PAPER • OPEN ACCESS

Diagnostics vehicle's condition using obd-ii and raspberry pi technology: study literature

To cite this article: J V Moniaga *et al* 2018 *J. Phys.: Conf. Ser.* **978** 012011

View the article online for updates and enhancements.

Diagnostics vehicle's condition using obd-ii and raspberry pi technology: study literature

J V Moniaga, S R Manalu, D A Hadipurnawan and F Sahidi

Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480

Email: jurike@binus.edu, smanalu@binus.edu dionisius.hadipurnawan@binus.ac.id firda.sahidi@binus.ac.id

Abstract. Transportation accident rate are still being a major challenge in many countries. There are many factors that could be cause transportation accident, especially in vehicle's internal system problem. To overcome this problem, OBD-II technology has been created to diagnostics vehicle's condition. OBD-II scanner plugged to OBD-II port or usually called Data Link Connector (DLC), and after that it sends the diagnostics to Raspberry Pi. Compared from another microcontrollers, Arduino, Raspberry Pi are chosen because it sustains the application to receive real-time diagnostics, process the diagnostics and send command to automobiles at the same time, rather than Arduino that must wait for another process finished to run another process. Outcome from this application is to enable automobile's user to diagnostics their own vehicles. If there is found something unusual or a problem, the application can told the problem to user, so they could know what to fix before they use their vehicle safely.

1. Introduction

On-Board Diagnostic is a computer-based system developed by Automobile manufacturers for diagnosing car. It focuses on diagnosing the performance of car's engine to check any errors in car's engine components [1]. OBD-II connect with DLC in the vehicle. Almost all the problem with the vehicle can be detect by OBD-II, such as exhaust manifold, ABS brake function, airbag (iSRS & SRS), inle performance, and so on. However, OBD was only for scanning the vehicle's diagnostics. To reading, processing, and showing the data, it needs Microcontroller. Microcontroller is a microprocessor system that installed in the chip. Microcontroller is different with microprocessor that used by PC because usually general microcontroller already filled with supporting minimal component system, such as memory and I/O interface. In another side, microprocessor usually only filled by CPU [2]. Nowadays a lot of tools that can control device digitally. However, we have choosen two of the popular models and compared their feature. Those two models are, Arduino Uno and Raspberry Pi. Arduino is an open source microcontroller board that able to connect to computer and others devices via universal serial bus (USB). Arduino can be powered from any power supply such as computer, battery, and more. [3]. There a lot of various of Arduino to choose for every needs, to conform with our research we chose to use Arduino Uno.

Figure 1. From left to right: An Arduino Uno [4], a Raspberry Pi [4], iSaddle Bluetooth OBD-II Scanner[5]

The Raspberry Pi is an inexpensive credit-card sized computer, that has 700Mhz processor, 512MB of SDRAM, on board Ethernet, HDMI and RCA interfaces, a dual USB connector and an SD card slot. In addition is maintains an array of input/output pins that make the Raspberry Pi extraordinarily conductive to tinkering and device innovation [6].

Table 1. Comparison of Arduino Uno and Raspbery Pi [4]

Indicator	Arduino Uno	Raspberry Pi
Model Tested	R3	Model B
Processor	ATMega 328	ARM11
Clock Speed	16Mhz	700Mhz
RAM	2KB	256MB
Ethernet	N/A	10/100
USB Master	N/A	2 USB 2.0

Raspberry Pi has been chosen rightly as the single board computer for this application because it has the highest performance compared to Arduino (Arduino still use ATMega 328 for its processor and Raspberry Pi already use ARM11 which has higher specification) and Arduino cannot run multiple process at the same time since it only has 2KB size of RAM. In addition, Raspberry Pi has input port for Ethernet and USB, which make it easier to make it connected to internet connection and connect to another device.

