

Automated BFT testing in LABCAR using ECU Test Tool

Divya Perumal¹, Dr. M. Suresh²

¹Student, Dept. of Instrumentation and Control Engineering, St. Joseph's College of Engineering, Chennai, India.

²Associate Professor, Dept. of Instrumentation and Control Engineering, St. Joseph's College of Engineering, Chennai, India

Abstract - Real time test vehicles are responsible for testing automotive electronic control unit (ECU) in a virtual environment. Many functionality check and testing are done in virtual HIL environment before testing in real environment. The functional verification testing of ECU is becoming difficult now a days due to many interconnected ECU's from different suppliers. This complexity is expected to continue at an exponential rate in future. The manual process of verifying the functionality of ECU is a tedious procedure and subjected to many manual errors. The main aim of the project is to generate a Base Function Test (BFT) report of an Electronic Control Unit (ECU) using ECU Test tool. This test automation helps majorly in reducing the complexity, testing effort and time for executing. It also reduces the errors caused during the manual testing. Therefore, the developed test automation approach can be beneficial to the embedded system developers/testers to increase the test coverage and develop HIL testing framework.

Key Words: ECU, BFT, Real-time testing, Automation, Simulation, Models, Hardware-In-The-Loop.

1. INTRODUCTION

The development of today's modern vehicles with increased level of performance is unimaginable without the use of ECU's. The components of vehicles are controlled and monitored by different ECU's. These ECU's are constantly becoming more powerful and thus more complex in terms of their functionality. Accordingly, the risk of software errors also increases. Hence, testing the functionality of the ECU is very important before integrating it into the vehicle. Another problem encounters the increasing number of interconnections between the different ECUs. The malfunctioning of one ECU can easily cause malfunctioning of another. Hence, testing plays a crucial and important role before manufacturing an ECU. If anything goes wrong in the testing, it causes a major defect in the vehicle. Most errors are caused during manual testing process. Therefore, an automated testing environment is necessary to increase the efficiency of an ECU. In order to validate an ECU network successfully, the manufacturer must combine innovative automated test methods with an integrated development process and incompatible set of tools.

2. HARDWARE IN THE LOOP (HIL)

Hardware-in-the-loop (HIL) simulation is a method used in the development and test of complex real-time embedded systems (ECU's). The HIL system operates by testing ECU's in

a closed-loop system with components that are simulated in a virtual environment so that they can be completely tested without needing to have the complete product available. In this project, a car is under control to the test platform and the ECU to be tested interacts with this "virtual car" in real time. Fig-1 represents the HIL principle in ECU testing.

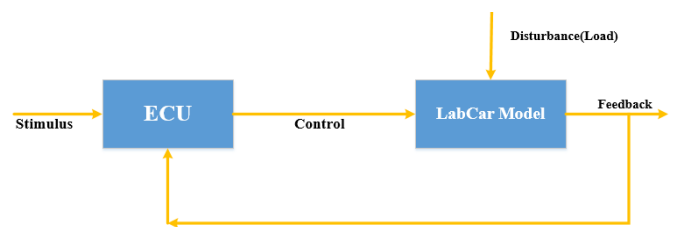


Fig -1: HIL Principle

3. TESTING IN A LABCAR ENVIRONMENT

LABCAR is a real-time test bench system for automotive Embedded Control Units. A LABCAR provides a virtual lab environment for testing an ECU. Testing in the live environment can sometimes prove costly – with respect to life and property. And in some cases, it may not be exhaustive to test all the scenarios. Also, time precision will be an issue during heavy traffic in the gateway. LABCAR is used to simulate network management signals as well as hardwired signals without any manual interruption in hardware. To improve time precision, multiple sensor calculations are possible in LABCAR at the same time and it provides real time simulated outputs. Testing at LABCAR requires complete software. Since unit testing involves testing of individual components without building complete software it is not preferred to use LABCAR at this level. The ECU to be tested is connected to the LABCAR Hardware. The ECU in the real car receives information from different sensors and the real vehicle is simulated by the model in the LABCAR PC. Thus, LABCAR provides a virtual car environment to the ECU under test. The driver inputs are sent by the tester to the LABCAR virtually through the GUI. Based on the GUI inputs, the LABCAR will calculate and send the sensor values to the connected ECU, which are provided by various sensors in the real vehicle. The LABCAR receives the output signals (from the ECU) which are processed by different actuators. Based on the ECU output, the vehicle behavior will be virtually simulated by the LABCAR. The implementation of these signals must simulate the behavior of a real vehicle so accurately, that the ECU can change into emergency mode without detecting errors.

