

Universal Methodology for Embedded System Testing

Swapnili P. Karmore

Department of Computer Science & Engineering
G. H. Raisoni College of Engineering
Nagpur, Maharashtra, India
swapnali.karmore@raisoni.net

Dr. Anjali R. Mahajan

Department of Computer Science & Engineering
P.I.E.T. Nagpur,
Maharashtra, India
armahajan@rediffmail.com

Abstract: This paper describes testing framework that is capable of testing heterogeneous embedded systems. There are three key contributions. The first is the introduction of a new approach of embedded system testing. The second is simple and efficient embedded system classifier interface, named as discovery interface device for embedded system. The third key contribution is a method for combining classifiers in one module, provides environment and testing methodology for embedded system testing. A set of different experiments in the domain of testing of embedded system is presented. This system yields of embedded system testing comparable to the best previous system implemented on traditional embedded software testing tools. This approach is capable to testing of host based and target based embedded systems.

Keywords: DID-Discovery interface device, HBT- Host based embedded system testing, and TBT- Target based embedded system testing.

I. INTRODUCTION

Testing is most commonly used method for determining quality of software. In the embedded world testing is a great challenge. In test plan sole characteristics of embedded systems must be reflected as they are application specific systems. It gives embedded systems testing exclusive and distinct flavor. There are two important classes of embedded systems, safety critical embedded systems and technical scientific algorithm based embedded systems. Host based embedded devices and target based embedded devices are sub classification for embedded systems. This paper brings together new approach and methodology to construct a frame work for heterogeneous and extremely complex embedded testing. Toward this end we have constructed a discovery interface device which provides classification and identification of embedded system as well as responsible for selection of testing methodology, this discovery interface device is the new concept in the embedded world. In the traditional testing approaches decision has to be made that whether to test hardware or software, but instead of finding heterogeneous testing techniques focus is on developing a mechanism which will assemble together all techniques of testing. This approach is capable to test host based and target based embedded system testing. Reminder of paper contains characteristics of embedded systems for soft real time and

hard real time, literature survey, proposed methodology, experimental setup, conclusion, future scope and references.

II. CHARACTERISTICS OF SOFT REAL-TIME AND HARD REAL TIME EMBEDDED SYSTEMS

Before performing actual embedded software testing or hardware testing, embedded system testing characteristics are identified. [5] Embedded characteristics are different for soft as well as hard real time system, on the basis of which we can design test cases for testing.

For the soft real time system, first characteristics is *embedded characteristics*, as embedded system is application specific and dedicated to specific task in a specific runtime environment, it is very difficult for testers to test the system in host based or target based pattern and run the host based system to target based and target based on host based.

Second characteristic is *interaction*; it is used for system application and execution. An interaction is very important characteristics in the operation of embedded system. Inter task communication, parallel execution of multiple users task, resource sharing and strict priority based execution among multitask application are playing important role in testing. Failures of interactions among these may lead priority inversion and other execution anomalies.

Third characteristic is *development practice* where supporting platform and interfaces plays an important role.

Fourth characteristic is *timeliness*, timeliness is critical and a major concern in both design and testing phases in hard real time embedded systems. Timeliness is used to determine whether the system is meeting real time constraint or not. In soft real time embedded systems sometimes timing issues are not so much important.

This testing approach gives solution to testing of heterogeneous embedded systems. [3] In a safety critical environment testing is risk based testing. [8] We can test ten numbers of different embedded systems in a one universal embedded system testing platform. Our experimental result shows that our method is promising for deployment in real applications.

