Connected car: Engines diagnostic via Internet of Things (IoT)

A F A Hamid, M T A Rahman, S F Khan, A H Adom, M A Rahim, M H N Ismail and A Norizan

School of Mechatronic Engineering, Universiti Malaysia Perlis, Pauh Putra Campus, 02600 Arau, Perlis, Malaysia.

E-mail: azarul.fahmin@gmail.com

Abstract. This paper is about an experiment for performing engines diagnostic using wireless sensing Internet of Thing (IoT). The study is to overcome problem of current standard On Board Diagnosis (OBD-II) data acquisition method that only can be perform in offline or wired method. From this paper it show a method to determined how the data from engines can be collected, make the data can be easily understand by human and sending data over the wireless internet connection via platform of IOT. This study is separate into three stages that is CAN-bus data collection, CAN data conversion and send data to cloud storage. Every stage is experimented with a two different method and consist five data parameter that is Revolution per Minute (RPM), Manifold Air Pressure (MAP), load-fuel, barometric pressure and engine temperature. The experiment use Arduino Uno as microcontroller, CAN-bus converter and ESP8266 wifi board as transfer medium for data to internet.

1. Introduction

Engine diagnostic with Internet of Things (IoT) is a system to overcome the shortcomings of the conventional engine diagnostic systems. Specifically, it is a system that sends data over the On Board Diagnosis (OBD-II) connector using a wireless system that connects to the Internet. The device used for accessing and transmitting the data is simple, low-cost, and easy-to-install. Stakeholders worldwide are struggling to finalize standards to allow specialized wireless on-board units (OBUs) to directly interact with the control area network (CAN-bus) and to communicate with other OBUs and with roadside units, under any propagation condition, intermittent connectivity, and traffic density [1]. There are many wireless network platform that can be used such as Big Data collector through mobile ad hoc network (MANET), wireless sensor networks (WSN) or cyber physical network (CPS) but the better network type is CPS that is the IoT type of network [2]. As for this project, the use of a wireless input and output terminal capable of accessing the Internet to transmit and receive any data from the Electric Control Unit (ECU) and any data from sensors that are attached to the car. Data transmitted from the racing car also can be analysed for purposes unrelated to engine or car component problem. For example, the data can be collected and analysed in real-time to analyst driving patterns, racing car part reliability, and emission characteristics. By using this IOT system, it able to overcome the problem of condition car monitoring. The idea in using IOT system is to perform a wireless dyno test. By using this application, it can perform dyno test without being on the chassis dyno machine. These augmented data can be processed by remote control centres for multiple purposes, such as to inform fleets of vehicles cooperatively driving [3], to detect possible in-vehicle malfunctions, to collect traffic statistics, to get...
car maintenance tips and service information. This paper presents a proof-of-concept for probe vehicle monitoring and for the exploitation of the collected data. Three system platform are considered: CAN-bus data collection, CAN-bus data conversion and sending raw data from converted CAN-bus data to cloud for storage.

2. Methodology

As for this project, there are three system platform is considered. CAN-bus data collection for the purpose of collecting data from ECU to Arduino with using CAN-bus converter. CAN-bus data conversion for converting byte data that receive by Arduino into decimal raw data that can be processed, calculate and easy to understand by human. Also the part of sending converted CAN-bus data to cloud storage via IoT. Diagram 1 shows the flow chart of experiment to achieve the target.

Diagram 1: Block diagram for engines diagnostic via internet of things (IOT).

2.1. CAN-bus data collection.

CAN-bus was first engendered for automotive use, so its most prevalent application is in-conveyance electronic networking. However, as different industries have completed the responsibleness and benefits of will over the past twenty years, they need adopte d the bus for a good form of applications. These physical layers relegate bound aspects of the will network, like electrical levels, signaling schemes, cable electric resistance, most baud rate rates, and more. As for automotive, all types of information area unit constantly collected by sensors on board of mundane cars and trucks [4]. These information area unit communicated over a CAN-bus. Depending on the kind of conveyance the data can be read through the OBD plug or by an immediate connection with the CAN-bus itself utilizing a CAN-bus clamp. Several implements were tested for the perpetual monitoring of data on the CAN-bus through the OBD plug. Several communication techniques and set-ups were tested for retrieving the data. A relevant selection CAN-bus data was made and an algorithm was devised in order to collect engine data parameter. During the experiment there two method has been tried. The first method is CAN-bus transceiver TJA1050 as in Figure 1 and Figure 2 that collect data from CAN-bus then send it to microcontroller via serial communication. The figure show block diagram that consist several part that such as part of the board communicate with CAN-bus from ECU via connection of CANH and CANL.
Figure 1: TJA1050 integrated circuit(IC) diagram

Figure 2: TJA1050 board use to collect data from CAN-bus port on ECU.

