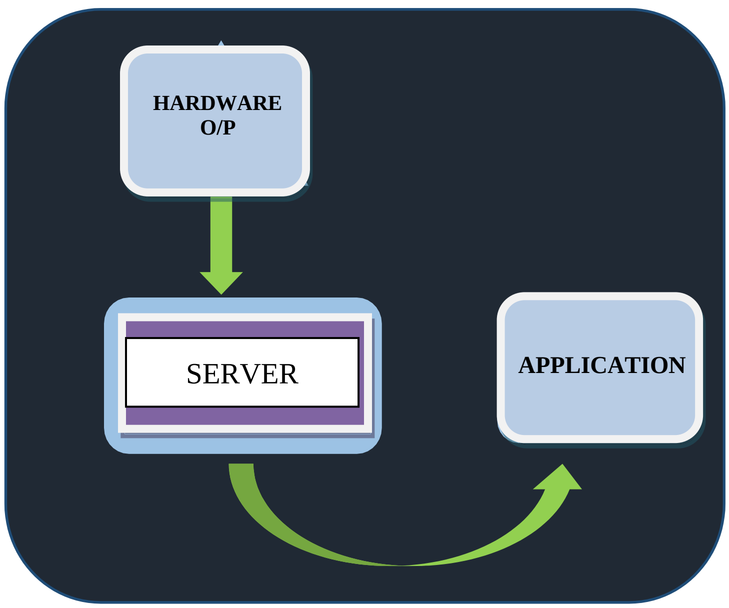

FIGURE:3.1: METHOD FLOW DESIGN

The figure illustrates the flow of the project, i.e. output of hardware component is sent to database as its input. Where it is processed upon. As per the algorithm of android application , data on server is processed and presented in human readable form.

3.2. Hardware design

The hardware part consists mainly of microcontroller, camera and GPS. Camera and GPS is interfaced with microcontroller using open source library codes. The details about each hardware component are illustrated below.

3.2.1. Resberry Pi Zero Wireless

The microcontroller used for our project is RESBERRY PI ZERO WIRELESS. It belongs to the family of Resberry pi one chip computer. Though different microcontrollers are available in market but the specifications and technical support required for completion of our project is only provided in Resberry pi zero wireless.

Techical specifications of RPI are as under

- Supports 02.11 B/G/N wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GHz, single-core CPU
- 512MB RAM
- Mini HDMI and USB On-The-Go ports
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers
- CSI camera connector

FIGURE:3.2: RESBERRY PI ZERO WIRELESS

3.2.2 GPS

GPS is commonly known as Global Position System. It is a satellite based system which provides the exact location of its receiver using some satellites and few of ground stations for navigation purpose. The location is calculated on Earth. It is also known as Navigational system with Time and Ranging which I abbreviated as NAVSTAR GPS.

At least 4 satellites are needed for GPS receiver to accurately determine its location on earth. GPS Receiver do not transmit any information to the satellite. So communication is just in one direction.

Thus, knowing about the importance of GPS receiver, we can say that it is used in many applications like Cabs, Fleet management and smartphones etc.

So, GPS Receiver is main component in this project. Latitudes and longitudes are given by this component ultimately precise location on Google maps. The frequency range for satellite to transmit to the receiver is 1.1 to 1.5 GHz. GPS Receiver if receives signal over said range is capable of computing the exact position on Earth.

NMEA output from GPS receiver contains the visited lat longs of the places andeatch are separated by commas. Each line starts with '$' and ends with carriage return feed sequence.

