ISSN 2320-9984

Volume 2 Issue 1 | March 2014 ISSN: 2320-9984 (Online)

International Journal of Modern Engineering & Management Research
Website: www.ijmemr.org

Android Based Universal Vehicle Diagnostics and Tracking System

Pol Sharad Ganapati
Department of Electronics & Tele Communication Engg.
Pravara Rural Engineering College,
Loni, Ahmednagar, [INDIA]
Email : sharadpol91@gmail.com

Bobe Anup Dattatraya
Department of Electronics & Tele Communication Engg.
Pravara Rural Engineering College,
Loni, Ahmednagar, [INDIA]
Email : anupbobe21@gmail.com

Gholap Siddharth Popatrao
Department of Electronics & Tele Communication Engg.
Pravara Rural Engineering College,
Loni, Ahmednagar, [INDIA]
Email : gholapsid21@yahoo.com

Prof. R. A. Kadu
Professor
Department of Electronics & Tele Communication Engg.
Pravara Rural Engineering College,
Loni, Ahmednagar, [INDIA]

Abstract— *This system aims to provide a low-cost means of monitoring a vehicle's performance and tracking by communicating the obtained data to a mobile device via Bluetooth. Then the results can be viewed by the user to monitor fuel consumption and other vital vehicle electromechanical parameters. Data can also be sent to the vehicle's maintenance department which may be used to detect and predict faults in the vehicle. This is done by collecting live readings from the engine control unit (ECU). An electronic hardware unit is built to carry-out the interface between the vehicle's OBD system and a Bluetooth module, which in part communicates with an Android-based mobile device. The mobile device is capable of transmitting data to a server using cellular internet connection.*

Keywords:—*AVR microcontroller, Bluetooth module, Remote server setup, Android application, control unit.*

1. INTRODUCTION

Nowadays, vehicle navigation and location concept is well defined and is frequently used. That was a result of a lot of research efforts which were invested in developing its components and ideas. Also, in recent years, the problem of diagnoses of defects and faults on a remote vehicle has received considerable attention as a result of several factors:

The relatively unprecedented advancement of communications technologies and the availability of various data services.

The current changes in the automobile industry trends where vehicles are developed with a particular focus on eco-friendliness safety, besides comfort. Additional services and safe and convenient automatic internal control systems have been introduced to satisfy the customers' needs.

Collection and analysis of diagnostic data from electronic control units is of paramount importance in the automotive industry, both from a life cycle support perspective post production and sales, and as a tool in the product development.

For pre-series test vehicles, access to diagnostic data is crucial in order to be able to track problems as early as possible in the development process, preventing serious faults to pass undetected into production Vehicles. Therefore, Information technologies must be added as software to enable, facilitate, and enhance achievement of the above goals.

Built-in automobile navigation systems are being equipped with various multimedia capabilities to satisfy the customer expectations of modern functions and features and offer diverse services in the current greatly competitive market state.

This reflects on the overall system cost and limits the advantages to high-end vehicle models. A new business oportunity that provides customer-oriented vehicle diagnosis and remote maintenance can be offered.

Vital vehicle information can be displayed to users, or/and communicated to remote maintenance departments via cellular networks [5], [6],where relatively newer cellular access technologies offer future solutions endless choices for high-end applications [7], [8]. Hence, if a correct diagnoses (or prediction) of a defect is available, instructions and commands can be sent to the driver on how to proceed in that situation. Also, if the system has knowledge of the location, in addition to an error category, the system can direct the vehicle to the closest vehicle service location in the vicinity.

However, these suggested solutions and systems need external dedicated hardware components and specialized software applications running on a laptop to display obtained data to the user as most vehicles do not have built in screens. This can pose an added cost and may be applied to a limited group of vehicles.

We present in this paper a low-cost, simplified, yet versatile vehicle diagnostic system that is compatible with all vehicles manufactured after 1996. The graphical user-interface (display and command) is Andriod-based and utilizes the popular personal area network (PAN) communications standard Bluetooth to facilitate extraction and relaying of readings, diagnostic trouble codes (DTC), and commands. We have chosen a smart phone as the computing device for the obvious growth and demand on such mobile devices, in addition to reducing the overall system cost by utilizing the built-in functionalities that is integrated in such compact devices. Also, when installed properly, such devices can be a low cost alternative to integrated navigation systems [1]. Moreover, our choice of Android as our operating system platform is consistent with current market trends and shares and user acceptance of such platform [9].

2. SYSTEM OVERVIEW

The purpose of our Android-based user-interface vehicle diagnostic system implemented in this work is the execution of diagnoses on a remote vehicle using internationally agreed data trouble codes. We implementation of a simple diagnostic system which can be readily available to the average user and because the system offers easy to understand information for drivers and for specialists.

