

# TEST SUITE DEVELOPMENT USING LABVIEW AND NI TESTSTAND FOR OS BASE SOFTWARE

Ashwini. A. Pandit<sup>1</sup>, Prof. P. B. Borole<sup>2</sup>

<sup>1</sup>M. Tech, Electronics and Telecommunication, <sup>2</sup>Assoc. Professor, Electronics and Telecommunication,  
<sup>1,2</sup>VJTI, Mumbai, Maharashtra

---

**Abstract:** Different modules in a software are developed by different teams or team members in any software development firm; due to this, when the software is compiled, there are chances for different software errors commonly known as ‘bugs’ in the software. Thus, Software Testing plays very important role before releasing any software. Software testing is the process of thoroughly verifying the desired outputs for all the estimated input parameters that are defined by the software. For intended ECM (Engine Control Module) various inputs/outputs drivers like analog input, discrete input, frequency input, pulse width modulation input, etc. are tested working if they are as per core software defined. Before starting any test, we need to consider supply voltage, type of input, and current range, tolerance, etc. Manual testing is time consuming, monotonous, less reliable and non-programmable also it requires huge investment of human resource. So we switch to automated testing which is fast, most reliable, programmable in less intervention of human. Human errors are avoided in automated testing. For automated testing, we used software called LabVIEW, NI TestStand of NI (National Instrument). In which we have written a sequence which calls API’s (Application Program Interface) using windows script host (WSH). We generate test inputs using NI cards of Test bench and observe output using LabVIEW VI (Virtual Instrument File) or read value using WSH. Test bench used is NI PXI-1045 General purpose 18 slot. Our goal is to reduce human efforts and errors in software testing using Test Automation.

**Keywords:** Automated Software Testing, NI TestStand, LabVIEW.

---

## I. INTRODUCTION

Based on test execution, Software testing is classified into two categories.

1. Manual Software Testing
2. Automated Software testing

### 1. Manual Software Testing:

Manual testing is the process of detecting software defects manually. Tester plays the role of an end user and uses most of all features of the application to make sure correct behaviour and to ensure completeness of testing, the tester often follows a written test procedure that leads them through a set of important test cases.

There are two stages of software testing: Regression testing and parametric testing. In regression testing, we are testing software against too many set points and in parametric testing we are testing software at tri voltage and tri temperature with limited set points.

Regression test is only for the special occasion that something in the application/code has changed. It can be design, code or anything at all that dictates the overall framework of the system. A retest that is conducted in this situation to make sure that modifications has not made an impact on anything that was already working before is called Regression Test. Regression testing can be done manually.

## 2. Automated Software Testing (AST):

As time goes on applications get piled on with more and more functionality which keeps increasing the scope of regression testing, the development time does not necessarily grow with releases, but the testing time does. We cannot reduce the time it takes to test by increasing the test team size more money and training is required. So regression testing is most often automated. [11]

We can categorize the tests and tools as [9]

1. Developer oriented tools
2. Functional testing tool
3. Load testing tools
4. Performance monitoring and maintenance tools

As we are performing functional testing; WinRunner, Astra QuickTest, SilkTest, RationalSuite [11] and NI TestStand are some of the test tools. We used NI TestStand, in which we designed a sequence. This test suite is tested on Test bench NI PXI-1045 General purpose 18 slot chassis for PXI holding cards like Analog, Digital, FIU (Fault Insertion Unit) and Timer-counter. The vision of automated testing aims at 100% automation [3].

## II. LITERATURE SURVEY

The goal of software testing is to find bugs as early as possible before they proliferate and to fix them prior to the software's release.

According to M. Jain and D. Gopalani [4], Types of testing are Aging Testing, Agile Testing, Alpha Testing, Application programming Interface Testing and Automated Testing.

