ASM Sc. J., 11, Special Issue 3, 2018 for SANREM, 59-65

Development of Instantaneous Fuel Consumption Measuring System using Arduino Uno

Muhammad Iftishah Ramdan[1], Teoh Cheng Lay [1], Lim Zhi Wey [1], Mohd Azmi Ismail [1] and Mohamad Yusof Idraos[1]

[1]*School of Mechanical Engineering, Universiti Sains Malaysia, Malaysia*

Instantaneous Fuel Consumption (IFC) measurement systems play a key role in keeping our environment clean, by allowing drivers to constantly monitor and optimize their driving behavior, in order to minimize fuel consumption and emission. This paper introduces an affordable method of measuring vehicle IFC by using an Arduino microcontroller to conduct pulse wave measurement of the injector signal. Multiple tests indicate that the developed measuring system is accurate and reliable, with the percentage difference between the values obtained using the Arduino measuring system's measuring value and the burette measurements as the reference being less than 8.2%.

Keywords: fuel consumption, affordable green technology, Arduino, vehicle efficiency.

I. INTRODUCTION

Currently, efforts are being made to produce highly efficient vehicles, driven by fluctuations in crude oil prices and increased global energy consumption. These efforts, however, are expensive, and take a long time to develop. The world is in need of a more short-term method of minimizing the use of fuel by vehicles. Adjusting the driving behavior is one of the tools to save fuel.

A study by Greene *et al.* shows that fuel prices are often not the main concern among consumers, when choosing between personal vehicles with different fuel economy ratings (Greene *et al.,* 2005). Until now, many believed that choosing a vehicle with better fuel economy reduces fuel expenditures in the long run. Alcott *et al.* have challenged this belief in a study, which introduces a phenomenon known as the "Energy Paradox" (Allcott & Wozny, 2014). The paradox states that by not considering the rise in gasoline prices in the future, when purchasing vehicles, consumers might spend even more on fuel (Allcott & Wozny, 2014). This shows that a more fuel-efficient car may not be the absolute answer to the rise in fuel consumption.

Driving patterns have always been related to vehicle fuel consumption. For instance, extreme acceleration and late gearshifts have substantial effects on fuel consumption (Ericsson, 2001). In addition, according to Vlieger *et al.* (2000), the fuel consumption of a spark-ignition vehicle is more sensitive to driving behaviors than that of a diesel-fueled vehicle.

Some studies have been done to categorize real-time driving behaviors. For instance, by

---

*Corresponding author's (Muhammad Iftishah Ramdan) Email: shahramdan@usm.my using a vehicle's accelerometer and Global Positioning System (GPS) signal, driving behaviors can be categorized as normal, moderate and aggressive (Castignani *et al.,* 2015; Meseguer *et al.,* 2013; Khanapuri *et al.,* 2015). The results of these studies have been used in research on obtaining real-time fuel consumption data for drivers to improve the vehicle fuel economy. The "Instantaneous Fuel Consumption" (IFC) is the crucial data for the studies, as this shows the volume of fuel consumed at any particular moment. Knowing the vehicle's IFC, a driver is able to minimize it by continuously optimizing his/her driving patterns, which will results in higher fuel economy in the long run. However, built-in IFC systems are currently only available for newly-manufacture model vehicles, and not widely available to drivers having older-model vehicles.

Generally, there are two methods to obtain an engine's IFC, namely, the direct method and the indirect method. With the direct method, the IFC data are obtained from the vehicle's engine control unit (ECU) via an On-board Diagnostics-II (OBD-II) port (Aessandrini *et al.,* 2012). Although most vehicles are equipped with OBD-II ports, not many have the sufficient sensor data from the engine than can actually be used to obtain the IFC. As a result, many other studies were ultimately designed to obtain the IFC of the engine indirectly. The indirect method of the IFC estimation uses data from sensors or actuators, without going through the vehicle's ECU. One example of obtaining the IFC indirectly is by estimating it using the vehicle speed and acceleration data that are obtained from the GPS (Ahn *et al.*, 2002; Skog & Händel, 2014). This method is based on a physical model describing the relationship between the car dynamics, the engine speed, and system's energy consumption (Ahn *et al.*, 2002; Skog & Händel, 2014). This paper presents the development of an indirect method of obtaining the IFC data of a light-duty passenger vehicle, Perodua Myvi, by using the signal from its fuel injector.