III. LITERATURE SURVEY

Embedded system testing is not a simple task, it varies application to application. Embedded systems can be tested manually as well as automatically. Some of the embedded system testing uses simulation based verification when new multi core application arises, so in order to improve the quality of software the test generation frame work is used static analysis, mutation testing and the model checking is used for the message passing programming libraries for distributed embedded system. The verification simulation testing model checking could be used in order to check the equivalency between the original program and the program with inserting fault depend on the result .The fault is detected by model checker used in order to generate the test cases. No time out were used for C bounded model checker, no scalability will explored in order to test. Unique test cases are used MCAPI which has no longer benchmark. [01]

Due to diverse architecture of software between the embedded systems, make it difficult to reuse, portability and dependability. Middleware is software between operating system and an application to solve any beginning problem some middleware consists of API module service manager and content manager module. API module is used with interface to communicate with lower and upper layer the application of touch screen for the embedded, middleware system is used in order to verify usability. functionality and the integrated test of embedded middleware used communication between system to system, operating system to libraries ,libraries to application were the issues related to the diverse architecture embedded system such issues were solved somewhere in some extend.[02]

Some embedded software testers use supporting tool for embedded software testing. Embedded system requires hardware for testing purposes having very high development cost. Supporting tool can automatically generate test cases and test drivers, and supports coverage test and unit test which are based cross testing technology and multiple round mechanism results can be shown graphically by using constructed testing environment. The (ATEMES) Automatic testing environment for multi core embedded software is composed of four parts pre-processing module (PRPM), Host-side auto testing module (HSATM), Target-side auto-testing module (TSATM) and the last is post processing module. [6]

The construction and application of electronic port gate system, which is vivid application scenario of internet of vehicles, will benefits the entry and exit of vehicles for quick clearance guidance. It is also a great help for the monitoring of comprehensive import the efficiency of entry and exit port is a significant improvement on administrative work. [11]

Traditional PLC software about programming and monitoring all were designed in serial port debugging mode based on local. So the remote monitoring and controlling function couldn't be realized in the mode. In the article the virtual serial port (VSP) technology was imported into the system and it turned the PLC serial port software from local control function to remote one. [12]

IV. PROPOSED METHODOLOGY

The proposed methodology for research work is as shown in Figure 1. This is a new design approach for embedded systems testing, several embedded testing activity are performed simultaneously.

The designing of embedded system testing will contain the designing of the following systems, test interface design

- A. Design of embedded systems
- B. Test code for module testing
- C. Module testing interface design

Test Automation and flow Design

- A. Test cases and test code
- B. Test script statement and flow
- C. Test flow script interpreter

The testing of embedded system can be performed here on the basis of two types Host based testing and Target based testing. The very first module is the development of discovery interface design. The proposed plan is related to the testing of embedded devices.

This testing mechanism is divided in to three parts such as general testing, internal testing, and external testing. General testing is performed on components that are basically used in embedded devices and without that the circuit can't be complete, such as, capacitor, sensor, etc.

Internal testing is testing of each component .This testing can be performed with the help of some algorithms which is used for embedded system testing. One of them is Hit and Jump algorithm, which is combination of random walk and the structured algorithm.

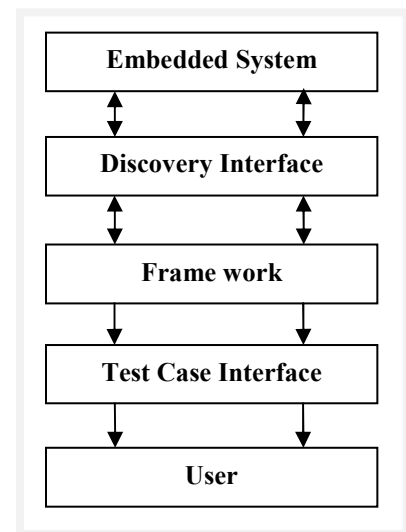


Fig. 1. Proposed Plan of work

Discovery interface design is responsible to identify and to discover drivers and API interfaces for circuit. Figure 2 shows the block diagram of discovery interface design. Microsoft has urbanized one operating system called WinCE for embedded systems. Device makers and OEMs are licensed by Microsoft. The OEMs and device makers provides functionality to transform and generate their own user

interfaces. They also support MIPS, and ARM processors, and also the current version of Windows Embedded Compact supports Intel x86 and compatibles.

Windows CE kernel can also work with minimum capacity memory, which is Megabytes of memory. Devices may be configured as a closed system that does not allow for end user expansion, example can be given as; it can be burned into ROM as they are frequently configured without disk storage space. [10] Windows CE competes to a real-time operating system characterization having deterministic interrupt latency. From Version 3 and forward, the system needs priority inheritance. Priority inversion levels are 256. For this system thread is the fundamental unit of execution so execution time gets reduced and simplifies the interface.