Benefits and Limitations of Automated Software Testing are explained in detail in survey paper in [3]. S. Mohanty, S. Sengupta and S K Mohapatra also explained significance and advantages of Automation for verification of software testing in their paper. [5]

A test bench or testing workbench is an (often virtual) environment used to verify the correctness or soundness of a design or model. Test benches are of different types Stimulus only, Full test bench, Hybrid test bench and Fast test.

Paper [6] performs Automation Testing using Software Testing Tool Selenium.

Automated Test Summit, 2005, an Industry Event hosted by National Instrument summarized benefits of Teststand as focusing on developing a modular test software framework for test software architecture you are able to balance your technical requirements with your business requirements. Test software framework such as TestStand can greatly reduce your development time as the framework already provides the modularity and the structures and features required to build a complete test system. [1]

Use of Automated Test Bench for an Induction Motor using Lab VIEW is presented in [1].

Automatic testing of multimedia units using programming language LabVIEW and Teststand software tool from National Instruments is explained in [7]

Information about TestStand environment, benefits of using TestStand as Automation tool is explained in.

## III. BLOCK DIAGRAM

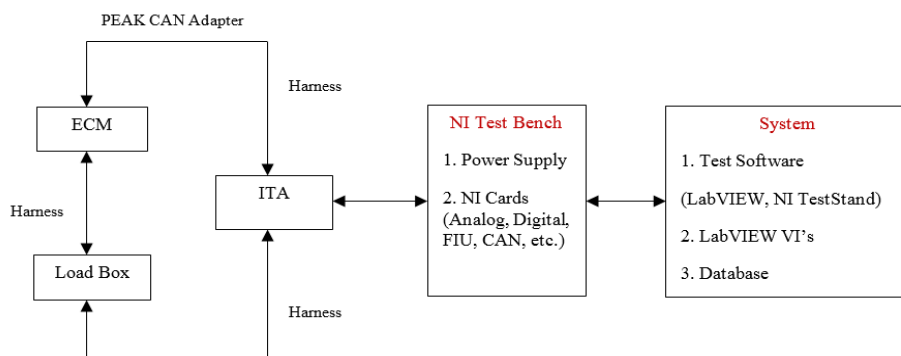


Fig.1. Block Diagram Automated Testing

Where ITC: Interconnect Test Adaptor

ECM is connected to Load box and NI Test bench using harness via ITA. NI Bench includes Power supply pin, slots for NI cards which can be analog, digital, CAN, FIU, etc. To observe inputs and outputs, we use test software like LabVIEW, CALTERM III. Peak CAN is used to establish communication between ECM and Test bench.

Test bench has OS windows 7 and NI-PXIE-8115 Embedded Controller. Software installed are LabVIEW 2013, WSH and NI TestStand 2013. Hardware used are Testing Module/ ECM, Harness for connection, PCAN adapter for communication, Manual loadbox for user interface, Breakout box and importantly NI Test bench with NI cards.

## IV. SOFTWARE

### A. LABVIEW:

LabVIEW stands for Laboratory Virtual Instrument Engineering Workbench. It is a graphical programming environment where one can use sophisticated measurement, test and control systems. LabVIEW is a platform-based method for measurement and control that connects with a wide variety of hardware. We could use it across different hardware targets and operating systems. It provides built-in-analysis libraries to help extract useful information from raw data. LabVIEW's appearance and operation imitate physical instruments as oscilloscope and multimeter. VI consist of main 3 parts: front panel, block diagram and icon/connector pane.

### B. NI TESTSTAND:

NI TestStand is a ready-to-run test management software dedicated to develop automated test and validation systems faster. NI TestStand can be used to develop, execute, and deploy test system software. In addition, we can develop test sequences that integrate code modules written in any test programming language. Finally, we can deploy test systems to production with easy-to-use operator interfaces. The redesigned Call Executable step type provides a more powerful interface for calling common scripting languages such as Python, Perl and Tcl. [10]

Teststand is compatible with LabVIEW, C/C++ DLL, .NET, ActiveX/COM, LabWindows / CVI, HTBasic adapters. Sequence editor screen provide us insertion pallet Sequence file editor screen, list of main sequence with subsequence and Variables window.