II. MATERIALS AND METHOD

The method used to measure a vehicle's fuel consumption involves using an "Arduino UNO" microcontroller's "Timer" and "Interrupt" functions to measure the injector signal's pulse width. Once the injector pulse width is obtained, the instantaneous fuel consumption can be obtained by manipulating the data to a set of mathematical formulas.

"Timer" is an Arduino function, responsible for precise time counting (in microseconds), independently of the program currently being executed. The "timer" uses a counter, calibrated from 0 to a specified value, and resetting once the maximum value is reached. This is important, as an "interrupt" function can be initiated when the counter resets. As this measuring system requires very precise time counting, the "interrupt" function can ensure that the code is executed exactly at the moment when it is needed.

"Interrupt" is an Arduino function that is able to stop the program execution, and force the processor to execute the code from the "interrupt" handler, when a specified event occurs. This specific event is based on the state of the "interrupt" pin. In this measuring system, the "interrupt" is programmed to occur whenever there is any change in the state of the "interrupt" pin, namely, a transition from "high" to "low" and vice versa. In this system, pin 2 of the Arduino is used as the "interrupt" pin.

The injector signal is needed to measure a vehicle's fuel consumption. The injector signal comes from the Engine Control Unit (ECU), to manage the work of the fuel injectors. This signal experiences "HIGH" and "LOW" state periods. The injector is either open or closed, with no intermediate states. The injector closes at the "HIGH" state, when the voltage is at 12V, and opens at the "LOW" state, at 0V. Figure 1 shows a general signal flowing into the engine's fuel injector.

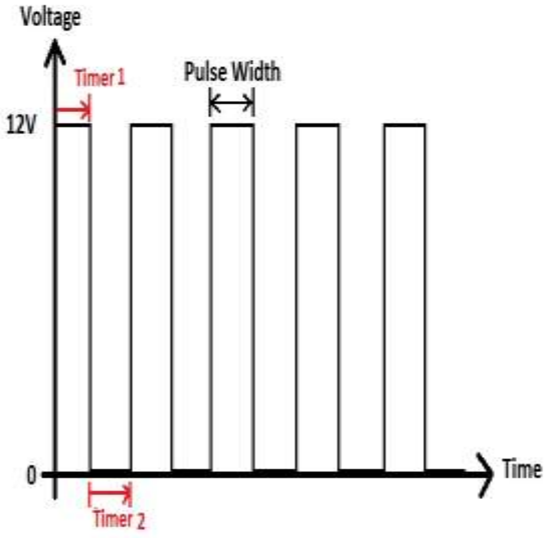

Figure 1: Injector signal of the fuel injector

Pulse width measures the duration of the injector opening for a particular cycle, and is generally controlled by the ECU. Pulse width may vary between cycles. For example, a longer pulse width is sustained if more fuel is needed to produce a higher engine torque. Therefore, measuring the pulse width can identify the fuel consumption.

The "timer" and the "interrupt" functions, explained in the previous section, are used to measure the pulse width. As stated, an "interrupt" function will be initiated if there is a change in the "interrupt" pin's state. For instance, if the pin state is at "HIGH" initially, the timer will start, under the name "Timer 1". Once the pin state changes to "LOW", an "interrupt" function will stop "Timer 1" and start another timer, called "Timer 2". "Timer 2" will continue to run until the pin state changes to "HIGH" state again, activating another "interrupt" function, which runs a program to subtract "Timer 1" from "Timer 2", to obtain the pulse width for one cycle. This process repeats itself for every signal cycle.

The IFC can be obtained by determining the total pulse width (in microseconds) produced in one second when the timer is reset to zero. The total pulse width is then multiplied by the injector rate, which is the volume of fuel that the injector is able to inject into the combustion chamber over time. The units for the injector rate are normally cc/min or liter/s, which can be calculated using the developed measuring system. Once the IFC is obtained, the total fuel consumption per trip can also be calculated, by summing up the IFC at every one-second interval. The complete flowchart for the program code written for the measuring system can be seen in figure 2 and figure 3.

