

ANoalMethod of Test Automation for Testing Embedded SoftwareAjay Roy¹, Srinivas Perala² and Koushik Barman³*School of Electronic and Electrical Engineering**Lovely Professional University, Punjab**Email:ajoy.22652@lpu.co.in¹, srinivaslabview@gmail.com², koushik.15737@lpu.co.in³****Abstract-***

The focus of this paper is about the method of test automation for testing embedded software using NI's Test Stand and LabVIEW. Software testing is a process to check the functionality of the software application to know whether it is developed properly to meet the specified requirements or not. It is an important phase in the software development life cycle because any product before being released into the market needs to be defect free. To produce the quality product and meet costumers needs it must undergo this process. For the testing of software, tester generates several different test cases based upon the requirement. software requirement specifications include both functional and non-functional requirements of the software. In this paper, we describe the current method of manual test process and same how to automate this manual testing process using NI's Test stand and LabVIEW.

Keywords-LabVIEW, Test Stand, Validation

I. INTRODUCTION:

Embedded systems are widely used in many fields nowadays like defense, automotive, aeronautics and in different sectors of industries. Embedded software is a software system which is essential for the system. This permanently resides in the device; whose operations it is controlling. Embedded software developers require embedded software tools to develop the software. And it is difficult to handle the complexity of the code, increasing size, increasing time, increasing requirements. so, there is a huge difficulty in implementing the reliability testing for the embedded software. Verification is performed through testing on the software to check whether the software is of high quality or not[1]. Verification can be defined as a static practice of verifying every type of functionalities like documentation, design analysis, code, program, and inspection. It is just like an objective process. Verification gives us a good detailing about its quality and its concern towards the requirements of the market, whether the system is well-engineered and error-free, but it will not ensure if it is useful or not. The evaluation process during the development or after development determines whether it can meet the specified product requirements. Validation is defined to be a process of evaluating the final product whether it reached the expectations and conditions which were set initially. it is a dynamic approach of testing the actual product. Test cases are essential for ensuring quality and code behavior as specifies. But, writing test cases don't have only the benefits they also have some disadvantages, as the test cases which we make can be in large numbers and if it is done through manual testing process time, money and people requirement will be high. In a situation like startups or small enterprises, cost becomes their major issue, they won't prefer testing instead they go for the development of new functionalities. Automating manual testing process is a

solution for this. Manual testing helps to thoroughly go through the test cases, which requires a lot of time and a large number of people. In manual testing test cases are executed by a human tester and software, whereas in automated testing human interference will be less. Automated testing is a reliable method and is significantly faster than the manual approach. Similar kind of works has been found in various literature. On-Board Diagnostics Counters for an Automotive On-Road Application has been mentioned in [8]. An alternative to traditional software rehosting has been mentioned in [7]. Implement TPS with ATML and NI TestStand mentioned in [6].

II. TEST STAND and LabView

The test stand is a test management software from National Instruments. It is designed to quickly develop validation systems and automated test. One of the important features of the test stand is that it provides a consistent feel and look to the developers. Any language can be used to develop the software using the test stand. The main purpose of using this software is that it reduces the implementation time and requirement of people is also less this helps in testing and verifying software as fast as possible than the manual method. As per the demands of market and requirements if we can produce a high-quality product in less amount of time with all the required specifications this would be a great advantage for increasing the profits. The Test Stand Engine is a facility used to characterize, develop and test engines. It uses module adapters to invoke code modules. A code module is like a program module from an ADE or programming language, which contains one or more functions that perform a specific test or some other actions which are specified. Module adapters call code modules, pass inputs; outputs and required parameters to code modules, the status of the code module and values obtained. It handles the core test management functionality, includes the management, generation, database logging, definition and execution of the overall testing process and more.