API (application programming interface) is an interface used by software components to talk with each other. This may include data structures, specifications for routines and object classes. An Application programming interface describes the way in which particular function is performed.

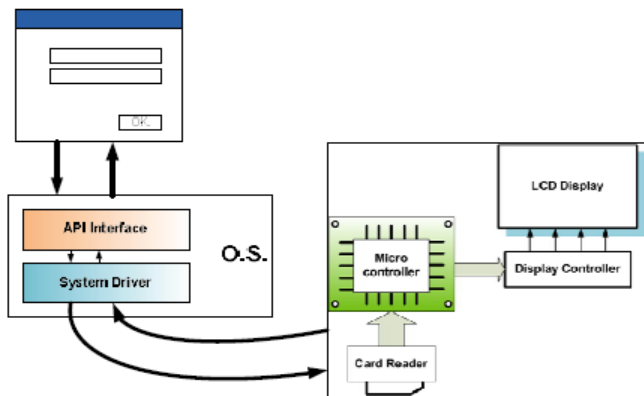


Fig. 2. Block Diagram of Discovery Interface Design

We can use MSP SDK 6.0, the MSP development SDK is an add-on. Basic function is providing documentation. It is used for supervision of the execution of .NET programmers. Common language runtime (CLR) is used, it is a machine component of Microsoft's .NET framework and it is virtual machine component. The compiled code is converted into machine instruction that, in turn, is executed by the computers CPU. Memory management, type safety, exception handling and some additional functionality are also given by CLR.

The basic aim of discovery interface design is to test whether hardware is present on embedded or not. Detection of system in order to categorize it as, in which group it is lying safety critical or of technical scientific.

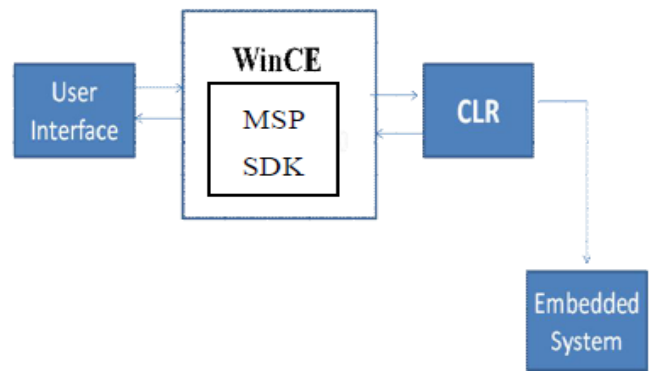


Fig. 3. Proposed Architecture for DID(Discovery Interface Design)

A. Host based testing in Embedded Systems

In the host based testing, we are performing performance and load testing of hardware. [5] Basic aim is to find out which feature and facilities are provided by embedded system. In future we can also find out the details of processor used in embedded systems. To describe host based testing, small experiment is performed that is named as memory management.

To interact with hardware plug-ins are required. This is dynamically linked file which interact with system driver like mobile O.S. and will fetch low level information for hardware. In computing, a plug-in (or plug n) is a set of software components that adds specific abilities to a larger software application. Emulator provide virtual environment for mobile OS.HBT is used for all types of mobile phones, for tabs, mobiles having different operating system like android based symbion based window based on so on .

