# Semi Automted IoT Vehicles

S B Rudraswamy
Dept of ECE, SJCE
Mysuru, India

Ashruth Burli
Dept of ECE, SJCE
Mysuru, India

Ananya M B
Dept of ECE, SJCE
Mysuru, India

Nisarga N C
Dept of ECE, SJCE
Mysuru, India

Vardhini V Hebballi
Dept of ECE, SJCE
Mysuru, India

***Abstract*—Developing countries face the problem of crowded and congested roads because of inefficient implementation of traffic rules.There is a lack of awareness about the rules to the commutator. In addition, Drivers also tend to ignore the rules often. An intelligent traffic system that can automatically implement the traffic and road safety rules to the IOT based cars is proposed in this paper. The system proposed first identifies all the road safety rules to be implemented in the current location, followed by alerting to the driver. In case of no response by the driver in the way of following the rule, an automated imposition of rules is done. The recognition of the rules is done by sensing the current location through GPS and searching the Virtual Traffic Management System for all the rules that are applicable in that region which are predefined. Upon having known all the rules to be followed, the driver is alerted about the rules. Following which, the control signals are sent to control system to orient the functioning along the lines of following the traffic rules thereby automating the response directly by the vehicles.**

***Keywords—Intelligent transportation system, Web services, Speed limit, Road safety rules, IoT, Haversine algorithm, MySQL, Cloud service, Global Positioning System (GPS), Raspberry Pi.***

## I. INTRODUCTION

Traffic and road safety rules have been developed to promote traffic safety and efficiency.Developing countries face the problem of crowded and congested roads because of the inefficient implementation of traffic rules. There is a lack of awareness about the rules to the commutator. In addition,

Drivers also tend to ignore the rules often. Also, it is particularly challenging to new drivers operating on unfamiliar roads to follow rules. This is one of the main reasons for accidents.Hence, it is necessary to develop a

system that assists the driver about the traffic or road safety rules while steering and to automate the response of the automobile upon the driver's ignorance to follow the rules. The major concern of traffic sense system is to ensure road safety to avoid accidents due to driver's negligence resulting in losses of lives. Thus, an intelligent traffic system that is proposed is advantageous because of its unique features and capabilities of the Internet of Things (IoT).

### *A. Problem statement*

The problem this project intends to solve is to overcome all the hassles caused by either lack of awareness of the driver towards the road safety rules or their deliberate negligence towards it. The problem statement is to propose an intelligent traffic system that first identifies the traffic rules ahead from the current location of the vehicle to alert the driver about the road safety rules and upon driver's ignorance to follow the rules, to automate the response of the automobile.