Before starting a sequence we can select adaptor. In our case it is LabVIEW. In Main sequence, we can add different types of steps like insert a LabVIEW VI, Label, write a function, message pop-up for accepting input from user and call a subsequence. We can add delay as well as flow controls/loops like For loop, Switch case, If-Else, Do-while, Break, etc. This feature provide more control over Sequence execution.

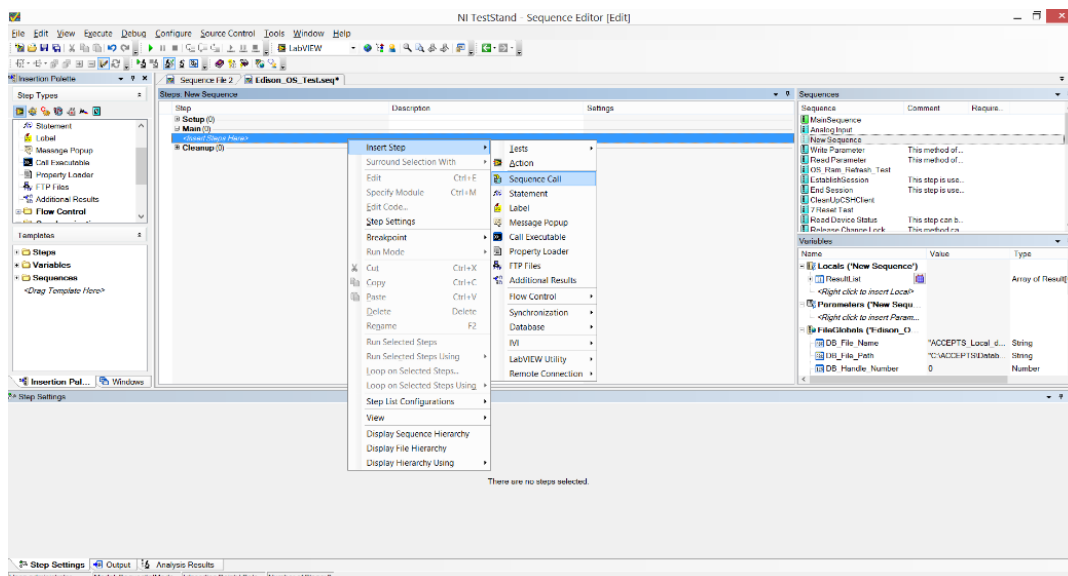


Fig. 2. Step selection in NI TestStand

Like any other programming tool we define variables before using it. There is hierarchy of defining variables as Locals, Parameters, FileGlobals and StationGlobals. In main sequence, we can write pre-expression as well as post-expression before and after execution of sub-sequence. This helps to execute a sequence.

## V. HARDWARE

### A. ECM (ENGINE CONTROL MODULE):

Electronic Control Unit (ECU) is a generalised term for any embedded system that controls one or more electrical system or subsystems in a vehicle. ECM is a type of ECU. It is designed as per client's requirements. Recent ECUs use a microprocessor which processes the inputs from the engine sensors in real-time. An electronic control unit contains the hardware and software (firmware). The hardware comprises of electronic components mounted on a printed circuit board (PCB), ceramic substrate or a thin laminate substrate. The main component on this circuit board is a micro controller chip (CPU). The software is kept in the microcontroller or other chips on the PCB, usually in EPROMs or flash memory so the CPU can be re-programmed by uploading updated code/calibration file or replacing chips.

### B. PCAN (PEAK CAN APPLICATIONS) ADAPTOR:

PCAN hardware allow us to communicate with ECM using CAN (Controller Area Network) protocol. CAN is a vehicle bus standard which is intended to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, we used CAN based protocol J1939 for communicating with ECM, which is used in heavy-duty vehicles such as trucks and buses, mobile hydraulics, etc.

