International Journal of Engineering & Technology, 7 (2.7) (2018) 303-307

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper



Comprehending testing of distributed embedded systems

Chaitanya Kilaru [1] *, Dr. K RajaSekhara Rao [2]

[1] Research scholar, Department of CSE, JNTU Hyderabad, Telangana
[2] Director, UshaRama College of Engineering and Technology, Telaprolu, Krishna District, Andhra Pradesh
**Corresponding author E-mail: kilaru.chaitanya84@gmail.com*

Abstract

It has been stated in the literature that a test case that needs to be used for testing a distributed embedded system be decomposed into elementary level test cases that can be tested at different computing locations of the system as conducting testing considering entire distributed system is complex as the reliability of testing depends on the environment setting made at each of the location and also the complexity of the communication interface established at each of the distributed location. There should be strategy for conducting testing at each of the locations and comprehend the results to provide wholesome working position of entire distributed embedded systems.
In this paper, a strategy has been presented that explain how the testing can be undertaken at individual locations using different test methods and how the results are joined such that the final test results reflects the state of testing a distributed embedded system
.

Keywords: *Testing Distributed Embedded Systems; Integrating Test Results; Comprehending Test Re.*

1. Introduction

Testing distributed embedded systems is complex for various reasons that include the need for establishing a test environment at each of the location, combined working of test environments, Total working of entire distributed embedded system, and complexity of communication interface through which communication between the distributed embedded systems can be undertaken. Testing is complex as there is heterogeneity from the point of hardware, software at each of the computing location. It is thus proved that testing distributed embedded system considering the total system as whole is complex and sometimes not feasible.
Testing of an embedded system involves testing hardware, hardware independent code, and hardware dependent code. Each segment of testing requires use of the embedded hardware, embedded software and the HOST at which the firmware is developed and cross compiled. Each segment can be tested using one or more standard methods such as the methods scaffolding, assert macros and instruction set simulators are used for testing Hardware independent code involving only HOST, Logic analysers are used for testing hardware in evolving only hardware, in-circuit emulators are interfaced to the TARGET are local and the entire test case is administered considering entire system as a whole. However, when it comes to testing of distributed system, a test case must be administered considering entire distributed embedded systems which are complex and sometimes not feasible due to existence of many embedded system that are networked through a protocol.
Thus, there should be strategy or model using which testing is carried at individual locations and combining / merging process is used to generate test results that reflecting the testing of entire distributed embedded system as a whole.

2. Related work

In literature methods have been proposed for testing distributed system considering a set of test cases that require same testing method. Chaitanya et al. [1] have presented the way testing of distributed embedded system can be carried using scaffolding method particularly when the distributed network is established through RS485 based network. They have shown the way testing must be performed at the HOST considering hardware independent code of the embedded systems. They have also shown the way the test results obtained at different location after holding testing using scaffolding could be merged to get overall status of testing entire distributed embedded system using scaffolding method.
Chaitanya et al. [2] have presented the way testing of distributed embedded system can be carried using assert macros particularly when the distributed network is established through USB based network. They have shown the way testing must be performed at the HOST considering hardware independent code of the embedded systems. Assert Macros based testing is carried to find whether the required environment exists for carrying further processing. They have also shown the way the test results obtained at different location after holding testing using Assert Macros could be merged to get overall status of testing entire distributed embedded system using assert macros.
Chaitanya et al. [3] have presented the way testing of distributed embedded system can be carried using instruction set simulators particularly when the distributed network is established through CAN based network. They have shown the way testing must be performed at the HOST considering hardware independent code of the embedded systems. Instruction set simulators allows testing to be carried simulating as if the code is being executed on the target. They have also shown the way the test results obtained at different location after holding testing using instruction set simulators could

Copyright © 2018 Chaitanya Kilaru, Dr. K RajaSekhara Rao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

be merged to get overall status of testing entire distributed embedded system suing CAN based networking.
Chaitanya et al. [4] have presented the way testing of distributed embedded system can be carried using in-circuit emulators particularly when the distributed network is established through I^2C based network. They have shown the way testing must be performed considering both TARGET and the HOST considering both hardware independent and dependent code of the embedded systems. They have also shown the way the test results obtained at different location after holding testing using in-circuit emulators could be merged to get overall status of testing entire distributed embedded system suing I^2C based networking.
Chaitanya et al. [5] have presented the way testing of distributed embedded system can be carried using Logic Analysers particularly when the distributed network is established through I^2C based network. They have shown the way testing must be performed considering Hardware at the TARGET. They have also shown the way the test results obtained at different location after holding testing using Logic Analysers could be merged to get overall status of testing entire distributed embedded system using I^2C based networking.
David E. Simon [6] has detailed in his book various testing debugging techniques which in way have become the industry standard testing techniques for testing standalone embedded systems.
Justyna et al., [7] have presented a method for testing an embedded system through projection of the same as a model and enumerating different paths existing in the model. Each path is considered to be a test case. Bruno [8] in his text book has explained several tools using which comprehensive testing of stand-alone embedded systems can be carried.

