LDRA

Product Brochure

Delivering Software Quality and Security through Test, Analysis and Requirements Traceability

LDRA
Software Technology

Companies producing safety-, security-, and mission-critical software must meet the most stringent industry standards while measurably improving developer productivity and software quality. The *LDRA tool suite*® and LDRA Certification Services provide a comprehensive solution to address today's demands for traceable and demonstrable requirements-driven development and verification.

The tool suite, which has an unrivalled pedigree of over 40 years in the software quality market, is a completely integrated yet open and extensible solution enabling customers to build quality into their software from requirements through to deployment. Integrating the *LDRA tool suite* into the software development process has enabled critical software development programs to achieve certification or approval under rigorous standards such as DO-178B/C Level A (Aerospace), IEC 61508 SIL 3 (Industrial), ISO 26262 (Automotive), EN 50128 SIL 3/4 (Rail Transportation), and IEC 62304 (Medical Devices, Class II and III).

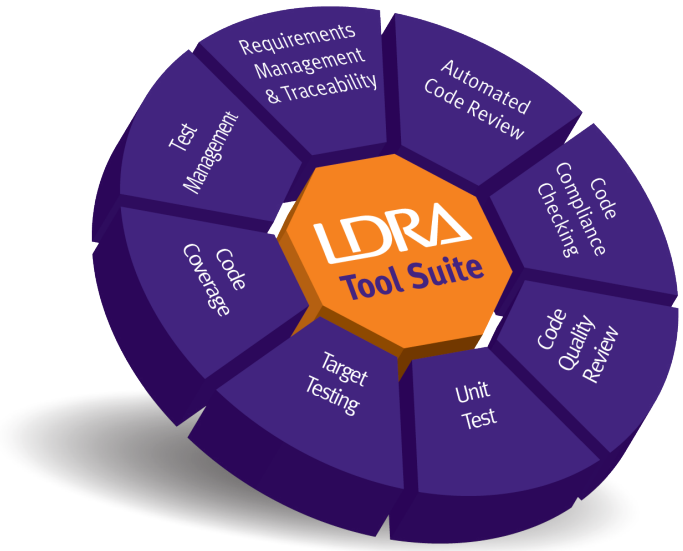
The technologies and services offered by LDRA have been successfully deployed across hundreds of projects in these domains expediting development, verification, and certification of critical software.

The *LDRA tool suite* can be effectively leveraged by an entire software project team throughout the development lifecycle. Daily users include project management, systems engineers, developers, QA managers and test and maintenance engineers.

As leaders in the critical software verification and validation market, LDRA and its employees participate in numerous standards bodies such as the MISRA C/C++ committees, the SC205/WG71 (DO-178C) committee within the avionics software market, the ISO software vulnerabilities working group, and the working group developing the "C Secure Coding Rules" annex to the SC 22/WG 14 C language group. In this role, LDRA consistently contributes to advancing state-of-the-art software engineering practices to help ensure the delivery of high quality safety, security, and mission critical systems.

LDRA
Software Technology

Automated Activities



LDRA Lifecycle Automated Activities

LDRA Tool Suite – Automating Best Practices in Critical Software Development

LDRA has developed its tool suite to enable the application of lifecycle best practices which produce software that meets the highest quality standards. The *LDRA tool suite* is unique in its integration of lifecycle traceability, static and dynamic analysis, together with unit and system-level testing on virtually any host or target platform.

The *LDRA tool suite* provides an open architecture and optimised integrations with various tools within the customer's selected tool chain such as requirements management tools, modelling tools, software configuration management tools, compilers, debuggers, integrated development environments (IDEs), real-time operating systems (RTOSs), communication protocols, processors, and target connections.

LDRA Tools and Services:

- Expedite and ensure a successful certification/approval process
- Provide insight and guidance throughout the software development lifecycle and through the certification process

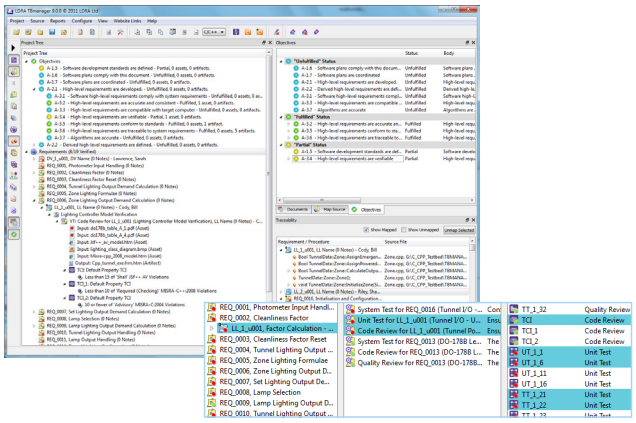
by helping customers to:

- Apply best principals in the software development lifecycle
- Comply with coding standards (industry and user-defined)
- Perform coverage analysis on all code down to the target level
- Automate unit and system level testing
- Execute requirements-based testing
- Trace all requirements and design artefacts throughout the development lifecycle
- Automatically produce certification/approval evidence

Lifecycle Best Practices Enabled with the LDRA Tool Suite

1. Requirements Management and Traceability

The *LDRA tool suite* can be used to bridge the gap between requirements, source code, verification plans, and verification results to create a Requirements Traceability Matrix (RTM). Requirements can come from sources such as IBM® Rational® DOORS®, IBM® Rational® RequisitePro®, a Microsoft® Word document or Excel spreadsheet. This RTM may then be verified by the *LDRA tool suite* and mapped to LDRA analysed source code or flow down from mappings performed in modelling tools such as MathWorks® Simulink®, IBM® Rational® Rhapsody®, Esterel® SCADE® and Artisan Studio®. With either scenario the requirements can be fully verified using an automated process that provides round-trip engineering across the entire software development lifecycle. With the *LDRA tool suite* the reporting, resolution and management of defects are all complementary to requirements management, and incorporated into the RTM, bridging a major gap in software development.



TBmanager GUI's

TBreq® and *TBmanager*®, key enabling technologies within the *LDRA tool suite*, provide a task-oriented management capability that aggregates verification results and data as the individual verification tasks are assigned and executed. At any given time *TBreq* is capable of generating reports including and describing activities, results, and artefacts, with little or no overhead. The reports are designed to leverage user-definable templates which can be used to quickly and easily generate documentation in a variety of formats.

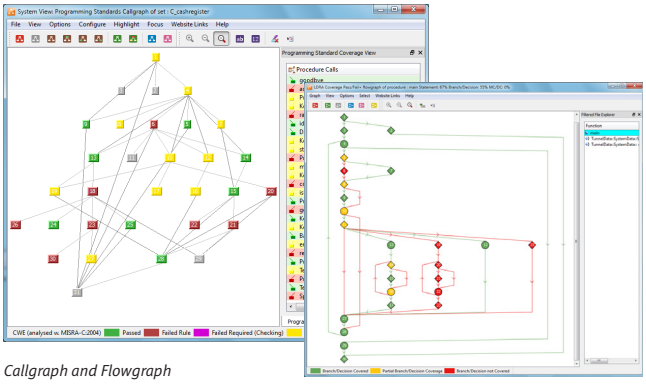
Through automation, the *LDRA tool suite* and the RTM automatically ensures that any change to requirements, design, or source code is easily traced and managed. This capability greatly reduces the costs associated with impact analysis, verification, debugging and change management, which together contribute the largest aggregate cost in software development today.

2. Automated Code Review

LDRA's support for automated code review includes automated analysis of the structure, interfaces, parameters, and data objects used in the code. Effectively, interface, coupling, control flow and data flow analysis of the source code under test is automated, presenting the results graphically or textually and greatly enhancing design conformance (or non-conformance) to design standards or rules. These results are available in many common development environments (such as Eclipse) as well as in reports for certification.

LDRA provides graphical code visualisation which raises the level of abstraction and improves code comprehension. This serves as an extremely powerful technique that helps developers identify and address issues earlier in the development cycle where they are less costly to fix. The *LDRA tool suite's* colour-coded diagrams simplify analysis and greatly enhance the code review process where:

- Callgraph diagrams provide a hierarchical display of the application and system components.
- Flowgraph diagrams provide a graphical display of the control flow across program blocks.