CANoe/CANalyzer tools from Vector can simulate the network management signals and not hardwired signals. E.g. Vehicle speed in GUI will be reduced by LABCAR when brake pedal value is increased in GUI. Fig-2 illustrates the block diagram of LABCAR.

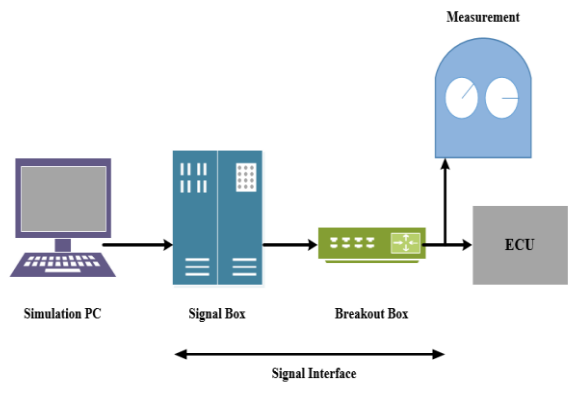


Fig-2: An illustration of LABCAR

4. INTEGRATED CALIBRATION AND APPLICATION TOOL (INCA)

INCA is a tool set for measurement, calibration and diagnostic with various products offering a wide variety of functions for ECU development. It has user-friendly experiment environment with graphical measurement displays, oscilloscopes and editors for the calibration of characteristic values, curves and maps. Also, it has easy integration of user-specific display and control elements. It supports of many interface and measurement modules and drive recorders. INCA contains databases for consistent management of ECU projects, hardware configurations and experiments. It acts as an open interface for automation and integration with the tool environment.

5. BASE FUNCTION TEST (BFT)

The Base Function Test is a black box test which ensures the basic functionality of the software (injections, engine running, etc.). As a part of the system test, it serves as the first step for ensuring the basic functionality of ECU. Communication, start functions and the interplay of software and hardware are verified after the complete package integration through this test.

5.1 BFT Test Steps

Flash Programming Test - Detection of errors during programming (BFT supports flashing via INCA, DIAGRA and Customer specific flash tool.

FCM Test - Detection of error memory entries and resets which cause shut-offs

ECU States Test - Check of special system functions which are important for engine operation (ECU-Initialization,

detection of the engine states, synchronization, idle speed detection, overrun, after run condition)

Core Values Test - Detection of physical implausibility

Runtime Performance Test - Detection of real-time violations and resets

Engine in Operation Test - Detection of defective injections which are able to cause a damage of the engine

EEPROM Emulation Test - Check data saving to EEPROM with regard to block and sector changes (data should not be overwritten from data) and to write order (detection of not authorized mass data saving)

Error Memory End Test - Check that there are no unexpected error memory entries occurred during the complete Test run.

Test step	Characteristic / function checked	Test objective
BFT_1_1	Flash programming Test	Detection of errors during programming
BFT_2_1	Read and clear the error memory	Detection of error memory entries and resets which cause shut-offs
BFT_3_1	ECU States Test	Check of special system functions which are important for engine operation (ECU-Init, detection of the engine states, synchronization, idle speed detection, overrun, after run, ...)
BFT_3_2	Functionality check for "core values"	Detection of physical implausibility
BFT_3_3	Runtime-Test	Detection of real-time violations and resets
BFT_4_1	Engine in operation	Detection of defective injections which are able to cause a damage of the engine
BFT_4_2	EEPROM-Test	Check data saving to EEPROM with regard to block and sector changes (data should not be overwritten from data) and to write order (detection of not authorized mass data saving)

Table -1: BFT Test Steps

6. ECU TEST TOOL

ECU TEST is a software tool developed for test and validation of embedded systems. It aims at the specification, implementation, documentation, execution and assessment of test cases. It automates and controls the whole test environment and also supports a wide range of test tools. Fig-3 shows the ECU Test tool workspace.

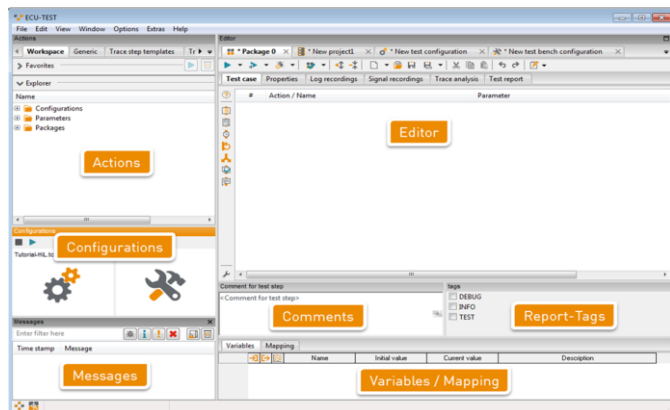


Fig-3: ECU Test Tool

7. RESULTS AND DISCUSSION

7.1 INCA Testing

The A2L and HEX files generated during the build process are flashed and tested in the INCA software. The testing is done as shown below in Fig-4.