### *B. Literature survey*

Jessen Joseph Leo uses GPS technology to find the location of the vehicle with respect to the hairpin bend to decide the priority in which vehicles have to move in[1], but this system puts no effort in solving the problems of traffic congestion in other areas. A model by Benoit Vanholmeprovides real-time advisory alerts to the driver when approaching mapped points of interest such as speed-limited zones, intersections and speed bumps. Here determination of driver's approaching POIs is done using the K-Nearest Neighboralgorithm in [2], but the K- nearest neighboring algorithm requires to calculate a parameter K that is difficult. The legal safety rules for the autonomous vehicle such as distance keeping, intelligent speed adaptation, and lane changing are implemented using cameras and radar in [3]. An android-based application collects data from a vehicle such as latitude, longitude, speed, and accelerometer data and sends it to its nearest fog server for processing. Algorithms were developed which find the location of road anomalies and accident-prone area. The result of this is shown in Google map of driver's smartphone in [4].The concept of IoT and Cloud based technology is used in car parking services in [5], but this only identifies whether the given zone is a parking or not and no other rules are considered. In [6], RFID technology is used for guiding the emergency vehicles with reduced congestion path. The IoT based car by SaliqAfaqueis remotely controlled through internet by a web browser. The RPi camera captures the live video and RPi transmits this video for live processing over the web browser in [7],but only the forward movement of the car is taken care and the dark vision(i.e., night time) using RPi camera is difficult.In [8], automated vehicles communicates with other automated vehicles using the pure ALOHA system. In [9], the vehicle is embedded with the GPS, accelerometer sensors, crash sensor vehicle chip and sim module for communication. The information collected using these sensors is updated to the cloud at definite intervals. In [10], the system designed uses GMM/GMR acceleration model in the curve entry section and the curve exit section and the velocity tracking algorithm in the mid curve. The GMM/GMR method considers local automatic driving behavior based on time series information. Eye blink sensors and over speed controller sensors are used for speed control mechanism in [11], but detection is not early and blink sensors are not very precise which can lead to accident. In [12], the system provides priorities to the vehicles at the curves and controls the speed based on the type of the bend and the amount of bend to be taken. The location of the vehicle is matched with the maps to find if the vehicle is nearing the bend. In [13], a mobile application is developed which is connected to the cloud designed for the user to know about the availability of parking spaces, but has ambiguity in cloud computing when many users uses the application at a time. The system which analyses the driver's curve driving data and considers the road construction by JianmingXie can be used for road condition management [14]. In [15], a smart car is structured with a DC motor for running driver and a steer motor for turning control, which can run along a black line at a high speed. A control strategy for smart car turning is based on the road visual, through the image of road, the smart car can track along the road and the running speed is adjusted according to the road shape. Here, the control strategy based on road shape is established mainly depending on image process which is not always efficient.In [16 and 17], the system designed enables the user to preserve parking slot from remote place with the help of mobile application. Authentication of the valid booking is incorporated to benefit valid user. This system is implemented using low-cost IR sensors, Raspberry-Pi model 3b for real-time data collection and E-Parking mobile application.

### C. *Objective*

The main objectives of this project is to develop a real time driver assistance system which assists the drivers by informing about the rules ahead and in case of no response by the driver,the system actuatesthe response.The system can performs the following functions:

• Control the speed of the vehicle in accordance with the speed limit by using DC geared motor in the speed limit zone.

• Notify the driver in case of speed breaker ahead.

• Disable the horn system of the vehicle in case of no horn area.

• Notify the driver in case of no parking area.

• Notify the driver about other signboard rules ahead such as school zone, one-way, deep left/right curvatures etc.

## II. PROPOSED SYSTEM

The proposed model centers on the elements of automated vehicles, which comprises of automating the vehicle based on the traffic rules and regulations when the vehicle approaches a point where the rules are defined. This defines how the flow of processes in the working of the model is designed starting from the user inputs to all the computations in the local device flowed by storage in the database present in the server which is hosted in the cloud and to the response from the system back to the user. The model comprises of 2 modules namely, the Policy maker's site and the Automobilists' site.

### A. *Policy maker's site*

This is the interface for the policy makers (rule makers) which will be the part of road safety department to formulate and deploy the set of rules corresponding to geographical locations. A GUI is designed for road safety department to input and update road safety rules by policy makers. The road rules are recorded in terms of its geographic location i.e., its latitude point and longitude point, type of the rule, for how much distance the rule is to be implemented from the initial coordinates. This data is used to create database of road safety rules of a particular geographic location using MySQL. MySQL database of road safety rules is uploaded to the server in cloud which forms the Virtual Traffic Management System. The policy maker is authenticated with login credentials to access the Virtual Traffic Management System. The policy maker is provided with privileges to certain functions like add new rules with new coordinates, define new rules at already present coordinates, define new coordinates for already present rules and delete the existing rules.

The flowchart depicted in figure 1 provides the detailed working of the policy maker's side. The Graphic User Interface provide the authorised traffic management unit, a web page access to perform the above-mentioned privileges.

In figure 2, the flowof how the policy makers and the user site is built and how the rules are stored in the database is explained.A process flowis a list of event steps which defines theinteractions between a role and a system, to achieve a goal.The actor can be a human or other external system.

