On Board Diagnostics System for Vehicle Monitoring and Tracking

[1]T. C. Jermin Jeaunita, [2]Sarasvathi V, [3]N. Pooja Hipparagi

[1,2,3]Department of Computer Science and Engineering
PESIT Bangalore South Campus, Bangalore, India

Abstract : Vehicle monitoring system is essentially required for the safety of lives as well as vehicles. It is required to monitor the vehicle's health for its longer life. Monitoring how, when and where one's vehicle is being driven helps in various real time applications. To do so, On Board Diagnostics (OBD) System is a sole technology for vehicle monitoring. This paper presents vehicle monitoring system which makes use of Raspberry Pi and ELM327 USB cable. Using these two devices and specific parameter ID (PID), the vehicle's data (rpm, speed, engine load, and engine temperature) has been collected and then sent to the remote server. The system also implements real time vehicle tracking system using GPS device. To store collected data, a database is used at the remote server and to display analyzed data, graphical user interface (GUI) is developed. OBD provides access to the vehicle's operating parameters to the owner of the vehicle or repair technician, which will help to recognize health status of the vehicle.

***IndexTerms* – On Board Diagnostics (OBD), Parameter ID (PID), Global Positioning System (GPS), Raspberry Pi.**

I. Introduction

The Internet of Things (IoT) is an arrangement of interconnected devices, advanced machines, creatures or people which are identified uniquely and also the capability to exchange data over Internet while not expecting human or human-to-computer association. On Board Diagnostics (OBD) is an automobile term alluding diagnosis of the vehicle. Basically it monitors how, when and where your vehicle is being driven. OBD provides access of vehicle's systems to the owner of the vehicle or repair technician which will help to recognize health status of the vehicles. From 1996, the State of California began emission control system for all the vehicles. Earlier vehicles had capability to diagnose the vehicle but it was not having option to send information to the owner to notify it. If a problem occurs then the indication was through a light by which user was unable to decide that in which part of the vehicle, problem has occurred.

The objective of this paper is to discuss on OBD system which monitors various vehicle parameters using vehicle's sensors. The paper is organized as follows. Section I describes introduction, section II describes literature survey, section III describes system design, section IV describes results, section V describes conclusion and future scope.

II. Literature Review

In earlier vehicle system whenever fault occurs, it was just indicating user with some indication light in vehicle dashboard. With this type of indication, user of the vehicle was unable to find out what problem has occurred and in which place. In (Sang Hyun Park et al, 2012) authors have developed an OBD device which will help us getting various vehicle parameters by connecting it with the vehicle's engine control unit. This OBD device helps user to understand various vehicle parameters and troubles involved. The authors (S. H. Baek et al, 2015) concentrated on collecting real time parameters such as speed and rpm of a vehicle. The OBD device is connected to the vehicle's engine control unit which in turn connected with the controller area network. The collected information was in hexadecimal format which was not readable for the user. So, some decoding technique was needed to read the data. The main drawback of this paper was diagnostic data in hexadecimal format and decoding collected data, which is difficult for the normal users. In (H. S. Kim, et al, 2015) authors have developed OBD Bluetooth scanner to diagnose and collect vehicle related information. To collect vehicle diagnostic data an android device is developed. The authors mainly focused on defining protocols for transmitting and receiving data from various sensors continuously using Bluetooth device. The main drawback of this paper is Bluetooth connectivity problems. Whenever you turn on the vehicle to collect the data, waiting period is required for the Bluetooth to connect with OBD device and the android device. Drawback of this paper is, the collected information is only displayed to the user who is handling that particular android device. Owner or some other user is not privileged to see the collected data. Authors (J. E. Meseguer, et al, 2015) accessed fuel consumption with the help of vehicle parameters like speed and mass air flow. An android device was used as a user interface for the entire application. Using a web interface the collected data has been sent to a remote server. Advantage of this system is it enables access to the data collected from the vehicle, and the disadvantage is fuel tank capacity of the vehicle depends on the vehicle manufacturer model. Authors (J. Zaldivar, et al,2011) focused on accident detection using force experienced during accident while travelling. When an accident occurs, authors calculated force experienced by the traveler through speed and throttle position. The main drawback of this system is that we cannot always determine using force that accident occurred and also there is no direct parameter id which determines such situations. Example-

- There is no direct PID for calculating distance travelled.
- We cannot check for gear (which gear vehicle is running currently).