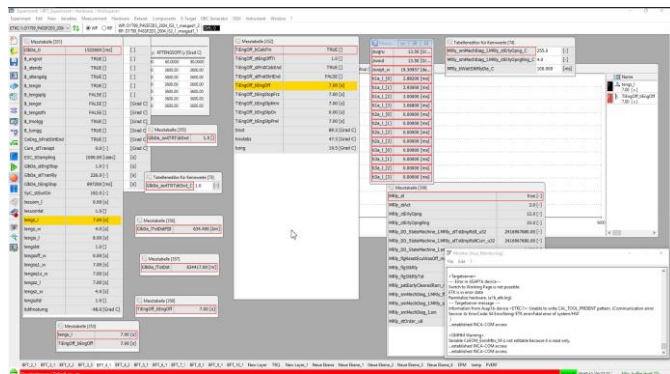


Fig-4: INCA window

Manual INCA testing is difficult because the exact variables should be added and monitored for different values of brake and accelerator pedal. But, by using ECU-Test Test Package, INCA testing will be automated and also very easy to implement.

The values of the LABCAR model in LABCAR EE while testing the HEX and A2L file in INCA is shown below in Fig-5.

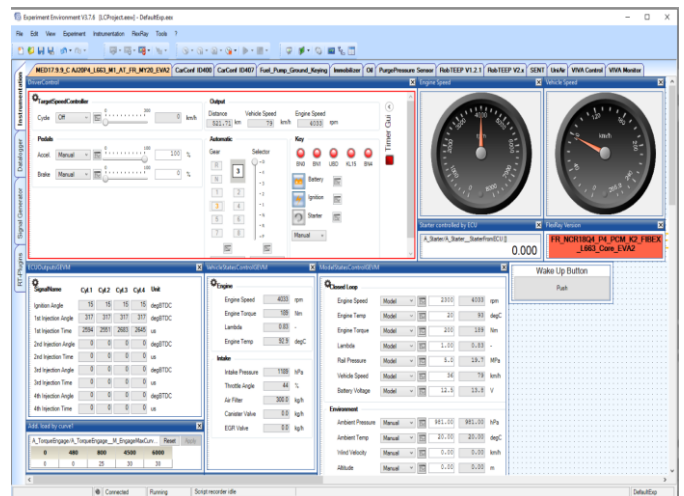


Fig-5: LABCAR EE window

7.2 ECU Test Results

The following figures shows that how the BFT test cases are carried out when we execute the project wizard of BFT package. After configuring TBC and TCF, the LABCAR model of the particular project is updated in LABCAR EE and INCA is loaded with the properties of A2L and HEX file. Then, the project wizard or the particular testcase should be executed. This command will run the testcase one by one by executing its test commands. Fig-6 depicts the execution of testcases one by one while running the BFT Project wizard.

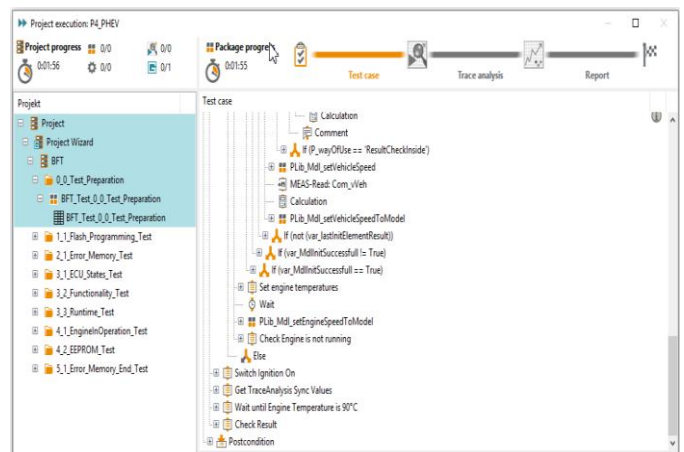


Fig-6: Project wizard execution

Execution of all the test cases, followed by trace analysis which is mentioned in every test case. Finally, a test report will be generated where it can be converted to any pdf or xlsx form depends upon user's convenience. Fig-7 shows the test report execution of a test case to determine Maximum Engine Speed.