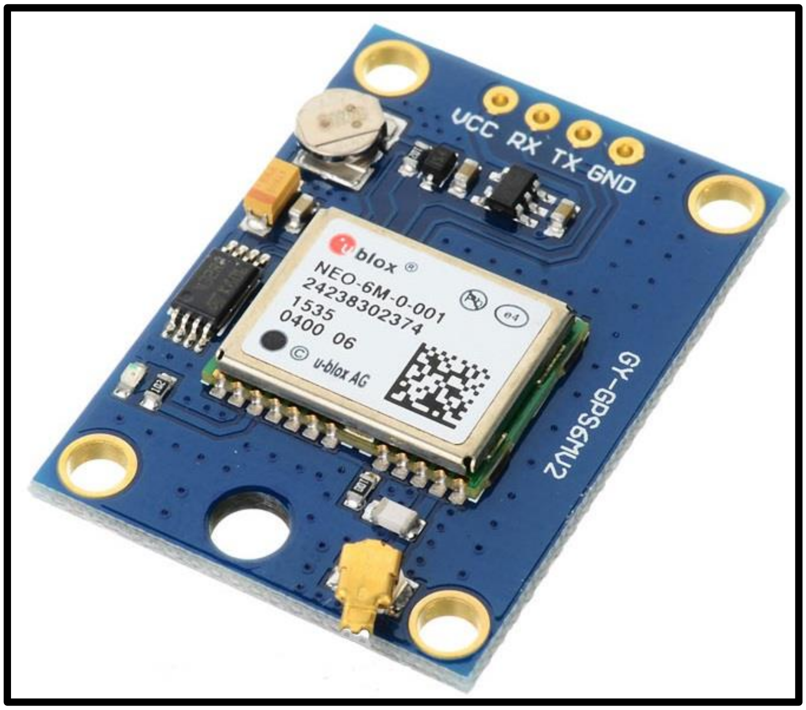

FIGURE:3.3: GPS RECEIVER MODULE

For example

$GPGGA,184237.000,1929.9699,N,07347.6174,E,1,05,2.1,707.1,M,-64.7,M,,0100*7D

$GPGSA,A,3,25,25,18,26,12,,,,,,,,5.3,22.1,4.8*36

$GPGSV,3,1,21,15,42,132,46,25,44,326,45,18,47,239,45,26,34,097,40*72

$GPGSV,3,2,11,12,27,194,45,25,02,174,26,29,68,359,,05,26,034,*7F

3.2.3. Camera:

The Raspberry Pi Camera v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras. The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable. The high quality Sony IMX219 image sensor itself has a native resolution of 8 megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and

640x480p90 video.

FIGURE:3.4: RESBERRY PI CAMERA MODULE

3.3 Software development

Many platforms are available for application development. The platform I selected for the application development is android. Following are main steps to get application integrated with server and get it done working in desired way.

- On start we have map screen with live vehicle tracking and we call this screen Main Activity in Android.

- On Main Activity we have live vehicle tracking on Google Maps and we also have a side menu from which we can go to other screens by selecting the related options from the side menu. We call it navigation drawer in Android in which we can add options as we want. In this app we have map screen option, graph screen option, driver image screen option and profile option in the navigation drawer.

- I'm using native Google Maps for Android in Main Activity to show live vehicle tracking on map. To show live location on Google Map I'm fetching vehicles live data from Firebase Cloud Firestore using Firebase Realtime listeners to get Realtime updates of vehicle location on google map by getting latitude and longitude from the Firebase Listeners.

- When I select graph option from side menu, I'll go to the graph screen on which there is also Firebase Cloud Firestore used for fetching latest speed data from Firestore after every 2 seconds and showing it on the graph. Here I'm using the graph's 3rd party library i.e **MP Android Chart** from which I'm using Bar Chart to display live speed on graph.

- When I select profile option from side menu, I'll move to profile screen on which I see users data that is also coming from Firestore. Here I'm using Firebase Firestore get data listeners that gives us fresh data from Cloud Fire store.

- When I select Driver Image option from the side menu, I'll redirect to driver's image screen on which I fetch data from Firebase storage in which driver's latest image stored. For displaying image from that Firebase Storage URL, I'm using Picasso library for Android that is used to display images from the URLs in Android.