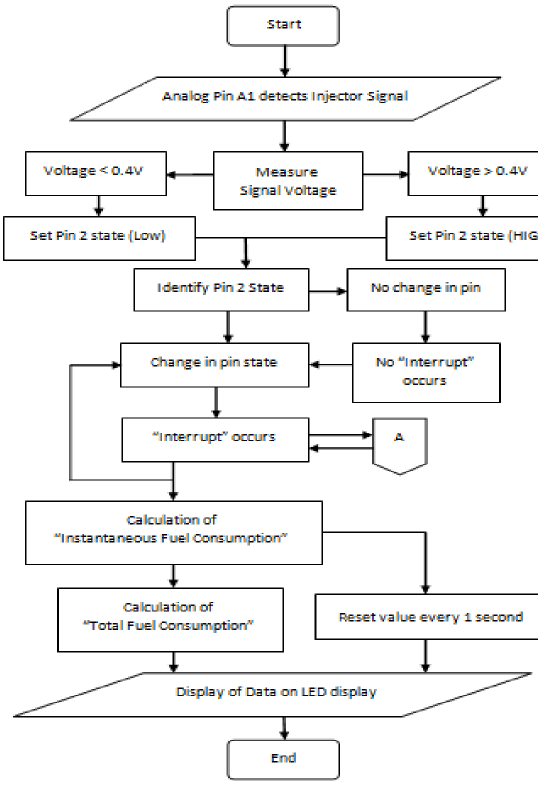

Figure 2: Flow chart of fuel consumption measuring system

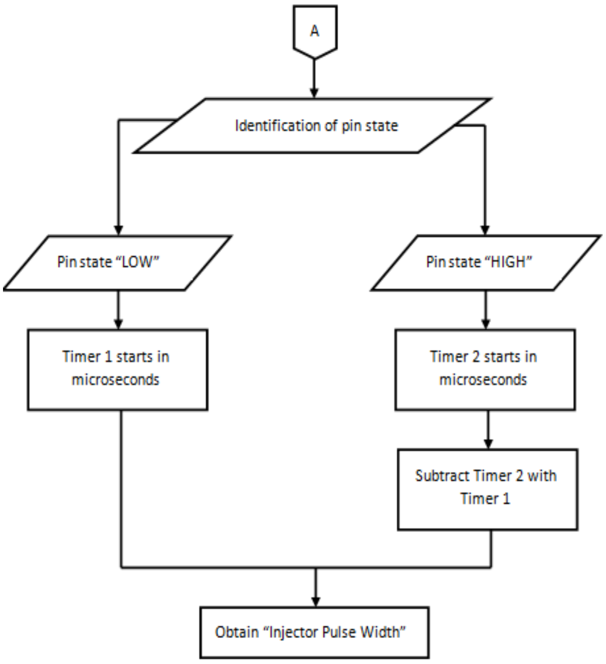

Figure 3: Flow Chart of "Interrupt" Function subsystem

Circuit Connection and Wiring

As an Arduino microcontroller can only support up to a maximum of 5v of voltage, the 12v injector signal must be directed into a voltage regulator before directing the signal into the microcontroller. Therefore, a 5v voltage regulator 7805 IC is used. Also, a 16 x 2 LED display is used to display the data obtained by the system. Figure 4 shows the complete wiring of the measuring system.

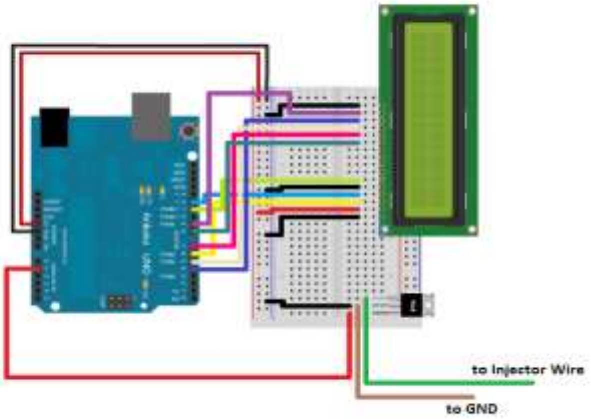

Figure 4: Schematic diagram of system's components

Obtaining Injector Flow Rate

The Injector Flow Rate (IFR) is a crucial value that needs to be obtained experimentally, to ensure accurate fuel consumption reading. Even though the IFR can be found from the fuel injector specifications, the measured IFR would result in a more accurate fuel consumption reading. The measured IFR can be obtained through a simple experiment, in which the total amount fuel sprayed by the injector is collected and measured using a burette, then divided by the total injector open time. To obtain the injector signal, the injector wire is tapped from the ECU (figure 5). Based on the IFR experiment, the injector flow rate for the K3-V3 engine fuel injectors is 3.148 ml/s. Equation 1 shows the formulation used to obtain the IFR.

