IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Vehicle Health Monitoring System

E.Chandrasekhar, *Assistant Professor, Department of Electronics and Communication Engineering,*

Chaitanya Bharathi Institute of Technology(A), Gandipet, Hyderabad, India.

Abstract: Now-a-days, rented cars are being widely used by people and the business growth pertaining to this sector has been tremendously increasing in recent years. However, the cars have been the primary requirement for running the firm; it is absolutely required to keep track of these valuable assets. The monitoring of cars is very much important as the firm holds large number of cars and the tracking of each car's health being used by the customers simultaneously is the primary concern. These challenges can be tackled with the help of On-board diagnostics. On-board diagnostics is an automotive term referring to a vehicle's self-diagnostic and reporting capability. OBD systems give the vehicle owner or repair technician access to the status of the various vehicle subsystems .The amount of diagnostic information available via OBD has varied widely since its introduction in the early 1980s versions of on-board vehicle computers. Early versions of OBD would simply illuminate a malfunction indicator light or "idiot light" if a problem was detected but would not provide any information as to the nature of the problem. Modern OBD implementations use a standardized digital communications port to provide real-time data in addition to a standardized series of diagnostic trouble codes, which allow one to rapidly identify and remedy malfunctions within the vehicle. A car generally is equipped with OBD-I or OBD-II bus to specify the reports or diagnostics of the vehicle.

Keywords: *On-board diagnostics, self-diagnostic, idiot light, real-time data, diagnostic trouble codes, malfunctions.*

I. Introduction

The total car sales raised to 2.6 million units in India alone by the year 2009.With the increase in number of cars along with other modes of transport such as public transport system, vehicles for supply chains and two wheelers on road, the issues like safety, fuel consumption, pollution check are of utmost importance which depends on vehicle condition, road infrastructure and driver behavior [1].

The paper aims at developing an embedded system for detecting the vehicle condition by monitoring the internal parameters that are used in evaluating the vehicle‗s current health condition. These parameters are obtained using OBD2 protocol through a port provided by the manufacturers to the vehicles.

A real time evaluation system is being defined that can be used for rapid condition screening and provide reliable information about the vehicle conditions. This real time evaluation system can be called Vehicle Health Monitoring System. The information that is obtained about the vehicle conditions can be accessed using a scan tool known as the OBD adapter by inserting the adapter into the OBD port provided inside the car.

OBD stands for “On-Board Diagnostics”. It is a computer-based system originally designed to reduce emissions by monitoring the performance of major engine components [3]. The system model being developed is a standalone on-board model which will be a black box for outside world. This model can be extended to identify and report the faults in car to the authorized service centre through wireless communication and IoT, a concept of remote diagnostics.

A. HARDWARE SPECIFICATIONS

1) ***Atmel AT89S52 microcontroller:*** The AT89S52 microcontroller is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89S52 provides the following standard features:

- 8K bytes of Flash
- 256 bytes of RAM
- 32 I/O lines
- Watchdog timer
- Two data pointers
- Three 16-bit timer/counters
- Six-vector two-level interrupt architecture
- Full duplex serial port
- On-chip oscillator
- Clock circuitry

2) ***Wi-Fi module – ESP 8266:*** The ESP 8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP 8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP 8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your adapter device

and get about as much Wi-Fi ability as a Wi-Fi Shield offers. The ESP 8266 module is an extremely cost effective board with a huge and ever growing community. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area.

Specification of ESP8266:

- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- 19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- 1MB Flash Memory
- Integrated low power 32-bit CPU could be used as application processor
- Standby power consumption of < 1.0mW

3)***16x2 LCD Display:*** LCD (liquid crystal display) is the technology used for displays in notebook and other smaller computers. There are some preset commands instructions in LCD, which we need to send to LCD through the microcontroller.

4) ***Crystal Oscillator:*** It is an electronic circuit that is used to generate an electrical signal of precise frequency by utilizing the vibrating crystal's mechanical resonance made of piezoelectric material.

5) ***Bridge rectifier:*** A bridge rectifier is a type of full wave rectifier which uses four or more diodes in a bridge circuit configuration to efficiently convert the Alternating Current (AC) into Direct Current (DC).