2. OBD-II Scanners

Recently, On Board Diagnosis-II (OBD-II) scanner have been introduced to use a vehicle network (OBD-II) for the diagnosis and checking of a vehicle, which has previously used many devices such as black boxes, vehicle diagnostic equipment and vehicle supply management [7]. OBD is a standard tool that allows driver to check car's engine status by plug the OBD to the porst that placed under the driving wheel. Mostly OBD can get the status of emission control, driving speed, voltage control, engine coolant temperature, idle time, and more [8]. OBD-II is mostly used with extension tools such as gas analysers and vehicle telematics [9]. The OBD scanner is a costly device and is mainly owned by workshops [10]. We are able to access this car's ECU with mobile device by connected the OBD-II to the port and connected to the mobile device via OBD-II connector. Then OBD-II will export the data to the mobile device after CAN-Bus convert the data. [11]. There are many scanners that could be connected to OBD-II, for the example, one of the most basics scanner, U480 CAN OBD-II Diagnostic Scanner. This scanner has a basic LCD display to give information about vehicle on board diagnostics that connected to it, but need wire to connect to the vehicle. The developed scan tool is compliant with various SAE (CAN) standard communication protocols to connect with the ECU installed in the vehicle [12]. On board diagnostics is a tool that have a function to diagnosing and reporting the car's engine condition. It is the interface between cars Controller Area Network (or CAN bus) and external devices, e.g., a OBD scan tool that connects the OBD port and a laptop. All components of car's vehicle are able to communicate with each other with CAN-Bus. For example, a position value is sent to the Electronic Control Unit (ECU) after human driver press the gas pedal, and the ECU adjusts air/fuel injections according to the position value. Therefore, we can read this position message transmitted over CAN bus from the OBD port [13]. Now scanners mostly are wireless for example there are 3 scanners that most used for now, first is ERUSUN Bluetooth Mini OBD-II Scanner, that serving a wireless connection between user vehicle's computer and user device's via Bluetooth, then there is iSaddle Bluetooth / Wi-Fi Wireless

OBD-II Scanner, is a scanner that serving a connection both Wi-Fi or Bluetooth and can connect to tablet, smartphone and computer. ELM-327, is one such adapter where the data read from the OBD2 port are transmitted via Bluetooth upon pairing [14]. Last, there is iSaddle OBD-II Wi-Fi / Bluetooth Scanner that connect also with both Bluetooth or Wi-Fi [15].

3. iSaddle Bluetooth OBD-II Scanner on Android

It is necessary to know the OBD2 protocols and convert the OBD2 signals to the serial data standard for the PC [16]. iSaddle Bluetooth OBD-II Scanner is more suitable to this application than iSaddle WI-FI OBD-II Scanner because this application require internet connection from the devices. When the device connect to this application via Wi-Fi, it would make impossible for the device to connect to the internet via Wi-Fi. In this case, user must choose between connect into this application but using mobile data, or not using this application at all. Furthermore, Bluetooth wireless technology is an open specification that enables low-power and short range wireless connection [17]. Since this application will be used while user in the vehicle that does not has any electricity to charge the smartphone, low-power wireless technology would be a great solutions. By using the OBD2 Bluetooth interface, the information on real instantaneous fuel consumption is also acquired from the vehicle engine [18]. OBD2 speed is taken as benchmark for true vehicle speed. Bluetooth based OBD2 device is installed in the vehicle, which collects vehicle speed at 1sec interval and transmits it a mobile phone [19]. OBD-II does not provide a user friendly data and meaningful, although there are a lot of research that claim there is a possibility to analyse driver behaviour using real-time data from OBD-II's diagnosing data. [20]. Nowadays those researches focusing on to use multi-source information fusion to get better interpret of relationship of driver's behaviour and car's engine performance.[21] Unfortunately the analysis part is done off-line after the data gather, so OBD-II can't give a real-time driver's support [22].

iSaddle Bluetooth OBD-II Scanner has a small size, compact structure, and supports all OBD-II protocols. Besides, stable wireless connection, no wires, and no batteries are some extra feature that iSaddle OBD-II has [23]. To work, iSaddle Bluetooth OBD-II Scanner require the vehicle that has 16-pin Data Link Connector (DLC) under the vehicle's dash [24]. Also, that must be an information on vehicle's emission control label that the vehicle is OBD-II compliant. This scanner then get plugged to DLC and it connects to any computer, tablet, or smartphone [25]. There are many diagnostic software that associated with iSaddle Bluetooth OBD-II scanner. With these software, it would be able to read diagnostic trouble codes, clear trouble codes, turn off the check engine light, set the maximum number of responses to obtain, removes spaces from Engine Control Unit (ECU) responses (to make data transmission more speed), store unique identifier in the Electronically Erasable Programmable Read Only Memory (EEPROM), set the CAN mask and filter with one command, easily switch between variable and fixed CAN message lengths, send CAN remote frames (RTRs), and many more [23].

The car's engine trouble indications or commonly called by Diagnostic Trouble Codes (DTC) is save in the system. Those codes are not the same in all vehicles, and not all country that produce vehicle use the manufacturer codes. However, now everyone is able to connect OBD-II to the port and diagnosing the car's engine trouble then identify the trouble data [26]. Nowadays freeware and commercial software packages are available, that allow to monitor OBD-II vehicle data by using just a smartphone and off-the-shelf scan tools. There a lot of software that able to monitor car's engine by using smartphone and some external tools. However all on-board diagnosing systems only display the low-level data. OBD-II doesn't provide a user friendly data and meaningful, although there are a lot of research that claim there is a possibility to analyse driver behaviour using real-time data from OBD-II's diagnosing data. Nowadays those researches focusing on to use multi-source information fusion to get better interpret of relationship of driver's behaviour and car's engine performance. Unfortunately the analysis part is done off-line after the data gather, so OBD-II can't give a real-time driver's support.