$$IFR = \frac{Total\ Fuel\ Consumption\ in\ 10\ seconds}{Total\ Injector\ Open\ Time\ in\ 10\ seconds} \quad (1)$$

The fuel consumption measuring system's accuracy and reliability are tested by comparing them with the 3D fuel consumption engine map data that have been acquired experimentally.[11]

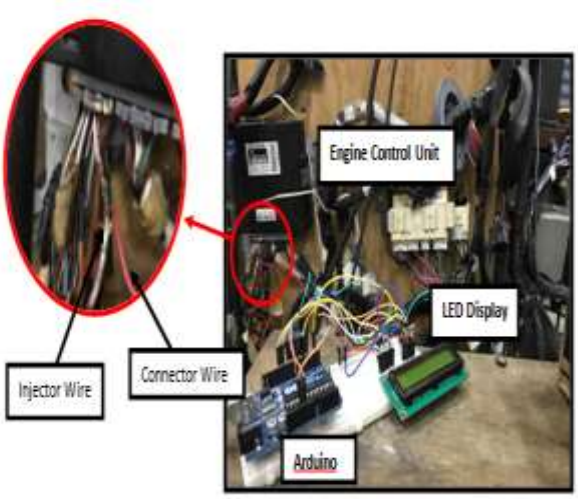

Figure 5: Laboratory set up of fuel consumption measuring system

III. RESULTS AND DISCUSSIONS

The Arduino measuring system's fuel consumption data are compared with the data from the existing fuel consumption 3D engine map (Ramdan, 2017). The measuring system shows great accuracy, as the differences between the two fuel consumptions data are within the 1 – 9 % range. The complete data and the comparison can be seen in Table 1. As the table demonstrates, the percentage difference tends to decrease as the engine torque increases. This is due to the fact that as the engine torque increases at a particular engine speed, the pulse length also increases, while the pulse frequency remains the same. This enables much more precise "interrupts" to occur, resulting in better pulse width measurement by the Arduino.

The opposite trend in the percentage difference is observed as the engine speed increases because increasing the engine speed would increase the pulse frequency detected by the Arduino. Although the "interrupt" function is supposed to work immediately, there is a slight delay in microseconds, due to the codes operation. Thus, a higher-frequency pulse width could lead to more interruptions and may increase the error slightly. However, this error is not sufficient to have a significant effect on the measurement system. Figure 6 shows the overall trend of the percentage difference between values obtained experimentally and by the measuring system with respect to engine torque and engine speed.

Table 1: Fuel consumption data obtained from Arduino measuring system and from fuel consumption engine map (Ramdan, 2017).

Torque (Nm)	Engine speed: 1000 RPM			Engine speed: 2000 RPM			Engine speed: 3000 RPM			Engine speed: 4000 RPM		
	Exp	Arduino	%Different	Exp	Arduino	% Different	Exp	Arduino	%Different	Exp	Arduino	% Different
10	0.333	0.322	**3.320**	0.746	0.723	**3.046**	0.970	1.022	**5.368**	1.309758	1.221	**6.777**
20	0.471	0.452	**3.950**	0.867	0.835	**3.662**	1.139	1.075	**5.642**	1.669449	1.531	**8.293**
30	0.533	0.519	**2.688**	1.075	1.043	**2.975**	1.375	1.448	**5.306**	2.163332	2.293	**5.994**
40	0.667	0.645	**3.250**	1.292	1.253	**3.049**	1.621	1.531	**5.556**	2.256700	2.381	**5.508**
50	0.769	0.759	**1.330**	1.417	1.382	**2.500**	1.933	**1.854**	**4.079**	2.675585	2.562	**4.245**

Table 1 demonstrates that the measuring system works best at lower engine speed and higher engine torque values. The percentage differences obtained at lower engine speeds of 1000 rpm and 2000 rpm range between 1.3% and 4.0 %. These results are considered excellent, as daily vehicle usage is normally within this range of engine speed. Although the percentage differences are greater at higher engine speeds, such as 3000 rpm and 4000 rpm, between 4.0% and 8.2%, they are still within the measuring system's acceptable tolerance limit.

