

WIND RIVER

Case Study: Ultra Datal Safety-Critical Avionics Upgrade Using COTS

Mark James, Marketing Manager, LDRA

Paul Parkinson, Senior Systems Architect, Wind River

Tom Roberts, Embedded Software & Systems Engineering Manager, Ultra Electronics Datal

Executive Summary

This case study presents the midlife upgrade of a pre-existing, uncertified, avionics system and highlights the significant challenges due to requirements for DO-178B Level B safety certification coupled with a migration to a commercial off-the-shelf (COTS) hardware platform. It focuses on the advanced test techniques that were applied and specifically how the LDRA tool suite was used to overcome the challenges to develop a safety-certifiable platform running on Wind River's VxWorks.

Introduction

In recent years there has been a dramatic increase in the availability of COTS software designed for safety certification under standards such as the RTCA/DO-178B avionics standard. This has been due in part to the steady proliferation of COTS-based hardware platforms, since the Perry memorandum in 1992¹ paved the way for the use of COTS on U.S. Department of Defense programs.

The use of certifiable COTS software is often just considered for new development programs that have an explicit safety certification requirement at the outset. It is also becoming

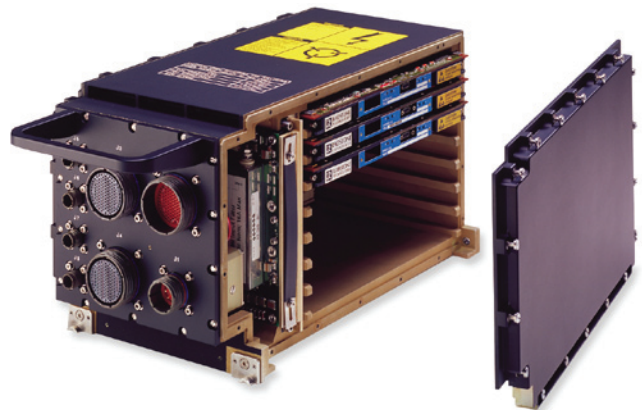


Figure 2: GE Intelligent Platforms rugged VME chassis

evident that there are programs that need to undergo technology refreshes or midlife upgrades. Safety certification requirements may be introduced at these points because of increased functionality or dependency on the system. Such programs require that certification evidence be developed to prove the correct, safe operation of the entire system, not just the new functionality, and this presents interesting challenges for both development and certification.

Project Background

Ultra Electronics Datal has undertaken the upgrade of a pre-existing avionics system where a DO-178B Level B safety certification requirement has been introduced in addition to a need to migrate from proprietary hardware to a ruggedised COTS-based hardware solution, using a GE Intelligent Platforms Intelligent Platforms PowerPC single board computer (SBC). This project lasted for 18 months, starting with a team of six staff and rising to a peak of 18.

This case study considers how Datal used the LDRA tool suite to implement MISRA-C:2004 programming language standards conformance on Wind River's VxWorks and achieve DO-178B Level B testing objectives.

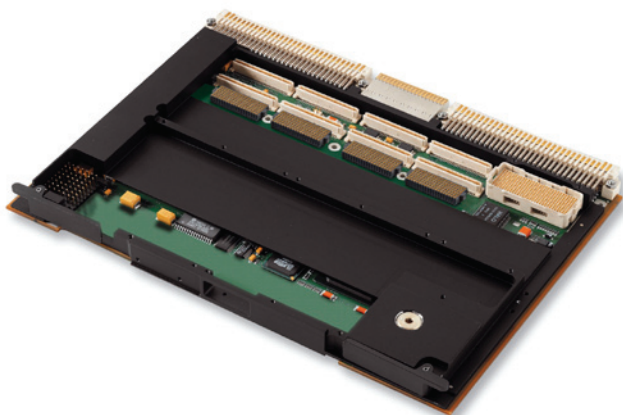


Figure 1: GE Intelligent Platforms PowerXtreme PPC9A

GE Intelligent Platforms

Development Challenges

This project faced a number of development challenges because the pre-existing software and device drivers were not developed with safety certification in mind, and code needed to be re-engineered and modified to meet safety certification requirements.

Challenge 1: Porting to the VxWorks DO-178B Safety-Critical Subset

In practical terms, in order to achieve DO-178B Level B safety certification, a subset of the full VxWorks real-time operating system (RTOS) application programming interface (API) is used. This safety-critical subset excludes functionality that could compromise predictability and determinism (e.g., SCSI). In order to determine the dependencies of the existing project code on nonsubset functionality, Datel performed static analysis of the source code.

This approach also provides visibility of the impact of the changes to the code because if companies are not actively working toward the implementation and enforcement of coding standards or best practices a number of problems can arise. These can include inconsistency of coding styles and implementation, which can make source code peer review difficult and therefore adversely affect the ongoing maintainability of the code.

Challenge 2: Reduction of High Cyclomatic Complexity

The static analysis of the source code performed by the LDRA tool suite also revealed that some of the existing project code exhibited high cyclomatic complexity, which reflects a complex decision-making structure in terms of a directed graph. High cyclomatic complexity is undesirable for safety certification because the associated high number of paths through the code and complex conditions mean that it is complex and time consuming to perform functional and code-coverage testing.

As such, the development team used the results of the static analysis at the individual function level to determine which functions should be considered for modification or reimplementing in order to reduce cyclomatic complexity and better facilitate testing and safety certification. In some cases the risk of change had to be carefully weighed against the risk of failure.