4. Process in OBD-II

To set up the OBD scanner tool is not a uniform process since there are many varieties of types and brands of vehicle and the OBD scanner itself. But generally the process can be described like this, first user must download the application for the mobile and it depends on the OBD scanner tool itself, because different tool may require different application. This step should be done well before user's smartphone connect to the OBD scanner tool. Next step user must connect the OBD scanner to the DLC (Data Link Connector), usually it was located under the steering wheel. DLC consist of 16 pins. This step also must

be done well because it will connect directly to the OBD scanner tool. As a concern when user is installing the OBD scanner to the DLC, this is the following list of 16 Pin DLC and its function, These 16 pins include: (1) Manufacturer reserved pin, (2) The J1850, (3) Manufacturer reserved pin , (4) The chassis ground connector,(5) The signal ground , (6) The CAN high, J-2284, (7) The k line, ISO9141-2/DIS 14230-4connector port, (8) Manufacturer reserved pin, (9) Manufacturer reserved pin, (10) J1850 Bus, (11) Manufacturer reserved port, (12) Manufacturer reserved port,(13) Manufacturer reserved port,(14) CAN, J2284 PORT,(15) The L line, ISO 9141 and ISO/ DIS 14230-4, (16) The battery power connection. Since there are many varieties of types and brand of vehicle also the OBD scanner itself and this process is a general process, there are some

(a)   **(b)**   **(c)**
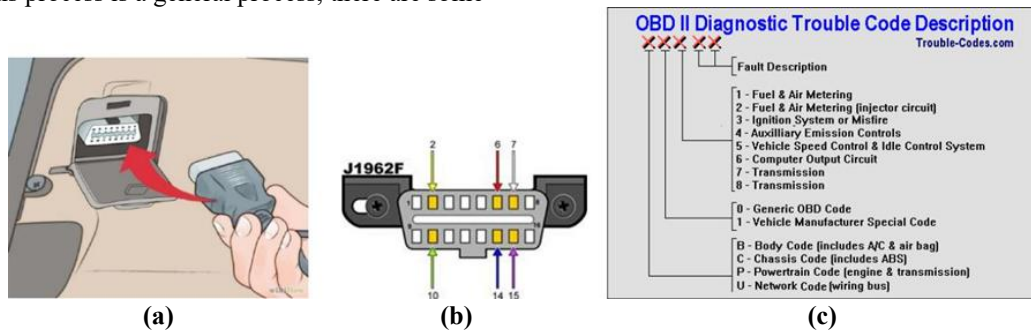
Figure 2. (a) Data Link Connector and OBD Wire, (b) 16-Pins of OBD-II [28], (c) OBD-II Diagnostics Trouble Code Description [29]

The other concern is DTC (Diagnostic Trouble Code), DTC is a set of code consisting of letters and numbers, where each code represents type of fault on user's vehicle. DTC originally made by SAE (Society Automotive Engineers) for use by all vehicle manufacturers who have to comply with OBD 2 emissions regulations in the U.S and adopted by European and Asian market. There are 4 basic categories of DTC, Powertrain (P) category covers functions that include engine, transmission and associated drivetrain accessories, Body (B) category refers to functions that are, generally, inside of the passenger compartment. These functions provide the driver with assistance, comfort, convenience, and safety. Chassis (C) category represent functions that are, generally, outside of the passenger compartment. These functions typically include mechanical systems such as brakes, steering and suspension. Network and Vehicle Integration (U) category is for functions that are shared among computers and systems on the vehicle. DTC also divided to 2 major groups, Generic OBD Codes are code that have "0" as second digit to represent they are common to all makes and models of vehicles also for basic emissions fault diagnosis, Vehicle Manufacturer Special Codes are code that have 1 as second digit to represent unique vehicle to a particular vehicle make or model [30].