### C. NI TEST BENCH:

NI's PXI-1045 chassis has 18-slot chassis, which is designed for a wide range of test and measurement applications. Due to its large slot capacity and modular structural design, It is well suited for high-channel-count, multi-instrument benchtop or rack-mounted systems. The combination of accurate timing features and incorporated trigger buses provides the best platform for multiple-instrument synchronization and automated testing.

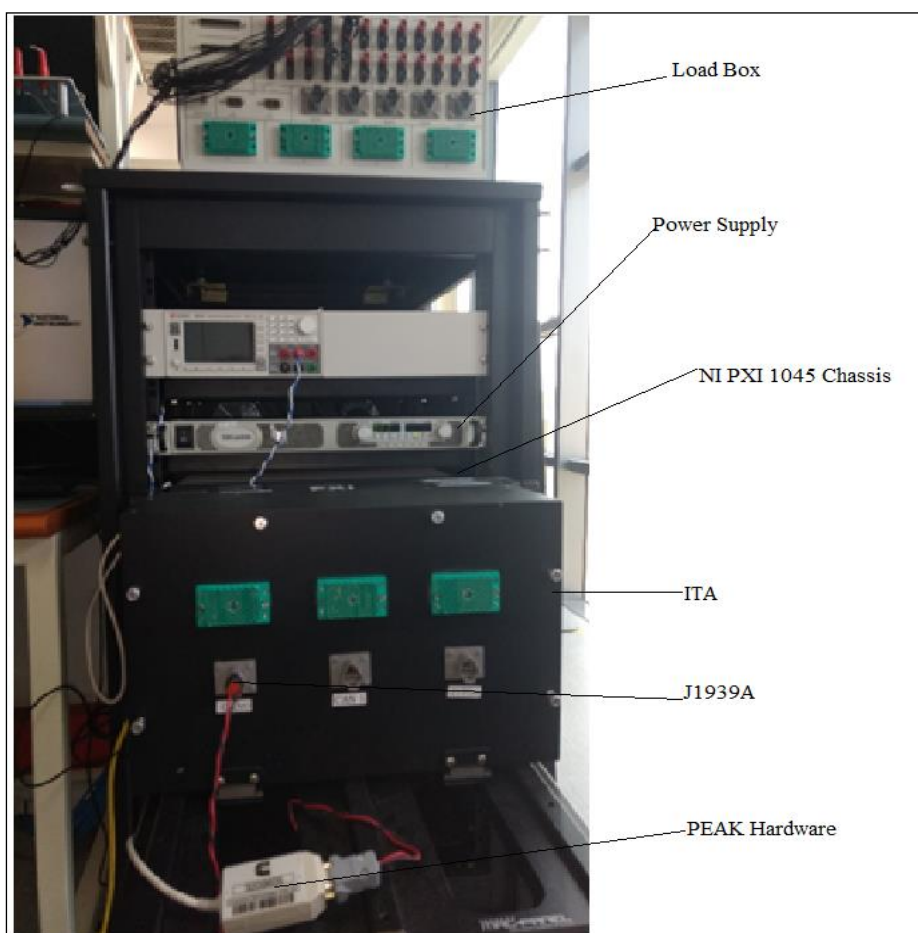


Fig. 3 NI Test Bench

We can see cards inserted into chassis, Network devices connected, devices connected to COM port, USB ports in application NI MAX.