In addition to verifying that the software is properly structured, this static analysis approach may be used to ensure that programming standards are uniformly enforced and the obvious benefits are recognized by many regulatory authorities that approve the use of such techniques for the development of safety-critical software.

Challenge 3: Programming Language Subset Compliance

The project selected a subset of the C programming language MISRA-C:2004² for the upgrade in order to reduce the risk of coding errors, facilitate ease of maintenance, and enable future portability. MISRA-C (Motor Industry Software Reliability Association) was originally developed through the collaboration between automotive manufacturers, engineering consultancies, and tools developers to promote best practice and commonality in the development of safety-related automotive electronics and other embedded systems. However, since its publication, it has quickly become regarded as a “best practice” for the development of C in embedded and safety-related systems and has been widely adopted in the aerospace, defense, and industrial sectors.

The MISRA-C language subset has been designed to promote portability and ensure that there is no reliance placed on compiler-specific or platform-specific constructs that could lead to unexpected or unpredictable results. It also restricts the use of certain C language constructs that are known to be a common source of errors and reduces program complexity that helps improve software maintainability. When applied to DO-178B software development, this can help to make the software more suitable for testing, which can provide tangible benefits in terms of reducing the effort required to perform functional testing and code-coverage testing as part of the safety certification process.

The project development team, however, was faced with a further challenge because the original code had not been written to conform to the MISRA-C:2004 subset or its predecessor. Therefore, many of the functions violated multiple MISRA-C rules. The team was faced with the decision of whether to change the code to comply with MISRA-C rules and risk failure through inadvertently changing the behavior of the code.

Difficult decisions had to be made to justify which rules should be adhered to from the MISRA-C:2004 standard and which rules could be relaxed. For example, some functions contained code with extensive and complex control decision logic and multiple return statements. To comply with strict MISRA-C:2004 enforcement it would be expected that such code should be refactored. However, in so doing there is always the risk that the control flow could be altered, and Datel was therefore left to decide whether much would actually be gained under such circumstances. Once these choices were made, Datel was able to create a custom profile within the LDRA tool suite to reflect its choice of MISRA-C rules.

Challenge 4: Code Coverage Challenges to Meet DO-178B Level B Objectives

DO-178B Level B safety certification requires that block, statement, and decision-level code coverage be undertaken on the software. However, the existing software comprised 80,000 source lines of code (SLOC), which had not been written with code-coverage testing in mind, making it difficult to manually determine all of the test cases that would be required.

To overcome this issue, the Datel development team used LDRA TBeXtreme for source code analysis and automated test case generation and dramatically reduced the time required for unit and code coverage testing. In addition, the development team members were able to perform unit testing of C packages in parallel and integrate the results, reducing the overall testing time.

Datel also utilized the extensive command-line automation facilities of the LDRA tool suite to perform nightly regression test runs to ensure that code changes did not introduce errant behavior. This approach was used in conjunction with real target hardware for certification testing rather than a simulated environment that might exhibit different behavior to the actual hardware.

Finally, a comparison of the baseline and modified code was required to verify functionality. The LDRA tool suite was again used to automate this process and highlight source code changes between baseline and modified versions and report on untested source code that was affecting the overall code coverage analysis metrics.

Conclusion

Datel made a number of significant achievements in this safety-critical avionics upgrade project through use of the LDRA tool suite and Wind River's safety-critical VxWorks operating system.

It was able to reduce risk associated with modifications to the project source code through the information provided by the source code analysis and MISRA-C rule checking. It was also able to reduce testing time through automated test case generation and parallel working, resulting in a starting baseline of approximately 90 percent code coverage. The support for automated regression testing also enabled them to introduce changes and verify their impact quickly and with traceability.

The outcome of the project was the creation of a DO-178B Level B certified safety-critical software system, which was delivered to the customer, who in turn will perform system integration testing with its application, also making use of the LDRA tool suite.

About Ultra Datel

Ultra Electronics Datel is part of the Ultra Electronics Group and has been in business for more than 25 years. Datel specializes in mission- and safety-critical software development to a variety of civil standards, including RTCA/DO-178B and UK MOD Defense Standard 00-55/00-56. Datel also provides services in systems integration, consultancy, and secure information management solutions.

About LDRA

For more than 30 years LDRA has developed and driven the market for software used for the automation of code analysis and software testing of safety-critical applications. The LDRA tool suite is widely used in the aerospace and defense technology, nuclear energy, and automotive industries. Through the use of the LDRA tool suite, companies ensure that their systems are built in accordance with prescribed standards and are durable and reliable in use. The LDRA tool suite is available for a variety of programming languages and supports a wide range of host and target platforms. LDRA is represented worldwide, with its head office in the UK and subsidiaries in the United States, supported by an extensive distributor network. For more information on the LDRA tool suite, visit <http://www.ldra.com>.

About Wind River

Wind River, a wholly owned subsidiary of Intel Corporation, is the global leader in embedded technologies. Wind River enables companies to develop, run, and manage device software faster, better, at lower cost, and more reliably. Wind River platforms are pre-integrated, fully standardized, enterprise-wide development solutions. They reduce effort, cost, and risk and optimize quality and reliability at all phases of the device software development process, from concept to deployed product. Founded in 1981, Wind River is headquartered in Alameda, California, with operations worldwide. To learn more, visit Wind River at www.windriver.com.

Notes

1. W. Perry, U.S. Secretary of State for Defense, "Specifications & Standards – A New Way of Doing Business," U.S. DOD Memorandum (1994).
2. "Guidelines for the Use of the C Language in Critical Systems," MISRA-C:2004, Motor Industry Software Reliability Association, <http://www.misra.org.uk>.