

## Adopting Hardware-in-the-Loop for Testing Vehicle Instrument Panel Using Economical Approach

Wan Shahmisufi Wan Jamaludin<sup>1</sup>, Tan Wei Ren<sup>2</sup>, Bakhtiar Affendi Rosdi<sup>3</sup>,  
Dahaman Ishak<sup>4</sup>, Noor Hafizi Hanafi<sup>5</sup>, Muhammad Nasiruddin Mahyuddin<sup>6</sup>

<sup>1,2,3,4,6</sup> School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia,  
14300, Nibong Tebal, Pulau Pinang, Malaysia

<sup>5</sup> Continental Automotive Components Malaysia Sdn. Bhd, MK 1, 2455, Tingkat Perusahaan 2A, Prai Industrial Estate,  
13600 Perai, Pulau Pinang, Malaysia

---

### Article Info

#### Article history:

Received Jan 13, 2018

Revised Mar 15, 2018

Accepted Mar 29, 2018

---

#### Keywords:

Commercial Off-The-Shelf

Device-Under-Test

Hardware-In-The-Loop

Vehicle Instrument Panel

Signal Conditioning

---

### ABSTRACT

An economical approach for testing Vehicle Instrument Panel is proposed in this paper due to high expenditure of purchasing the available Commercial Off-The-Shelf Hardware-In-The-Loop. Vehicle Instrument Panel is designated as the Device-Under-Test in this paper. The Hardware-In-The-Loop, designated as the test equipment, will simulate the assigned input signals controllable via designed Graphical User Interface. The resulting display is shown on the Graphical User Interface and the Device-Under-Test. The speedometer gauge measurement showed the highest disparity of 4 km/h which is within the tolerance of the pre-determined specification of the Device-Under-Test.

Copyright © 2018 Institute of Advanced Engineering and Science.  
All rights reserved.

---

### Corresponding Author:

Muhammad Nasiruddin Mahyuddin,  
School of Electrical and Electronic Engineering,  
Engineering Campus, Universiti Sains Malaysia,  
14300, Nibong Tebal, Pulau Pinang, Malaysia.  
Email: nasiruddin@usm.my

---

## 1. INTRODUCTION

Vehicle Instrument Panel (VIP) for a modern car involves many operation, as it displays information's like driving condition, fault diagnostics, warning signals, navigation, reminder, infotainment etc [1]. It is essential for the instrument panel to test all of its functionality before it is incorporated together during car manufacturing. False information conveyed by the VIP to the driver may harm the vehicle, the driver, and the occupants.

Before the product is handed over to the end customer, test equipment has been developed and established by the test engineers to ensure the VIP is verified to be in good condition. The product which is VIP is assigned as the Device-Under-Test (DUT) and Hardware- In-the-Loop (HIL) device is the test equipment, together with PC and several communication tools between PC and the HIL. Hardware-in-the-Loop Simulation (HiL) is a technique designed to test embedded systems in a comprehensive and reproducible way, thus reducing time and cost in the process of systemic testing [2]. The proposed solution is based on a host PC executing the simulation and connected via USB, to an electrical interface module that provides the interfacing signals to the system-in-test. The application of HIL has been widely practiced by the industries as the means of simulation and testing, either in automotive sector or in other field of interests as found in [4]–[13]. Based on Table 1 and Figure 2, HIL manufactured by National Instruments (Figure 2a), dSPACE (Figure 2b), Opal-RT (Figure 2d) and other HIL platforms are used as the conventional equipment for simulation and testing. Most Commercial-Off-The-Shelf (COTS) HILs obtainable are usually very costly

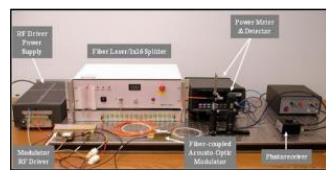
and sophisticated [14]. On the contrary, the economical approach of the proposed system is implemented using a low cost HIL system. The DUT's pre-requisite is this HIL system's foundation which eliminates the unnecessary functions and features installed inside the other COTS HIL.