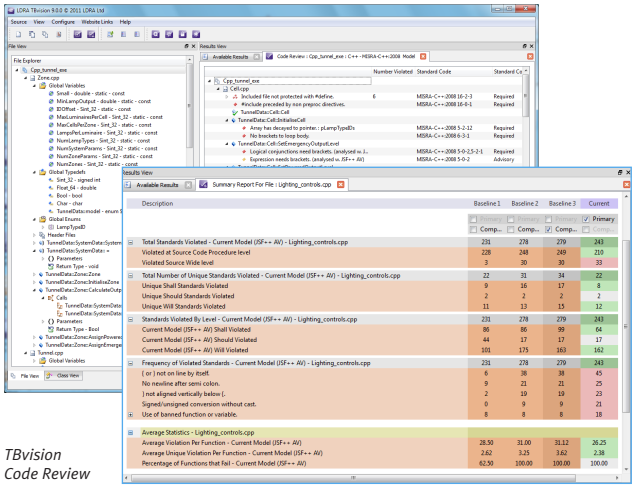
Callgraph and Flowgraph

The key deliverables automated by the *LDRA tool suite* are a systematic analysis linked to a graphical representation of the as-built application. These include system callgraphs and procedure flowgraphs, data flow analyses reports (from both parameter and variable usage viewpoints) and path level analysis reports employing LDRA's test path - linear code sequence and jump (LCSAJ) technology. These code review artefacts can be used by developers and management to improve the overall performance, efficiency, and quality of the code. Furthermore, these artefacts are also automatically carried forward into LDRA's unit testing and test verification components.

3. Code Compliance Checking

As part of the automated code review capability, the *LDRA tool suite* provides a powerful static analysis capability that automates the procedure of checking the code for compliance with user-defined rules or industry specific standards. Furthermore, through sophisticated reporting capabilities, compliance issues are quickly identified and resolved saving significant effort, time, and money when compared with the common experience of applying a traditional manual code review process. Available automated compliance checking solutions for industry recognised standards include MISRA-C:1998, MISRA-C:2004, MISRA C++:2008, MISRA AC, HIS, JSF++ AV, High Integrity C++, IPA/SEC C, Netrino C, CERT C or CWE.

Users may choose any of these standards or create and apply their own customised standard, optionally based on an existing standard. The code review facilities of the *LDRA tool suite* provide the user with the ability to quickly identify violations early in the software development lifecycle, often revealing latent errors that would not be identified through the standard testing process. This improves the quality of code, reduces the errors found during formal testing and helps to ensure on-time delivery within budget.

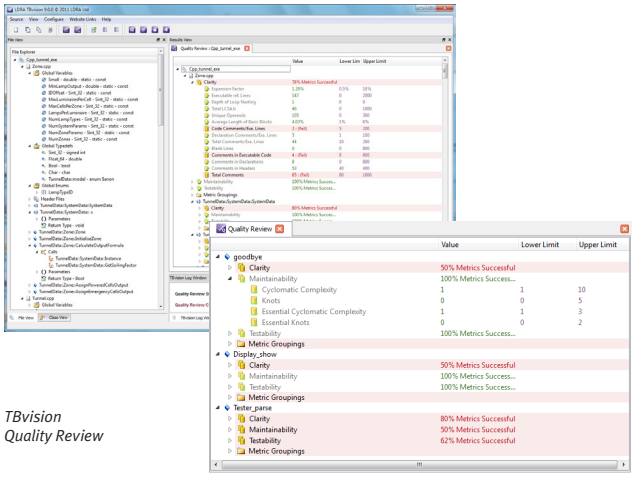


TBvision Code Review

4. Code Quality Review

The *LDRA tool suite* enables the developer to quickly determine the quality of the software with respect to testability, maintainability and clarity. In so doing, the available reporting goes beyond traditional complexity metrics to determine "essential complexity", i.e. the sections of the source code which require refactoring. Coupled with this, code density metrics, which are uniquely analysed by the *LDRA tool suite*, highlight potential future maintainability problems.

The key deliverables of the code quality review process are the visual quality model, system quality metrics and code refactoring guidelines that can all be used to improve overall code quality. The consistent enforcement of a user specified quality model which generates reports for the development team and management improves project communication as well as the overall quality of the application.