## VI. METHODOLOGY

Our design goal is to design a test suite using LabVIEW, NI TestStand and embedded c for OS base software. Which includes following tasks:

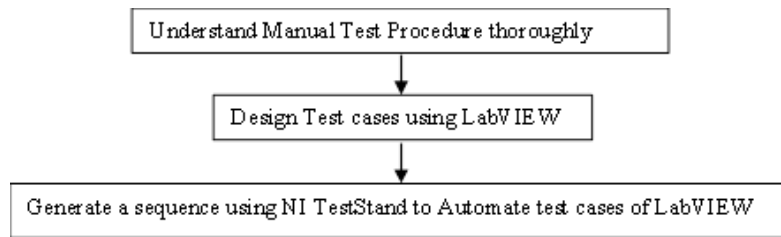


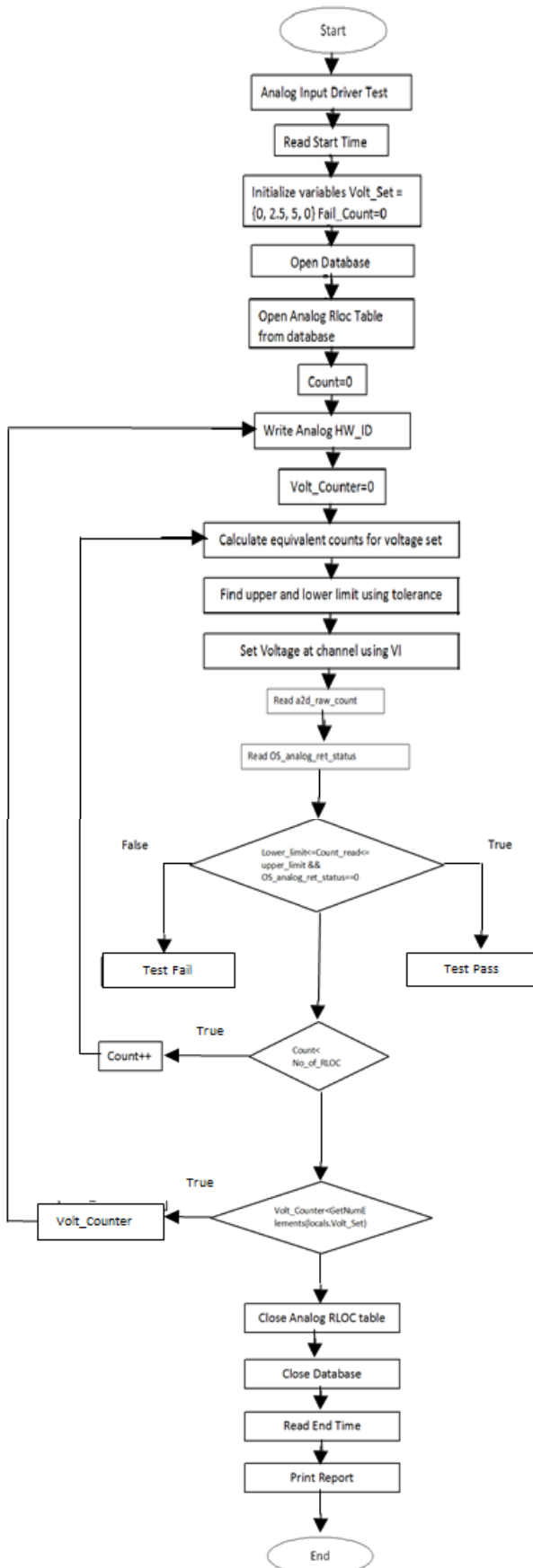
Fig. 4. Design Methodology

### A. PARAMETERS UNDER TEST:

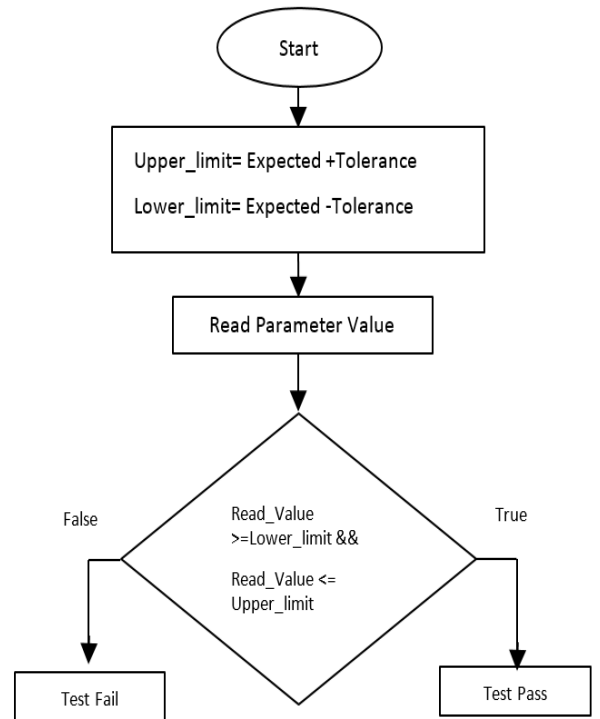
TABLE I: PARAMETERS AND TEST PROCEDURE

SR. NO	TEST NAME	Test Procedure	SR. NO	TEST NAME	Test Procedure
1.	Basic Bench Test	Monitor parameters as ECM Run Time, ECM Status, Reset Count, Power up and Down Count, Calibration version number. If this test fails no other test is going to perform.	2.	PWM Diagnostics	Create a fault (STB, STG, Open connection) and check return status. Remove fault after each test.
3.	Timer Test	The purpose of this test is to verify the functionality of the Timers using the OS Time	4.	OS Shutdown Test	Check different types of Shutdown and its return status
5.	Analog Input	Feed input from range 0 to 5 and read respective raw count from 0 to 1023 (10 bit binary)	6.	OS Reset Test	Check different types of Reset and its return status, Reason for reset and Reset count
7.	Discrete Input	Set pin high or low, Check if pin toggles for respective RLOC(Resource locator)	8.	External and Internal Watchdog Test	Ensure that the internal and external watchdog times out as expected
9.	Discrete Output	Set pin 0 or 1, Check if its feedback pin toggles	10.	Over speed Test	Check Over speed detection functionality
11.	Frequency Test	Set different frequencies and read period count for different mode of operation.	12.	PWM input	Set period, Duty cycle, mode and run. Read if intended duty and period is set using LabVIEW VI.
13.	Multiple calibration download	Check if multiple calibration files can be downloaded on ECM			