Figure 1. VIP displaying important information to the vehicle real-time condition [3]

Table 1. Previous research using HIL for testing and evaluation purposes [4]–[13]

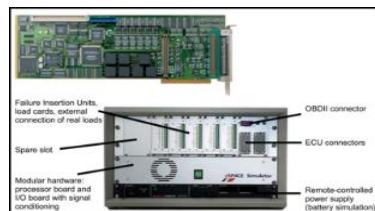
Literature Journal	Device-Under-Test	Test Equipment
Wang et al. 2015	Hybrid Electric Bus Control System	dSPACE TTC200 controller
Fasano et al. 2013	Multisensor Sense and Avoid System	i. Panchromatic and color camera (Allied Vision Technologies Marlin) ii. Infra-Red Camera (FLIR A40V)
Chen et al. 2012	Electronic Control Unit (ECU) of Dual Clutch Transmission Vehicle	i. Data Acquisition card PCL-818LS ii. Control hardware FreeScale MC9S12DP256
Kim et al. 2012	Light Detection and Ranging Sensor	i. Low Jitter Delay Generator DG535 ii. A321 AOM Driver iii. Acousto-Optic modulator
Rad et al. 2012	Actuation System of a 2-DOF Parallel Robot	i. Development board FiO Std ii. MATLAB Simulink
Maxwell et al. 2010	GM Two-Mode Hybrid Electric Vehicle	i. dSPACE MicroAutoBox (MABX) ii. National Instruments PXI.
Mouzakitis et al. 2009	ECU Diagnostic Software	i. MATLAB ii. dSPACE ControlDesk iii. Fault insertion unit
Huang et al. 2008	Instrument Cluster	i. dSPACE Simulator ii. DS2211 HIL I/O Board
Skjetne & Egeland 2006	Marine Control Systems	CyberSea Simulator
Ramaswamy et al. 2004	ECU for a Hybrid Electric Vehicle	i. dSPACE Simulator ii. DS2210 HIL I/O Board



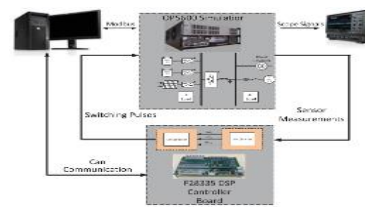
(a) Maxwell et al. 2010



(b) Ramaswamy et al. 2004



(c) Kim et al. 2012



(d) Shi et al. 2017

Figure 2. Conventional HIL utilisation for testing and simulation purposes

Hence this research aims to adopt HIL system for testing VIP using an economical approach. This paper is outlined in six different sections. Section 2 explains the testing device's overall design. Section 3

explains the Graphical User Interface (GUI) applied to run the simulation using either manual as shown in subsection 3.1 or using Speed Profile Selection in subsection 3.2. Section 4 explains the hardware and software framework of the signal generation and Section 5 explains the results of the speedometer testing and the cost expenditure of the COTS HIL. Finally, this paper summarizes the economical approach for testing VIP using HIL system.

## 2. OVERALL DESIGN OF THE TESTING DEVICE

Figure 3 shows the design architecture of the proposed test equipment. The architecture consists of three main sections which are the Test Input, HIL system and the DUT unit. The Test Input section is where the test personnel inputs the test parameter and performs the test simulations via the GUI designed using Qt Creator.