TBvision Quality Review

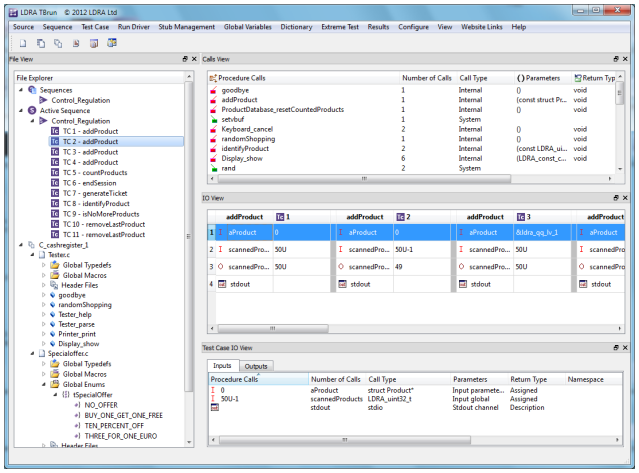
5. Unit Test

LDRA's unit testing capability assists developers in generating the test vectors, test harnesses and code stubs necessary to ensure complete testing for both unit testing and regression testing. Integrated coverage analysis provides the feedback necessary to ensure that all of the code under test has been fully exercised before transitioning to system integration and system test. LDRA's unit testing functionality contributes to the elimination of defects in the code that may not be identified until later on in the product development lifecycle. Once the test cases are defined, the tool automatically generates a test harness containing the sequence of test cases that are used on both the code under development and later on for regression testing. These tests can be executed on either the host or target-based system.

Too often development teams are forced to rush their product to market without sufficient verification and validation being performed. Compressed development schedules often prohibit the manual generation of a complete set of unit test cases for verification purposes. LDRA's extreme testing tool, *TBeXtreme®*, helps to ensure that the essential job of unit testing is completed by using the knowledge of the code structure gained from the static analysis to automatically generate test vectors with the objective of exercising as much of the code under test as possible. Once generated, test harnesses for test case execution and regression testing are then produced, eliminating the traditional time and resource problems associated with bottom-up testing. These test cases may then be regressed in a host or target environment via a fully automated process.

Benefits of LDRA's unit test solution:

- Faster, more comprehensive test generation and execution on host or target
  - Automated test vector and driver / harness generation expedites the generation and execution of quality tests while eliminating the requirement for manual scripting
  - Intuitive graphical and command line interface options
- Shorter testing time and reduced testing effort
  - Sophisticated automated analysis reduces manual testing and review freeing up developers and empowering testers
- Automated Regression Testing
  - Test data and test results are stored and maintained for fully automated regression analysis
- Improved product quality, documentation accuracy, and less time performing impact analysis
  - Automatic detection and documentation of source code changes
- Automatically generated test case documentation including pass/fail and regression analysis reports

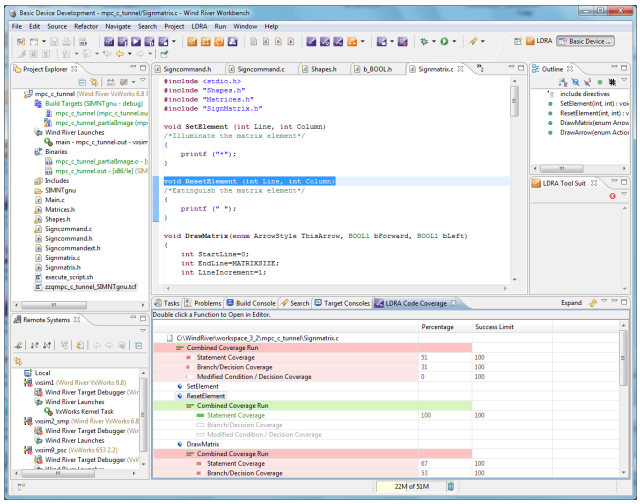


Unit Test GUI

6. Testing on Target Hardware

A key capability of the *LDRA tool suite's* target testing is the high degree of flexibility and adaptability offered by the tool, virtually eliminating the risks associated with tweaking tools, parameters, and integrations for unique target constraints and evolving target systems.

The *LDRA tool suite* provides highly automated solutions for a wide range of processor / IDE combinations, which addresses the significant challenges of performing test execution on a specific target. Moreover, the *LDRA tool suite* provides Object Code Verification (OCV), a proven, single-tool solution to directly compare code coverage at the source code level with that achieved at the object code level.