3. The strategy

A test case that must be tested using entire distributed embedded system can be decomposed into a set of elementary test cases such that each elementary test cases can be tested at a specific location using a specific method. The test cases that must be tested at a location can be carried using testing methods such as scaffolding, assert macros, instruction set simulators, in-circuit emulators and Logic analysers. While some test cases help in testing hardware, some help in testing hardware independent code and some other methods can be used for testing hardware independent code.
The test results produced through undertaking testing using elementary test cases are to be merged and an overall test status that represent testing considering the entire distributed embedded system as whole must be arrived in order to assess the overall functioning of the entire embedded system. The merging of the test results must be done considering a master test case irrespective of location where testing is carried and the method used for undertaking testing

4. The pilot project

A pilot project has been developed which is a distributed embedded system which is meant for monitoring temperatures within nuclear reactors and also controlling the same through flow of coolants into the nuclear reactors. The distributed embedded system is developed using 5 Microprocessors. While two microcontrollers are used for sensing two temperatures in each of the reactor tube, two Microcontrollers are used for controlling the pumps that are used for flushing the coolants into nuclear reactors. One Micro controller is used to for coordinating with other micro controller and interfacing with the user for updating h the users the status of the process happening within the nuclear reactor.

5. Overall test cases and breaking and combining elementary test cases

The functional requirements of the pilot project, the related test cases, the elementary decomposed test cases, the input-output variables related to test cases and the test data to be used for undertaking test cases are shown in Table 1. Only 5 functional requirements have been shown out of 20 functional requirements which are specified for original project.
The test results obtained at different locations are e combined and grouped considering the test cases that belong to the same master test case. The grouping will bring together all the test results related to the same test case. Table 1 shows the grouping of test results that have been obtained at different locations and testing Methods.

Table 1: Decomposing Test Cases and Combining Test Results

Functional requirement Number	Functional Requirement	Test Requirements	Test Case serial	Sub-Test case serial	Split test cases	Input Variable-1	Input Variable Value 1	Input Variable-2	Input Variable Value 2	Test Output Variable	Test Output Values	Expected output	Test Pass/fail	Location	Test Type
1	Read Temp-1 and write to LCD	Test Temp-1 read is written to LCD	1	1	Test Temp-1 read is written to LCD	TEMP-1	32			LCD W	32	32	P	ES-1	Scaffolding

		Test for proper sensing of Temp-1 signal at the output of the temperature sensor	2	2	Test for proper sensing of Temp-1 signal at the output of the temperature sensor						TRUE	TRUE	P	ES-1	Assert Macro
2	Test the Communication based communication between the 89C51 (ES-1) and the central ES-5)	Test for the equivalence of output data sent (output Register) through ES-1 resident communication interface and received at the communication port of ES-5 (Comm. Input register)	3	3A	Test for proper outputting data on output communication port of ES-1	COMM-REG1	32	COMM-REG2	21	OUT-STA	Y	Y	P	ES-1	Scaffolding
				3B	Test for proper data read at the input communication on output communication port of ES-5	COMM-REG1	32	COMM-REG2	21	OUT-STA	Y	Y	P	ES-5	Scaffolding

Functional requirement Number	Functional Requirement	Test Requirements	Test Case serial	Sub-Test case serial	Split test cases	Input Variable-1	Input Variable Value 1	Input Variable-2	Input Variable Value 2	Test Output Variable	Test Output Values	Expected output	Test Pass/fail	Location	Test Type
3	Read-Temp-1 and send to Central Micro Controller	Test for the equivalence of output data sent through ES-1	4	4A	Test for Reading of a particular Temparature-1 @ES-1	COM M-REG1	32	COM M-REG2	21	OUT-STA	Y	Y	P	ES-1	Scaffolding

		resident communication interface and received at the communication port of ES-5	4	4B	Test for Reading of a particular Temparature-1 @ ES-5	COMM-REG1	32	COMM-REG2	21	OUT-STA	Y	Y	P	ES-5	Scaffolding
4	Read Temp-1 and measure through-put	Test for number of times temperature is read in one minute	5	5	Test for throughput	THRU-T1-TIME	10			TEMP1-THRU	10	10	P	ES-1	Instruction Set Simulator
			6	6	Test for through put					TEMP1-THRU	12	12	P	ES-1	Scaffolding
5	Test the Communica-tion based communica-tion between the 89C51 (ES-3) and the Central Micro Controller (ES-5)	Test for the equivalence of output data sent (output Register) through ES-3		7A	Test for proper outputting data on output communication port of ES-3	PUMP1-STA	ON			PUMP1-STA	ON	ON	P	ES-3	Scaffolding
		resident communica-tion interface and received at the communica-tion port of ES-5 (Comm. Input register)	7	7B	Test for proper data read at the input communication on output communication port of ES-5	PUMP1-STA	ON			PUMP1-STA	ON	ON	P	ES-5	Scaffolding

6. Merging test results

The test results produced due to testing a group of test cases that relate to same master test case are merged to find the overall test result containing a master case that should have been tested considering entire distributed embedded system as a whole. The Merged test results will clearly reveal buggy areas either in hardware or software are both. The test results also are helpful for assessing overall reliability of the entire distributed embedded system. Table 2 shows the overall test results obtained for the pilot distributed embedded system. The overall reliability of the system with the test cases used appears to be nearing 100%.