**B. FLOW CHARTS:**



**Fig. 5. Test Procedure**



**Fig. 6. Criteria to Pass/ Fail Test**

### **C. REPORT GENERATION AND VALIDATION:**

TestStand is capable of reporting results in ASCII text, HTML, XML and even in the new ATML standard reporting. This reporting is completely customizable and uses a plan to create a report using the test results obtained in TestStand. This particular report is in HTML. We can see report format selection and other settings in TestStand report option window. Our report contains Station ID, Date, Time, Overall test Result at the heading of report file. If there is failure in any sub sequence, it is reported in header. For individual test, we can add/change content to be displayed in report.

## **VII. TECHNICAL SPECIFICATION**

### **A. ADVANTAGES OF TEST AUTOMATION:**

1. Improved product quality: Fewer defects present in the software product. [2]
2. Test coverage: High coverage of code (e.g. statement, path or branch) is achieved through automation.
3. Reduced testing time: Time required for testing, so it runs more tests within a timestamp.
4. Reliability: When repeating tests as variance in outcomes, AST is more reliable. It can be due to the manual tester is running tests in a different way, but he cannot make use of the knowledge of the tester.
5. Increase in confidence: Confidence in the quality of the system is rise.
6. Reusability of tests: When tests are designed with maintenance in mind they can be repeated frequently, a high degree of repetition of test cases leads to benefits, not a single execution of an automated test case.
7. Less human effort: Automation reduces human effort that can be used for other activities (in particular ones that lead to defect prevention).
8. Reduction in cost: With a high degree of automation cost are saved.
9. Increased fault detection: Effectiveness as the ability to detect a large portion of defects in a system due to high fault detection ability.