6) ***MAX 232 IC:*** Max 232 is an IC(integrated circuit) that converts TTL(Transistor Transistor logic) logic level signal into its equivalent RS-232c level signal and RS-232c level to its equivalent TTL level signal. This IC is very important in case when we need to make connection and transfer data between devices that work on different signal level wave forms (TTL, Rs232c).

7) ***Heat sink:*** Heat sink is an electronic component or a device of an electronic circuit which disperses heat from other components of a circuit into the surrounding medium and cools them for improving their performance, reliability and also avoids the premature failure of the components.

8) ***Transformer:*** A transformer is a static electrical device that transfers electrical energy between two or more circuits through electromagnetic induction**.** A varying current in one coil of the transformer produces a varying magnetic field, which in turn induces a varying electromotive force (emf) or voltage in a second coil. Power can be transferred between the two coils through the magnetic field, without a metallic connection between the two circuits. Transformers are used to increase or decrease the alternating voltages in electric power applications.

8) ***LM35 temperature sensor*****:** LM35 is a temperature sensor that outputs an analog signal which is proportional to the instantaneous temperature. The output voltage can easily be interpreted to obtain a temperature reading in Celsius. The advantage of LM35 over thermistor is that, it does not require any external calibration. The coating also protects it from self heating.

9) ***Triggers:*** We use potentiometer to measure speed and a push button as Ignition key. These act like triggers to the parameters of the vehicle.

B. SOFTWARE SPECIFICATIONS

Thingspeak: ThingSpeak is an open IoT data platform based on public cloud technology. ThingSpeak enables real time data collection, analysis and actuation with an Open API. According to its developers, "ThingSpeak is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates". It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plug-ins and apps for collaborating with web services, social network and other APIs. The core element of ThingSpeak is a 'ThingSpeak Channel'.

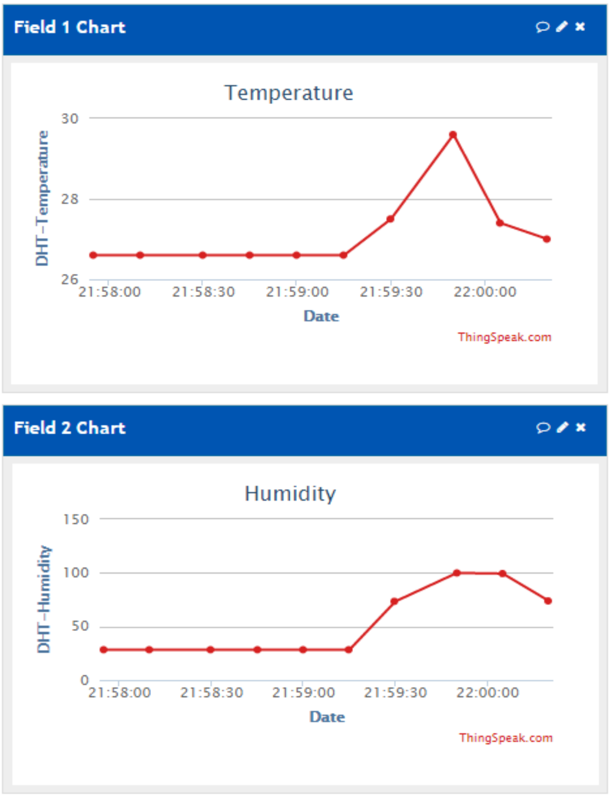


Fig 1 Thingspeak channels

A channel stores the data that we send to ThingSpeak and comprises of the below elements:

- Eight fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.
- Three location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
- One status field - A short message to describe the data stored in the channel. To use ThingSpeak, we need to sign

up and create a channel. Once we have a channel, we can send the data, allow ThingSpeak to process it and also retrieve the same.

The retrieved information from channels is displayed in the form of charts. The smart charts are shown in the Fig 1.Each of the dots corresponds to the value and the time at which the value was posted to the channel. Place the mouse over a dot to get more details on the exact date and the GMT offset from which the value was posted. We can send any data here; say the periodic readings from a temperature sensor or RPM values from a motor. The Y-axis shows the names that we specified to each of the labels.

