LDRA

Software Technology

Ultra ELECTRONICS

WIND RIVER

Case Study

Ultra Datel Safety-Critical Avionics Upgrade Using COTS

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

Abstract

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

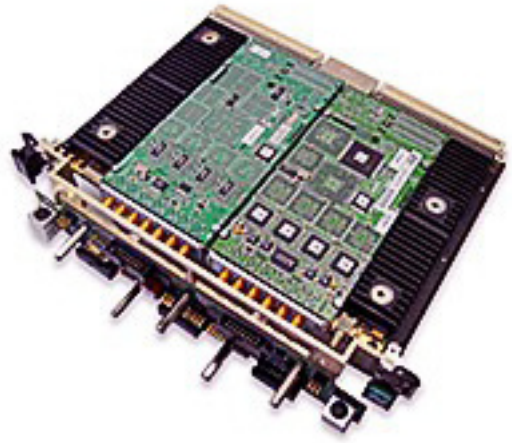
Introduction

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 1992 [1] paved the way for the use of COTS on U.S. Department of Defense programmes.

The use of certifiable COTS software is often just considered for new development programmes which have an explicit safety certification requirement at the outset. It is also becoming evident, however, that there are programmes which need to undergo technology refreshes or midlife upgrades and safety certification requirements may be introduced at these points due to increased functionality or dependency on the system. Such programmes 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 Datel has recently 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 Fanuc 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. In the remainder of this case study, we will consider how Datel used the LDRA tool suite to implement MISRA-C:2004 programming language standards conformance on VxWorks and achieve DO-178B Level B testing objectives.



GE Fanuc Intelligent Platform

Development Challenges

This project faced a number of development challenges, due to the fact that the pre-existing software and device drivers had not been developed with safety certification in mind and code needed to be re-engineered and modified to meet safety certification requirements. These development challenges are considered in turn in the following subsections.

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

In practical terms, in order to achieve DO-178B Level A 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 which could compromise predictability and determinism (e.g., SCSI). Hence 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 changes to the code because if companies are not actively working towards 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 means 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 reimplementation in order to reduce cyclomatic complexity and hence 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 utilised to ensure that programming standards are uniformly enforced and the obvious benefits are recognised by many regulatory authorities which 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:2004 [2] 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, defence and industrial sectors.

“ ...if companies are not actively working towards the implementation and enforcement of coding standards or best practices a number of problems can arise. ”

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 which could lead to unexpected or unpredictable results. It also restricts the use of certain C language constructs which are known to be a common source of errors and reduces program complexity which helps to 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 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 which could lead to unexpected or unpredictable results. It also restricts the use of certain C language constructs which are known to be a common source of errors and reduces program complexity which helps to 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, were faced with a further challenge due to the fact that the original code had not been written to conform to the MISRA-C:2004 subset, or its predecessor, and therefore many of the functions violated multiple MISRA-C rules. The team were faced with the decision of whether they should change the code in order to comply with MISRA-C rules and, as a result, risk failure through inadvertently changing the behaviour 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 had been 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 which would be required.

To overcome this issue the Datel development team used LDRA TBeXtreme for source code analysis and automated test case generation and hence 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 required further.

Datel also utilised 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 behaviour. This approach was used in conjunction with real target hardware for certification testing, rather than a simulated environment which might exhibit different behaviour to the actual hardware.

Finally a comparison of the baseline and modified code was required in order 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 which was affecting the overall code coverage analysis metrics.

Project Achievements and Conclusions

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.

They were 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. They were 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 their application, also making use of the LDRA tool suite.

"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. "

References

[1] W. Perry, U.S. Secretary of State for De fense, "Specifications & Standards – A New Way of Doing Business," U.S. DOD Memorandum (29 June 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.

LDRA Headquarters
Portside, Monks Ferry,
Wirral, CH41 5LH
Tel: +44 (0)151 649 9300
e-mail: info@ldra.com

LDRA Technology Inc. (US)
Lake Amir Office Park
1250 Bayhill Drive Suite # 360
San Bruno CA 94066
Tel: (650) 583 8880

LDRA Technology Inc. (US)
74 Main St
Suite 209
Maynard MA 01754
Tel: (978) 405 3180