But the system have a data clocking problem due to the TJA1050 just send data without segregate it by every data packet. Another method that use TJA1050 and MCP2515 IC as in Figure 3 and 4. MCP2515 IC is a serial communication to serial peripheral interface (SPI) bus is tested.
Figure 3: MCP2515 and TJA1050 circuit diagram

Figure 4: MCP2515 AND TJA1050 CAN-bus converter board.

With this board the system is able to clocking the data packet. Every byte will not messed up among them and it will help the data packet selection and conversion part. CAN-bus contrivances send data across the CAN-bus network in packets called frames [5].

A CAN-bus frame consists of the following sections:-
(a) CAN-bus Frame - an entire CAN-bus transmission: arbitration ID, data bytes, acknowledge bit, and so on or it can be call data packet. Frames additionally are referred to as messages.
(b) SOF (start-of-frame) bit – designates the commencement of a message with an ascendant (logic 0) bit.
(c) Arbitration ID – identifies the message and betokens the message's priority. Frames come in two formats -- standard, which utilizes an 11-bit arbitration ID, and elongated, which utilizes a 29-bit arbitration ID. For this experiment the system get 11-bit ID such as 0x360, 0x3E1 and etc.
(d) Data Field – contains 0 to 8 bytes of data.

2.2. CAN-bus data conversion.
Data acquisition is one amongst the most effective tools to extend the understanding of car behaviour. The network most commonly employed in automotive applications is that the CAN-bus. As mention in CAN-bus data collection the data that being communicate in form of byte [6]. To make the other device understand the data that being transferred the system need a data converter that can convert data in form of byte to decimal [7]. In this experiment there are two method of converting data and five type of data from ECU as in Table 1.

| Method            | Explanation                                      |
|-------------------|--------------------------------------------------|
| Conversion method 1 | Test condition based on every byte and program choose the decimal value for every decision. If(byte[i] == F){ Dec[i]=16; } If (byte[i] == E){ Dec[i]=15; } Total=Dec[2]^3+Dec[1]^2 |
| Conversion method 2 | String to long integer converter. long int strtol(const char *str, char **endptr, int base); |

2.3. Sending raw data to cloud.
There are many sensoring network that has been introduced such as mobile ad-hoc network (MANET) Global System for Mobile Communications (GSM) and wireless detector networks (WSN). A lot of recently, the cyber physical system (CPS) has emerged as a promising direction to counterpoint human-to-human, human- to-object, and object-to-object interactions within the physical world in addition as within the virtual world [2]. CPS with Represenational State Transfer (REST) architecture is used as a medium for sending data to cloud. Basically REST is associate art form, associated an approach to communications that's usually utilized in the event of internet services. the utilization of REST is commonly most well-liked over the additional heavyweight SOAP (Simple Object Access Protocol) vogue as a result of REST doesn't leverage the maximum amount information measure, that makes it a far better acceptable use over the net [8]. The SOAP approach needs writing or employing a provided server program (to serve data) and a shopper program (to request data).REST is commonly utilized in mobile applications, social networking websites, mash up tools, and automatic business processes. Flexibility is provided by distribution resources their own distinctive Universal Resource Identifiers (URIs). As a result of every verb encompasses a specific that means (GET, POST, place and DELETE), REST avoids ambiguity. As in Table 2, this experiment done with two method that is HTTPrequest with cloud data.sparkfun.com and Fusion Tables with google cloud storage.
Table 2: Sending data to cloud method.

| Method                                      | Explaination                                                                 |
|---------------------------------------------|-----------------------------------------------------------------------------|
| Sending data to cloud method 1              | HTTP request method that need a public key and private key for channeling the path of network communication and fill it in one url to be accessed by wifi module ESP8266. http://data.sparkfun.com/input/[publicKey]?private_key=[privateKey]&barometric=[value]&fuel=[value]&map=[value]&rpm=[value]&temperature=[value] |
| Sending data to cloud method 2              | Fusion table in google cloud system api: apiKey=vtyff7566ffvge insertRow=? insertColumn=? |

2.4. Hardware setup.
After platform system experiment is done. The final hardware setup as in Figure 5 is made. The ECU will transfer data in CAN-bus communication to the Arduino UNO microcontroller via MCP2515 CAN-bus converter. MCP2515 CAN-bus converter will connect to the Arduino through Serial Peripheral Interface (SPI) and CAN Hi/Low to the ECU Haltech Sprint 500. Data from ECU will be in form data packet that consist CAN-id and another data in Hex value. The Arduino will be need to separate that data packet to obtain the engine raw data. Data that obtain from ECU will be sending to cloud by using ESP8266 wifi module in REST architecture method. ESP8266 is communicating with Arduino by software serial that can make algorithm sending the data by HTTP request script that will fill data.sparkfun.com cloud.