Test Stand supports the following module adapters:

. ActiveX/COM Adapter

C/C++ DLL Adapter

HTBasic Adapter

LabVIEW Adapter

Lab Windows/CVI Adapter

.NET Adapter

Sequence Adapter

Labview helps us to visualize every aspect of your application, it is just like a graphical programming approach. It includes hardware configuration, measuring data, and debugging. This makes visualization simple to represent complex logic on the diagram, develop data analysis algorithms. As Test stand supports different module adapters, LabVIEW has been used.

III. Research Methodology for the Proposed Technique

In this paper, Dishwasher embedded software considered for explaining about Testing automation using Test stand and LabVIEW. Let's have a small introduction to the Dishwasher has been given in this section. Dishwashers require fine-grained control over water pumps, complex system control over valves, and so on. Generally, Dishwasher has One main microcontroller which takes inputs from User Interface panel which controlled by another microcontroller which called UI microcontroller. Embedded software is written for these two microcontrollers. As per User Interface panel instructions, UI microcontroller sends Message to the Main controller through any one of standard protocol [2,3,4]. Commonly used Protocol is Ethernet. It depends on the Manufacture Requirement design. The Main controller understands UI message and sends signals to different sensors and actuators according to message functionality. Manual Testing Method: Validation testing method starts with test case writing from Software requirement document. It is required manual test board emulator to operate different functions. Because we cannot test directly on the product. So, per functions requirements, a testing board will be designed. Here in this example, Dishwasher manual test board connected with User Interface panel and Main Microcontroller.

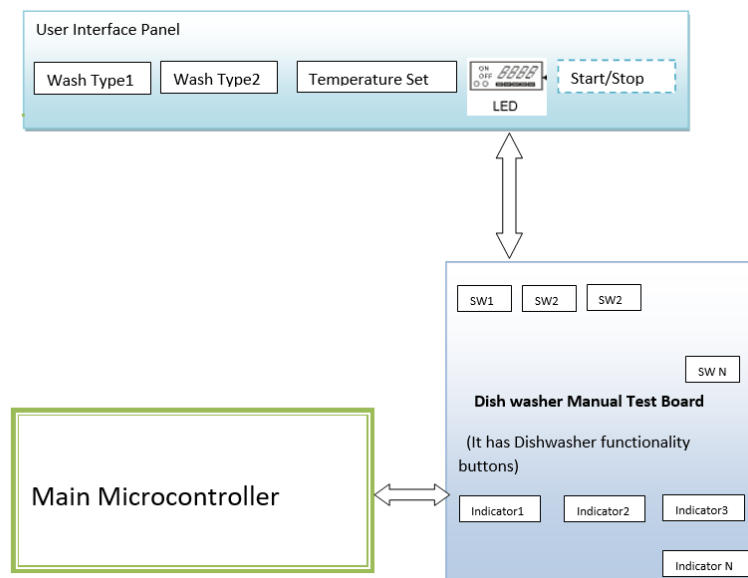


Fig.1. Block diagram of Manual test board

Manual test process for the sample test case:

Pre-Conditions: Flash Dishwasher embedded software which we need to test into a Main microcontroller and UI microcontroller.

Step1: Turnon Power; the tester must turn on/off button at Manual test board

Step2:Select wash type1 cycle and start; Tester can select washtype1 at UI panel to start the washtype1 cycle.

Step3 Check Output at LED panel as 1hr time display; Tester must check the display and if is 1 hr. time display test is passed otherwise test failed.

In this way, the validation engineer writes typically more than 300 test cases to test all functionality of the software.

Each test case takes a minimum of 20 to 30 minutes depending on no of steps in that test case. To complete all test cases, it takes 3 to 4 months with 4 to 5 test engineers.

And if we want repeat testing, the same time it will take as it is required test engineer manual support.

Most of the test cases can be scripted using NI Test stand and the scripted test case is called sequence.