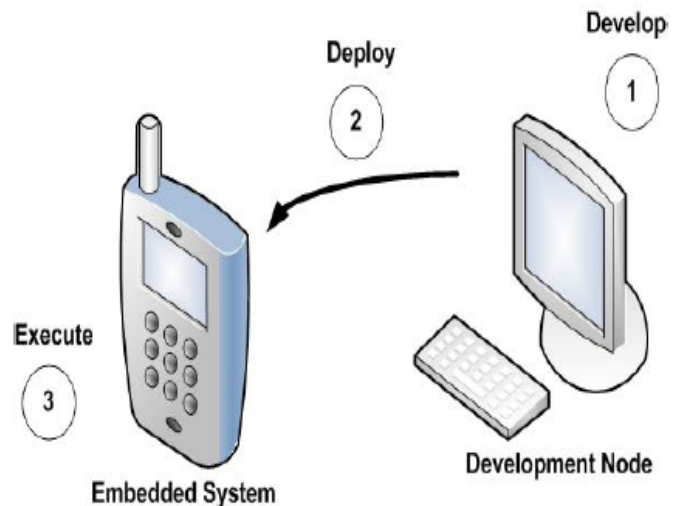


Fig.4. Approach-1 HBT, host based testing

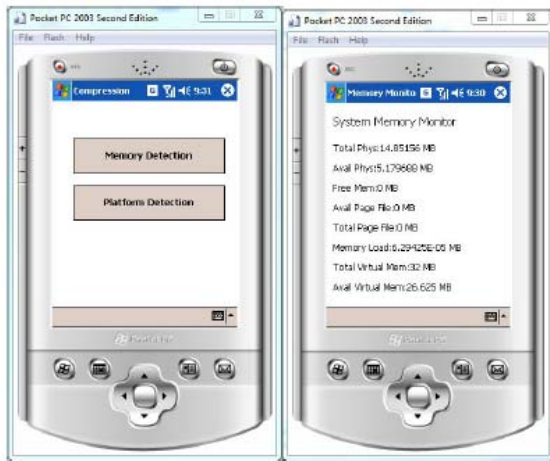


Fig. 5. Approach-1 Results for host based testing

Here experimental results are shown to demonstrate host based testing, software program is written and embedded via cable to Mobile phone containing window CE operating system responsible to perform memory detection and platform detection.

B. Target based testing

Figure 6 introduces an approach for target based testing. [5] Embedded system contains application software which varies with changes of low level software component; the testing framework of target hardware may contain set of LCD display, a keypad and a touch panel. The application software is responsible to display graphics on the LCD screen, and basic function is to reads user input from the keypad and touch panel.

In a changing and fast growing world all the testers are well known with the fact that perfect and accurate testing results only make them successful tester,

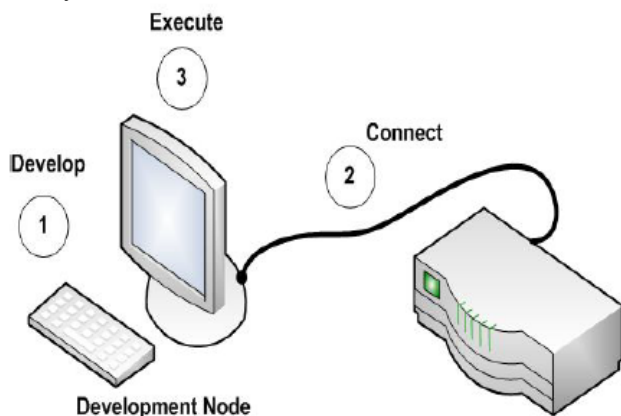


Fig. 6. Approach-2 Target based testing for embedded system testing.

Buying hardware reference boards for each specific and dedicated application development of embedded system leads increase in product development costs. In target based embedded system testing, for the development of hardware reference board connection probes and advanced development tools are required, which is further responsible for increase in

the development cost. Specialized Hardware reference boards for testing are difficult to set-up and use. All of these factors are responsible for making hardware reference boards an unfavorable choice for embedded system testing and application software development. [9]

V. EXPERIMENTAL SETUP

Justification of research is achieved after performing smaller experiments on host based and target based embedded devices. These results are associated with target based testing. Testing is performed on different embedded devices and its components over one environment with the help of serial port data transmission, USB port data transmission, and parallel port data transmission. In the USB Port data transmission again host based embedded system testing and target based embedded system testing are performed

In host based mobiles, laptops are tested and in target based all type of printers are tested. Due to application specific characteristics, it is not possible for tester to test embedded device with one tool.

Case Study 1:

Serial port Testing-

Testing components of embedded device like LED, Relay, motor etc.