II. PROPOSED MODEL

A smart monitoring technology to perform task on daily basis have been proposed .The central idea is to design and implement a cost effective and easy to handle vehicle health monitoring system. Internets of things (IoT) has enhanced the flexibility for incorporating discrete components and supervise the same through an IoT platform called Thingspeak. The proposed system consists of an adapter board which consists of an Atmel AT89S52 microcontroller which acts as a central unit of the system to which a Wi-Fi module is interfaced in order to send the received microcontroller parameters to the IoT based platform where the data will be saved.

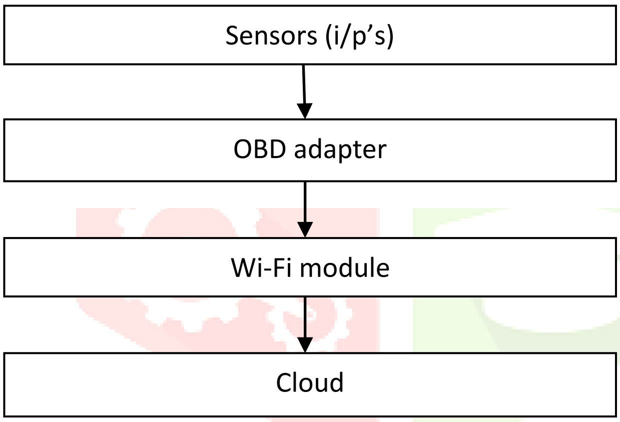


Fig 2 Design Flow

DESCRIPTION

1) An OBD-II adapter is designed and implemented on a circuit board with all the required components. Car parameters will be acquired with the help of this device from the car input sensors.

2) The output of the adapter is connected to a Wi-Fi module where the data is transferred and acquired via IoT platform.

3) The Wi-Fi module will send the data acquired from the OBD adapter to an IoT platform where the data will be saved for the user to access. This data will be saved from time to time without any delay for the most part.

III. EXPERIMENTAL SETUP

Implementation of the project includes working with the embedded system and software implementation. The whole embedded system of the vehicle health monitoring system is integrated as shown below in the Fig 3.

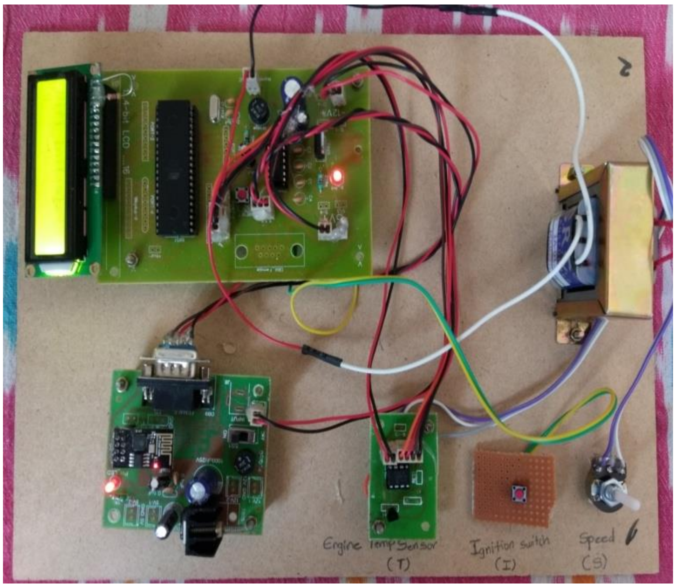


Fig 3 Vehicle health monitoring system setup

The setup consists of an OBD adapter, an ESP8266 Wi-Fi module, the three car sensors and triggers, and a transformer which powers up the entire setup.

A. OBD ADAPTER

The Fig 4 below shows an OBD adapter which is the main module of the whole setup.