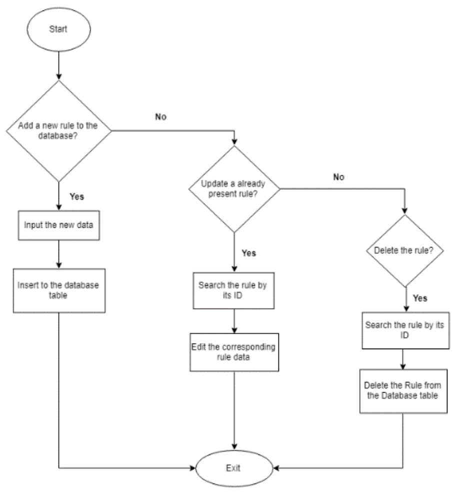


Figure 1: Flowchart for policy maker's site of the model

Use case analysis is an important and valuable requirement analysis technique that has been widely used in modern software engineering. The actor or admin (Policy maker) logins to the webpage where he can enter the details of the rules to the cloud. These entered values are stored in the cloud and fetched whenever it is needed. The stored values are speed limited values, parking mode area details, hump area details, no horn zones at a particular location. Only the authenticated policy maker has control over speed limitations location, no horn zones, parking zones and on the data entry of the location in the database.

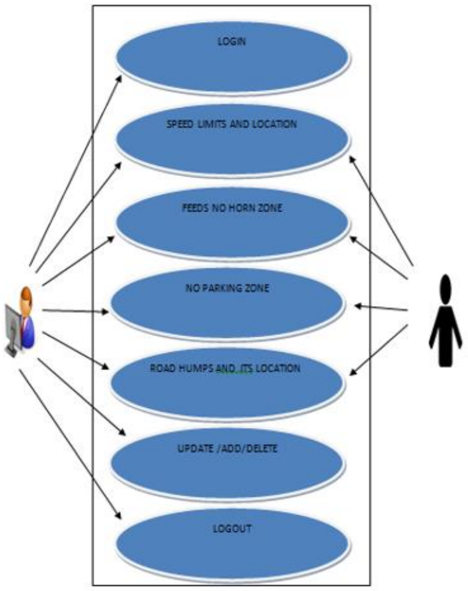


Figure 2: Use Case for the model

B. *Automobilists' site*

This module is present at the user end providing the users with option to switch between the two modes of operation i.e., the Driving mode and the Parking mode. A Touch screen display is provided to the vehicle driver to switch between the two modes. The flowchart depicted in figure 3 provides the detailed working at the Automobilists' site.

- Driving Mode

When the driver switches to driving mode, GPS module senses the live location of the vehicle and the live location coordinates are sent to Raspberry Pi dynamically. Raspberry Pi is the processor for the model where the cloud access program runs to access the Virtual Traffic Management System for all the traffic rules that are applicable in that area which are predefined.

Using the HAVERSINE distance algorithm, an additional column is created in the table which holds the distance between the present coordinates of the vehicle and coordinates of the corresponding rule in the virtual traffic management system. The implementation of Haversine distance algorithm is explained further in this chapter. The nearest five rules are retrieved from the cloud and are processed. Upon having known about the nearest rule to be followed, the driver is alerted about the rule through a speech or/and text on touch screen display. In addition, the control signals are sent to the Raspberry Pi, which in turn sends control signals to the interface drives to automate the nearest rule on the vehicle along the lines of traffic rules.