Table 2: Merged Test results

Functional requirement Number	Functional Requirement	Split test cases	Input Variable-1	Input Variable Value 1	Input Variable-2	Input Variable Value 2	Test Output Variable	Test Output Values	Expected output	Test Pass/fail	Overall test status
1	Read Temp-1 and write to LCD	Test Temp-1 read is written to LCD	TEMP-1	32			LCDW	32	32	P	Pass
		Test for proper sensing of Temp-1 signal at the output of the temperature sensor					Signal validity	TRUE	TRUE	P	
2	Test the Communication based communication between the 89C51 (ES-1) and the central ES-5)	Test for proper outputting data on output communication port of ES-1	COMM-REG1	32	COMM-REG2	21	OUT-STA	Y	Y	P	Pass
		Test for proper data read at the input communication on output communication port of ES-5	COMM-REG1	32	COMM-REG2	21	OUT-STA	Y	Y	P	
3	Read-Temp-1 and send to Central Micro Controller	Test for Reading of a particular Temparature-1 @ES-1	COMM-REG1	32	COMM-REG2	21	OUT-STA	Y	Y	P	Pass
		Test for Reading of a particular Temparature-1 @ ES-5	COMM-REG1	32	COMM-REG2	21	OUT-STA	Y	Y	P	
4	Read Temp-1 and measure throughput	Test for throughput	THRU-T1-TIME	10			TEMP1-THRU	10	10	P	Fail
		Test for through put					TEMP1-THRU	12	10	F	
5	Test the Communication based communication between the 89C51(ES-3) and the Central Micro Controller (ES-5)	Test for proper outputting data on output communication port of ES-3	PUMP1-STA	ON			PUMP1-STA	ON	ON	P	Pass
		Test for proper data read at the input communication on output communication port of ES-5	PUMP1-STA	ON			PUMP1-STA	ON	ON	P	

7. Conclusions

Testing a distributed embedded system considering the entire system as whole is complicated due to several reasons especially the difficulty involved in setting the environment required for undertaking the testing. In a distributed embedded system while a cross section might function while the other might be malfunctioning. Failure of hardware and communication interfaces will make it complicated to administer a test case. One should know precisely where a failure can occur.

Decomposing test cases into elementary test cases such that the test cases can be executed locally and then combining the test results will provide a basis and framework for undertaking testing of an embedded system as a whole.

Acknowledgement

I acknowledge KLEF (deemed to be university) for having given the opportunity and ambiance required for undertaking the experimentation related to this work. I acknowledge Dr. JKR Sastry, for guiding me in completing this work and Dr. A. Srinath, for extending his moral support, and research related thought provocation.

References

[1] K. Chaitanya, J.K.R. Sastry, K. Rajasekhara Rao, D.B.K. Kamesh, An Effective Model for Testing Distributed Embedded Systems using Scaffolding Method, PONTE International Journal of Sciences and Research, issue.8, Vol.73, 2017.

[2] K. Chaitanya, Sastry JKR, K. N. Sravani, D. Pavani Ramya and K. Rajasekhara Rao, Testing Distributed

[3] Embedded Systems using Assert Macros, ARPN Journal of Engineering and Applied Sciences, 2017, page no. 3011-3021.

[4] K Chaitanya, K. Rajasekhara Rao, JKR Sastry, DBK Kamesh, Testing Distributed embedded systems through instruction set simulators", PONTE, International Journal of Sciences and Research", issue.7, vol.73, July 2017, page no. 353-382.

[5] K Chaitanya, K. Rajasekhara Rao, JKR Sastry, DBK Kamesh, An Efficient Method for Testing Distributed Embedded Systems using In-circuit Emulators, "PONTE, International Journal of Sciences and Research", issue.7, vol.73, 2017, page no. 390-422.

[6] K. Chaitanya, JKR Sastry, K. Rajasekhara Rao, Testing Distributed Embedded Systems Using Logic Analyzer, International Journal of Engineering and Technology, issue.1, vol.3, March 2018.

[7] David E. Simon, an Embedded System Software Primer, Pearson Publications.

[8] Justyna Zander, Ina Schieferdecker, Pieter J, Model-Based Testing for Embedded Systems, CRC Press, and page no. 245- 249.

[9] Bruno Bouyssounouse, Embedded Systems Design: The ARTIST Roadmap for Research and Development, Springer, 51-55.