OBD-II-Based Vehicle Management Over GPRS Wireless Network for Fleet Monitoring and Fleet Maintenance Management

Hartono Pranjoto, Lanny Agustine, Mimi Mereditha
Electrical Engineering Department, Widya Mandala Catholic University, Indonesia.
pranjoto@ukwms.ac.id

***Abstract*—The method of extracting parameters of the engine, electronics, body and power of commercially manufactured vehicles using On-Board Diagnostic version 2 (OBD-II) has been widely used. Important parameters from the vehicles are extracted using standard protocols. General Packet Radio Service (GPRS) is an 'always on' data service, and it is a part of Global System for Mobile Communications (GSM) cellular phone and always on when GMS service is on. With GPRS, a computer connects to the Internet via the wireless network as long as there is GSM network. A computer with a Bluetooth® connects to Bluetooth OBD-II onboard a vehicle. The data extraction; P for powertrain (engine and transmission), B for body, C for chassis, and U for the network; uses AT command in the RS-232 connection. A computer processes the data and then transmits the data to a data center via GPRS wireless network using SIM808. This system uses GPRS because the data is small and the 'always on' feature. Data received by the data center is processed and stored in MYSQL format for further use and display with a web browser. An operator with a web browser can monitor the vehicle parameters and later remedial actions (maintenance or management) if necessary.**

***Index Terms*—Fleet Management; GPRS; Internet of Things; OBD-II Monitoring; Vehicle Monitoring.**

I. INTRODUCTION

Late model light vehicles build after 1996 usually have self-diagnosing/self-reporting capability via digital communication called OBD-II (referred to as On-Board Diagnostics version 2). OBD systems provide means to access the status of various data from the vehicle subsystems using standardized digital communication to supply real-time data for immediate or later diagnostic purposes, which is called DTC (Diagnostic Trouble Codes). DTC provides the user (either the owner or technician) information to identify and remedy the trouble/ malfunction of the vehicle. [1]

The connector is under the steering wheel either on the left-hand or right-hand side of the steering wheel. Figure 1 shows the location of a car sold in Indonesia which in this case is right under the steering wheel as shown with the arrow.

OBD-II connectors used to obtain digital signal from the vehicle is a 16-in connector as shown in Figure 2(a), 2(b), and 2(c). Figure 2(a) is the photograph of the connector used in a 12-Volt vehicle that is also called OBD-II Female type A connector, in which this connector will have a 12-Volt supply from the vehicle. OBD-II Male type A as its counterpart is shown in Figure 2(b). OBD-II Male type B is for vehicles with 24-Volt supply as shown in Figure 2(c). These two photographs show the difference of the connector, in which the 24-Volt connector has a slot so that the female connector can only fit with the male type B and not vice versa [2].

Figure 1: Illustration of the location of OBD-II connector indicated by arrow

(a) (b) (c)

Figure 2: (a) OBD-II Female type A, (b) OBD-II Male type A and (c) OBD-II Male type B.

OBD-II specifies the standard of the data communication between the vehicle and the reading unit as well as the electrical specification of the signal and in addition is the physical standard of the connector and the location of the signals. The connector is a 16-pin connector located inside the car and located within 2 feet from of the steering wheel and access to the connector must be direct (no tools required).

The OBD-II connector physical specification, its mechanical characteristics and its pins are standardized and described in the SAEJ1962 Standard [2]. The pin connection and their signal names are also indicated in the specified standard and for convenience rewritten again in Table 1.

The OBD-II signals and the standard for the data returned are described in the standard [1-3]. There are ten different modes of operation of the normal OBD-II based on the SAE-J1962[3-4] that are used. The data engine of the OBD-II are shown in Table 2.

Table 1
OBD-II pin connection

Pin	Signal
1	Manufacturer discretion: GM:J2411 GMLAN/SWC/Single-Wire CAN VW/Audi/BMW: Switched +12V to tell a scan tool whether the ignition is on. Ford, FIAT: Infotainment CAN High
2	Bus Positive Line of SAE J1850 PWM and VPW
3	Manufacturer discretion: Ford: DCL(+) Argentina, Brazil (pre OBD-II) 1997-2000, USA, Europe, etc. Ford: Medium Speed CAN-High Chrysler: CCD Bus(+)
4	Chassis ground
5	Signal ground
6	CAN-High [3, 4]
7	K-Line [5, 6]
8	Manufacturer discretion: BMW: Second K-Line for non-OBD-II (Body/Chassis/Infotainment) systems. FIAT: Infotainment CAN-Low.
9	Manufacturer discretion: BMW: TD (Tachometer Display) signal aka engine RPM signal. GM: 8192 bit/s ALDL where fitted.
10	Ford: Infotainment CAN-Low Bus Negative Line of SAE J1850 PWM only (not SAE J1850 VPW)
11	Manufacturer Discretion: Ford: DCL(-) Argentina, Brazil (pre OBD-II) 1997-2000, USA, Europe. Ford: Medium Speed CAN-Low Chrysler: CCD Bus(-)
12	Manufacturer discretion: GM: Diagnostic codes to DIC (1994-2004 Corvette)
13	Manufacturer discretion: Ford: FEPS – Programming PCM voltage
14	CAN-Low [3, 4]
15	L-Line [5, 6]
16	Battery voltage: Type "A" 12V/4A refer to Figure 2b Type "B" 24V/2A refer to Figure 2c