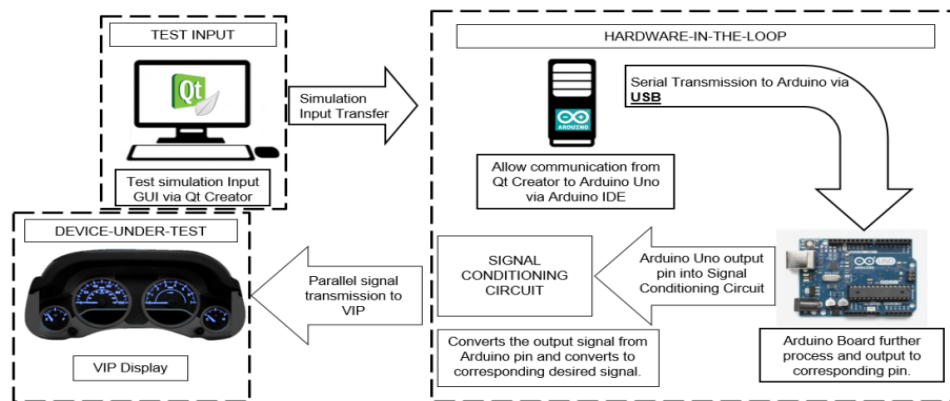


Figure 3. Overall Design of the Testing Device

The HIL section consists of two main subsection blocks which are the GUI design software and the signal generator hardware. User interface is where the HIL system receives the test input from the Test Personnel. It is then transferred to the microcontroller which further transmits to the Signal Conditioning Circuit (SCC) for further process and transmits the corresponding signal generation of frequency-based, resistance-based or digital logic. The proposed test system is able to perform testing and verification as required by the DUT with less expenditure compared to the conventional HIL product available in the market.

## 3. GRAPHICAL USER INTERFACE

User interface which is part of the HIL system is used to launch the testing operation. In this paper, Qt Creator will be utilized as the software IDE to design the GUI via two main modes which are the Design mode and Edit mode.



Figure 4. Qt Creator's Edit Mode to input the desired source code [14]

Figure 4 shows the Edit Mode inside the Qt Creator IDE, which contains the source code to program the GUI based on the interaction of the input and output widget. Files directory as shown in Figure 4a is selected and the related source code will be displayed in the Editor's View as shown in Figure 4b. Figure 5 shows that the design of user interface is conducted inside the Design Mode. It comprises of vast amount of Input / Output widgets such as Push Button, Tool Button and Check Box as shown in Figure 5a. Any action taken towards the input widget i.e. clicking the push button, toggling the radio button, or ticking the check box will be processed according to the source code programmed resulting to the output responded by the output widget i.e. opening a dialog box, rising an analog reading or showing the digital reading.

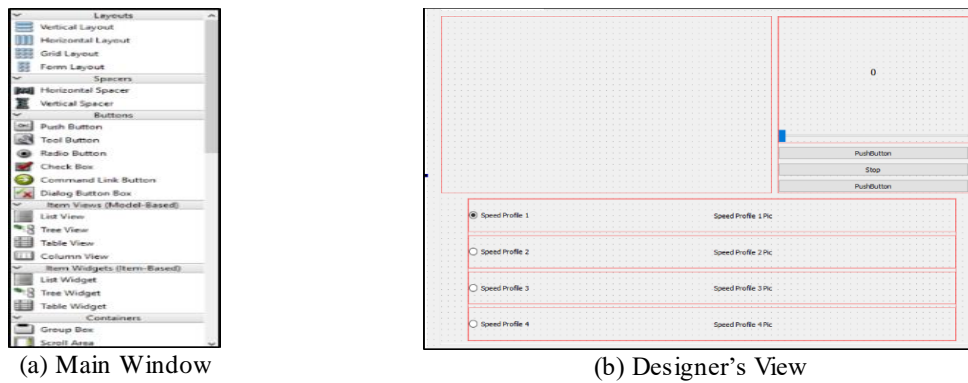


Figure 5. Qt Creator's Design Mode to input and allocate the desired input/output widget [15]

### 3.1. Manual Input Test Using Slider Widget

Figure 6 shows how the manual input by the slider widget produces corresponding display at the GUI and transmits frequency data into Arduino IDE. Based on Figure 7, the GUI consists of a slider widget as the input widget used to control the speed reading which will be displayed on the speedometer gauge. The speed range is from 0 to 160 km/h as the test personnel moves the slider from right to left. Besides resulting the speedometer gauge to move, the movement of the slider will also display the corresponding frequency output display which is positioned on top of the slider widget. The frequency output is displayed since the DUT receives frequency signal in order to display the speed of the vehicle in actual condition. The corresponding value of frequency for every speed reading displayed on the speedometer is according to equation 1.

$$V = \frac{s}{f} F \tag{1}$$

$V$  is the resulting speed value displayed on the Speed gauge and the digital display inside the GUI.  $s$  is the maximum value of the speed on the DUT and  $f$  is the maximum value of frequency required to generate the maximum speed value.  $F$  is the required frequency value to display the desired speed value of  $V$ . In this particular case,  $s$  is 160 and  $f$  is 707.78.