The vehicle's ECU stores only diagnostic trouble codes when there is a reading that is out of range but all other readings are not being made use of. If these readings are analyzed they can be used to predict faults and prevent them. The readings obtained from the sensors need to be transmitted to a remote server held at the maintenance department, where they can be processed and analyzed to:

1) Detect faults present in the vehicle

2) Predict faults that may occur in the future.

The geographical position of the vehicle is also transmitted to make it possible to service minor faults at the site. Moreover, realtime data collection and storage in a dedicated database can be easily configured.

3.WORKING

Block Diagram gives generalized outline description of the project as to how we are going to design the further steps. The purpose of our Android-based user-interface vehicle diagnostic system implemented in this work is the execution of diagnoses on a remote vehicle using internationally agreed data trouble codes

Here we are using four types of sensor temperature sensor, diesel level sensor, speed sensor and seat belt sensor. Whatever signals coming from sensor are analog in nature so they are connected to ADC which is built in AVR microcontroller that digital signals are given to AVR microcontroller the relays, buzzer etc. are connected to the AVR microcontroller through device driver block.

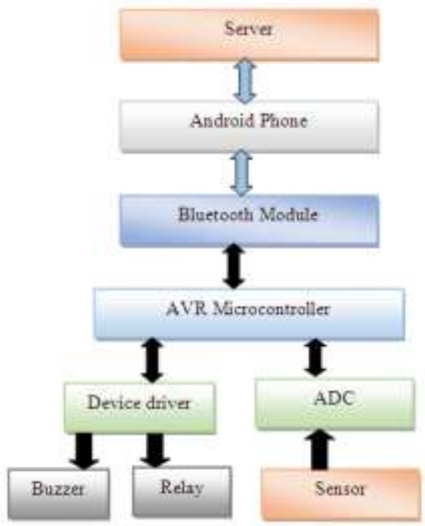

Figure 1: Block Diagram

Microcontroller is connected to Bluetooth module which is synchronized with android mobile phone. On android mobile phone, we built an application which shows the different parameters readings which we are going to sense these reading are send to the remote server (laptop) through internet or SMS the readings can be viewed in graphical manner on the server to detect the malfunctioning of the vehicle.

This block diagram represents combination of a low cost hardware unit and user friendly android based mobile application software utilized to create on board vehicle diagnostic system. The server is held at the maintenance department where it receives the readings in the form of HTTP/TCP packets. The server should have a static public IP address to which the application packets are destined.

4. SENSORS

Sensor	Component
Fuel sensor	Potentiometer
Speed sensor	Optocoupler
Temp. sensor	LM 35
Seat belt sensor	Mechanical switch

Table No.1: List of Sensors

Temperature Sensor

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package

Fuel level sensor

In our project we are using the potentiometer as the fuel level sensor. The Floating material is used for the variation in the voltage of potentiometer. Electro-mechanical liquid level sensors are often designed with a float arm pinned to the center of a rotary potentiometer. This design concept is used in most automotive fuel level senders because it offers the potential for both long life and low cost. The float arm is mounted vertically, and liquid level changes produce a rotary motion for the potentiometer contacts. This change of position alters the resistive value of the sensor.

Speed sensor

In this project optocoupler used as a speed sensor Basically the simplest way to visualise an optocoupler is in terms of its two main components: the input LED and the output transistor or diac. As the two are electrically isolated, this gives a fair amount of flexibility when it comes to connecting them into circuit. All we really have to do is out a convenient way of turning the input LED on and off, and using the resulting switching of the phototransistor/ diac to generate an output waveform or logic signal that is compatible with our output circuitry. Vehicle wheel has hole circular manner. On one side LED is present and other side photodiode present. As one cycle completes, LED light passes through hole on phototransister. Pulse gets generated at output. Number of pulses equal to number of cycles, according to this speed measurement is done.

Seat belt Sensor

We are using the simple mechanical switch as a seat belt sensor. In one type of micro-switch, internally there are two conductive springs. A long flat spring is hinged at one end of the switch (the left, in the photograph) and has electrical contacts on the other. Seat belt sensor is used for the safety.

V. MICROCONTROLLER *Atmega16*

In this section, we describe the overall architecture of the Atmel AVR ATmega16.We begin with an introduction to the concept of the reduced instruction set computer (RISC) and briefly describe the Atmel Assembly Language Instruction Set. A brief introduction is warranted because we will be programming mainly in C throughout the course of the book. We then provide a detailed description of the ATmega16 hardware architecture.