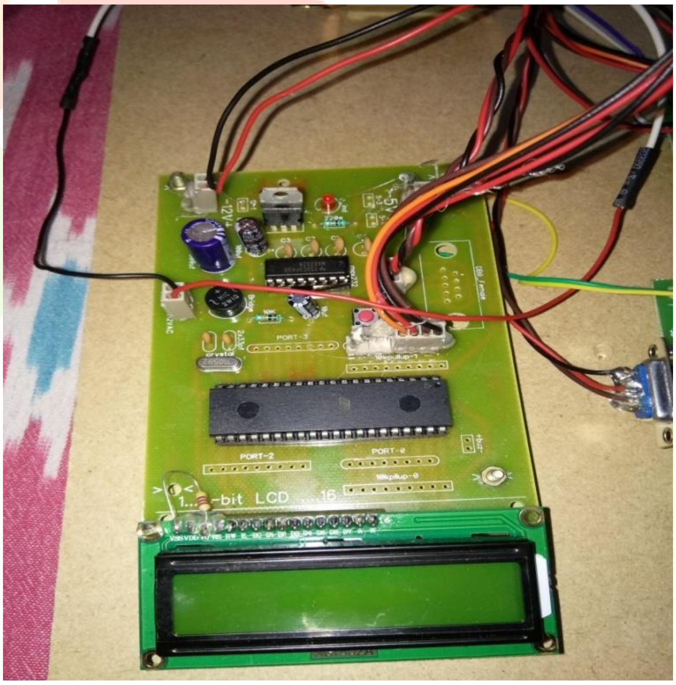


Fig 4 Integrated OBD adapter

This OBD adapter consists of a microcontroller Atmel AT89S52 which controls, processes and sends data to cloud with the help of intermediate components. The code is dumped into the microcontroller which enables LCD and Wi-Fi initialization. There are other components like LCD screen, Crystal oscillator, Max232 IC, a bridge rectifier and heat sink. The LCD screen shows all the parameters that are being varied and showcases the present scenario of the car parameters with a time interval of 20 seconds as shown in Fig 5.

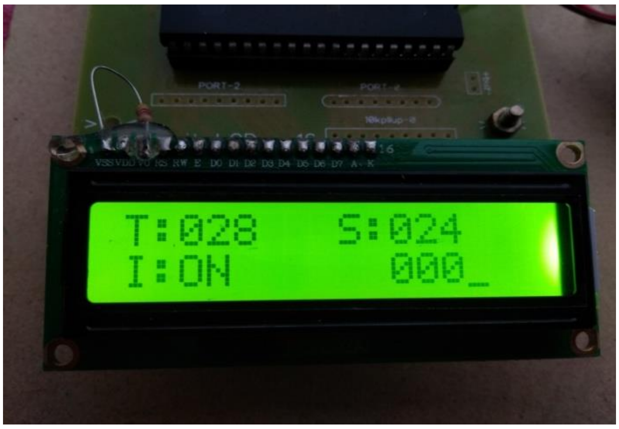


Fig 5 LCD screen parameters

B. ESP 8266 Wi-Fi MODULE

The Wi-Fi module consists of ESP 8266 Wi-Fi microchip, Max232 DB9 connector, a switch and heat sink.

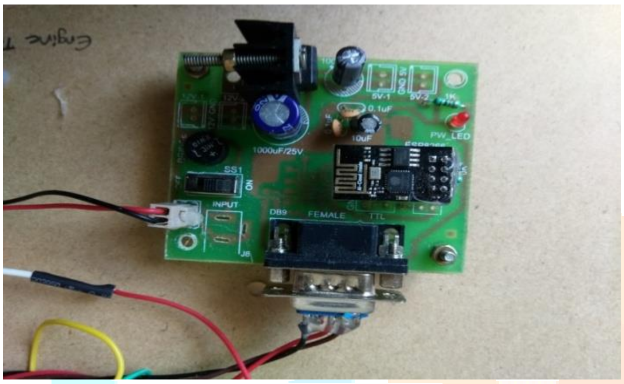


Fig 6 ESP 8266 Wi-Fi module

C. VEHICLE PARAMETERS

Three vehicle parameters that are being used includes Temperature, Speed and Key ignition.