![Figure 5: Hardware connection that consist arduino and can converter.](image)

3. Result
There are five type of data that are considered in this experiment that is Revolution per Minute (RPM), Manifold Air Pressure (MAP), Barometric pressure, Load-fuel and engine temperature. The result come with two different method of CAN-bus data collection, CAN-bus data conversion and sending raw data
to cloud. From this experiment also there are two user interface (UI) for end user uses that is UI that give by the cloud storage provider and the web interface that consist graph and meter created with using google open chart in the H.T.M.L script. For CAN-bus data collection result is obtain from offline data acquisition when the vehicle is in static condition After the experiment it should get an output like in Figure 6. In the figure it show result of data collected from ECU via CAN-bus.

![Data collection from Arduino and CAN converter board.](image)

**Figure 6:** Data collection from Arduino and CAN converter board.

For CAN-bus data conversion is obtain from offline data acquisition with combining data collection and data conversion algorithm. The experiment got two type of output from different method. In Table 3 the system is tested with method 1 and the data seems unstable due to conversion algorithm is wrong. For Table 4 the data is stable and has same value when compared with the software that provide the ECU manufacturer.

**Table 3:** Data conversion method 1 result.

| Time             | RPM | Manifold Air Pressure | Load-Fuel | Barometric Pressure | Temperature |
|------------------|-----|-----------------------|-----------|---------------------|-------------|
| 24/01/2017 15:53:10 | 0   | 6912                  | 65280     | 6912                | 7424        |
| 24/01/2017 15:53:12 | 6912| 0                      | 0         | 0                   | 0           |
| 24/01/2017 15:53:14 | 6912| 0                      | 0         | 0                   | 0           |
| 24/01/2017 15:53:19 | 0   | 0                      | 0         | 0                   | 0           |
| 24/01/2017 15:53:24 | 0   | 0                      | 0         | 0                   | 0           |
| 24/01/2017 15:53:29 | 0   | 0                      | 0         | 0                   | 0           |
| 24/01/2017 15:53:35 | 0   | 0                      | 0         | 0                   | 0           |
| 24/01/2017 15:53:41 | 0   | 0                      | 0         | 0                   | 0           |
| 24/01/2017 15:53:43 | 0   | 818176                 | 784128    | 818176              | 0           |
| 24/01/2017 15:53:48 | 818176| 0                      | 0         | 0                   | 0           |
| 24/01/2017 15:53:53 | 0   | 0                      | 257536    | 0                   | 0           |
| 24/01/2017 15:53:58 | 0   | 0                      | 0         | 0                   | 0           |
Table 4: Data conversion method 2 result.

| Date/Time         | RPM  | Manifold Air Pressure | Load-Fuel | Barometric Pressure | Temperature |
|-------------------|------|-----------------------|-----------|---------------------|-------------|
| 25/01/2017 12:42:57 | 4096 | -50.32                | 25        | 13                  | 65          |
| 25/01/2017 12:42:57 | 4128 | -54.33                | 29        | 13                  | 65          |
| 25/01/2017 12:42:58 | 4223 | -56.67                | 34        | 14                  | 65          |
| 25/01/2017 12:42:58 | 4096 | -50.32                | 25        | 13                  | 65          |
| 25/01/2017 12:42:58 | 4096 | -50.32                | 25        | 13                  | 65          |
| 25/01/2017 12:42:59 | 4096 | -50.32                | 25        | 13                  | 65          |

After this platform experiment is done there are too many error in method 1 data converting due to detection of SOF as 0 value at the beginning of data packet and this experiment run smoothly with method 2. In sending raw data to cloud that obtain from quarter mile test the result is analyze to show data transmitting timing difference between two method that tested. Figure 7 and 8 show the transmit data timing diagram for two different method. In figure 7 it use method 1 to transmit the data and the data collecting from ECU is every 2ms then it’s transmit data to the cloud in every 12ms. Figure 8 for the method 2 shows that the system capable to collect data form ECU every 2ms and transmit it to the cloud for every 20ms.