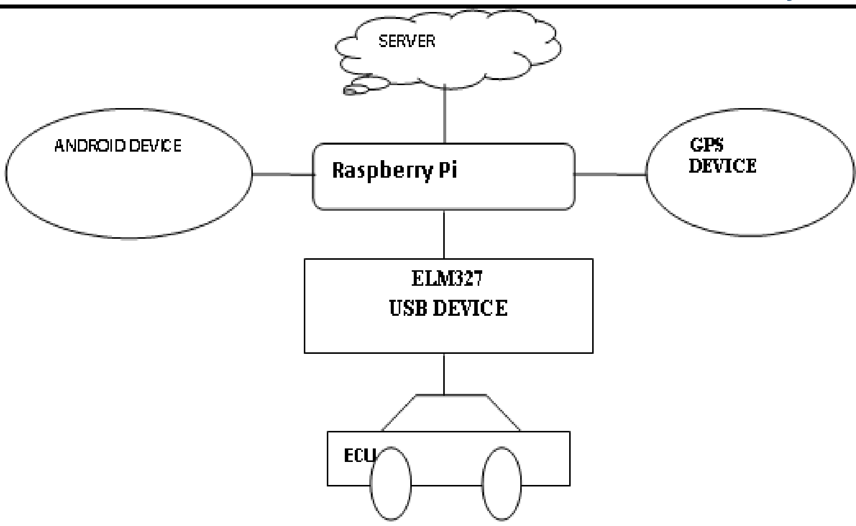

Figure 1 System Architecture

Figure 2 ELM327 USB Device

Figure 3 OBD Commands

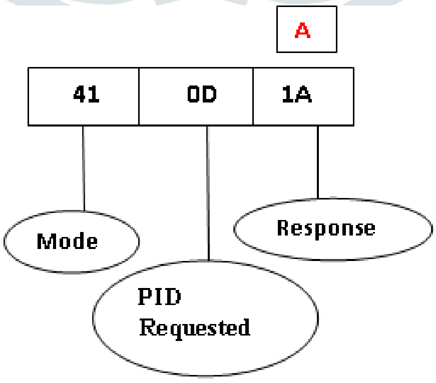

Figure 4 Explanation of Response Message

III. System Design

The basic architecture of the system is given in Figure 1. The proposed system allows collecting data from the vehicle through OBD device into the raspberry pi and then the details will be sent to the particular server for the further analysis. To track the vehicle location GPS device and android device has been used.

3.1 Data Collection

First step in the system design is to gather information from the vehicle. Once if a device needs to communicate with different parts of a vehicle, the device has to send parameter ID to the vehicle. To gather information from the vehicle, each vehicle is connected to its internal controller area network (CAN). ELM327 USB cable is used to extract data from the vehicle. The Figure 2 shows ELM327 USB device. The female side of this ELM327 USB cable is connected with vehicle's controller area network. To check whether the device is connected properly with the vehicle, OBD commands are used which is shown in Figure 3 and are explained below.

- ATI - This command will help in identifying device.
- AT/N - This command retrieves serial number of ELM327 USB device.
- ATZ - Resets the device and returns the identification of the ELM-USB device.
- ATSP - This command sets specified protocol.
- ATDP – This command will display protocol used by the vehicle to communicate.
- 0100 – This command ensures communication between vehicle and the OBD device.

3.2 Data Mapping

Once the data is collected from the vehicle, next step is to map the data into user readable form because the data generated from CAN bus will be in hexadecimal form. Hence it is required to convert the data into decimal form using standard formulas. For example if a request is made for vehicle's speed using parameter id (PID) 010D then response is 410D1A.

In the response bytes, ignore the most significant 2 bytes as they are associated with the mode and requested PID. Consider next byte as A (group1) as shown in Figure 4. Here 1A is response byte. Consider 1A as A and Converting A into decimal, will get speed as 16.15 km/h.

Gathered data is stored in a database for further access. Using suitable user interface, data can be displayed. Data flow diagram shows details about from where the input is coming and where the data is being stored and what will be the output from the system. The following section describes sequence of events that takes place through the entire process starting from the data collection till the data visualization. Figure 5 shows data flow diagram.