Automation Testing Method:

Manual test board needs modifications to replace manual interaction with software interaction. An empirical software testing technique as been mentioned in [5]. NI DAQ hardware has all kind of signals acquisition and generation capacities. LabVIEW programming can trigger NI DAQ hardware to automate manual actions. LabVIEW Programming used to talk with Manual test board through NI DAQ hardware. And NI test stand sequences uses these LabVIEW programming Vis to automate steps in test cases.

Automation testing board has add-on NI DAQ hardware to Manual test board as shown in Fig.2.

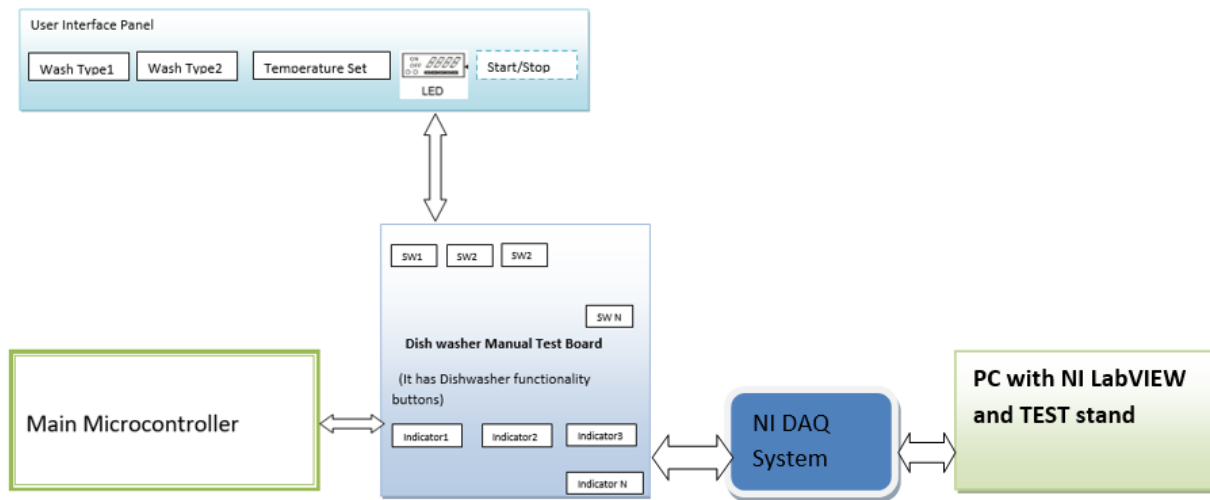


Fig. 2. Block diagram of Automation Testing Board

Test Stand is not an IDE like LabVIEW. Instead, Test Stand provides a generic environment for developing test sequences that already includes many commonly useful default behaviors, such as execution flow control, error handling, report generation, etc., while allowing the user to override or extend this default behavior to meet application. As per the sample test case, each step must be scripted in the test stand as shown below.

Pre-conditions steps are scripted in setup block, Test steps are scripted in the main block and postcondition steps scripted in the cleanup.

Test stand sample sequence for the sample test case shown in Fig.3.