Example Target Test System

7. Code Coverage

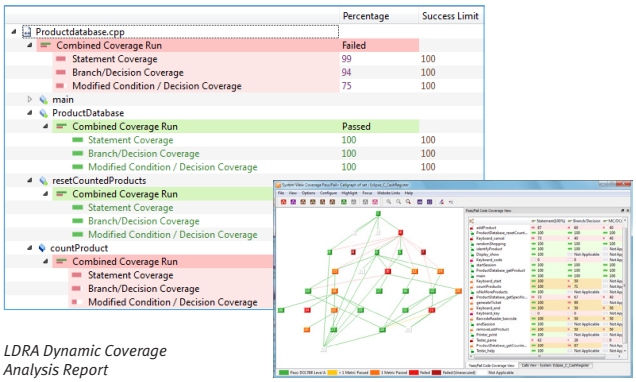
When building software to meet the highest quality standards, it is essential to ensure that the system is fully tested. LDRA's coverage analysis measures test data effectiveness by determining how much of the software code structure under test has actually been exercised. Structural coverage has become a best practice for quality-conscious development teams who must ensure that their products are safe, reliable and secure for use in application areas such as avionics, medical devices, industrial control systems, rail transportation, and critical automotive electronic control units.

LDRA provides an extensive range of coverage analysis measurements in its *LDRA Testbed*® and *TBrun*® products, ranging from straightforward statement coverage to the Modified Condition / Decision Coverage (MC/DC) required for most safety-critical avionic systems. Also included is Linear Code Sequence and Jump (LCSAJ) coverage – which is a test path metric that offers a more thorough test data assessment than decision coverage yet avoids the exponential difficulty of full path coverage.

In the avionics market, the DO-178B/C software development guidelines demand that the most safety-critical applications (Level-A classification) provide 100% coverage analysis to the MC/DC level on the source code, in addition to ensuring 100% coverage of the object code (assembler). The *LDRA TBobjectBox*® solution augments the source level coverage solution by adding object code coverage to provide the industry's most complete structural coverage analysis solution. *LDRA TBobjectBox* enables users to create test cases for structural coverage of source code and apply these exact same test cases to the corresponding object code. *LDRA TBobjectBox* comes in a number of variants to support different processor instruction sets. These reports can show both assembly and high level language code – making them an essential artefact for verifying the correctness of code generation.

The ability to quickly pinpoint inadequately or untested software is vitally important to improving the quality of the testing process and ultimately the product. Furthermore, this productivity enabler reduces regression testing costs by saving time, resources, and shortening the time-to-market while at the same time improving product quality. LDRA's test verification provides coverage metrics for the following:

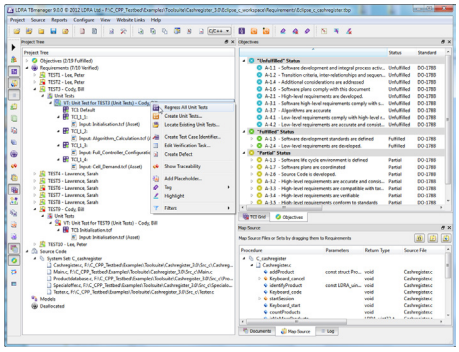
- Statement
- Branch/Decision
- Procedure/Function Call
- Branch Decision Condition
- Branch Condition Combination
- Modified Condition / Decision Coverage (MC/DC for DO-178B/C Level A)
- Dynamic Data Flow
- LCSAJ



LDRA Dynamic Coverage Analysis Report

8. Test Management

Being able to fully understand a system can take a long time and require large amounts of resources. The *LDRA tool suite* assists developers and test engineers in understanding, documenting and maintaining large complex systems. Additionally, LDRA's Test Management automatically detects changes to the source file(s) and performs a validation of associated regression test cases.



TBmanager GUI

Some of the key automated features include:

- Software change traceability which identifies and tracks significant source code changes and enables project teams to accurately monitor the impact of these changes on the testing process and analysis of the code.
- Detection of change in the lifecycle assets such as requirements, design, or code, to assess impact and optimise regression testing and suspect link resolution.
- Detection, validation, and regression of unit/module tests.

A key benefit of LDRA's test management facilities is the automated production of full system documentation for audits and version-control analysis, which also contributes to the reduction of ongoing maintenance costs.

LDRA
Software Technology

The *LDRA tool suite*

LDRA Testbed®

LDRA Testbed provides the core static and dynamic analysis engines for the LDRA suite of tools. This technology analyses both host and embedded software in a rigorous and systematic manner. LDRA Testbed is the proven technology foundation for supporting all software development efforts requiring certification or formal regulatory approvals.