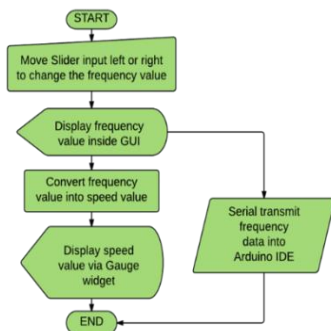


Figure 6. Flowchart of the GUI Manual Input Process

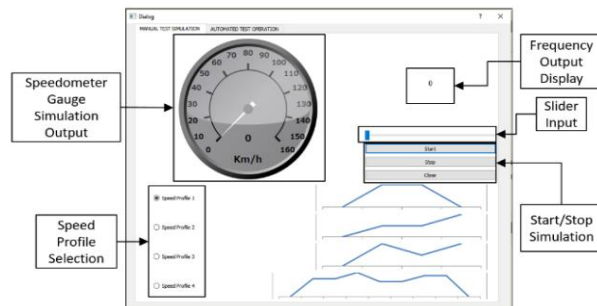


Figure 7. GUI for performing Speedometer testing using Speed Profile simulation or Slider Input widget [3]

### 3.2. Simulation via Selection of Speed Profile

Figure 7 also displays an option to perform driving profile simulation. 4 speed profiles are introduced for the testing purpose of the speedometer. The test personnel may select one speed profile by clicking at the radio button of the desired speed profiles before clicking the start button. Beside the radio buttons are the speed profile graphs showing whether the speed will be either decreasing, increasing or remain at the same speed.

Speed Profile 1, 2 and 3 consists of 3 stages each with different slope of either increasing, decreasing or constant speed. All three of the Speed Profiles have the similar simulation time. Meanwhile Speed Profile 4 consists of 8 stages and has a longer simulation time. As the simulation of the Speed Profile is performed, the changes in frequency value at the top right and the changes in speed value inside the speedometer are observable.

## 4. SIGNAL GENERATION FRAMEWORK

This section will explain the generation of the desired signals prior to testing the DUT. As the GUI performs simulation using the speed profile and manual input using the slider widget, this data is then transferred to the Arduino microcontroller which is further conditioned to the desired signal using Signal Conditioning Circuit. The signal generation framework consists of a Personal Computer (PC), PC-to-DUT communication means and Signal Conditioning Circuit (SCC).

### 4.1. Host PC

The PC is used as the medium to install the utilized software and IDE such as Qt Creator IDE and Arduino IDE. Arduino is an open source physical computing platform based on a simple input/output (I/O) board and a development environment [16]. Qt is a C++ toolkit with an extension for QML and JavaScript [14]. All the testing operation will be conducted by the test personnel using the PC via the GUI explained in Section 3. As the test personnel performs the test either using the speed profile simulation or the slider widget, the value of speed and frequency inside the GUI will be changed and displayed accordingly. Further, the value of frequency output displayed is also be processed by the Qt Creator to obtain meaningful data prior to the signal generation. The data are then converted to character strings before serially transmitted into the Arduino Uno by embedding several lines of source codes and header file to establish the Qt-Arduino connection as shown in Figure 6.

### 4.2. Device-Under-Test Communication Interface

It is vital to have a communication platform between a PC host and DUT. At this point, the PC host is unable to process further to generate the desired signal input to the DUT. Since the DUT consists of male harness connectors and no USB female ports or any PC-compatible connectors to be linked directly to the PC as shown in Figure 8. Thus, Arduino IDE performs the interconnection between the PC and the DUT via USB connection between the PC and the Arduino USB port. The output pin of Arduino Uno is then connected to the pinout of the DUT via Signal Conditioning Circuit. Arduino IDE is utilized to program the source code to communicate with the Qt Creator and to generate the digital signal via output pins or SPI bus. Arduino is an open-source platform that can be used to build and program electronics, design and invent wide range of mechatronics project [17]. Due to its simplicity and compatibility to communicate with PC, Arduino Uno is chosen from the other Arduino board members.



(a) Back view consists of two separated male connectors receiving +5 V, +12 V and ground signals



(b) Side view consists of two male connectors receiving frequency and resistance-based signals

Figure 8. Back and Side view of the DUT

Source code is developed and uploaded from the IDE into the Arduino Uno board by the means of USB. By using the several commands such as Serial begin(9600), Arduino communication with PC are established and PC will be started transmitting the signal into the Arduino board. This open-source prototyping circuit board will be used to generate a digital logic signal based on the information obtained from the PC. It will generate two types of output; the digital signal and the Serial Peripheral Interface (SPI) signal. This signal generation will then be transmitted to the SCC for further processing.