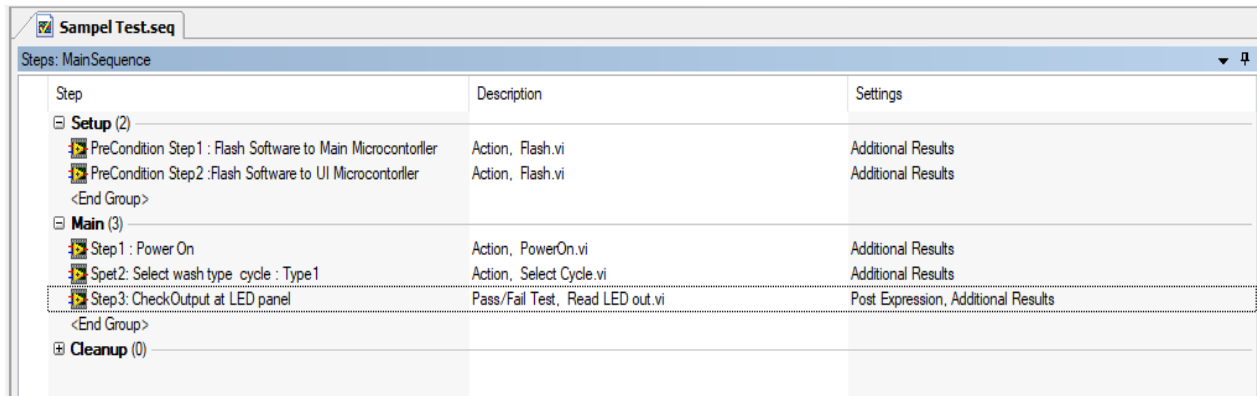


Fig. 3. Sample Test Sequence.

Once we run this test sequence we get the report as shown in figure 4. And this report has a clear explanation of each step.

Step	Status
PreCondition Step1 : Flash Software to Main Microcontorller	Done
TestResults/Data	
Flashed MC Version 1.0.1:	Flashed MC Version 1.0.1
PreCondition Step2 :Flash Software to UI Microcontorller	Done
TestResults/Data	
Flashed UI version 1.0.1:	Flashed UI version 1.0.1
Step1 : Power On	Done
TestResults/Data	
Step1:	Power Turn On
Spet2: Select wash type cycle : Type1	Done
TestResults/Data	
Step2:	wash type1 cycle Selected
Step3: CheckOutput at LED panel	Passed
TestResults/Data	
Step3:	Read Time Display is 1 hr.

End Sequence: MainSequence

Fig. 4. Sample Automation report

Once the new software is released, we can execute and take reports without any efforts. Usually, it takes 3 to 4months to do a manual test of 300 test case. Once we develop test sequences for test cases, we can reduce time to 10 – 14 days from 3 months.

Conclusion:

It is not possible to make 100% test automation as some of the steps are not possible to replace with software triggering. In such cases, hardware modification required to make automation. But it is depending on a cost-effective solution. In such cases, we can go for 90% automation and 10% can depend on manual testing. And scripting test cases also a big challenge. In the future, I am going to focus on automating the automation. It means the system can generate test scripts/ sequences by reading a test case using Machine learning and AI.

References:

- [1] S. Yalamanchili and K. S. Kumari, "Comparison of manual and automatic testing using genetic algorithm for information handling system," proceeding of 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), Paralakhemundi, Jan. 2016.
- [2] M. Wahler, E. Ferranti, R. Steiger, R. Jain and K. Nagy, "CAST: Automating Software Tests for Embedded Systems," proceeding of IEEE Fifth International Conference on Software Testing, Verification and Validation, Montreal, QC, Feb. 2012.
- [3] E. P. Enoiu, A. Cauevic, D. Sundmark and P. Pettersson, "A Controlled Experiment in Testing of Safety-Critical Embedded Software," proceeding of IEEE International Conference on Software Testing, Verification and Validation (ICST), Chicago, IL, Dec. 2016.
- [4] M. Bajer, M. Szlagor and M. Wrzesniak, "Embedded software testing in research environment. A practical guide for non-experts," Proceeding of 4th Mediterranean Conference on Embedded Computing (MECO), Budva, Apr. 2015.
- [5] N. Shete and A. Jadhav, "An empirical study of test cases in software testing," proceeding of International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, Apr. 2014.
- [6] Fan Shuyi, Mengchen and Jiang huixia, "Implement TPS with ATML and NI TestStand," 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, Apr. 2010.
- [7] K. Ham, "An alternative to traditional software rehosting," in IEEE Aero. El. Sys. Mag., vol. 20, Apr. 2005.
- [8] P. Kadekar and A. Wakankar, "Automated Test Environment for On-Board Diagnostics Counters for an Automotive On-Road Application," proceeding of 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, Apr. 2018.