This is the basic hardware configuration of ATmega16.PORT A is configured with eight tact (momentary) switches with accompanying de-bouncing hardware. PORTB is equipped with an eight-channel tristate LED indicator. For a given port pin, the green LED will PORTA of the controller, there are power (pins 10, 30, and 32) and ground (pins 11 and 31) connections

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers

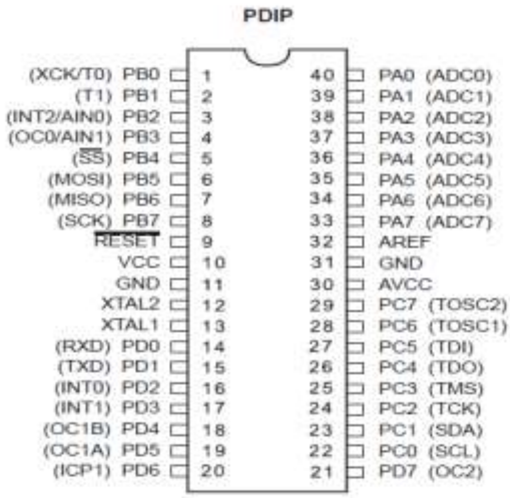

Figure 2: Atmega16

VI. BLUETOOTH TRANCEIVER

The microcontroller is programmed to send the measurements on its UART to the Bluetooth module. Bluetooth provides a means to connect devices such as mobile phones over a secure, globally unlicensed short-range radio frequency (2.45 GHz) and to enable the exchange of information between them. We used the BluCOM-18 Bluetooth transceiver module, which is a class 1 (20 dBm) model that has an approximate range of 100 meters. The asynchronous data from/to the Atmega 16 microcontroller is delivered to/from the Bluecomb18 Bluetooth module on the serial port at a speed of 9600 bps. The Bluetooth module is configured as a Slave and the mobile phone is considered to be functioning as a Master. The microcontroller sends/receives data to/from the Bluetooth module, which transmits/receives data continuously as raw binary bytes. Bluetooth employs a radio technology called frequency hopping spread spectrum, where data transmitted is chopped into chunks, which are transmitted on up to 79 bands, each with a bandwidth of 1 MHz centered from 2402MHz to 2480MHz. Equations

Figure 3 :Bluetooth module

VII. CONFIGURING Atmega16 and BluCOM-18

Prior to installing the Atmega16 and BluCOM-18 in the complete system circuit as shown in Figure 4, they needed to be programmed to serve their intended purpose and functions. They was performed using an ASCII terminal software to issue configuration commands. Since both the Atmega16 and the BluCOM-18 are TTL/CMOS based ICs, while the personal computer's serial communications port uses RS232 voltage levels, an interface is needed to convert between the different voltage levels. For this purpose, a properly connected a MAX232 IC was used.

8. MOBILE APPLICATION SOFTWARE

Android is an operating system which is based on the JAVA programing language and runs on Linux kernel. The Android platform is made up of the operating system, middleware, user interface and application software [1]. In addition to Android, there are several different operating systems in the market for smart phones such as Symbian, Windows Mobile, RIM, iPhone OS (iOS), … etc. Android has four distinguishing advantages when compared with the other mobile phone operating systems:

1) It is an open mobile platform; Users can customize and expand applications according to their needs.

2) All applications are equal, where all applications are run in virtual machine resources.

3) The application programs have no boundaries. The developers can combine the data of the World Wide Web and the locally available (stored) in the Android platform because Android can access the core mobile devices and Internet through the standard API.

4) The application development is quick and easy since the Android platform extends a great deal of useful libraries and tools to the developers. Our Android mobile application software was designed to perform the following tasks:

a) Connect to the Bluetooth module.

b) Send request messages to the OBD system.

c) Receive responses from OBD system.

d) Display the responses to the user in a user friendly form (values in decimal).

e) Be able to upload the values to a remote server when desired. The request messages are within the application software formed as explained previously in the message format section. Some indicators need to be calculated from the parameters obtained from the vehicle such as "fuel economy." The geographical position is obtained from the GPS sensor incorporated within the mobile device and combined with other parameters in the message.

Development of the Android Application Software

We have developed our Android mobile application software on a Windows® 7 platform for an Android 2.2 driven Samsung Galaxy S phone. We installed development environment preparation software which included the java development kit (JDK), Eclipse, Android software development kit (SDK), Android virtual devices (AVD). Meanwhile, Android development tools (ADT) is the plug-in through which Eclipse is customized for Android applications development. It provides a powerful integrated environment and extends the functions of Eclipse that allows users to create applications quickly and add components on the API. AVD is a collection of virtual devices where each AVD simulates a virtual device to run the Android platform and test the application software before they are run (tested) on the actual physical mobile phone (device).