**4.3. Conditioning Signals from Arduino to the DUT**

Figure 9 displays the block diagram of the signal generation which is the interconnection of the Signal Conditioning Circuit between Arduino Uno and the DUT. Since some of the DUT gauges and tell-tales displays cannot be triggered by simply feeding +5 V signals from the Arduino Uno, thus certain signal conditioning process is required to be designed. The circuit consists of two sections of input which is the digital logic with value of 0 V or 5 V, and the SPI bus connection.

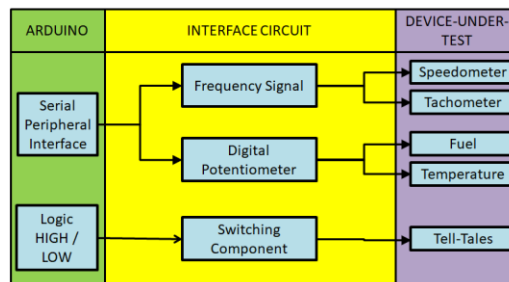


Figure 9. Signal Conditioning interfacing circuit connecting from Arduino to the DUT

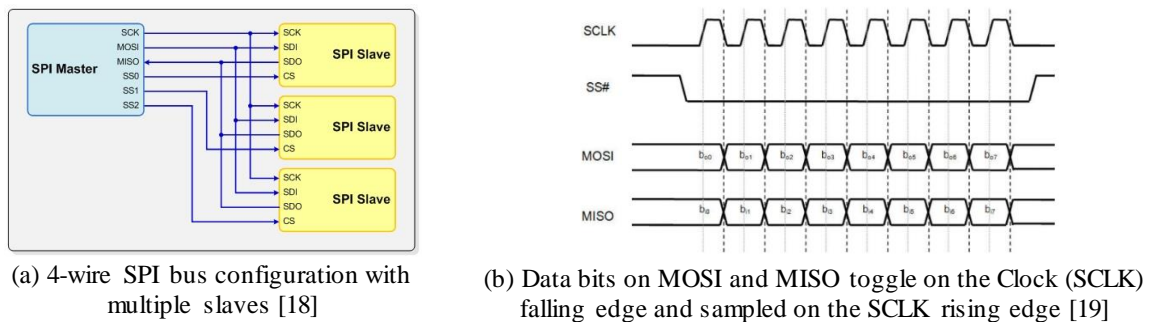


Figure 10. SPI configuration and Timing Diagram

Digital logic signals representing the tell-tale input signals generated from the Arduino Uno is then processed accordingly. 5 V signal is fed directly, via the DUT pin connection, to the 'Left' and 'Right' Turn tell-tale signal. However, a higher voltage supply requirement is needed for activating the Upper beam headlight. 12 V signal is required to be fed to the DUT and external grounding is required. For both Battery and Hand-Brake signal, a ground signal is required. By using optocoupler for each case, the current flow according to the required standard as above can be achieved. Optocoupler is an optical switch (power electronic switch) which is designed to isolate electrical output from input for the purpose of protection from such intermittent voltage and noise [20].

SPI bus is used for two types of signal which are frequency and resistance. SPI has a simple 4-wire synchronous interface protocol that enables controllers and peripheral devices to intercommunicate [21]. Multiple SPI connections can be achieved by connecting to the respective pins of Master- In-Slave-Out (MISO), Master-Out-Slave-In (MOSI), Clock and Slave Select (SS) to the SPI slave pins of the related IC chips accordingly. Figure 10a shows the multiple slave connection, with the SPI master consists of multiple Slave Select pins to control the desired SPI slave. To select the desired SPI slave, the corresponding SS is pulled low and the master activates the clock signal, hence generates the information on to MOSI line as the slave generates another information through MISO line as shown in Figure 10b [19].

For generating a frequency with a resolution of 0.01 Hz, the SPI connection from Arduino is connected to the Frequency Signal Generator Board. This board then generates the required frequency output for the Speedometer and Tachometer. A dedicated frequency generator solution will guarantee deterministic square wave signal generation in its own accord avoiding the extra loading on the main microcontroller.