Manual Testing Results:

Figure 7 shows embedded device testing over serial port, for device testing it is connected to com port one. After getting ok signal from controller we are ready to test different component of embedded device. For LED testing we have to send "LEDON" command on port, Controller listen to this command and perform task as given after completion will get the acknowledgement signal Relay testing also performed in these module



Fig. 7. Serial Data Transmission

Coding for setting of Serial Port Communication:

```
SerialPortManager _spManager;
private void UserInitialization()
{
```

```

        _spManager = new
SerialPortManager();
        SerialSettings mySerialSettings =
_spManager.CurrentSerialSettings;

serialSettingsBindingSource.DataSource =
mySerialSettings;
        portNameComboBox.DataSource =
mySerialSettings.PortNameCollection;
        baudRateComboBox.DataSource =
mySerialSettings.BaudRateCollection;
        dataBitsComboBox.DataSource =
mySerialSettings.DataBitsCollection;
        parityComboBox.DataSource =
Enum.GetValues(typeof(System.IO.Ports.Parity))
;
        stopBitsComboBox.DataSource =
Enum.GetValues(typeof(System.IO.Ports.StopBits))
);

        _spManager.NewSerialDataRecieved
+= new
EventHandler<SerialDataEventArgs>(_spManager_N
ewSerialDataRecieved);
        this.FormClosing += new
FormClosingEventHandler(MainForm_FormClosing);
    }

```

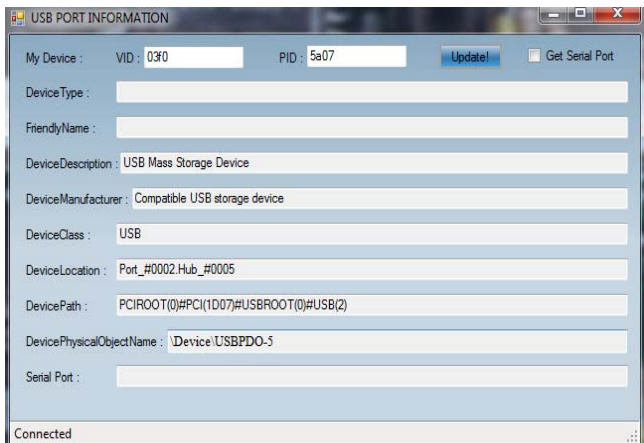


Fig. 8. USB port Testing

A major contributing factor in the success of USB as a peripheral interface has been the close adherence to the specification and the testing program that ensures USB interoperability. There are several instances in which it is essential to test the USB port, to test whether the USB is connected or not and if not working then to find out the cause.

Figure 8 is for extracting information of any USB device connected on USB device, for any device there is particular product ID and virtual ID number, with the help of which we can extract information of that particular device.

We can tests performance of any flash drive device with the help of this module, logic is on the runtime we are writing one dummy file of particular size in a defined loop and we are calculating average performance.

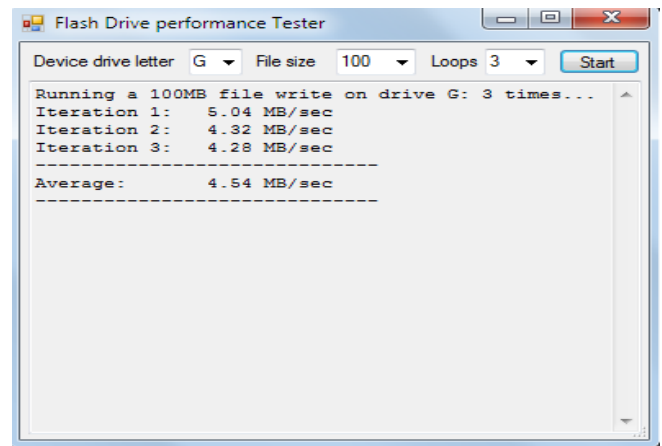


Fig. 9. Performance of Flash drive

Figure 9 shows flash drive performance testing, where different size of files; for example 100 MB or 200 MB can be selected from file size. File can be written on drive depending on number of loops selected from loops section. Average of time is calculated to determine flash drive performance in MB/sec.