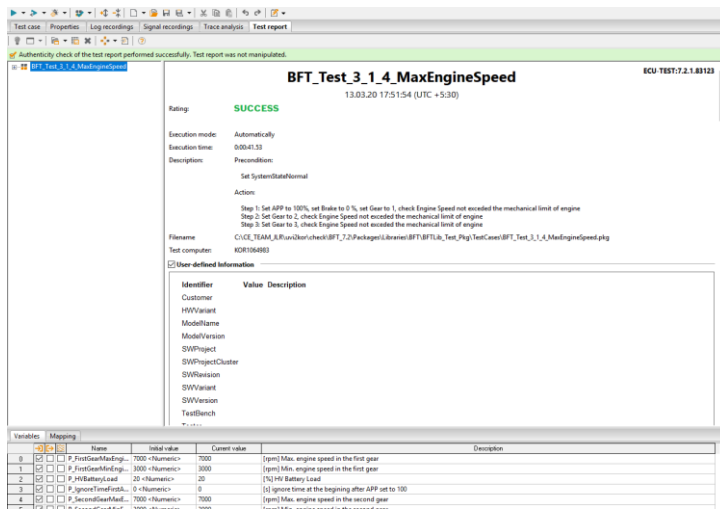


Fig-7: Single TestCase execution

The Test report contains the following,

Rating - Whether the testcase execution is SUCCESS or FAILURE.

Execution Mode - It depicts whether the testcase has been executed Automatically or manually.

Execution Time - It shows how much time it has taken for executing the particular testcase.

Description - It explains the precondition, action and postcondition for the particular testcase.

Filename - It contains the path of the test package.

7.3 Maximum Engine Speed Test Case Results

In Maximum engine speed test case, the following actions are performed, first gear is activated. Then, the variable APP to 100%. Then, wait for WaitMaxEngSpeed to activate. Again, set variable APP to 0%. These steps are repeated for second and third gear. Fig-8 shows the execution of test cases in a step by step manner followed by the **SUCCESS** command on the test case which is executed without any error.

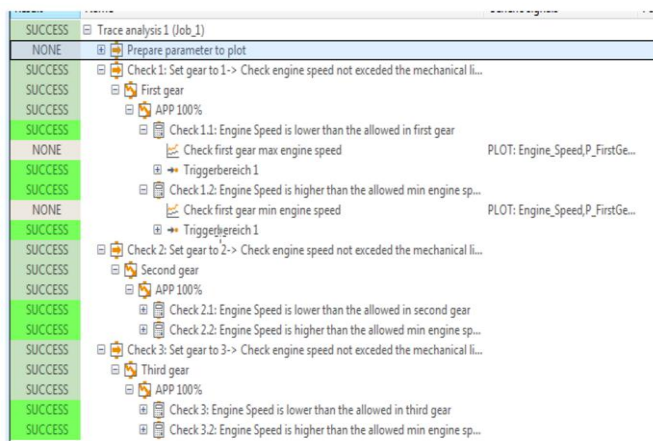


Fig -8: Maximum Engine Speed Test Case Package Execution

The value for maximum engine speed is 7000rpm. The value for minimum engine speed is 3000rpm. The wait period is 15sec. The maximum engine speed at first gear is recorded and seen through MDA tool is shown in Fig-9.

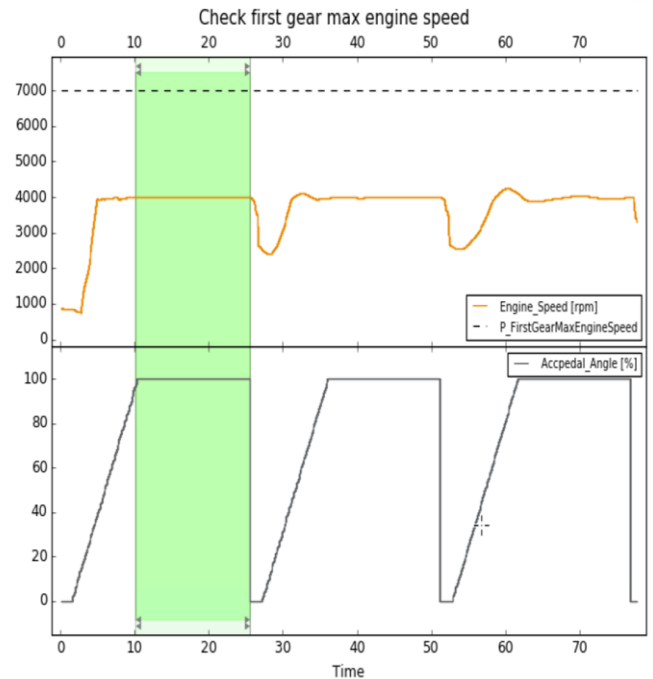


Fig-9: Maximum Engine Speed at first gear

The minimum engine speed at first gear is recorded and seen through MDA tool is shown in Fig-10.