TBrun®

TBrun provides a GUI-driven interface for creating unit and module test cases for either host or target-based software. Test harnesses are automatically generated and *TBrun,* also supports the creation of stubs for code that is outside the scope of the tests. Structural coverage analysis can also be included in the test harness, helping to ensure that the defined test cases execute 100% of the code under test. Formal test reports are produced and the generated test harnesses serve as a regression test environment for future development phases.

TBreq®/TBmanager®

TBreq supports the tracing of requirements through the software development lifecycle. *TBmanager* adds a task-oriented interface providing a common user experience across all *LDRA tool suite* modules and across the development team. High-level requirements captured in standard office documents or third party solutions, such as DOORS, IRQA or Requisite Pro can be traced through lower level documents to the code itself. The end result is a Requirements Traceability Matrix (RTM) that captures how each high level requirement was traced to the final software product, including the associated verification activities and results.

TBvision®

TBvision presents code standard violations and software flaws in the context of the original source code. The interactive environment, enabling the execution of both static and dynamic analysis on a user-defined scope, allows switching between reported violations, the original source code and any of the *LDRA Testbed* supported coding standards. In so doing *TBvision* clarifies why an issue is being reported and what remedial action is required. Software integrity can also be measured and reported in terms of quality, security, or simply the presence of defects (including dynamic memory errors). *TBvision* presents the identified software flaws from any of these perspectives and identifies the issues that need to be addressed to ensure that a software project meets its objectives.

LDRAcover®

LDRAcover utilises *LDRA Testbed* technology to provide extensive test effectiveness feedback generated through structural coverage analysis reporting facilities. *LDRAcover* addresses the rigorous structural coverage objectives of standards such as DO-178B/C (Avionics) up to and including Level A.

LDRArules®

LDRArules leverages *LDRA Testbed* to enforce compliance with industry or user-defined coding standards and provides clear visibility of software flaws that might typically pass through the build and test process to become latent problems. Most industry standard rule sets are supported off-the-shelf with additional compliance checks made available on an as needed basis.

Additional Options

TBmisra®

TBmisra provides coding standards checking against MISRA-C:1998, MISRA-C:2004, MISRA C++:2008, MISRA AC, JSF++ AV, High Integrity C++, DERA and IPA/SEC C.

TBsecure®

The *TBvision* plug-in *TBsecure* provides secure code standards checking against the Carnegie Mellon Software Engineering Institute (SEI) CERT C secure coding standard and the Common Weakness Enumeration (CWE) dictionary.

TBsafe®

TBsafe incorporates an additional set of high-integrity analysis tools to help rigorously test code to exacting standards such as those required for DO-178B/C, Def Stan 00-55 and IEC 61508. Features include Information Flow Analysis, Dynamic Data Flow Coverage, Modified Condition/Decision Coverage (MC/DC) for DO-178B/C Level A and Exact Semantic Analysis.

TBeXtreme®

TBeXtreme revolutionises unit testing by using the information gathered by *LDRA Testbed* to provide a totally automated solution for test vector generation. *TBeXtreme* eliminates the traditional time and resource problems associated with bottom-up testing.

TBobjectBox®

TBobjectBox provides an Object Code Verification (OCV) capability, as described in DO-178B/C, offering the only direct way to relate code coverage at the source code level with that achieved at the object code level. The tool also provides the mechanism to extend, where necessary, the code coverage at the assembler level.

TBevolve®

The *TBvision* plug-in, *TBevolve*, enables project teams to monitor the impact of code changes on their testing process. As the source code changes *TBevolve* will compare a baseline copy of a system with new versions and will highlight changed source code lines and report on untested source code which affects the overall code coverage analysis.

TBpublish®

TBpublish captures the analysis and test results from the *LDRA tool suite* and publishes the results via an HTML index, into a self-contained directory for easy navigation and collaborative reference.

TBaudit®

TBaudit offers development and test managers an automatically generated, user-configurable Microsoft Word report which contains the results of the review and testing activities carried out by the development, QA and testing teams using the *LDRA tool suite*. *TBpublish* is a prerequisite for *TBaudit*.

Target Licence Package (TLP)

The *Target Licence Package (TLP)* provides the right to use and to receive support for the *LDRA tool suite* when it is used for target testing. The *TLP* allows the user to receive assistance from LDRA in configuring the *LDRA tool suite* to interface with a specified embedded tool chain.