5. Process in Raspberry Pi
The Raspberry PI is not a fixed function device, unlike many mobile devices. Although a mobile device often has a camera, it is usually impossible to add extra hardware to most mobile devices. Using the Raspberry Pi, it is possible to play with rudimentary hardware. There are multitudes of the resources devoted to creating innovative systems using the Raspberry Pi as a foundation, from supercomputers to time-lapse cameras. One of the advantages of the Raspberry Pi is that they limit the resources available to the programmer, who must then develop the cleaner and better programming habits as the programming projects gain complexity. The restriction on resources will likely force the design of more efficient programs [6]. To work with the iSaddle Bluetooth OBD-II scanner, the Raspberry Pi needs a Bluetooth scanner to be pairing with. There are several commercial Bluetooth scanner, but in this application will use the Raspberry Pi Nano Bluetooth Dongle, which is the perfect low cost solution and easy to install. The first step before using The Bluetooth Dongle is to install the latest Raspbian Operating System (OS) in the Raspberry Pi which where the data will be shown and processed. Raspbian OS is a free operating system based on Debian, optimized for the Raspberry Pi hardware [31]. The next step is download pyOBD. pyOBD is an open source OBD-II compliant scan tool software that written

entirely in Phyton. It is made with OBD-II diagnostic interfaces. It will allow the user to see their vehicle's ECU data, display DTC, measured values, status tests, and more [32].

6. Application Process

The recommend system is able of diagnosis the vehicles, collecting, storing, and also retrieve the command from mobile application. Vehicle diagnosis data are collected using an iSaddle OBD2 Bluetooth scanner and are sent to Raspberry Pi via Bluetooth. Raspberry Pi needs Nano Bluetooth Dongle as extra tools that enabled it to retrieve and send data using Bluetooth. After the diagnostics data retrieved to Raspberry Pi, pyOBD will display the interface of the diagnostic data based from OBD-II diagnostics trouble codes and process it. To process the diagnostics data, this application is using Node.JS as server-side language. Compared to PHP that only handles small requests well, but struggles with large requests Node.JS has much better performance in high concurrency situation, no matter in benchmark test or scenario tests [33]. These data are encapsulated to JSON Objects, it will be store to MongoDB (NoSQL document oriented database management system [34]). MongoDB provides flexibility during the development process and easy to deploy and copy databases from one server another server compared than its competitor, Oracle Databases [34], which makes MongoDB is more suitable in this application. By connecting to internet connection, JSON Objects that stored in MongoDB can be parsed to user's smartphone through this application.

Users are also enabled to send their own command to their vehicle's. User must input their command in this application to do so. Next, the application will generated the input into JSON Objects and stored to MongoDB. The Node.JS API will process the JSON Objects into ELM-USB command list. These ELM-USB command lists are send to iSaddle OBD-II Bluetooth Scanner, and it will triggered the vehicle to do the command from the user.

(a)   **(b)**
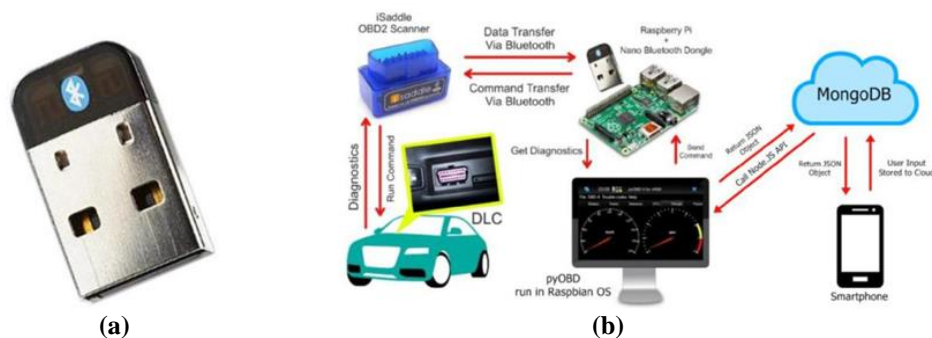
Figure 3. (a) Raspberry Pi Nano Bluetooth Dongle [31]; (b) Application Process Flow

7. Conclusion

Vehicle's diagnostic get much attention from industry and researchers in recent years. The variety and heterogeneity of vehicle diagnostics implementation has been the major reason which makes it interested. This paper presents a technique to analyse diagnostics from vehicle that connected to OBD-II and process the diagnostics data using Raspberry Pi. People can also send a command to their own vehicle by using this application. Although the process occurs only the delivery of vehicle diagnostic data to user's smartphone, Raspberry Pi is more suitable viewing the ability of Raspberry Pi that can be multitasking. In the future research that related to OBD-II scanner, its recommended to use OBD-II scanner with Bluetooth rather than Wi-Fi, because it's more energy saving, easier to use and more stable.