Table 2
Data description of OBD-II

Mode	Description
1	Current data
2	Frozen data
3	Diagnostic Trouble Codes data
4	Clear Diagnostic Trouble Codes and stored values
5	Test results, oxygen sensor monitoring (non-CAN vehicles only)
6	Test results, other component/system monitoring (Test results, oxygen sensor monitoring CAN vehicles only)
7	Diagnostic Trouble Codes (detected during current or last driving cycle)
8	Control operation of on-board component/system
9	Request vehicle information
0A	(Cleared DTCs)

On the current data (mode 01 hex0 there are many parameters that can be shown for different types of vehicles and also the type of output of the sensors. For the data extracted for this management is shown in Table 3. [3]

II. INTRODUCTION TO SIMCOM

SIMCOM808 is a module that comprises a GPS, a GPRS, and a Bluetooth module integrated into one larger module. This module also has another module important for the work that is the power management module – a module to manage internal rechargeable battery in case the external power is unplugged–embedded. The function block of this module is illustrated in Figure 3 [2]. As shown here, the module has complete GPS receiver, complete GPRS packet data connection along with GSM telephone and power management unit to control an internal rechargeable battery and the use of a Bluetooth wireless connection.

All of the function of SIM808 (GPS, GPRS and Bluetooth) are connected via one serial connection of USART, therefore, a microcontroller with one serial connection can handle all three components. This is contrary to the previous version of this device (SIM908) that requires two USART connection to connect to the GPS and the GPRS units. In order to connect to the different unit, the USART command must make the connection first using the simple AT command for each different unit in the system such as the GPS, the GPRS or the Bluetooth modules. [7]

(a) (b)

Figure 3: (a) Bottom side of SIMCOM808 with GSM slot, (b) Top side of SIM808 module.

Connection to the SIM card to connect to the GSM provider is external and connected to the external connector. All the GPS, GSM/ GPRS and Bluetooth sub-system are controlled via one UART port that can multiplex among, GPS, GPRS and Bluetooth connection. All controls of the system use the AT command which is already described in detail in [8-12] for the GPRS sub-system, GPS sub-system and the Bluetooth sub-system. In general, there are six commands for the GPS and may set of commands for the Bluetooth and the GPRS sub-systems. [13,14]

The GSM module is Quad-band with the power to the SIM card (1.8–3.0 V) compatible with all providers in Indonesia. The frequency of the GSM sub-system uses the standard frequency of 850, 900, 1800 or 1900 MHz with GPRS multi-slot calls 12/10 for better data connection stability.

The serial port connection supports data rate from 4800 – 115200 bit/ second (bps). Figure 3a shows the GSM SIM card side of the module, which also the main connection for all three antennas (GPS, GPRS, and BT antennas) in I-Pex connector labeled respectively. Figure 3b is the photograph of the top side of the module [7, 14].

Table 3
OBD Data used for the vehicle management

PID	Data bytes	Parameter	Min	Max	Unit
0C	2	Engine RPM	0	16,383.75	rpm
0D	1	Vehicle speed	0	255	km/h
0F	1	Intake air temperature	-40	215	°C
10	2	MAF air flow rate	0	655.35	grams/sec
11	1	Throttle position	0	100	%
45	1	Relative throttle position	0	100	%
46	1	Ambient air temperature	-40	215	°C

III. CONNECTION TO OBD-II VIA BLUETOOTH

OBD-II dongle with Bluetooth module is available very widely in the market. The system described here uses a simple OBD-II dongle with the ELM327 interface. Figure 4 shows the OBD-II dongle with the ELM327 integrated circuit and Bluetooth connection. This interface will allow the OBD-II protocol – which is CANBUS protocol – communicate with other peripherals using serial interconnection with simple 'AT' command. With the simple 'AT' command, the parameters can be extracted from the vehicle.

Figure 4: OBD-II dongle with the ELM327 integrated circuit and Bluetooth connection

Before any extraction, the pairing between two Bluetooth devices must be established by normal Bluetooth pairing and code exchange to ensure security. After the pairing, the serial connection will be set at 38.4 kbits per second (kbps) with 8 data bit and 1 stop bit. After the setting, the connection can be set and run at one-second interval to obtain the data.