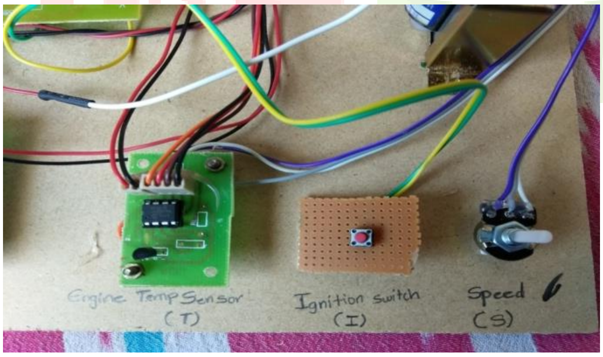


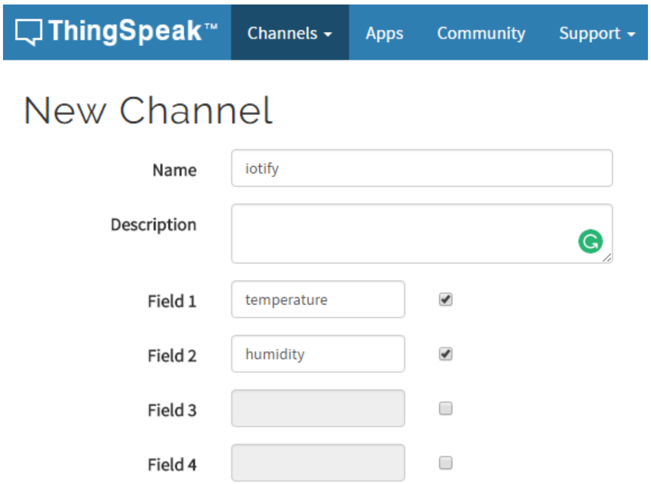
Fig 7 Vehicle input parameters

- Engine temperature can be measured with the help of a temperature sensor LM35.
- Speed can be measured with the help of a potentiometer.
- Ignition switch can be switched ON/OFF with the help of a push button.

The same parameters are displayed on the LCD screen for observation purpose.

D. THINGSPEAK

ThingSpeak to be of any meaning, it is imperative to have a micro-controller with an internet connection. The type of micro-controller is completely up to the user, as ThingSpeak is able to communicate with any type, as long as it is networked. We will be using an ATMEL AT89S52 microcontroller and ESP8266 Wi-Fi module for connection. First of all we must sign up for a new user account log in and create new channels. When logged in, create a new channel by selecting Channels > My Channels and then Create New Channel. The new channel will show the details which include channel name and field names.



New Channel

Name: iotify

Description:

Field 1: temperature ☑

Field 2: humidity ☑

Field 3: □

Field 4: □

Fig 7 Thingspeak new channel description

After setting up the channel, the channel produces a unique identifier key which is used to identify the channel when reading or uploading data.

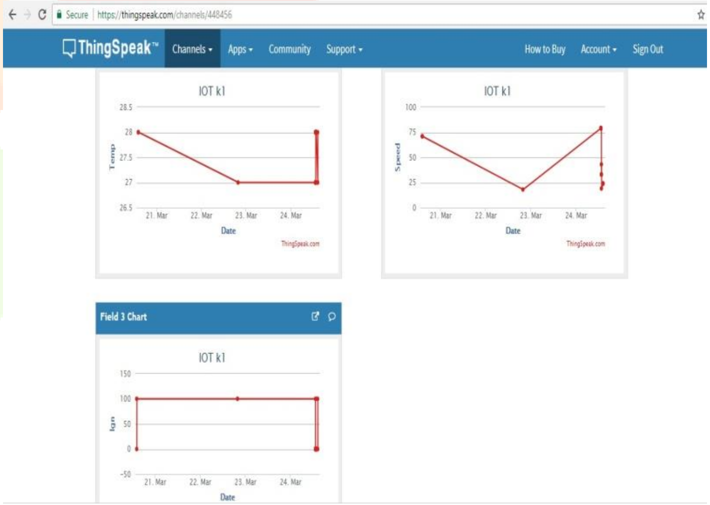


Fig 8 Thingspeak channel setup

All entries are stored with a unique identifier and a date and time stamp. Existing data can be imported from a "Comma-Separated Values" (CSV) file, which is a popular format for storing tabular data.

IV. EXPERIMENTAL RESULTS

A. INITIALIZATION PROCEDURE

First the whole setup is switched ON and the hotspot on the phone is enabled with the same username and password given to the Wi-Fi module in the code. The LCD screen shows the text saying "Wi-Fi Initializing" once the setup is ON as shown in the Fig 9.