![Data transmitting timing diagram method 1](image-url)
When the experiment is done it show sending data to cloud method 1 is better in case of size of data packet, data packet transferring time and the setup was easier than Fusion Tables in method 2. Faster data packet transferring time will increase processing decision based on condition was set in algorithm [9]. Figure 9 and 10 is a user interface for the cloud and website.

**Figure 8:** Data transmitting timing diagram method 2

| barometric | fuel | map | rpm | temperature | timestamp       |
|------------|------|-----|-----|-------------|----------------|
| 4          | 49   | 476 | 482 | 88          | 2017-09-17T00:16:482Z |
| 13         | 13   | 295 | 3116| 0           | 2017-09-17T07:20:439Z |
| 4          | 49   | 476 | 482 | 88          | 2017-09-17T07:14:561Z |
| 13         | 13   | 295 | 3116| 0           | 2017-09-17T07:10:419Z |
| 13         | 13   | 295 | 3116| 0           | 2017-09-17T07:13:595Z |
| 13         | 13   | 295 | 3116| 0           | 2017-09-17T07:06:413Z |
| 4          | 49   | 476 | 482 | 88          | 2017-09-17T07:08:044Z |
| 4          | 49   | 476 | 482 | 88          | 2017-09-17T07:07:452Z |
| 13         | 13   | 295 | 3116| 0           | 2017-09-17T07:04:713Z |
| 4          | 49   | 476 | 482 | 88          | 2017-09-17T07:02:390Z |
| 13         | 13   | 295 | 3116| 0           | 2017-09-17T07:01:019Z |
| 4          | 49   | 476 | 482 | 88          | 2017-09-17T06:50:578Z |

**Figure 9:** UI from data.sparkfun.com.
4. Conclusion

From the experiment result it can be conclude that the best way to collecting data from ECU is by using the MCP2515 TJA1050 CAN-bus converter board due to data clocking for every frame of data packet. Then the wireless layer network is CPS by using REST architecture with HTTP request method and for the improvement the study need to be include the Big Data cloud storage study for better data management and volume of data. Furthermore, security system for CPS will be useful for protection of network layer in crowd sensing.

Acknowledgments

Thanks to UniMAP Automotive Racing Team (UniART) for trusting, help and continue give a funding for develop this system.

References

[1] Van Geem, C. Bellen, M. Bogaerts, B. Beusen, B. Berlmont, B. Denys, T. Hellinckx,. “Sensors on Vehicles (SENSOVO) - Proof-of-concept for Road Surface Distress Detection with Wheel Accelerations and ToF Camera Data Collected by a Fleet of Ordinary Vehicles”. Transportation Research Procedia, 14, 2966–2975, 2016.
[2] F.J Wu, Y.F. Kao, Y.C. Tseng. “From wireless sensor networks towards cyber physical systems”. Pervasive and Mobile Computing, 7(4), 397–413, 2011.
[3] O. Briante, C. Campolo, A. Iera, A. Molinaro, S.Y. Paratore, & G. Ruggeri. “Supporting augmented floating car data through smartphone-based crowd-sensing”. Vehicular Communications, 1(4), 181–196, 2014.
[4] K. Ismail, A. Muhamad, M. Pratama. “Design of CAN bus for research applications purpose hybrid electric vehicle using ARM microcontroller”. Energy Procedia, 68, 288–296, 2015.
[5] J. Xia, C. Zhang, R. Bai, L. Xue. “Real-time and reliability analysis of time-triggered CAN-bus”. Chinese Journal of Aeronautics, 26(1), 171–178, 2013.
[6] X. Hong, W. Jianhua. “An extendable data engine based on OPC specification”. Computer Standards and Interfaces, 26(6), 515–525, 2004.
[7] G. Sánchez-Martínez, E. Koutsopoulos, N.H.M. Wilson. “Optimal allocation of vehicles to bus routes using automatically collected data and simulation modelling”. Research in Transportation Economics, 59, 1–9, 2016.

[8] A. Navarro, A. Da Silva. “A metamodel-based definition of a conversion mechanism between SOAP and RESTful web services”. Computer Standards and Interfaces, 48, 49–70, 2016.

[9] H.L. Yuan, J. Wang, J. Liu, S.L. Li. “Research of Zigbee and big data analysis based pulse monitoring system for efficient physical training”. Procedia Computer Science, 80, 2357–2361, 2016.