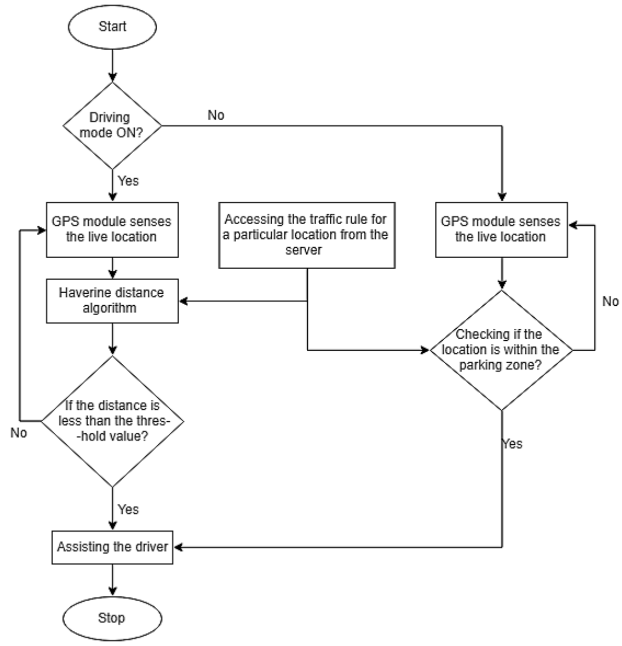


Figure 3: Flowchart for automobilists' site of the model

- Parking mode

When the driver wishes to park the vehicle, the driver has to switch to parking mode using the Touch screen display provided. The live location coordinates of the vehicle sensed by GPS module is compared with the parking zone coordinates predefined in the Virtual traffic management system by Raspberry Pi to indicate whether the location of car is in parking zone or not.

HAVERSINE ALGORITHM

The haversine formula is used to find the great-circle distance (arc length) between two points of interest on a sphere whose longitudes and latitudes values are known.Let the Θ be the central angle between any two points of interest on a sphere. It is given by,

$$\theta = \frac{d}{r}$$

where, *d* is the distance between the two points on sphere and *r*is the radius of the sphere i.e. Earth. The *haversine formula* hav(Θ) is given by:

$$hav(\theta) = hav(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\, hav(\lambda_2 - \lambda_1)$$

where $\varphi_1, \varphi_2$ are the latitude of point 1 and latitude of point 2 respectively and $\lambda_1, \lambda_2$ are the longitude of point 1 and longitude of point 2 respectively. The haversine function of an angle $\theta$ is given by:

$$hav(\theta) = \sin^2\frac{\theta}{2} = \frac{1-\cos\theta}{2}$$

To get the distance *d*, the arc haversine (inverse haversine) is applied to the central angle Θ.

$$d = r\, archav(h) = 2r \arcsin(\sqrt{h})$$

where *h* = hav(Θ). Also,

$$d = 2r \arcsin(\sqrt{hav(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\, hav(\lambda_2 - \lambda_1)}\,)$$

Thus, the required value of d is obtained using Haversine formula.
A data flow diagram (DFD) i.e., a graphical representation of the flow of data in the processes involved in modelling aspects of the policy maker's site is depicted in the figure 4.

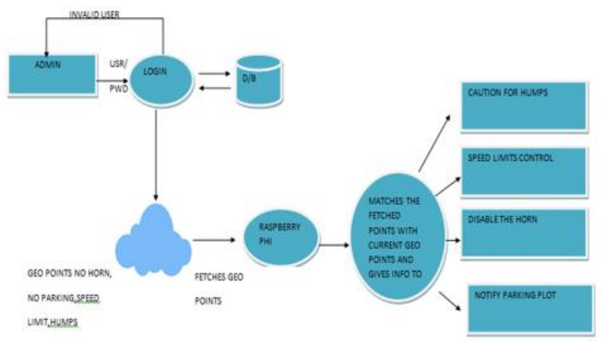


Figure 4: Data Flow Diagram for the model

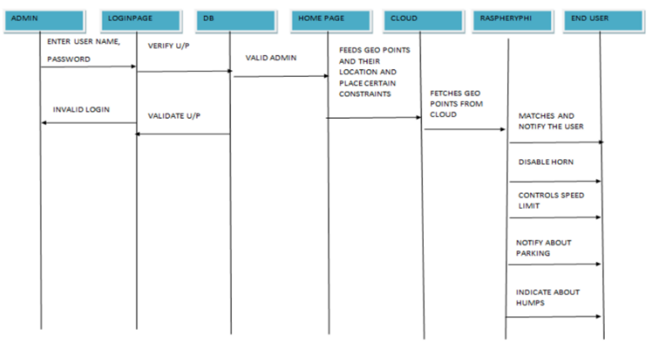


Figure 5: Sequence Diagram for the model

The sequence diagram is depicted in figure 5 it shows the sequence of tasks that should occur right from login to registering and adding, modifying or deleting the rules.