References

[1]   Graham N J 2008 *Wireless Automotive Data Link Connector* US7584030 B1.
[2]   Nurfalah M 2014 *Makalah Arduino dan Raspberry Pi Mikrokontroler* p 9.
[3]   Monk S 2012 *Programming Arduino Getting Started with Sketches* (McGraw-Hill Education)

[4]   Brandon 2013 Arduino Uno vs Raspberry Pi vs BeagleBone Black *MCM Electronics*. [Online]. Available: http://blog.mcmelectronics.com/post/Arduino-Uno-Raspberry-Pi-and-BeagleBone-Black#.Ua753EDLpWI.

[5]   Amazon 2017 iSaddle Super Mini Bluetooth OBD2 OBDII Scan Tool Check Engine Light & CAN-BUS Auto Diagnostic Tool for Windows & Android Torque (Blue Color, Super Mini). [Online]. Available: https://www.amazon.com/iSaddle-Bluetooth-CAN-BUS-Diagnostic-Windows/dp/B00OMT6UEK. [Accessed: 07-Apr-2017].

[6]   Wirth M and McCuaig J 2014 Making Programs With The Raspberry Pi, *Proc. West. Can. Conf. Comput. Educ. - WCCCE '14* pp 1-5.

[7]   Baek S H and Jang J W 2015 Implementation of Integrated OBD-II Connector With External Network *Inf. Syst.* **50** pp 69–75.

[8]   Wengert J 2005 On-board diagnostics in engine management systems for motorcycles, *SAE Tech. Pap.* pp 1–6.

[9]   Hilpert H, Thoroe L and Schumann M 2011 Real-Time Data Collection for Product Carbon Footprints in Transportation Processes Based on OBD2 and Smartphones *Proc. of the Annual Hawaii Int. Conf. on System Sciences*.

[10]   Hasan N N, Arif A, Pervez U, Hassam M and Ul Husnain S S 2011 Micro-Controller Based On-Board Diagnostic (OBD) System for Non-OBD Vehicles *Proc. - 2011 UKSim 13th Int. Conf. on Modelling and Simulation* pp 540–544.

[11]   Türker G F and Kutlu A 2016 Survey of Smartphone Applications Based on OBD-II for Intelligent Transportation Systems *J. Eng. Res. Appl. www.ijera.com* **6** p 2248.

[12]   Villatico F and Zuccari F 2008 Efficiency comparison between FC and ice in real urban driving cycles *Int. J. Hydrogen Energy* **33** pp 3235–3242.

[13]   Kang L, Qi B, Janecek D and Banerjee S 2015 EcoDrive: A Mobile Sensing and Control System for Fuel Efficient Driving *Proc. of the 21st Annual Int. Conf. on Mobile Computing and Networking* pp 358-371.

[14]   Amarasinghe M, Kottegoda S, Arachchi A L, Muramudalige S, Bandara H L M D and Azeez A 2015 Cloud-Based Driver Monitoring and Vehicle Diagnostic With OBD2 Telematics *IEEE Int. Conf. on Electro Information Technology*.

[15]   Ntziachristos L, Fragkiadoulakis P, Samaras Z, Janka K and Tikkanen J 2011 Exhaust Particle Sensor for OBD Application *SAE Tech. Pap.*

[16]   Hermann A and Desel J 2008 Driving Situation Analysis in Automotive Environment *Proc. of the 2008 IEEE Int. Conf. on Vehicular Electronics and Safety ICVES* pp 216–221.

[17]   Yu SN and Cheng JC 2005 A Wireless Physiological Signal Monitoring System with Integrated Bluetooth and WiFi Technologies *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **3**.

[18]   Astarita V, Guido G, Mongelli D and Giofrè V P 2015 A Co-operative Methodology to Estimate Car Fuel Consumption by Using Smartphone Sensors *Transport* **30** pp 307–311.

[19]   Chowdhury A, Chakravarty T and Balamuralidhar P 2014 Estimating True Speed of Moving Vehicle Using Smartphone-Based GPS Measurement *IEEE Int. Conf. Syst. Man, Cybern.* pp 3348–3353.

[20]   Choi S, Kim J, Kwak D, Angkititrakul P and Hansen J H L 2007 Analysis and Classification of Driver Behavior Using In-Vehicle Can-Bus Information *Bienn. Work. DSP In-Vehicle Mob. Syst.* pp 17–19.

[21]   Quintero G C M, Lopez J A O and Rua J M P 2010 Intelligent Erratic Driving Diagnosis Based on Artificial Neural Networks *IEEE Andescon* pp 1–6.

[22]   Ruta M, Scioscia F, Gramegna F, Loseto G and Di Sciascio E 2012 *Knowledge-based Real-Time Car Monitoring and Driving Assistance* (Padova, Italia: Edizioni Libreria Progetto).

[23]   Amazon *iSaddle Bluetooth OBDII Scan Tool User Manual*.