Figure 5 shows that after the connection, the host computer will send command 'AT SP 0' to instruct the OBD-II run in the auto mode and the device will respond with 'AUTO' prompt. At the bottom of the terminal screen is the result of sending data '010C', '010D', '010D', '010F', and '0104', for engine RPM, vehicle speed. The data connection is obtained every second with simple delay interval in the program.

IV. CONNECTION TO THE INTERNET

Internet connectivity involves the initialization of the GPRS to attach to the server. Depending on the provider, there are different APN for different providers. There two other parameters that are provider dependent, which is the username and password. This connection is the code snippet of the program to connect to the Internet as shown in Figure 6. This figure shows the connection using Telkomsel as the provider using generic username and password given by the provider.

After the connection to the Internet, this system has to connect to the proper website to display the location of the vehicle using Google Map. The website is coded into the system, which is 'www.percobaan3.tk.' The protocol to connect to the web server is the HTTP protocol via port 80. The connection request will use the "GET /HTTP/1.1". If there is the HTTP Get request, the server will respond with '200 OK'. Most of the connection via the HTTP channel will use the HTTP Get request instead of the Post submission. The methods GET, HEAD, PUT and DELETE share similar property; also, the methods OPTIONS and TRACE will be similar. A detailed description of this connection has been discussed in [15, 16].

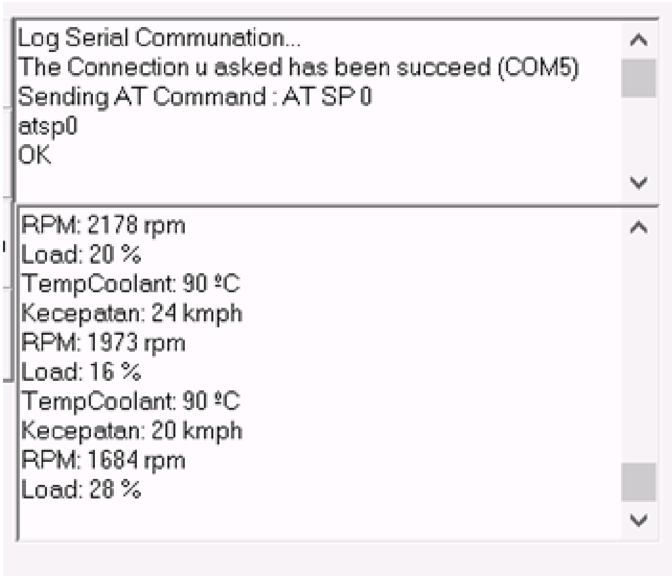

Figure 5: Prompt to send the auto command 'AT SP 0' and then respond with 'AUTO', then OBD-II engine command is send

V. IMPLEMENTATION OF THE SYSTEM

OBD-II-based vehicle management over GPRS (General Packet Radio Service) wireless network for fleet monitoring and fleet maintenance management has been designed and constructed using all the subsystem mentioned above. The system uses a Raspberry PI module to control and extract the parameters of the vehicles under the management using the Bluetooth module of the Raspberry Pi. Data extraction of the vehicle is conducted using the AT command available from the ELM327 which is onboard the OBD_2 Bluetooth dongle. Data extraction uses the command as described in Table 3. Extraction of data uses the prescribed formula as given by the OBD-II standard.

In Table 3, data for the engine rotation (RPM) is represented using two-byte data ranging from 0 to 65535 (which is 2^{16}-1). The actual engine rotation is calculated by dividing the two-byte value by four. Therefore the maximum value for the engine rotation is 16,353.75 RPM. The vehicle speed is the actual one-byte data, valued from 0 to 255 as the actual velocity in kph. Temperature values of air intake temperature and the ambient temperature is the one-byte value of the data offset by 40. Therefore a 0 value from the data represent the temperature of –40°C. and value of 255 represents the temperature of 215°C The throttle position, either the relative or actual position is represented by the bye value divided by 2.55. Therefore the lowest number is 0 and the highest is 100%. There are many more engine parameters for OBD-II, but the data is not extracted in this project and can be found in [17]. The formula and parameter extractions are also explained in detail in [17].

The illustration in Figure 5 is actual data gathered using a Toyota Agya 2016. The data showed there is already manipulated using the formula explained previously. The time interval for the data extraction is one second for this experiment, but for the actual data extraction and processing in the web will be conducted every five seconds.