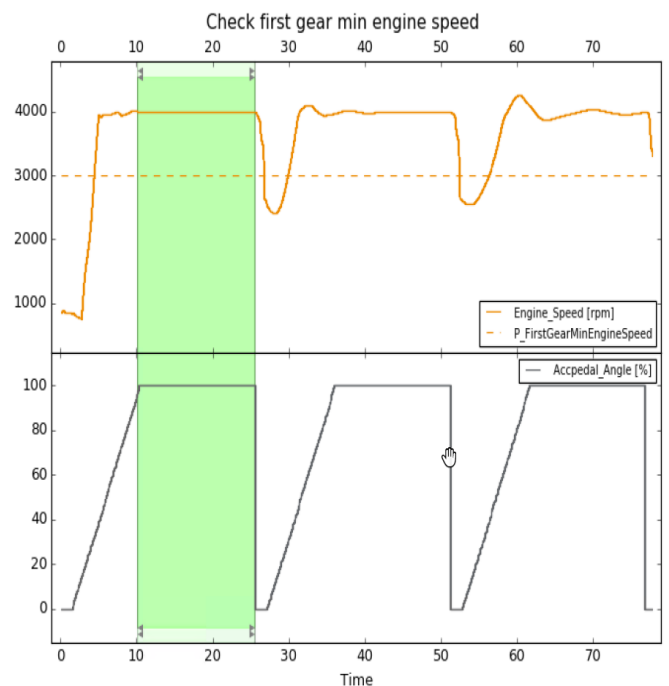


Fig-10: Minimum Engine Speed at first gear

8. CONCLUSIONS

Based on the analysis of results, a first step in this project is reducing the manual process of testing with INCA and LABCAR EE. The process of testing an ECU for BFT is very difficult and time consuming. Hence, introducing some testcases through ECU-Test and loading the TBC, TCF can reduce the manual work and also makes the result more accurate. In BFT testing, more than 12 tests are carried out to check the Boot-time, Engine start, Drivability, Low Idle and Overrun Detection test, Maximum Engine Speed test, Engine-Off test, Ignition On Test, Electrical Machine Idle Test, Drive Check Test, Accelerate Test, Recuperation Test, Engine Stop Test, Functionality check for core values, Runtime Test, EEPROM Test, Error Memory End Test. These tests are performed to check the functionality of the ECU. Hence, this project achieves in reducing the time consumed and manual work done by the user, increasing the accuracy and efficiency of testing, provides more clear results via MDA tool and can be imported into pdf or xlsx format, trace analysis can be done easily through the test cases, The project wizard can be easily loaded just by SVN software check in and check out process. Then it is added to the workspace of ECU-Test.

9. FUTURE WORK

As an extension of this project, the future work would include to perform analysis of data file format by automating LABCAR EE accelerator and brake pedal sliders with respect to data file values. Hence, certain testcase has to be written to read the data file variables and map that values of variables to the LABCAR model. This is a complete automation process where it fetches the values from the testing results and perform the functionality test.

REFERENCES

- [1] Prabodh Shende, Darren Scarfe, Wyatt Meek, Suresh "Automating HIL Test Setup and Execution" IEEE AUTOTESTCON 2008, Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989, Proceedings- 978-1-4244-2226-5. M.
- [2] E. Bagalini, M. Violante "Development of an Automated Test System for ECU Software Validation: An Industrial Experience" 15th IEEE Biennial Baltic Electronics Conference (BEC2016), Proceedings- 978-1-5090-1393-7.
- [3] Velibor Ilic, Srđan Popic, Milan Kovacic "Data flow in automated testing of the complex automotive electronic control units" IEEE 2016 Proceedings-978-1-5090-29570.
- [4] FAN Yue-zhen, JIANG Fa-chao "Hardware-In-the-Loop Simulation of Electronic Control Unit Used on Automated Mechanical Transmission" IEEE 2010, Proceedings- 978-1-4244-6349-7.
- [5] Krisztian Enisz, Denes Fodor, Istvan Szalay, and Laszlo Kovacs "Reconfigurable Real-Time Hardware-in-the-Loop Environment for Automotive Electronic Control Unit Testing and Verification" IEEE Instrumentation & Measurement Magazine 2014.