Tool Qualification Support Pack (TQSP)

Tool qualification is becoming increasingly important if not a requirement in some safety- and security-critical markets. The *TQSP* supports clients through the process of qualifying the *LDRA tool suite* for use as a verification tool in their project environments.

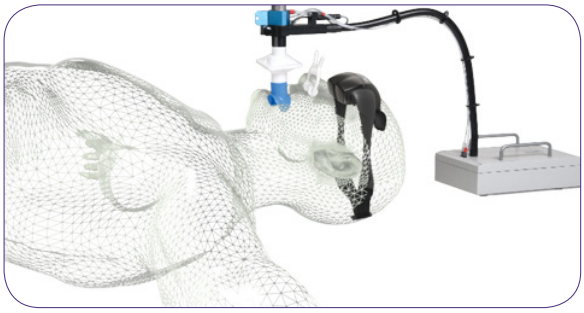
Client Testimonials

F-35 Lightning II

'LDRA has proven they will support us in any way to get the job done especially in meeting demanding milestones. They provided outstanding support for several F-35 teammates: Lockheed Martin (Fort Worth), BAE (Warton), Northrop Grumman (El Segundo), Seaweed, and Honeywell which directly contributed to a successful first flight of the AA-1 aircraft. We continue to work closely with LDRA to develop the needed automated process support to ensure that our software meets program cost, schedule, and quality targets.'

John H. Robb, Air Vehicle Software Senior Manager, LMCO

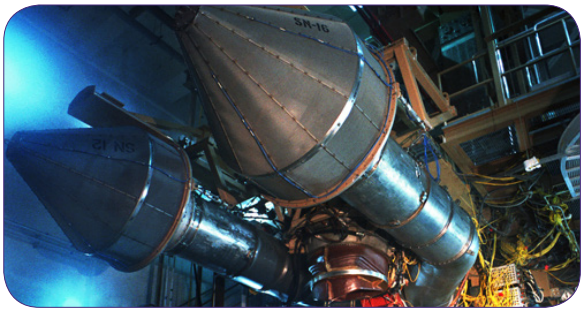
Synchro Module: Radiation therapy system for cancer treatment



'We configured the LDRA tool suite to meet the company's specific standard requirements as well as those of IEC 62304. The LDRA tool suite ensures that our products are compliant and that security is worth a premium for customers. Customers can be rest assured that any code we provide has been rigorously reviewed, achieves standards' compliance, and will be easier to maintain, reuse and add functionality to.'

Frédéric Rabouin, Embedded Software Developer, Esprit Lean

Pratt & Whitney F135 Engine for JSF



'We found that the graphical user interface was easy to work with and made developing a rapid, intuitive test process a lot easier than creating it manually. This saving was further increased through the repeatability of tests utilising the automated regression testing facilities. This automated solution made our job a lot easier. The LDRA tool suite resulted in a saving of £2 million.'

Tom Roberts, Engineering Manager, Embedded Software and Systems, Ultra Electronics Datel

Toyota's GT 86

'LDRA has the ability to work with limited target hardware which is important in the automotive sector in order to meet the demands for cost reduction and downsizing. We use the LDRA tool suite as a benchmark for other third-party and similar software platform products.'

Akihito Iwai, Project Manager DENSO Japan

Languages & Platforms

The *LDRA tool suite* is available for the following source code languages and host/target platforms.

Languages

Ada 83
Ada 95
C/C++
Java

Freescale Assemblers
Intel Assemblers
Texas Instruments Assemblers
ARM Assemblers

Languages shown in orange signify *TBrun* availability

Host Platforms

Windows 7/Vista/XP/2000/NT
Unix
Linux

Target Platforms

IDE:

Analog Devices
AONIX
ARM
Cosmic
Eclipse
Freescale
GNU
Green Hills
IAR
iSYSTEM
Keil
MPLAB X
QNX
TI
Renesas
TASKING
Wind River

Processor:

ARM
Freescale
Infineon
Intel
Microchip
MIPS
PowerPC
Renesas
TI

Note: This is not an exhaustive list as other languages and host / target platforms are available. Please contact LDRA for more information.

Certificate Number FM 26376

Designed by Young Greenwood Design (01260) 226541.

For more information or to receive a quote, contact us at salesdesk@logic.nl or by telephone: +31 (0)77 3078438

logic technology

Logic Technology B.V.
Van Horneplein 6
6019 BW Wessem
The Netherlands