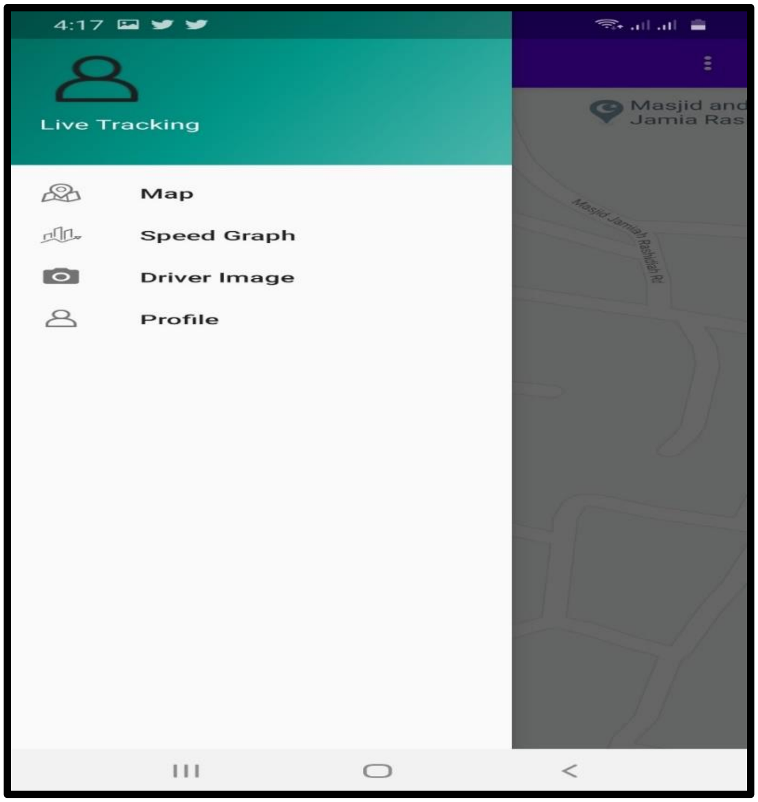

FIGURE:3.5: APPLICATION GUI

The above figure illustrates the following steps followed in the development of the algorithm,

1. Acquire input from Server/Database
2. Check whether vehicle is out or not
3. The selected frames are sent for processing as per algorithm
4. Speed is derived from Latitudes and longitudes in Km/Hr
5. Display required results on application

3.3.1 Application/Codes

. The code designed for the above-mentioned purpose is given below:

```
import time # time library t get milliseconds
import serial # pyserial library
import string # string library
import firebase_admin # firebase admin api library
import numpy as np #numpy library for haversine distance calculation
import os  #os library for executing camera command: raspistill -w 1920 -h 1080 -q 100 -o driver_image.png
from firebase_admin import credentials, firestore, storage # import firestor and storage client library from firebase admin api library
from pynmea import nmea # pynmea library for GPS data conversion
cred= credentials.Certificate('./live-tracking.json') #open firebase credential file
default_app=firebase_admin.initialize_app(cred)
db = firestore.client() #database instance
doc_ref = db.collection(u'Users').document('ecj7KGSiPQARqhZb81nm')#document name
#storage settings
bucket = storage.bucket("live-tracking-278815.appspot.com")#storage bucket name
blob=bucket.blob("driver_image.png") #driver image name that camera stores
imagePath = os.getcwd()+"/driver_image.png" #path in firebase storage

port = "/dev/ttyS0" # the serial port to which the pi is connectedcreate a serial object

ser = serial.Serial(port, baudrate = 9600, timeout = 0.5) #open serial port to read GPS data

def haversine_distance(lat1, lon1, lat2, lon2):
    r = 6371
    phi1 = np.radians(lat1)
    phi2 = np.radians(lat2)
    delta_phi = np.radians(lat2 - lat1)
    delta_lambda = np.radians(lon2 - lon1)
    a = np.sin(delta_phi / 2)**2 + np.cos(phi1) * np.cos(phi2) *   np.sin(delta_lambda / 2)**2
    res = r * (2 * np.arctan2(np.sqrt(a), np.sqrt(1 - a)))
    return np.round(res, 2)
def millis():
    milliseconds = int(round(time.time() * 1000))
    return milliseconds

last=millis()
last_lat=0
last_long=0
while 1:
    try:
    #print("open")
```

```
                if ser.is_open:

                    x = ser.readline()
                    data=x.decode('utf-8',errors="ignore")
                    values=data.split(',')
                    print(data)
                    if(values[0]=="$GPGGA" and values[1]!="" and values[3]!="" ): # the long and lat data are always contained in the GPGGA string of the NMEA data
                        #print("read data success")