Meanwhile, SPI bus is also connected to the digital potentiometer to generate the resistance output. The value of the potentiometer can be digitally controlled according to the data received from the Arduino Uno via the SPI bus. The lowest available digital potentiometer with an SPI interface is 5,000  $\Omega$  with 8-bit resolution (256 steps). To accommodate the range of the re-sistance signal required which is within 0  $\Omega$  to 200  $\Omega$ , multiple digital potentiometers are arranged and connected in parallel to enhance the resistance resolution but maintain the number of resolution steps. Thus, the resolution of the resistance value achieved when performing the parallel connection of the potentiometer is up to 3.906  $\Omega$ /step.

**5. RESULT AND ANALYSIS**

In this section, the result of utilizing the GUI and its corresponding signal generation for Speedometer test is discussed.

**5.1. Test Result for Device-Under-Test**

For this test, the personnel will be using the Slider widget to control the speed reading displayed on the GUI, which also displays the corresponding frequency value. The actual frequency value is also verified (but not as part of automated test process) using Agilent Digital Oscilloscope by probing the frequency output at the DUT Speedometer pin-out. The image of the DUT is also taken and measured to compare the actual Speedometer reading on the DUT compared to the GUI Speedometer reading.

Figure 11 shows the GUI display and the actual speedometer reading on the DUT. The right side shows the manual slider input which controls the frequency that is going to be generated. Meanwhile, the middle side shows the GUI display, which is the frequency of 100 Hz and speedometer value of 22.6059 km/h as calculate from the Equation 1. The left side of the figure is the image captured with the resulting speedometer reading when the frequency signal generated is 100 Hz, which is nearly 25 km/h. Figure 12 and 13 shows the other frequency generation which produce various speedometer reading. The accurate DUT speedometer measurement has been accomplished using image processing and the results are displayed in Table 2.



Figure 11. GUI and DUT corresponding speedometer reading for frequency generation of 100 Hz

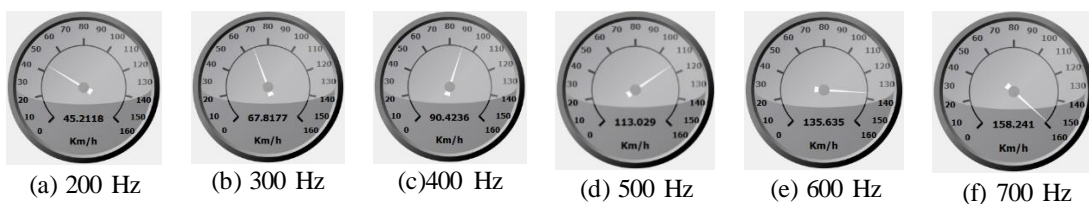


Figure 12. GUI Simulated Display

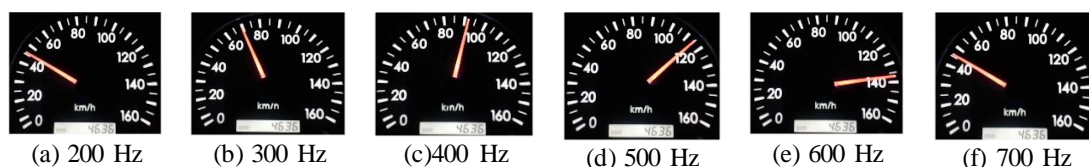


Figure 13. DUT (Actual VIP) Display

Table 2. Speedometer Test and Variation with Actual Result

Frequency Value (Hz)			Speedometer Value (km/h)		
GUI	Oscilloscope	Variation	GUI	DUT	Variation
100	100.15	0.15	23	23	0
200	200.30	0.30	45	46	1
300	300.45	0.45	68	69	1
400	400.64	0.61	90	91	1
500	499.96	0.04	113	115	2
600	600.91	0.91	136	139	3
700	702.18	2.18	158	162	4

Figure 14 shows the graphical representation for speedometer reading for both GUI and DUT. The lower section is the column chart showing the reading of speedometer, while the upper section is the line chart displaying the difference of the speed value between GUI and DUT. It is observed that the difference does not exceed the upper and lower limit of the tolerances pre-defined by the customer, which is the DUT's manufacturer.