The sequence is as follows

- First step is to gather various sensor data from the vehicle using specific parameter identification.
- Next step is data mapping, as the collected data will be in hexadecimal format so need to format it in user readable form using appropriate formula.
- Once data has been mapped, next step is to store it in server for further access.
- Final step is to make use of appropriate user interface to access and visualize data.

3.3 GPS Tracking

The GPS device is connected to the Raspberry Pi. GPS tracking requires collecting at least longitude and latitude parameters from National Marine Electronics Association (NEMA) to track the vehicle. The sequence of GPS tracking is as follows.

- GPS coordinates are collected using specific GPS device.
- Collected GPS data will be sent to the remote sever.
- User interface helps to alert the vehicle movement.
- In case of vehicle theft, OBD will help to track the vehicle easily as this system will continuously monitor vehicle position using GPS coordinates.

To gather longitude and latitude information from the NEMA, baud rate of 9600 kbps is set. Python code has been written to collect the GPS coordinates from the GPS device and stored it in a file for transmitting it to a remote server.

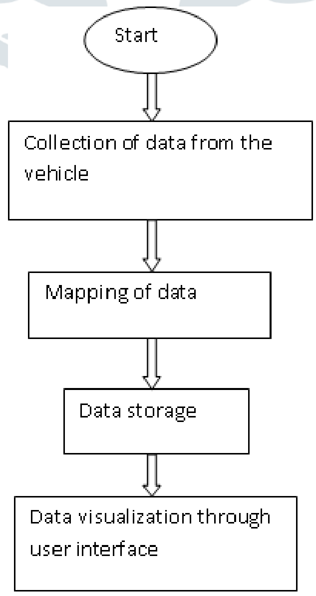

Figure 5 Data Flow Diagram

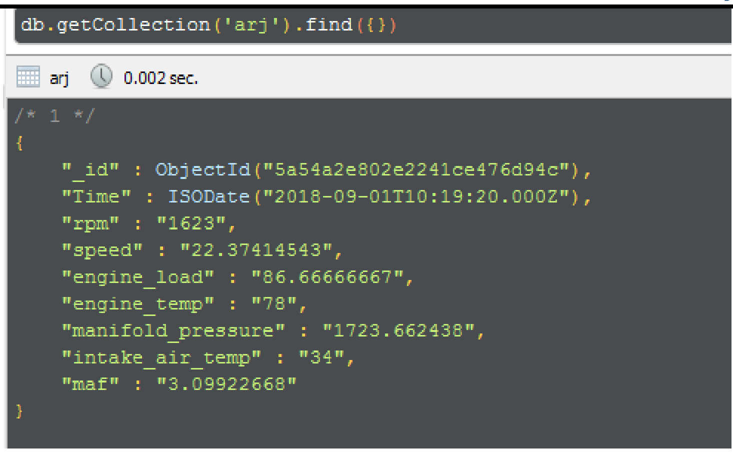

Figure 6 Sample Data Stored in MongoDB

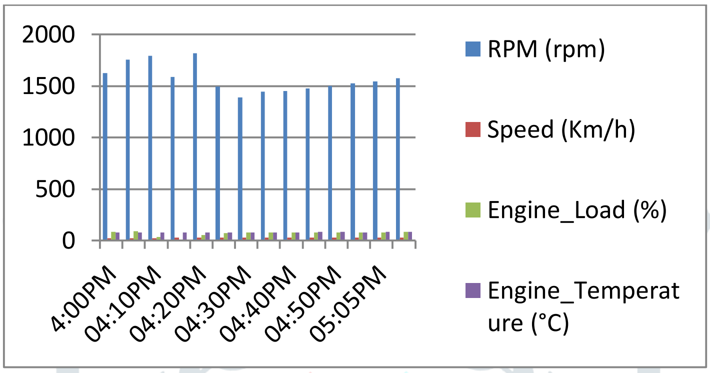

Figure 7 Bar Chart for Time Vs rpm, Speed and Engine load

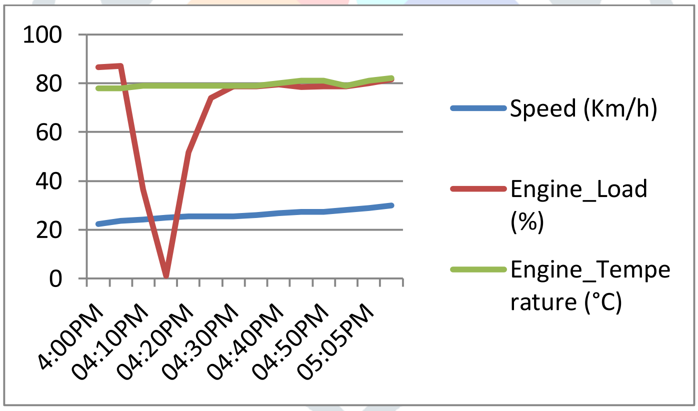

Figure 8 Graph for Time Vs Speed, Engine load and Engine temperature

3.4 Data Storage

In order to store gathered data from the vehicle and the GPS data, MongoDB database is used. MongoDB is a document based database which stores data as documents and collections. Figure 6 shows sample data stored in MongoDB database.

3.5 Data Access and Evaluation

In order to view the stored data, dashboard has been created. Stored data was then displayed to the owners using graphical representation by retrieving data from MongoDB and plotting graph with respect to parameters. To share the vehicle location, messaging model also has been implemented. In order to implement messaging model way2sms application interface has been used. For messaging, Google has its built in universal resource allocator (URL) except the GPS coordinates, so fetch the coordinate information from the GPS device and include it in the Google's URL.