                        #values=data.split(',')
                        #print(data)

                        gpgga = nmea.GPGGA()
                        gpgga.parse(data)
                        lats = gpgga.latitude
                        longs = gpgga.longitude

        #parse the latitude and print
                        #latval = msg.lat
                        #concatlat = "lat:" + str(latva)
                        #print lats
                    #parse the longitude and print
                    #longval = msg.lon
                        #concatlong = "long:"+ str(longval)
                        #print longs
                    #convert degrees,decimal minutes to decimal degrees
                        lat1 = (float(lats[2]+lats[3]+lats[4]+lats[5]+lats[6]+lats[7]+lats[8]))/60

                        lat = (float(lats[0]+lats[1])+lat1)
                        long1 = (float(longs[3]+longs[4]+longs[5]+longs[6]+longs[7]+longs[8]+longs[9]))/60
                        long = (float(longs[0]+longs[1]+longs[2])+long1)
                    #calc position
                        pos_y = lat
                        pos_x = long #longitude is negaitve
                    #plot the x and y positions

                    #shows that we are reading through this loop
                        print (pos_y)
                        print (pos_x)

                        location=firestore.GeoPoint(lat, long)


                        if (millis()-last>2500): #calculate distance in 2.5 seconds


                            distance=haversine_distance(last_lat,last_long,lat,long) #distance in 2.5
                            #print("The distance is %.2fkm." % distance)
                            last_lat=lat
                            last_long=long
                            last=millis()
                            speed=(distance*1000)/2.5 #speed in meter per second
                            speed=speed*3.6
                            print("The speed is %.2fkm/h." % speed)
                            doc_ref.update({
            u'Speed': speed,
            })




                        doc_ref.update({
            'Latlng': location,
            })
```

```



                    #time.sleep(1)

                if(doc_ref.get().to_dict()['update_image']):
                    print("update image here and set update image to false")
                    os.system("raspistill -w 1920 -h 1080 -q 100 -o driver_image.png")
                    blob.upload_from_filename(imagePath)
                    doc_ref.update({
        u'update_image': False,
        })
                    ser.flushInput()
        except KeyboardInterrupt:
            ser.close()
            break