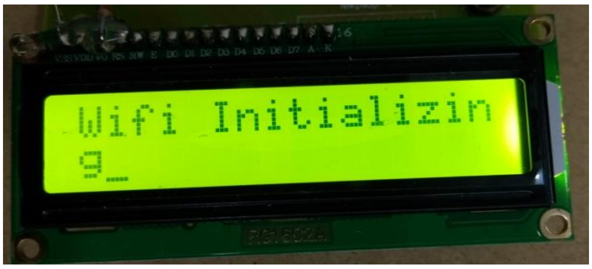


Fig 9 LCD Wi-Fi Initializing

Once the Wi-Fi module gets connected to the phone it shows by the text “CONNECTED” as shown in Fig 10.

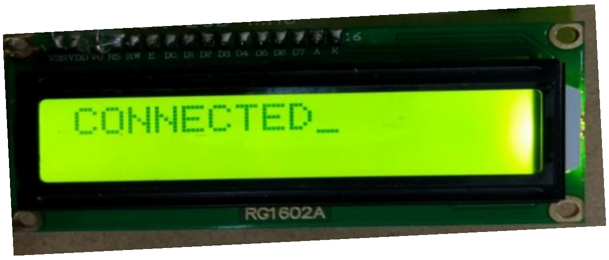


Fig 10 LCD Wi-Fi CONNECTED

With reference to LCD screen we can know the data that has been sent to the thinkspeak platform via the Wi-Fi module. The comparison between the two will make it clear about the real time data transfer.

B. OBSERVATIONS

Observation through the LCD screen is shown in Fig 11

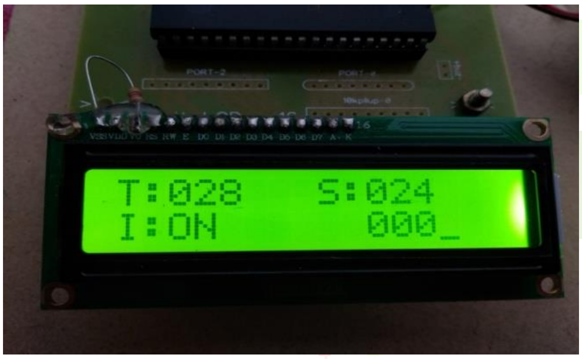
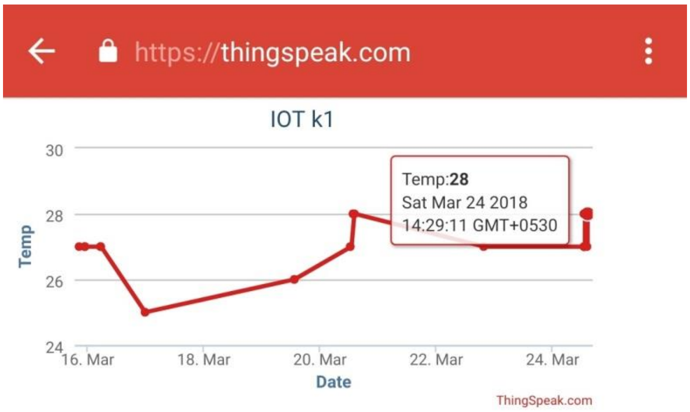


Fig 11 LCD Screen 2

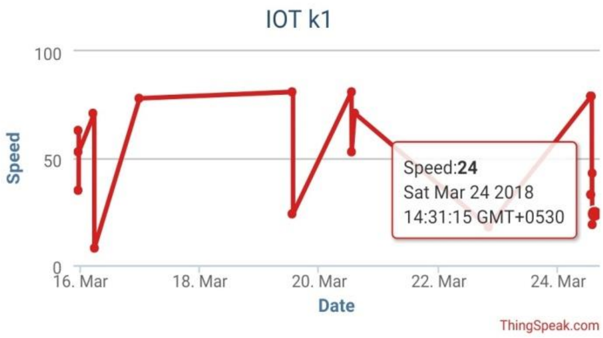
The LCD observations are
Temperature **T: 028**, Speed **S: 024**, Ignition **I: ON**
The thingspeak platform observations with reference to LCD screen.
1) **Temperature observation**



Temp:**28**
Sat Mar 24 2018
14:29:11 GMT+0530

Fig 12 Thingspeak – temp 2

The temperature is being shown as **28** at that particular time and date.
2) **Speed observation**