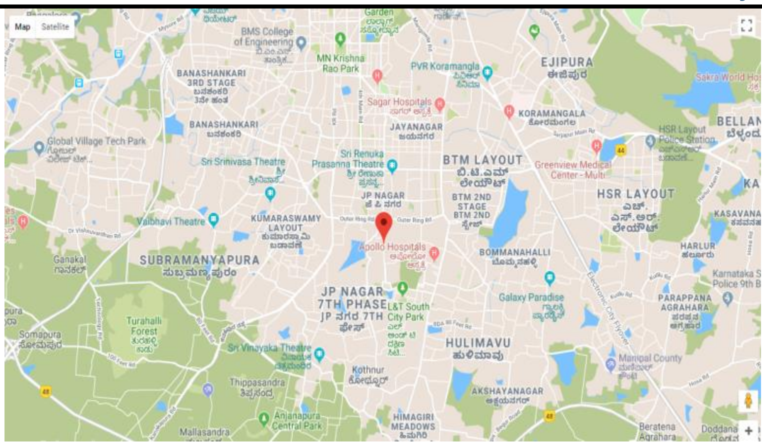

Figure 9 Vehicle Location on Google Map

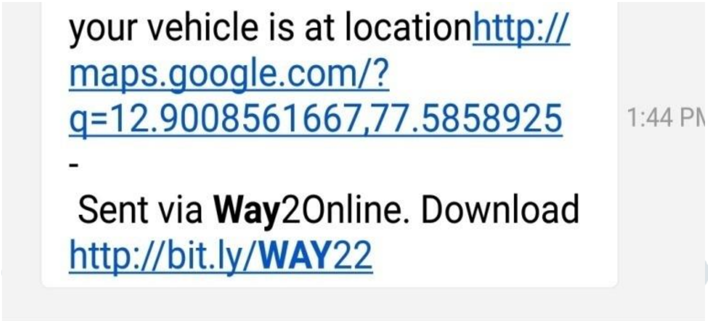

Figure 10 Location of the Vehicle Received Via Text Message

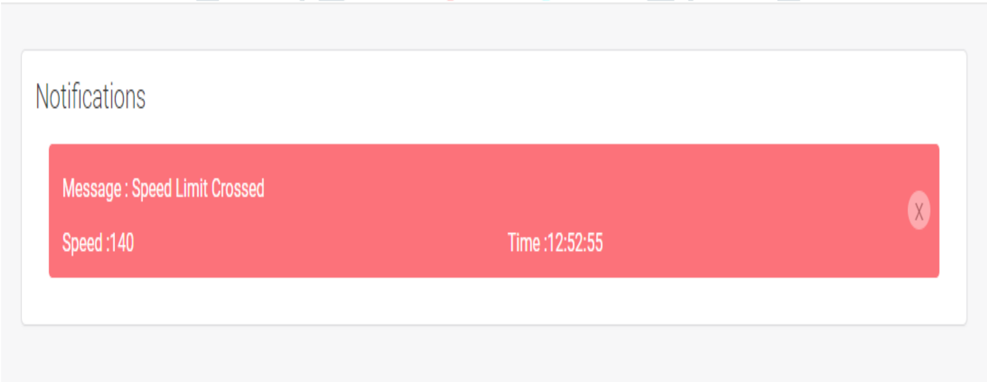

Figure 11 Notification of Vehicle Speed Limit Cross

IV. Result

In the proposed system, vehicle parameters like speed, rpm, engine load, and engine temperature are monitored. Figure 7 shows bar chart of Time vs. Rpm, Speed and Engine load. Owner can analyze that at a particular Speed what the rpm is. Owner can also analyze at any particular time, what the vehicle speed and engine load is. Figure 8 shows graph for the comparison of Time Vs Speed, Engine load, Engine Temp. By looking at the graph, owner can analyze vehicle's Speed, vehicle's engine temperature and vehicle's engine load.

Normally when the speed of the vehicle increases, automatically engine temperature also gets increased. If the vehicle is overloaded, then engine load will be higher than normal. Increase in the vehicle's speed will lead to increase in the vehicle's engine temperature and engine load which in turn may cause several vehicle problems.

To avoid occurrences of the problems in the vehicle one must ensure that vehicle is not over speedy driven. Whenever the engine temperature reaches maximum level, then immediate action has to be taken to cool the engine. Sometimes high engine temperature will cause the vehicle to damage seriously. "Higher the vehicle's load, slower the vehicle's speed", helps the owner to try not to overload the vehicle which impacts on vehicle's performance.

The proposed system has vehicle tracking which is shown in the Figure 9. To share the vehicle location, messaging model also has been implemented. Figure 10 shows location of the vehicle received via text message.

The proposed system also contains push notifications. Suppose, if an owner owns 10 vehicles and has instructed the drivers not to drive the vehicle with speed limit not more than 140 km. If the vehicle crosses given speed limit then owner of the vehicle will get a notification in the notification section saying that vehicle speed limit is crossed along with the speed and time. Figure 11 shows notification when vehicle crosses speed limit of 140 km. Whenever the vehicle is on move, owner of the vehicle must be worried about vehicle's safety like whether the driver is driving safely or not, how the person is driving the vehicle. Push notifications help the owner of the vehicle to keep track of vehicle's safety by knowing if the driver violates any given limitations.