## III. IMPLEMENTATION

This section discusses how the designed model and process flow is actually implemented. This shows the actual GUI designed on both policy maker and automobilist'side (displayed on the Touch Screen) and howthe software and hardware part are integrated and implemented. The implementation also has 2 parts:

### A. *Policy makers site*

This is the interface for the policy maker's which will be the road safety department to formulate and deploy the set of rules corresponding to geographical locations.

A web application is designed for road safety department so that policy makers input and update road safety rules. The web application is designed using ASP.NET framework of Visual Studio tool.

Figure 6 shows the index page of the application. Index page is loaded with necessary static content like type of signs and about the project as shown in figure 7.


Figure 6: Welcome Page of Policy Maker Site


Figure 7: About the project page

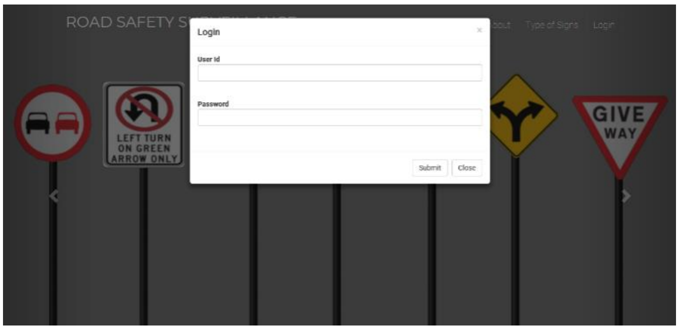


Figure 8: Login Page of Policy Maker Site to allow only authentic users to change the policies

On hitting login button of the index page, login page pops up as shown in figure 8. By entering valid login credentials in the appropriate textboxes and hitting submit button, policy makers are authenticated which directs to the Admin page. Admin master page has following pages

1. 'Latitude & Longitude' page
2. 'View lat&lon' page

Figure 9 shows the 'Latitude & Longitude' page to add rules at given coordinate. The Latitude and longitude co-ordinates should be entered and the type of rule at that point has to be chosenamong the various rules using the drop down menu under category. Also, value for that rule (say 40kmph for the speed limit) should be specified accordingly. Submit button inserts the rule details entered to the database table.

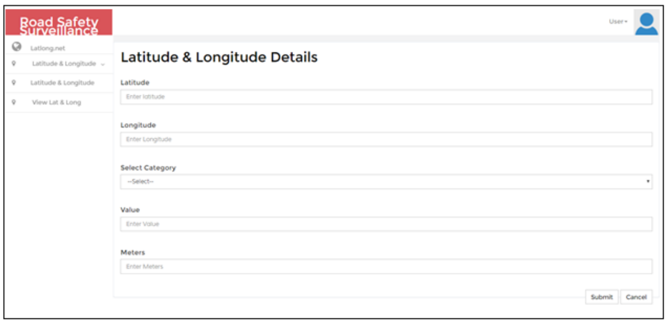


Figure 9: GUI at Policy Maker site to add rules at any co-ordinates

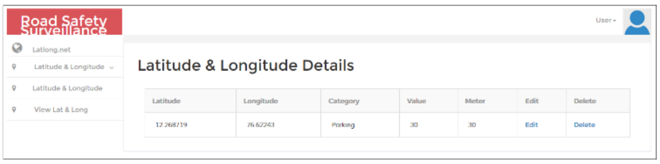


Figure 10: Detail of rules at any co-ordinates

Figure 10 shows 'View lat&lon' page designed to display the database table contents with edit and delete options for each entry. Edit button redirects to 'Latitude & Longitude' page where changes can be made. Update button updates the database table for that corresponding entry.