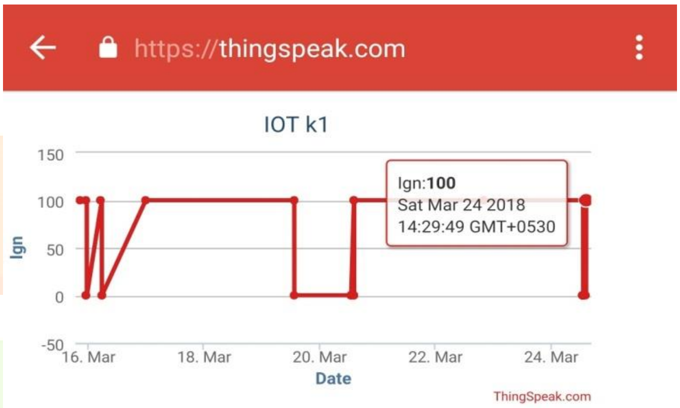


Speed:**24**
Sat Mar 24 2018
14:31:15 GMT+0530

Fig 13 Thingspeak – speed 2

The speed is being shown as **24** at that particular time and date.

3) **Ignition observation**

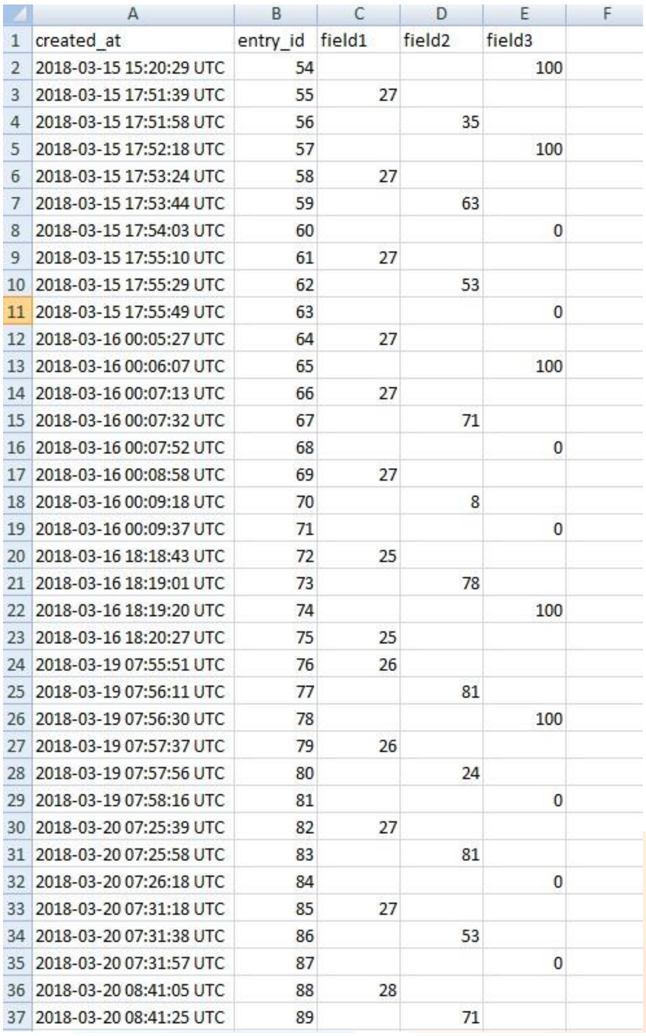


Ign:**100**
Sat Mar 24 2018
14:29:49 GMT+0530

Fig 14 Thingspeak – ignition 2

The ignition is being shown as **ON (100)** at that particular time and date.

The observations can also be made through a file called “Comma-Separated Values” (CSV) file, which is a popular format for storing tabular data. This file can store data with the values of the three parameters as shown in Table 15.