V. Conclusion

The proposed On Board Diagnostics system of vehicles makes use of ELM327 USB device to gather vehicle information and also it helps the owner of the vehicle to know vehicle status. Owner can get to know when, where and how a vehicle is being driven. Also this system helps the owner of the vehicle to monitor driving behavior of the driver like speed at which the driver is driving the vehicle and current location of the vehicle. The system also records, vehicle's various sensor data, performs analysis on it and also provides notifications to the owner of the vehicle. This system helps for the safety of the vehicle by continuously monitoring it. Dashboard has been successfully created to view the collected data from the vehicle. Graphical visualization like bar chart and line chart has been created to view the data from the vehicle.

In future, additional sensor data can be collected and can perform analysis on it. Suppose while driving, if the vehicle stops suddenly, immediate text message can be sent to the driver, regarding problems occurred in the vehicle and also suggestion to overcome that problem.

References

[1] Sang Hyun Park and Sang Yub Lee, 2012, Development of On-Board Diagnostics via CAN for a HVI (Human Vehicle Interface) technology, IEEE 10th International Symposium on Parallel and Distributed Processing with Applications, pp. 839-840.

[2] S. H. Baek and J. W. Jang, 2015, Implementation of integrated obd-ii connector with external network, Information Systems, Vol. 50, pp. 69–75.

[3] H. S. Kim, S. J. Jang, and J. W. Jang, 2015, A study on development of engine fault diagnostic system, Mathematical Problems in Engineering, Article ID 271374.

[4] J. E. Meseguer, C. T. Calafate, J. C. Cano, and P. Manzoni, 2015, Assessing the impact of driving behavior on instantaneous fuel consumption, in 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp. 443–448.

[5] J. Zaldivar, C. Calafate, J. Cano, and P. Manzoni, 2011, Providing accident detection in vehicular networks through obd-ii devices and android based smartphones, in IEEE 36th Conference on Local Computer Networks (LCN), pp. 813–819.