B. *Automobilists' site*

In this module, the hardware is resemblance of the vehicle whose response towards following the traffic rules is being controlled. The prototype of the proposed model consists of Raspberry pi which acts as a controller, a GPS module to obtain live location of the vehicle, a touch screen display which mimics the infotainment display available in the real time vehicles as in figure 11 and figure 12.

A user interface is provided for the vehicle driver over a Touch Screen display for the user to switch between the driving mode and the parking mode, which also displays the live location coordinates, the rule that is currently applied on the vehicle, the live location address and the total distance covered from the start of the model. The GUI also contains text box in which the live location i.e., the latitude and longitude values are provided. The touch screen display that had been used in the prototype is Raspberry Pi compatible 7-inch HDMI touch screen.

Raspberry Pi model 3 is used in the prototype which controls the response of the model and also processes the rules being fetched from the database. It also processes the latitude and longitude being obtained from the GPS Module.

A Java code is continuously run in Raspberry Pi which has the following features. The code consists of GUI design script, Raspberry Pi GPIO pin declaration, web services (to access the database), functions to trigger control signals to the output drives.

A constructor is executed as soon as the object of the class is created. Within the constructor, the text boxes in the GUI are filled with the live location that is received by the GPS module. Also, the horn switch and the stop button are enabled.

There two threads that run in the background to continuously update the location and to display the rules in the GUI. These threads are enables one the start button on the GUI is enabled.

When the start button is enabled, the motor (resemblance to the vehicle wheel) starts to rotate at a initially specified speed (40 kmph as used the prototype) and the first thread i.e., to update the location is continuously is created and is started which executes a function that has an infinite loop to get the GPS location of the vehicle and displays it in the GUI and these coordinates are reverse geocoded to get the current address which is also displayed in the GUI. And also, retrieves the rules for this location using the web services. All the rules that are retrieved are at a distance of 10 km (in the prototype) from the current location.

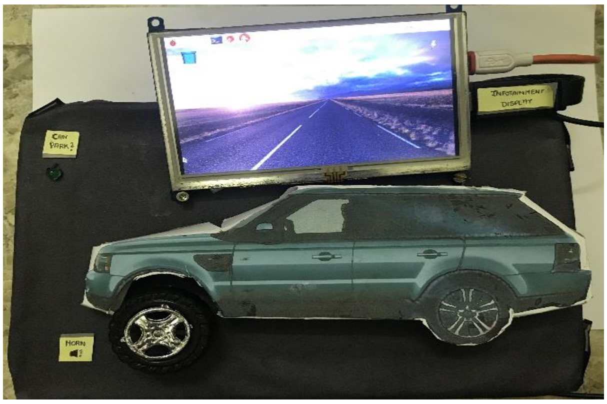

Figure 11: Complete prototype setup

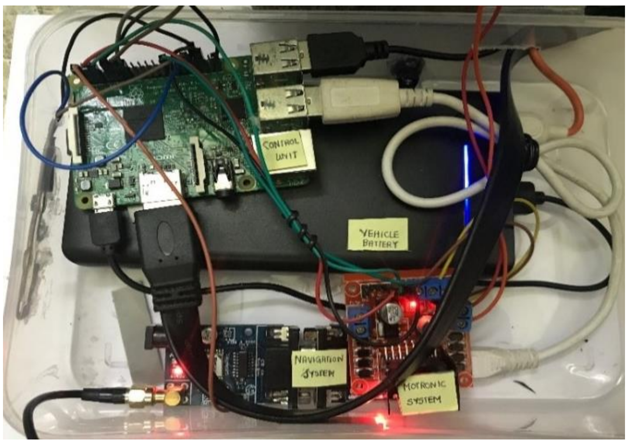

Figure 12: Components of prototype setup

All the rules that are obtained are stored in an array-list. The nearest among these rules is found out using the Haversine formula and the rule that is at a distance less the 20 meters is applied on the vehicle. The other thread that is run to display the rule that is enacted on the vehicle in the GUI. The rule to be applied is processed and found out if the rule is speed limit, no horn or a no parking zone.