	A	B	C	D	E	F
1	created_at	entry_id	field1	field2	field3	
2	2018-03-15 15:20:29 UTC	54			100	
3	2018-03-15 17:51:39 UTC	55	27			
4	2018-03-15 17:51:58 UTC	56		35		
5	2018-03-15 17:52:18 UTC	57			100	
6	2018-03-15 17:53:24 UTC	58	27			
7	2018-03-15 17:53:44 UTC	59		63		
8	2018-03-15 17:54:03 UTC	60			0	
9	2018-03-15 17:55:10 UTC	61	27			
10	2018-03-15 17:55:29 UTC	62		53		
11	2018-03-15 17:55:49 UTC	63			0	
12	2018-03-16 00:05:27 UTC	64	27			
13	2018-03-16 00:06:07 UTC	65			100	
14	2018-03-16 00:07:13 UTC	66	27			
15	2018-03-16 00:07:32 UTC	67		71		
16	2018-03-16 00:07:52 UTC	68			0	
17	2018-03-16 00:08:58 UTC	69	27			
18	2018-03-16 00:09:18 UTC	70		8		
19	2018-03-16 00:09:37 UTC	71			0	
20	2018-03-16 18:18:43 UTC	72	25			
21	2018-03-16 18:19:01 UTC	73		78		
22	2018-03-16 18:19:20 UTC	74			100	
23	2018-03-16 18:20:27 UTC	75	25			
24	2018-03-19 07:55:51 UTC	76	26			
25	2018-03-19 07:56:11 UTC	77		81		
26	2018-03-19 07:56:30 UTC	78			100	
27	2018-03-19 07:57:37 UTC	79	26			
28	2018-03-19 07:57:56 UTC	80		24		
29	2018-03-19 07:58:16 UTC	81			0	
30	2018-03-20 07:25:39 UTC	82	27			
31	2018-03-20 07:25:58 UTC	83		81		
32	2018-03-20 07:26:18 UTC	84			0	
33	2018-03-20 07:31:18 UTC	85	27			
34	2018-03-20 07:31:38 UTC	86		53		
35	2018-03-20 07:31:57 UTC	87			0	
36	2018-03-20 08:41:05 UTC	88	28			
37	2018-03-20 08:41:25 UTC	89		71		

Fig 15 CSV file tabular format of the monitoring parameters

Column A gives the date and time. Column B gives the entry IDs. Column C, Column D and Column E gives temperature, speed and ignition status parameters respectively. Hence these are the observations that are produced through thingspeak platform which can be very useful for effective health monitoring of the vehicle.

The following observations can also be seen with the help of a mobile app by the name "ThingView Free". Through this app, we can accurately view all the parameters from point to point.

Fig 16 ThingView mobile app

V. CONCLUSION

The paper presented a real time vehicle health monitoring system which makes use of IoT involving an OBD adapter which reads the sensor data and sends the data to the cloud via Wi-Fi module. This whole setup can be made available for people at low cost and the OBD adapter will help people keep track of their car's health through daily monitoring and also facilitates in enhancing the management of the vehicle. As the health system provides health of the vehicle, the insurance companies are planning to pay according to its health and usage. Therefore, OBD is one monitoring system which will be very much useful and this can be the next major future tool for effective monitoring.

VI. ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mentioning of the people whose constant guidance and encouragement made it possible. We express our earnest gratitude and indebtedness for the assistance and help given by Sri. E Chandrasekhar, Assistant Professor, Dept. of ECE, CBIT and we sincerely thank the Department of Electronics and Communication Engineering at Chaitanya Bharathi Institute of Technology, Hyderabad for providing us with all the required laboratories to carry forward the necessary work.

REFERENCES

[1] R. Rajamani, A. S. Howell, C. Chen, J. K. Hedrick, and M. Tomizuka, -A complete fault diagnostic system for automated vehicles operating in a platoon,‖ IEEE Transactions on Control System Technology, vol. 9, no. 4, pp. 553–564, July 2001.

[2] R. J. Patton and J. Chen, -Robust fault detection of jet engine sensor systems using Eigen structure assignment,‖ AIAA Journal of Guidance, Control, and Dynamics, vol. 15, no. 6, pp. 1491–1497, Nov-Dec 1992.

[3] W. H. Chung, Fault Detection and Identification with Application to Advanced Vehicle Control Systems. California PATH Research Report UCB-ITS-PRR-96-25, 1996.

[4] Automotive Electronics Handbook, Jurgen, R., McGraw-Hill, 1995.

[5] Alternative Cars in the 21st Century, Riley, R. Q., SAE, 1994.

[6] M.-A. Massoumnia, G. C. Verghese, and A. S. Willsky, -Failure detection and identification IEEE Transactions on Automatic Control, vol. AC-34, no. 3, pp. 316–321, Mar. 1989.