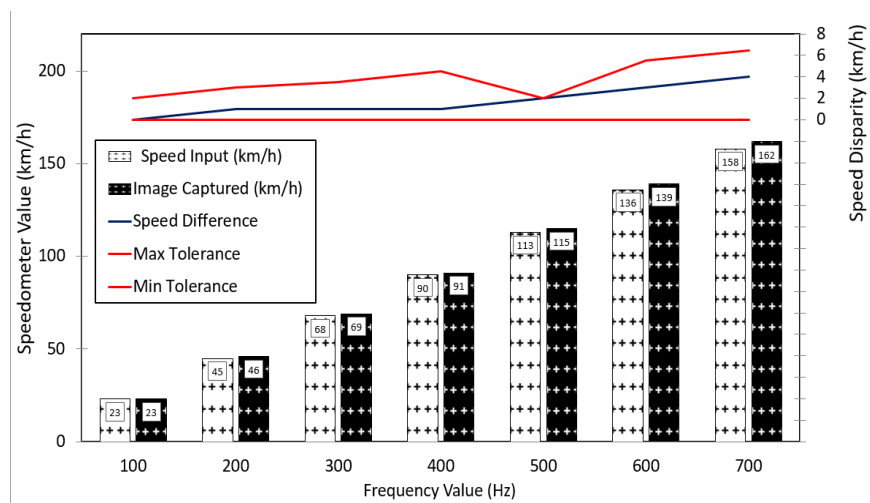


Figure 14. Speedometer reading comparison between GUI and DUT. Upper section shows the disparity does not exceed the maximum and minimum tolerance pre-assigned by the DUT manufacturer

5.2. Analysis of Cost Expenditure

The proposed HIL test equipment eliminates the requirements of utilizing costly COTS HIL. At affordable price, less expensive electrical components and microcontrollers were purchased and the test equipment is designed from ground up to meet with the testing requirements. Table 3 shows the cost of the project if the COTS HIL is used on this project. Excluding the charges such as after-sales service and spare parts, using the proposed system is definitely reduces the cost of the project. Moreover, the proposed system maintenance is expected to be much lower than the COTS HIL due to the fact that the knowledge about the system architecture is known to the designer/engineer.

Table 3. Available COTS HIL in market with the estimated budgetary price

Item	Specification	Description	Brand	Price (RM)
1	PXI-2720	PXI Programmable Resistor Module	NI	11,973.76
2	PXI-2586	PXI Relay Module	NI	6,530.66
3	cRIO-9063	4-Slot Integrated Dual-Core Controller	NI	6,871.98
4	NI 9482	4-Ch SPST Relay Module	NI	929.62
5	NI 9421	8-Ch Sinking DI Module	NI	523.64
6	sclx fs 100	Failure Simulation for 100 Functions	dSPACE	5,842.24
7	MOTIONDESK	MotionDesk 3D Animation Tool with USB dongle	dSPACE	11,389.68
8	ACE SCLX MC	Advanced Control Education Kit SCALEXIO MultiCore	dSPACE	32,124.98
9	POWER SUP DELTA	Delta Power Supply	dSPACE	18,008.70



## 6. CONCLUSION

In this paper, the test equipment designed to generate the simulation input for VIP has been described. The frequency generated had successfully displayed the corresponding speed reading inside the speedometer. Minor differences which is 4km/h at most are observed and is within the pre-assigned tolerance. Using Arduino Uno as the microcontroller with the addition of a specifically designated SCC for further signal conditioning, and Qt Creator IDE and Arduino IDE as the software framework, it is possible to adopt a HIL system for testing VIP using economical approach.

## ACKNOWLEDGEMENT

This work is supported in part by Universiti Sains Malaysia (USM) and Collaborative Re-search in Engineering, Science and Technology (CREST) Research Fund under Grant 304/PELE-CT/6050344/C121.