### **B. DISADVANTAGES OF TEST AUTOMATION**

1. Automation cannot replace manual testing completely [2]
2. Process of test automation needs time to mature

## **VIII. CONCLUSION**

Software testing plays very important role in software design. Whenever there is any change in software, it is necessary to test complete software or the part which has dependency on modified code. As name explains itself, it is a regression testing, where we test rigorously to find bug in software.

Manual testing doesn't require any tool to test software. But it has some drawbacks as less accuracy, large testing time, less test coverage. It is not even reusable. To overcome these drawbacks, we switch to automated testing. Automated testing is more reliable, accurate, less time consuming, reusable and having large test coverage. For this research paper we have designed a test suite which is combination of different software's like LabVIEW, WSH and NI TestStand. We can test our specific ECM and verify the correctness or soundness of OS software.

In this paper we have presented a way for Automating Software tests. Although 100 percent automation is not possible, this method will reduce tremendous testing time as well as human efforts. It is observed that, tests which would take half an hour by manual testing, they are tested in 3 minutes using this automated testing. Comparative to other automation tools LabVIEW is easier to design test case as compared to other programming languages. It allows to interface variety of hardware. We have developed test sequences in NI TestStand that integrates code modules written in LabVIEW programming language.

In future if we generalize this sequence, we can use same sequence for OS testing of any ECM.

### ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Professor P. B. Borole for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this chance to express my gratitude to Mr. Harish Tilak, Senior Electronic Engineer (TF, Embedded Software), Cummins Pune, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I am grateful to colleagues of Cummins, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Lastly, I thank my parents, brother, sisters and friends for their constant encouragement without which this assignment would not be possible.

### REFERENCES

- [1] K. B. Chavhan and R.T. Ugale, "Automated Test Bench for an Induction Motor using Lab VIEW", 1st IEEE International Conference on Power Electronics. Intelligent Control and Energy Systems (ICPEICES-2016).
- [2] Rafi D.M, Moses K.R.K, Petersen Kai and Mantyla M.V, "Benefits and limitations of automated software testing: Systematic literature review and practitioner survey", 2012 7th International Workshop on Automation of Software Test, AST 2012- Proceedings, art.no.6228988, pp.36-42.
- [3] A. Bertolino, "Software testing research: Achievements, challenges, dreams," in Proceedings of the Workshop on the Future of Software Engineering (FOSE 2007), pp. 85-03.
- [4] M. Jain and D. Gopalani, "Aspect Oriented Programming and types of software Testing", second conference on Computational Intelligence and Communication Technology, 2016.
- [5] S. K. Mohanty, S. Sengupta, S K Mohapatra. "Test Bench Automation to overcome Verification Challenge of SOC Interconnect", International Conference on Man and Machine Interfacing (MAMI), 2015.
- [6] Vishawjyoti and Sachin Sharma, "Study and Analysis of Automation Testing Techniques", Journal of Global Research in Computer Science, Volume 3, No. 12. December 2012.
- [7] National Instrument. Automated Test Summit, 2005, an Industry Event hosted by National Instrument. Rendl, Petr Sehnal, Frantisek Steiner, "Automatic Testing of Multimedia Units", Published in Electronics Technology (ISSE), 35th International Spring Seminar, INSPEC Accession Number 12945432, 2012.
- [8] I. Gronau, A. Hartman, A. Kirshin, K. Nagin, S. Olvovsky, "A Methodology and Architecture for Automated Software Testing", Technical report, in IBM Research Laboratory in Haifa.
- [9] P. Rathi, V. Mehra, "Analysis of Automation and Manual Testing Using Software Testing Tool", International Journal of Innovations & Advancement in Computer Science, IJIACS, ISSN 2347 – 8616, Volume 4. March 2015.
- [10] [www.peak-system.com/](http://www.peak-system.com/)
- [11] <http://www.softwaretestinghelp.com/>