```

FIGURE 3.6-3.7: CODE/APPLICATION ALOGORTHM

CHAPTER 4
PROJECT TEST RESULTS AND DETAILED ANALYSIS

4.1. Project Analyses and Evaluation

The project results are illustrated below.

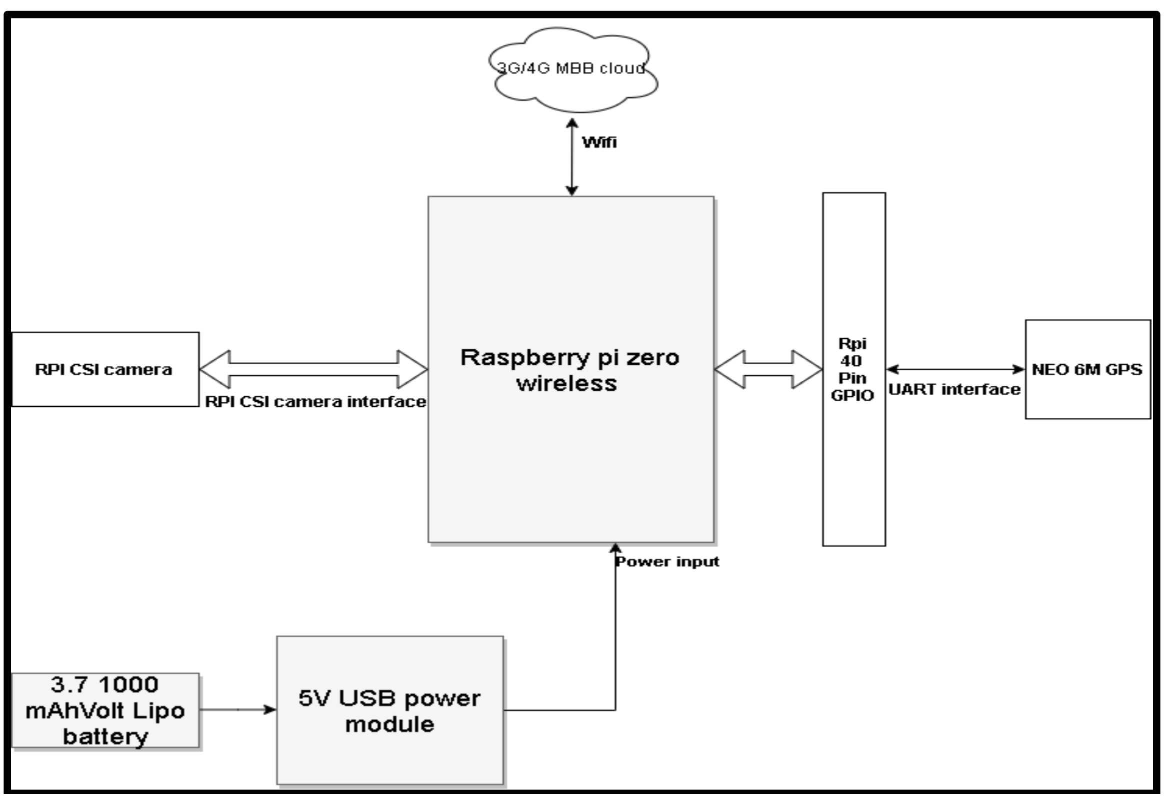

FIGURE:4.1: BLOCK DIAGRAM

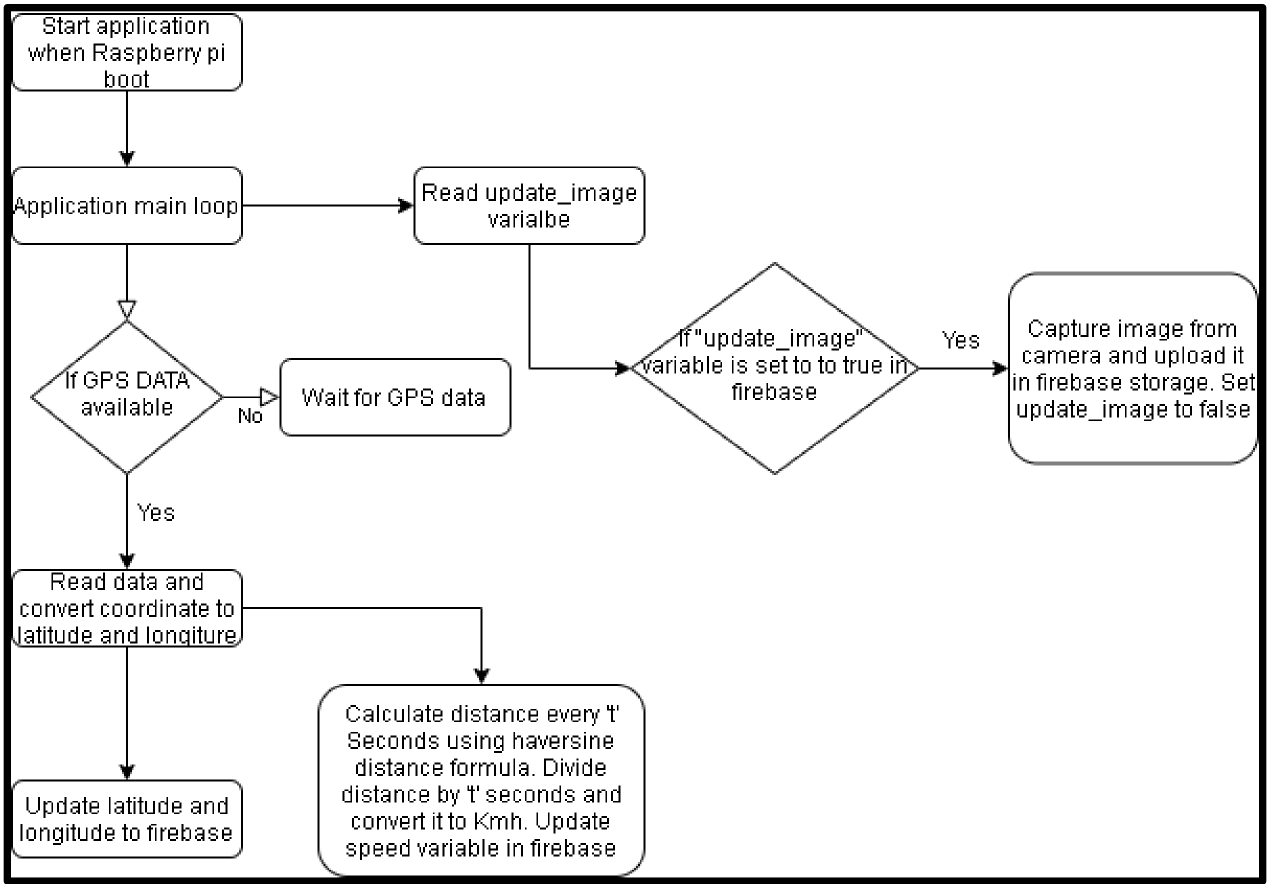

FIGURE:4.2: FLOW CHART

FIGURE:4.3: HARDWARE SETUP

4.2 Acquiring Data from Hardware:

The figure 4.1 shows the data being received by server. It is sent by hardware part of the project. Data is series of lat/longs after every 2.5 seconds. After 2.5 seconds the data is upgraded in the server. After every one minute data old data is replaced by new data.

FIGURE 4.4: LATTITUDES/LONGITUDES

4.3 Database Output:

Fig. shows the processed data as per algorithm of the application. It includes location in terms of latitudes and longitudes, crew member names, emergency contact numbers and the speed of vehicle at some particular time. Emergency numbers and crew information can be seen as well

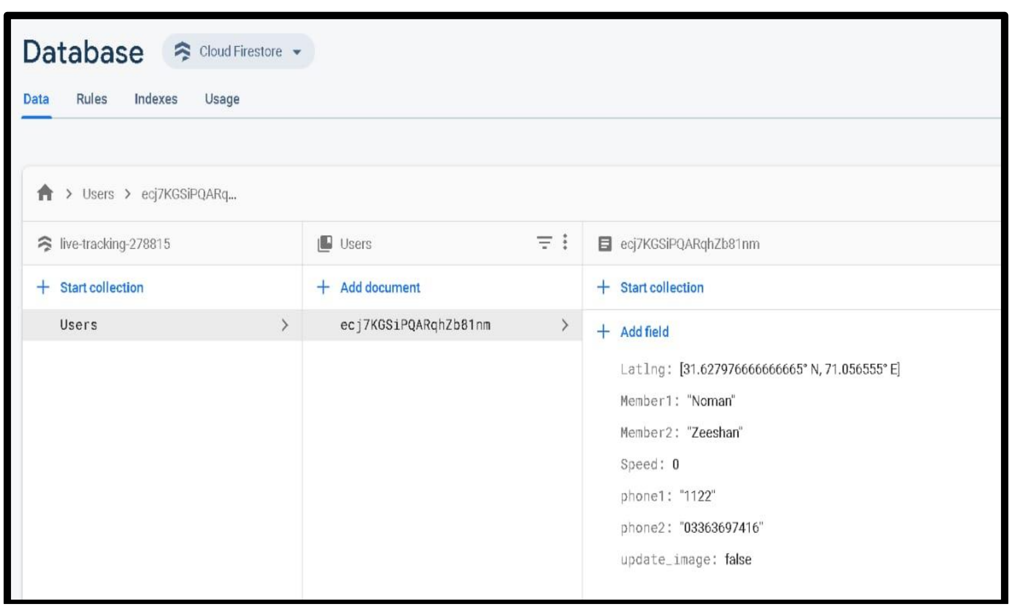

FIGURE 4.5: DATA AT SERVER

4.4 Vehicle Monitoring On Google Maps

The Project Main aim was to monitor automobile like car or truck remotely on google maps. The Lat/Longs or data being received from satellites by GPS receiver are the used to calculate the exact location. This as a burst of information are sent to database. This database updates the location after every 2 seconds. The updating of data can be seen on firebase as Lat/Longs turn red after every two seconds to update the existing Lat/Longs. The application ultimately process the Lat/Longs as per Algorithm and show the Location on Google Maps as test result shown