If the rule category is speed limit, the speed to which the vehicle has to change is scaled down by a factor 10. The scaled values of the speed ranges from 1 and 10. Accordingly, the PWM signal is sent to the motor driver which drives the motor in the specified speed. The change in speed is gradual. And also, the message for example, “There is a speed limit of 40 km/hr in this area” is displayed in the GUI.

If the rule category is parking, a message “You are currently in the parking zone” is displayed in the GUI. And checks if the switch provided to change to parking zone is enabled. If its enabled and the rule category is parking, the green LED is turned on indicating that the vehicle is in the parking zone and vice-versa.

If the rule category is no horn, the buzzer (resemblance to the horn in the vehicle) is disabled i.e., even when the horn button in the GUI is clicked, the buzzer is disabled. And also, the message “This is strictly a No Horn zone” is displayed in the GUI. The hardware prototype also contains power supply to power up the R-pi and the motor driver.

## IV. RESULTS

The model proposed was designed and implemented and was tested for all the test cases. A miniature vehicle was developed to perform the analysis. A set of rules were deployed in the database for different locations and was tested for the rules defined above namely speed limit, no horn and parking.

The results obtained were approximately as expected and were optimum. The rule access from the server and the processing time to change of rule totally took approximately 5 seconds with strong internet connection.

In regions where no speed limit is provided the speed of the motor (vehicle) changes in accordance with the input from the driver. It was seen that the motor run at full speed in these regions when user increased the speed. In the display no rules were displayed as those particular regions had no rules defined. In regions where speed limit is defined the speed of the motor only increased/decreased up to the specific rpm corresponding to the particular speed. Despite user trying to increase the speed beyond the given limit, the speed of the motor did not increase. The speed control in the regions defined with speed limit was tested.

In zones which are not defined as Parking zone, if we try to park (i.e., if the driver tries to switch from driving mode to parking mode) the green LED did not glow indicating the user that he cannot park in that zone. In the zones where it is defined as parking zone the user can successfully switch from driving mode to parking mode and the led will glow indicating that the automobilist can park in that area.


Figure 13: Screenshot of the touchscreen display

When the testing was done and executed the GUI in the touch screen showed latitude, longitude of the current location of the moving vehicle, distance covered by the vehicle, address of that location and also the rule that is fetched from the cloud and is executed as in the depicted in the figure 13.

## V. CONCLUSION

This system presents a real time road safety monitoring to solve the problems caused due to breaking of rules. The proposed system provides a new way to control traffic rules and a step towards autonomous vehicles which follows the traffic rules. The traffic administration department (Traffic Rules Department of Government) can use this real time road safety monitoring system to detect dangerous situations on the road and thereby react by imposing immediate actions. The data that are stored in database which is present in the cloud are all verified and validated. Automating the traffic rules over a vehicle will thereby cease the driver from violating the traffic rules. The system also helps to limit the speed of the vehicle thus by reducing the risk of accidents.

On the whole IOT along with our system, Traffic Sense will play an important role in road safety monitoring system in real time scenarios. On including many other features like traffic signals, inter-vehicle communication, etc an advanced vehicle can be designed.