The vehicle database involves the condition of the current vehicle based on the condition of the vehicle tracking system.

```
void initGPRS()
{        delay_ms(1000);
        printf("AT+CGDCONT=1,");
        putchar('"');
        printf("IP");
        putchar('"');
        putchar(',');
        putchar('"');
        printf("internet");
                    // APN
        putchar('"');
        putchar(13);
        wait_string("OK");
        delay_ms(5000);
        printf("AT+CSTT=");
        putchar('"');
        printf("internet");
//APN
        putchar('"');
        putchar(',');
        putchar('"');
        printf("");
                    // username
        putchar('"');
        putchar(',');
        putchar('"');
        printf("");
// password
        putchar('"');
        putchar(13);
        while(getchar()!='O') ;
}
```

Figure 6: Code snippet of the GPRS modem to the GPRS network.

VI. CONCLUSION

This work has shown the work of monitoring a vehicle performance via OBD-II using simple computer or microcontroller to obtain vehicle condition such as speed, engine RPM, coolant temperature and any other generic OBD-II parameters. The data is then sent via the GPRS to the data center for further analysis and storage.

This work has shown a great detail on how to connect the OBD-II dongle using the Bluetooth connection with computer and also a detailed description on how to obtain the data and then present the data – using some arithmetic conversion of the data – into a manageable description of the data.

ACKNOWLEDGEMENTS

This work is supported by: the Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jenderal Penguatan Riset Dan Pengembangan,Kementerian Riset, Teknologi dan Pendidikan Tinggi, sesuai dengan Surat Perjanjian Penugasan Pelaksanaan Program Penelitian Nomor: 003/SP2H/P/K7/ KM/2016 tanggal 25 April 2016

REFERENCE

[1] J. III Keenan, *Creating A Wireless OBDII Scanner*, Worcester Polytechnic Institute, 2009.

[2] SAE International Surface Vehicle Recommended Practice, “Diagnostic Connector Equivalent to ISO/DIS 15031, *SAE Standard J1962*, Rev Dec 2001

[3] International Organization for Standardization, “Road vehicles — Diagnostics on Controller Area Networks Part 4”, *Requirements for Emissions-related Systems ISO 15765*, 2005

[4] SAE International Surface Vehicle Recommended Practice, “High Speed CAN (HSC) for Vehicle Applications at 500 KBPS”, *SAE Standard J2284-3*, Rev. 2002.

[5] International Organization for Standardization, “Road Vehicles – Diagnostics Systems – CARB Requirements for Interchange of Digital Information”, *ISO 9141*, 1994

[6] International Organization for Standardization, “Road Vehicles - Diagnostic Systems - Keyword Protocol 2000 - Physical Layer”, *ISO 14230-1*, 1994.

[7] Sim Tech, *SIM 808 Hardware Design V. 2.00*, SIMCom Wireless Solutions, 2012.

[8] T. Halonen, J. Romero, and J. Melero (eds), *GSM, GPRS, and Edge Performance: Evolution Towards 3G/UMTS 2nd Ed.*, New York: Wiley, 2003.

[9] G. Maral and M. Bousquet, *Satellite Communications Systems, Systems, Techniques and Technology 5th Ed.*, West Sussex: Wiley, 2009.

[10] E. Seurre, P. Savelli, and J.P. Pietri, *GPRS for Mobile Internet*, Norwood, Massachussettes: Artech House, 2003.

[11] Eberspächer , J., Vögel, HJ., Bettstetter, C., Hartmann, C., GSM – Architecture, Protocols and Services 3rd Ed., Stuttgart: Wiley, 2009.

[12] European Telecomunications Standard Institute, *Technical Specification AT command set for User Equipment 3GPP TS27.007 version 11.5.0 Release 11 (2013-01)*, January 2013

[13] SIM Tech, *SIM800 GPS Application Note V1.00*, SIMCom Wireless Solutions LTD. 2015.

[14] SIM Tech, *SIM800 Series AT Command Manual V1.09*, SIMCom Wireless Solutions LTD. 2015.

[15] H. Pranjoto, L. Agustine, Y.S. Susilo, and R. Tehuayo, “GPS Based Vehicle Tracking Over GPRS for Fleet Management and Passenger/Payload/Vehicle Security”, *ARPN Journal of Engineering and Applied Sciences*, 9, 11, 2014.

[16] L. Agustine, E. Pangaliela, and H. Pranjoto, “Vehicle Security and Management System on GPS Assisted Vehicle Using Geofence and Google Map”, *Lecture Notes on Electrical Engineering* (ISBN 978-981-287-986-8, DOI 10.1007/978-981-287-988-2, 65), 2015, 215

[17] Wikipedia The Free Encyclopedia, s.v., *OBD-II PIDs* (accessed June 12, 2017, https://en.wikipedia.org/wiki/ OBD-II_PIDs