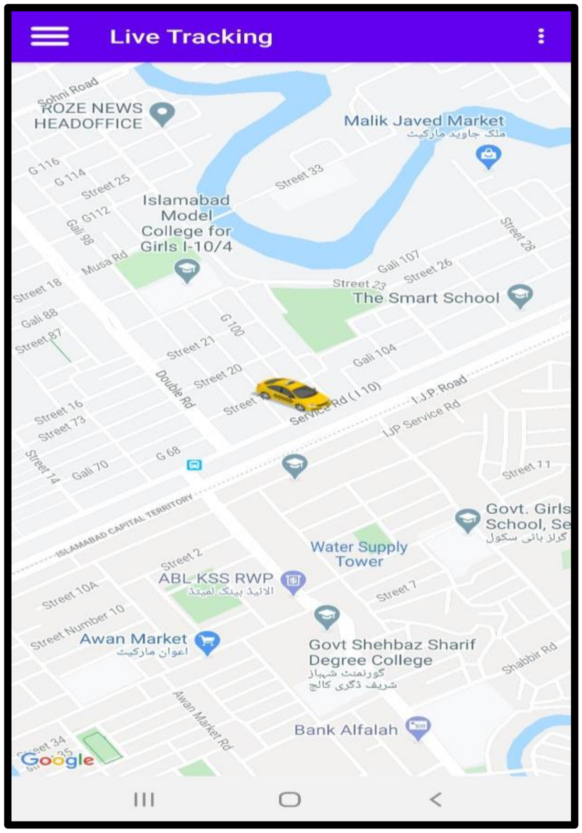

FIGURE 4.6: APPLICATION GUI FOR LIVE LOCATION

4.5 Speed Calculation from Lat/Longs:

Data in terms of Lats/Longs are used to calculate the speed of vehicle. As the Lats/Longs are continually updating after every 2.5 secs. So speed as per algorithm is calculated .

```
        distance=haversine_distance(last_lat,last_long,lat,long) #distance in 2.5
                            #print("The distance is %.2fkm." % distance)
                            last_lat=lat
                            last_long=long
                            last=millis()
                            speed=(distance*1000)/2.5 #speed in meter per second
                            speed=speed*3.6
                            print("The speed is %.2fkm/h." % speed)
                            doc_ref.update({
        u'Speed': speed,
```

FIGURE:4.7: SPEED CALCULATION CODE

4.6 Camera Test Result:

Resberry Pi Camera v2 and high quality 8Megapixel was integrated with Microcontroller Resberry Pi. It will be used for viewing diver's image at application end. Request is initiated by the application by clicking on view image. This command is then processed by resberry pi and picture is captured at that point of time. Then this image is sent to the server from where it can be viewed from application GUI.

Test result of integrated camera is as under

FIGURE:4.8: IMAGE CAPTURED BY CAMERA

CHAPTER 5
RECOMMENDATIONS

5.1 Recommendations and future work

1. The project has been employed on simple sensors, it may be gradually designed with different components like OBD2 scanner, Accident alert , and Lane keeping sensor , to make it more applicable and practical.
2. Camera and GPS used are basic components with inbuilt in accuracy, these may be replaced with the more modern and advanced components, to improve accuracy and sensitivity.
3. Advanced camera may be used to monitor the entire vehicle instead of simple driver's image.
4. OBD 2 scanner can be used for real time speed ,fuel and performance of vehicle.
5. Multiuser application can make this project be used in conveys
6. Implementing 3G/4G module may enhance the efficiency of project. Offline tracking would be possible
7. Ignition switch can be controlled to shut down vehicle at required point of time

5.2 Conclusion

1. The real time vehicle monitoring system guarantees the owner remotely about the timely and exact location of vehicle along with its speed and driver image .
2. Helping to know about drivers behavior at application end.
3. Hardware part of project can be hidden in vehicle which may not be visible to anyone.
4. In case of emergency inbuilt contact numbers can be contacted for early response like 1122, 15 etc.
5. Crew members information are visible at application end.

5.3 Drawbacks

1. GSM technology is an old and do not support the required data rate.
2. Existing GPS technology gives location in 30m of exact location. It is serious draw back for real time monitoring
3. Few of the components are interfaced which may not give full sense of security to the owner.

5.4 Advantages

1. Comprehensively developed the vehicle monitoring system with effective use of different sensors.
2. The project can be used to monitor vehicle in real time
3. Th project displays the real time location of vehicle.
4. It is an active technique of monitoring..