## REFERENCES

- [1] M. D. Raj and V. S. Kumar, "Vision based feature diagnosis for automobile instrument cluster using machine learning," in *2017 4th International Conference on Signal Processing, Communications and Networking (ICSCN - 2017)*, pp. 1–5, 2017.
- [2] P. B. Renaux, *et al.*, "Hardware-in-the-Loop Simulation Low-Cost Platform," *2017 VII Brazilian Symp. Comput. Syst. Eng.*, no. 2324, pp. 173–180, 2017.
- [3] W. S. W. Jamaludin, *et al.*, "Cost-Effective Microcontroller-Based Hardware-In- the-Loop Test Equipment for Testing Vehicle Instrument Panel," in *International Conference on Smart Instrumentations, Measurement and Applications*, pp. 6, 2017.
- [4] J. Wang, *et al.*, "Hardware-in-the-loop test for the design of a hybrid electric bus control system," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol/issue: 229(15), pp. 2760–2773, 2015.
- [5] G. Fasano, *et al.*, "Real-time hardware-in-the-loop laboratory testing for multisensor sense and avoid systems," *Int. J. Aerosp. Eng.*, vol. 2013, 2013.
- [6] R. Chen, *et al.*, "A New Hardware-in-the-Loop Test System for Electronic Control Unit of Dual-Clutch Transmission Vehicle," *Adv. Mater. Res.*, vol. 490–495, pp. 13–18, 2012.
- [7] H. J. Kim, *et al.*, "Hardware-in-the-loop projector system for light detection and ranging sensor testing," *Opt. Eng.*, vol/issue: 51(8), pp. 83609–1, 2012.
- [8] C. R. Rad, *et al.*, "Hardware-In-The-Loop (HIL) Simulation Used for Testing Actuation System of a 2-DOF Parallel Robot," *Appl. Mech. Mater.*, vol. 162, pp. 334–343, 2012.
- [9] T. Maxwell, *et al.*, "Hardware-in-the-loop testing of GM two-mode hybrid electric vehicle," in *2010 IEEE 12th Workshop on Control and Modeling for Power Electronics (COMPEL)*, pp. 1–5, 2010.
- [10] A. Mouzakitis, *et al.*, "Hardware-in-the-Loop System for Testing Automotive Ecu Diagnostic Software," *Meas. Control*, vol/issue: 42(8), pp. 238–245, 2009.
- [11] Yingping H., *et al.*, "Design validation testing of vehicle instrument cluster using machine vision and hardware-in-the-loop," in *2008 IEEE International Conference on Vehicular Electronics and Safety*, pp. 265–270, 2008.
- [12] R. Skjetne and O. Egeland, "Hardware-in-the-loop testing of marine control systems," *Model. Identif. Control*, vol/issue: 27(4), pp. 239–258, 2006.
- [13] D. Ramaswamy, *et al.*, "A case study in hardware-in-the-loop testing: Development of an ECU for a hybrid electric vehicle," *Sae Sp*, no. 724, pp. 33–42, 2004.
- [14] T. L. Tien, *et al.*, "An Evaluation on the Effective Processing Time of Raspberry Pi-3B-based Embedded Systems for Determining Forged Images," in *2017 4th NAFOSTED Conference on Information and Computer Science*, pp. 213–218, 2017.
- [15] "User Interface Qt Creator Manual," 2017. [Online]. Available: <http://doc.qt.io/qtcreator/creator-quick-tour.html>.
- [16] A. Taksale, *et al.*, "Low cost hardware-in-loop for automotive application," in *2015 International Conference on Industrial Instrumentation and Control, ICIC 2015*, pp. 1109–1114, 2015.
- [17] M. Banzi, "Getting Started with ARDUINO," Second Edi. USA: O'Reilly Media, Inc., 2011.
- [18] Corelis, "SPI Interface - Tutorial," 2017. [Online]. Available: [https://www.corelis.com/education/SPI\\_Tutorial.htm](https://www.corelis.com/education/SPI_Tutorial.htm).
- [19] "Introduction to I2C and SPI protocols – Byte Paradigm – Speed up embedded system verification," Byte Pradigm sprl, 2016. [Online]. Available: <http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/?/article/AA-00255/22/Introduction-to-SPI-and-IC-protocols.html>.
- [20] S. Materials, "Basic Components of Digital Protective Relays 1.1," pp. 1–53.
- [21] E. Lin, "AN 98542 Designing a Serial Peripheral Interface," California, 001–98542 Rev. \*A, 2015.