## REFERENCES

[1] Vehicle movement control and accident avoidance in hilly track, by Jessen Joseph Leo, R.Monisha, B.T.Tharani Sri Sakthi and A. John Clement Sunder, International Conference on Electronics and Communication System JCECS -2014

[2] Highly Automated Driving on Highways Based on Legal Safety by Benoit Vanholme, Dominique Gruyer, Member, IEEE, Benoit Lusetti, S´ebastien Glaser, and Sa¨ıd Mammar, Senior Member, IEEE , IEEE transactions on intelligent transportation systems, vol. 14, no. 1, march 2013

[3] Cooperative Driving of Automated Vehicles with Inter-vehicle Communications by Takeshi Sakaguchi, Atsuya Uno, Shin kato and Sadayunki Tsugawa, IEEE Intelligent Vehicles Symposium 2000 Dearborn (MI), USA October 3-5,2000

[4] Vehicle Control Algorithms for Cooperative Driving With Automated Vehicles and Intervehicle Communications by Shin Kato, Sadayuki Tsugawa, Member, IEEE, Kiyohito Tokuda, Member, IEEE, Takeshi Matsui, and Haruki Fujii, IEEE transactions on intelligent transportation systems, vol. 3, no. 3, September 2002

[5] Real-time Driver Advisory Model: Intelligent Transportation Systems by James OBUHUMA, Henry OKOYO and Sylvester MCOYOWO,

IIMC International Information Management Corporation, 2018 ISBN: 978-1-905824-60-1

[6] Smart Vehicles with Everything, by R.Srinivasan, A.Sharmili, Dr.S.Saravanan and D.Jayaprakas, 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)

[7] IoT-Driven Road Safety System Dasari Vishal , H. Saliq Afaque , Harsh Bhardawaj and T. K. Ramesh by Department of Electronics and Communication Engineering Amrita University, Bengaluru at 2017 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)

[8] Design of an Iot based autonomous vehicle with the aid of Computer Vision, by Mohammad Rubaiyat Tanvir Hossaim, Md. Asif Shahjalal and Nowroz Farhan Nur, International Conference on Electrical, Computer and Communication Engineering (ECCE), February 16-18, 2017, Cox's Bazar, Bangladesh

[9] Safe driving using IoT sensor system by A. Jesudoss, Muthuram .B.O and Lourdson Emmanuel .A , School of Computing Sathyabama Institute of Science and Technology at International Journal of Pure and Applied Mathematics , in Volume 118 No. 20 2018, 3745-3750.

[10] An Approach to IoT based Car Parking and Reservation system on Cloud by Vaibhav Hans, Parminder Singh Sethi and Jatin Kinra, Centre of information Technology University of Petroleum and Energy Studies Dehradun, India at 2015 International Conference on Green Computing and Internet of Things (ICGCIoT)

[11] Titus, Joby and S Sanjana, V and B, Saranya and Manikandan, Pravin and Professor, Assistant. (2017) at Automatic Road Surveillance to Implement Vehicle Traffic Rule. International Journal of Scientific Research. 4. 87-90.

[12] "Embedded controller for vehicle In-Front obstacle detection and cabin safety alert system" by V.Ramya, B.Palaniappan and K.Karthick, at International Journal of Computer Science and Information Technology, vol.4, No.2, April 2012. [2] V.Ramya, B.Palani

[13] IoT based smart and safe driving applications, by Dr. Thyagaraju G S, HMT Gadiyar and U B Sujit.

[14] A Personalized Curve Driving Model for Intelligent Vehicle, by Jianming Xie, Jianwei Gong ,Shaobin Wu, Guangming Xiong, and Chao Lu at IEEE conference 2017

[15] Control Strategy Design for Smart Car Auto-tracing with Visual by Hui Zhang and Yongxin Liu , College of Electronic Information Engineering Inner Mongolia University, Hohhot, 010021, China at 2014 26th Chinese Control and Decision Conference (CCDC)

[16] IoT Based Sensor Enabled Smart Car Parking for Advanced Driver Assistance System by Mahendra B M, Dr Savita Sonoli and Nagaraj Bhat, RV College of Engineering.

[17] Monitoring And Reporting Of Traffic Rules Violation Using Microcontroller Through Wireless Communication System by Hossain, Mohammmad Bhuiya, Mu and Ahamed, Jamal Uddin and Bhuiyan, Tanvir and Bhowmik, Suman. (2018).