References

[1]https://www.picprojects.info/content/projects/9-vehicle-tracking-system-using-gps-gsm.html

[2] Fleischer, Paul Benjamin; Nelson, Atso Yao; Bremag, Appah. Design and development of GPS/GSM Based vehicle tracking and alert system for commercial inter-city buses.

[3] Muruganandham and P.RMukesh (2010) "Real time Web based vehicle tracking using GPS" World academy of science, Engineering and Technology.

[4] Vehicle Tracking Bangalore, http://www.simbaprojects.org/download/india/present aion%20and%20feedback/ITS/IIT%Bangalore.

[5] Abid Khan & Ravi Mishra ―GPS – GSM Based Tracking System‖ International Journal of Engineering Trends and Technology- Volume 3 Issue 2- 2012 ISSN: 2231-5381.

[6]http://www.engineersgarage.com/arm-projects/introduction-to-arm-microcontroller-raspberrypi

[7]https://en.wikipedia.org/wiki/Temperature-sensor

[8]https://en.wikipedia.org/wiki/Gas-sensor

[9]https://en.wikipedia.org/wiki/Camera

[10] System," in Proc. International Journal of Engineering Trends and Technology- Volume3Issue2-2011,p.161-164.

Appendix A

Vehicle Monitoring System

Extended Title: Real time GSM-GPS Based Vehicle Monitoring Systems
Brief Description of VMS / Thesis with Salient Specs: The project aims at monitoring vehicle in real time, for obtaining live location, speed and drivers image at the android application The Lats/Longs of location, drivers image etc are made in a message burst by the microcontroller. Then this message burst is sent to the Database through 3G/4G cloud. Interfacing of different components are done using open source libraries. The algorithm of android application the processes the data at server. Displays the exact location, driver image and speed of vehicle at its GUI.
Scope of Work The project will render the detection of Live location, easier. Components used will make the project work effectively. Camera being used to display the driver image. Scope of the project includes the live monitoring of vehicle and enhanced sense of security for the owner
Academic Objectives The project will impart following skills into the syndicate: 1. Mobile communication techniques 2. Python language learning 3. Electronic Circuit Design 4. Data structures.

Project Application/ Objectives:

The objective of methodology is to provide an accurate mechanism for monitoring and tracking vehicle on road using GPS, Camera and microcontroller. This is online monitoring. This project has vast application for monitoring of vehicles and enhancing sense of security. Data available in database can be used for antitheft measures.

Our objective is to demonstrate the real time vehicle monitoring system and its application by integrating GPS, camera etc for said objective.

Previous Work Done on The Subject:

1. http://www.simbaprojects.org/download/india/present aion%20and%20feedback/ITS/IIT%Bangalore.
2. System," in Proc. International Journal of Engineering Trends and Technology- Volume3Issue2-2011,p.161-164.

Material Resources Required:

1. GPS
2. Resberry Pi Zero W
3. Resbery Pi Camera
4. 3G/4G Cloud
5. 3.7v 10000mAh Battery with dual USB module
6. Wires
7. Misc

No of Students Required: 4

Special Skills Required:

1. Mobile communication techniques
2. Python language learning
3. Electronic Circuit Design
4. Data structures.

Appendix B

Timeline

<table>
<tr><th></th><th>Nov 2019</th><th>Dec 2019</th><th>Jan 2020</th><th>Apr 2020</th><th>May 2020</th><th>Jun 2020</th><th>Jul 2020</th></tr>
<tr><td>Literature Study</td><td colspan="2">Completed</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Design And Procurement of Hardware</td><td></td><td colspan="3">Completed</td><td></td><td></td><td></td></tr>
<tr><td>Python Code</td><td></td><td></td><td colspan="3">Completed</td><td></td><td></td></tr>
<tr><td>Interfacing</td><td></td><td></td><td></td><td colspan="2">Completed</td><td></td><td></td></tr>
<tr><td>Android Application</td><td></td><td></td><td></td><td></td><td colspan="2">Completed</td><td></td></tr>
<tr><td>Finalization</td><td></td><td></td><td></td><td></td><td></td><td colspan="2">Completed</td></tr>
<tr><td>Testing</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Completed</td></tr>
</table>