B. Mobile Application Software Testing and Menus

In order to test and verify the proper operation and functions of this Android-based application software, the hardware interface unit was connected through an OBD connector to a Chevrolet Optra 2007, which operates on the ISO-9141-2 protocol. The circuit immediately responded to the connection as the LEDs light in the correct sequence, while a Bluetooth pairing with application software installed on the Android-based Samsung Galaxy S mobile phone was a success. The mobile applications configuration software menu allows for personal settings of features such as turning GPS *ON* or *OFF*, duration between consecutive reading from OBD, custom (specific) parameters for certain vehicles.

Also, the desired commands or requested parameters to be read and displayed are under the control of the driver, where they are selected from

Figure.4. Configuration menu on the mobile application

Software the function modules submenu options as depicted in Fig 8. Some vehicles real-time parameters include throttle position, intake manifold pressure, ignition advance angle, engine speed and vehicle speed. After the requested test parameters were selected, the test was a success, as the system functioned as expected displaying readings taken from the vehicle's sensors on the mobile screen. Figure 9 shows a screen shot taken from the Android mobile device displaying the measured and calculated readings. Note that an error was present, as the application failed to measure the Mass Air Flow. This is due to the vehicle's inability to measure this value, because this particular car model does not support this PID.

C. Remote Server Set-Up

The server is held at the maintenance department where it receives the readings in the form of HTTP/TCP packets. The server should have a static public IP address to which the application packets are destined. The mobile application transfers data via cellular internet from the integrated OBD system. The server performs analysis on the received parameters and uses this data to detect any malfunctions present in the vehicle [12], or predict faults that may occur in the future. The server is a PHP programmed server that contains a MY SQL database. The database holds a NULL table. In the table, there is a field for each PID where the readings from the OBD system are recorded. To make sure that the fault detection and prediction is done correctly the server needs to receive real time data, hence, the application at the mobile device needs to have a high data rate broadband connection such as that supported by 3G cellular Internet.

9. CONCLUSION

We implemented a universal integrated system which is composed of a combination of a low-cost hardware unit and a user-friendly Android-based mobile application software utilized to create an on-board vehicle diagnostic system. The mobile application software will interact with the hardware interface unit wirelessly via Bluetooth to acquire desired vehicle parameters from the ECU of the vehicle. These readings will be displayed locally to the user then can be sent to a remote maintenance server as HTTP packets via a cellular internet connection. The packets received will be tabulated in the server, then made use of by the maintenance department which holds the server.

REFERENCES:

[1] T. Menard, and J. Miller, "Comparing the GPS capabilities of the iPhone 4 and iPhone 3G for vehicle tracking using Free Sim_ Mobile," 2011 IEEE Intelligent Vehicles Symposiium (IV), pp. 278–283, June 2011.

[2] N. Navet, Y. Song, F. Simonot-Lion, C.Wilwert, "Trends in Automotive Commutication Systems", Proceedings of IEEE, Vol.93, pp 1204-1223,2005.

[3] S. Carter, Nissan Releases Andriod App for Leaf Vehicle, http:// nissanleaf.net/, August 4 2011.

[4] J. Barkenbus, "Eco-driving: An overlooked climate change initiative," Energy Policy, Volume 38, Issue 2, February 2010, Pages 762-769.

[5] M. Johanson, and L. Karlsson, "Improving vehicle diagnostics through wireless data collection and statistical analysis," IEEE Vehiclular Technology Conference (VTC-Fall 2007), pp. 2184-2188, September 2007.

[6] L. Ferhatovic, A. Lipjankic, A. Handzic, and N. Nosovic, "System for remote diagnostic of vehicle defects", *in Proc.17th Telecommunications Forum (TELFOR)*, pp. 1323-1326, November 2009.

[7] A. Tahat, M. Khalaf, O. Elmuhesen, "A solar energy water heater remote monitoring and control system," *International Conference on Electronic Devices, Systems, and Applications (ICEDSA)*, pp. 98-103, April 2011.

[8] A. Tahat, A. Sacca, Y. Kheetan, "Design of an integrated mobile system to measure blood-pressure", *in Proc. IEEE 18th Symposium on Communications and Vehicular Technology (SCVT)*, pp. 1-6, November 2011.

[9] Press release Canalys, "Andriod takes almost 50% share of worldwide smart phone market, Palo Alto, Singapore and Reading (UK), August 2011.

[10] Atmel AVR Microcontroller Primer: Programming and Interfacing Steven F. Barrett University of Wyoming Daniel J. PacK

[11] C. Haseman, *Andriod Essentials*, Berkeley, CA: A Press, 2008

[12] H. Gelgele, Kesheng Wang, "An expert system for engine fault diagnosis: development and

[13] application," Journal of Intelligent Manufacturing, pp. 539-545, 1998.