

Improving Embedded Software Test Effectiveness in Automotive Applications

Freescale Semiconductor
Author, D Brook

Document Number: CODETESTTECHWP
Rev. 0
11/2005





As the automotive industry introduces more and more safety-critical, computer-based systems (e.g., brake-by-wire), there is a growing convergence between the guidelines that the Automotive¹ and Aerospace² industries use for developing safe and reliable embedded software.

The Automotive industry implements strong software development processes, using software standards and certification in an effort to improve the overall quality and reliability of embedded software. The intelligent interconnection of tools, processes and methods should also be a central focus of developing better software. This discussion focuses on how structural coverage analysis techniques may be used to improve overall test effectiveness, one aspect of the Automotive and Aerospace software development guidelines.

Methodologies to improve the testing and testability of software have improved in recent years, but it is not unusual for an embedded application to have as little as 20-30 percent of its code actually tested before first customer shipment. In most projects this is a quality, time and cost issue, but in systems where safety is a major characteristic, the problem can be more acute.

Testing is an essential component of the software verification process, and it serves two primary functions: to prove that the software works correctly and to detect programming errors and mistakes in the implementation of the software design. For large embedded systems, the human capacity to comprehend complexity is limited, and defects can never be completely ruled out. The following examples highlight difficulties with software testing:

- > Defects in the software design are difficult to identify.
- > Systems interacting with the real world present unique issues as hardware and software are not continuous and testing boundary values is not guaranteed to ensure test completeness.
- > Traditional functional and black box testing does not guarantee that all executable code or states can be reached.

Good embedded software testing is to some extent an art form and requires effective methods and tools to support the deployed testing strategy. Typically, more than 50 percent of development time is spent testing the software, representing a significant cost. For safety-critical software, it is important that the effectiveness and quality of this testing be tracked and measured to ensure test completeness. For safety-critical avionics software, structural coverage analysis data must be submitted to the certification authorities to prove that the software has been tested completely.

The objective of structural coverage analysis is to determine which code structures have not been exercised by requirements-based test procedures. There are many coverage analysis techniques, with varying degrees of rigor, from statement coverage through to modified condition/decision coverage.

The data that coverage analysis provides helps to identify:

- > Missing test cases
- > Missing requirements—dead code
- > Unreachable code

By using the coverage data as feedback, test effectiveness can be dramatically improved, eliminating a lot of the guesswork associated with improving testing.

The DO-178B guidelines for developing avionics provide a very specific reference to deciding the required level of coverage, and a similar approach is echoed in the Motor Industry Software Reliability Association (MISRA) software development guidelines. DO-178B states that the more safety critical a system is, then the more rigorous the coverage analysis needs to be. Applying this approach to an example in the automotive market, this means that low impact automotive systems, such as an in-car entertainment system, would benefit from a high-level coverage analysis measurement such as statement coverage. At the other end of the scale, a throttle-by-wire system would require the most rigorous coverage analysis, such as modified condition/decision coverage. Choosing the right coverage level helps to ensure the beneficial use of development resources, ensuring that the more safety critical a system is, the more scrutiny it gets.

¹ Motor Industry Software Reliability Association "Development Guidelines for Vehicle Based Software"

² RTCA DO-178B/EUROCAE ED-12B "Software Considerations in Airborne System and Equipment Certification"

For safety-critical applications, system certification and tool qualification are increasingly used as a means to reinforce the most disciplined approach to software validation and the tools used for development and test. In the Avionics industry, software tools are categorized in reference to their ability to introduce and detect errors as follows:

- > Development tools have an output which can be part of airborne software and can introduce errors.
- > Verification tools cannot introduce errors but may fail to detect them.

The implication of this is that some tools are required to be qualified in order to be used for avionic application development. This means that the tool must be proven to produce deterministic results that are at least as accurate as the manual process that they replace. Freescale's CodeTEST™ software analysis tool is one such tool and is qualifiable under the US FAA³ Safety Critical guidelines for code coverage analysis tools.

CodeTEST uses a unique software instrumentation and data collection technology that works successfully with cache-based systems while minimizing intrusiveness. It adds statements or “tags” during the compilation process, leaving the source code intact. These tags are added at key execution points such as function entries and exits. The unique tag value is then written at run-time to a fixed memory location. A hardware data collection agent listens to this address and captures each tag write as it occurs. This allows efficient and effective execution analysis of software even on cache-based processors, providing the following analyses:

Software Execution Trace

Provides a detailed history of software execution allowing the root cause of software problems to be found. The information collected can be analysed and interpreted at the source code and RTOS level.

Memory Analysis

Monitors dynamic memory allocations and de-allocations, to help identify memory leaks and errors, and to provide the basic data needed to design a memory fragmentation solution.

Performance Analysis

Hardware data collection provides the highest resolution software execution speed measurements. The CodeTEST performance analysis measurements may be used to verify the performance of real-time code, or to focus optimization efforts. This allows evaluation of actual software execution against expectations and information about the impact of real-world characteristics such as external interrupts.

Code Coverage

CodeTEST provides three levels of coverage analysis: statement, decision and modified condition/decision coverage. The hardware data collection technique allows coverage measurements to be made at the system level with the minimum of intrusion.

It is worth restating that the primary objective of tools like CodeTEST is to reduce the cost of taking software to market through accelerated development time and improving the ability of the test process to identify defects. This objective can be achieved more easily if verification tools are part of a well integrated environment where time consuming, repetitive tasks such as test and verification can be automated. As the deployment of embedded software extends to more critical and complex areas of vehicles, such as braking and steering, the need for more sophisticated tools and methods is significantly more important. Tools which can interoperate through the design, development and test phases will have a dramatic positive impact on the cost and reliability of production software.

³ US Federal Aviation Administration

How to Reach Us:**Home Page:**

www.freescale.com

e-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor

Technical Information Center, CH370

1300 N. Alma School Road

Chandler, Arizona 85224

1-800-521-6274

480-768-2130

support@freescale.com

Europe, Middle East and Africa:

Freescale Halbleiter Deutschland GmbH

Technical Information Center

Schatzbogen 7

81829 Muenchen, Germany

+44 1296 380 456 (English)

+46 8 52200080 (English)

+49 89 92103 559 (German)

+33 1 69 35 48 48 (French)

support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.

Headquarters

ARCO Tower 15F

1-8-1, Shimo-Meguro, Meguro-ku,

Tokyo 153-0064, Japan

0120 191014

+81 3 5437 9125

support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.

Technical Information Center

2 Dai King Street

Tai Po Industrial Estate,

Tai Po, N.T., Hong Kong

+800 2666 8080

support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor

Literature Distribution Center

P.O. Box 5405

Denver, Colorado 80217

1-800-441-2447

303-675-2140

Fax: 303-675-2150

LDCForFreescaleSemiconductor

@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright license granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc., 2005.

Document Number: CODETESTTECHWP

Rev. 0

11/2005