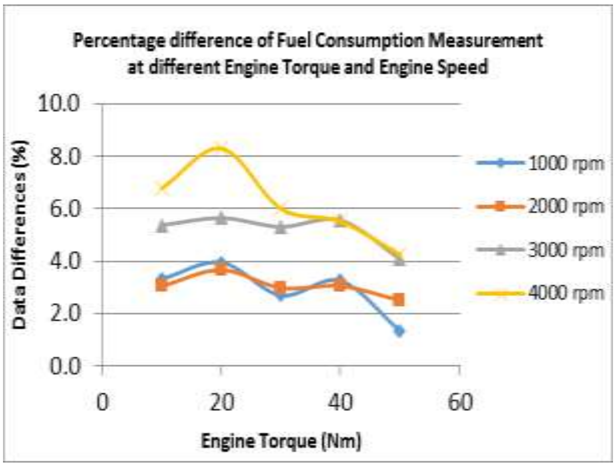

Figure 6: Difference percentage of fuel consumption measurement at different engine torque

IV. CONCLUSION

An IFC measuring system is developed using an Arduino microcontroller and the injector signal from the engine's ECU. By measuring the injector signal's pulse width, the total injector opening time can be determined. The injector flow rate has been determined experimentally to be 188.89cc/min. The fuel consumption can be calculated by multiplying the injector open time by the injector flow rate value. The instantaneous fuel consumption is calculated by determining the fuel consumption at each consecutive second. The Arduino measuring system works well, with the percentage difference between the measuring system and the experimental reference fuel consumption data being less than 8.2%.

V. ACKNOWLEDGMENT

The work described in this paper is funded by Universiti Sains Malaysia short term grant 304/PMEKANIK/60313026, and conducted under the supervision of Dr. Muhammad Iftishah Ramdan.

---

[1] Aessandrini, A, Filippi, F and Ortenzi, F 2012, ‘Consumption calculation of vehicles using OBD data’, in 20th International Emission Inventory Conference, 13-16 August 2012, Tampa, Florida.

[2] Ahn, K, Rakha, H, Trani, A and Aerde, MV 2002, ‘Estimating Vehicle Fuel Consumption and Emissions Based on Instantaneous Speed and Acceleration Levels’, Journal of Transportation Engineering, vol. 128, no. 2.

[3] Alott, H, and Wozny, N 2014, ‘Gasoline Prices, Fuel Economy, and the Energy Paradox’, The Review of Economics and Statistics, vol. 96, no. 10, pp. 779-795.

[4] Castignani, G, Frank, R and Engel, T 2015, ‘ Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring’, IEEE Intelligent Transportation Systems Magazine, vol. 7, no. 1.

[5] Ericsson, E 2001, ‘Independent driving pattern factors and their influence on fuel-use and exhaust emission factors’, Transportation Research Part D: Transport and Environment, vol. 6, no. 5, pp. 325-345.

[6] Greene, DL, Patterson, PD, Singh, M, and Li, J 2005, ‘Corrigendum to ‘Feebates, rebates and gas-guzzler taxes: a study of incentives for increased fuel economy”, Energy Policy, vol. 33, no. 10, pp. 757–775.

[7] Khanapuri, AV, Shastri, A, D’souza, G and D’souza, S 2015, ‘On road: A car assistant application’, in International Conference on Technologies for Sustainable Development (ICTSD), 4-6 February 2015, Mumbai, India.

[8] Manzoni, P 2013, ‘Driving Styles: a smartphone application to assess driver behavior’, in 2013 IEEE Symposium on Computers and Communications (ISCC), 7-10 July 2013, Split, Croatia.

[9] Ramdan MI 2017, ‚Perodua Myvi engine fuel consumption map and fuel economy vehicle simulation on the drive cycles based on Malaysian roads’, MATEC Web Conference, vol. 90, no. 01044.

[10] Skog, I and Händel, P 2014, ‘Indirect Instantaneous Car-Fuel Consumption Measurements’, IEEE Transactions On Instrumentation and Measurement, vol. 63, no. 12.

[11] Vlieger, ID, Keukeleere, DD and Kretzschmar, JG 2000, ‘Environmental effects of driving behaviour and congestion related to passenger cars’, Atmospheric Environment, vol. 24, no. 27, pp. 4649-4655.