VI. CONCLUSIONS

We have introduced a new approach for the testing of heterogeneous embedded systems in a real time environment. The host base and target base model serves as basis for heterogeneous embedded system testing. Testing of ten number of embedded system under one tool is not possible as testing of mobile is not similar to testing of card of washing machine, each embedded system requires its own specific tool.

Embedded systems are connected via serial port parallel port or via USB, are detected and identified by Discovery interface design. Discovery interface categorized it as safety critical embedded systems and technical scientific based embedded systems. For the host based embedded system memory management experiment is explained and for target based testing USB testing and flash performance testing is explained. Under one single environment target based as well as host based testing is achieved. We are proposing universal methodology for testing of heterogeneous embedded system

A set of different experiments in the domain of testing of embedded systems is presented. This approach is capable to test host based and target based embedded system testing. Low level testing methodology for embedded systems is proposed here. Our experimental result shows that our method is promising for deployment in real applications. To save energy, stamina and time; there is a requirement of advance methodology. This paper describes testing framework that is capable of testing several embedded systems and embedded system components. Future scope is development of testing tool for heterogeneous embedded system and embedding an artificial intelligence approach for improving embedded system testing.

REFERENCES

- [1] Alper Sen and etem Deniz, "Verification test for multicore communication API (MCAPI)", 2011-12th international workshop on microprocessor test and verification, 2011.
- [2] Yang-hassin fan and jan-ouw, "Middleware software for embedded system", 2012 26th international conference advance information networking and application workshop, 2012.
- [3] C. Reichmann; P. Graf; K.D. Müller-Glaser: "GeneralStore – A CASE-Tool Integration Platform Enabling Model Level Coupling of Heterogeneous Designs for Embedded Electronic Systems", 11th IEEE Conference on the Engineering of Computer Based Systems 2004, Brno, Czech Republic, May 2004.
- [4] Justyna Zander-Nowicka1, Zhen Ru Dai1, "Model Driven Testing of Real-Time Embedded Systems- From Object Oriented towards Function Oriented Development", IFIP 17th Intern. Conf. on Testing Communicating Systems – Test Com 2005, Montreal, Canada, ISBN: 3-540-26054-4, Mai 2005.
- [5] Hua-ming Qian; Chun ZhengA, "Embedded Software Testing Process Model Computational Intelligence and Software Engineering" , 2009. CiSE 2009. International Conference on 2009.
- [6] Alexander Krupp, Wolfgang Muller Paderborn University, "A systematic Approach to the Test of Combined HW/SW Systems", IEE Conference on the testing and automation of embedded systems, Nov2010.
- [7] Huang Bo, Dong Hui ,Wang Dafang and Zhao Guifan School, "Basic Concepts on AUTOSAR Development", School of Automobile Engineering, Harbin Institute of technology at wihai, Shandon Province, China, Princeton University, IEE International Conference on Intelligent Computation Technology and Automation, 2010.
- [8] Johannes Kloos, Tanvir Hussain, Robert Eschbach, "Risk-based Testing of Safety-Critical Embedded Systems Driven by Fault Tree Analysis Fourth International Conference on Software Testing", Verification and Validation Workshops 2011.
- [9] Mark Baker and Hong kong, "Design and Implementation of The Java Embedded Micro-kernel Software Architecture The Distributed Systems", Group University of Portsmouth, PO1 2EG, UK Mark.Baker@computer.org and Hong.Ong@port.ac.uk 2011.
- [10] Hasrul Ma'ruf*, Hidayat Febiansyah* and Jin Baek Kwon, "Supporting Java Components in the SID Simulation System", Journal of Information Processing Systems, Vol.8, No.1, March 2012.
- [11] Wenxuan Yin_yz, Xiang Gaoyz, Xiaojing Zhuyz, Deyuan Guox, "An Efficient Shared Memory Based Virtual Communication System for Embedded SMP Cluster", 2011 Sixth IEEE International Conference on Networking, Architecture, and Storage
- [12] Li Deng, Hai Jin, Song Wu, Xuanhua Shi, Jiangfu Zhou , "Fast Saving and Restoring Virtual Machines with Page Compression", International Conference on Cloud and Service Computing, 2011.