W H I T E   P A P E R

Dr. Oscar Slotosch
Validas

Dr. Marcel Beemster
Senior Software Engineer
ACE Associated Computer Experts bv

TEXAS INSTRUMENTS

Model-Based Tool Qualification of the TI C/C++ ARM® Compiler

Introduction

The SafeTI™ Compiler Qualification Kit is a modern, model-based qualification kit that offers a high degree of flexibility during application of the kit. This paper will describe challenges posed by the safety standards and processes, as well as illustrate the benefits of model-based tool qualification.

The second part of the paper focuses on the Texas Instruments C/C++ ARM® compiler specific aspects. Its model combines the C software programming language conformance tests provided by the SuperTest qualification suite with specific TI test cases for the TI C/C++ compiler. Using examples, we illustrate how the contained qualification support tool assists during qualification of the compiler and provide a description of the generated documents.

1. Requirements for Tool Qualification

There are many standards for the development of safety-critical systems, such as the ISO 26262 [ISO26262], DO-178C / DO-330 [DO330] and IEC 61508 [61508]. These functional safety standards require analyzing all tools that are used within the development process of the software. This includes the integration and verification of the software. All these standards have a three-phase approach for using tools safely:
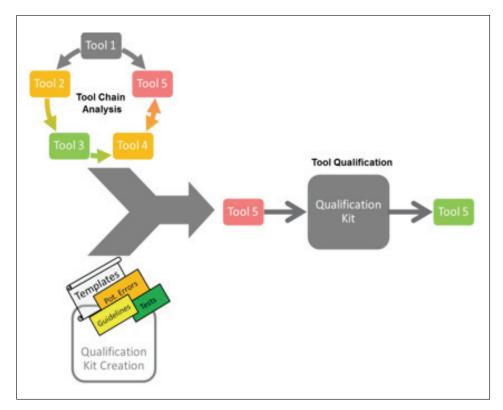
1. *Classification:* The tools are classified into categories that describe the confidence (certification credit) required in the development process of the system. The classification is based on the analysis of potential errors in the tool and their detection or prevention probabilities within the process. Note that the confidence categories for tools differ among the different standards: tool "confidence levels" in the ISO 26262, tool "criteria" in the DO-178C and tool "classes" in the IEC 61508. Tools that do not require confidence, since they have either no impact or a high-detection probability for all their potential errors in the process, can be used without qualification in the analyzed processes.

2. *Qualification:* Tools that require confidence in the analyzed processes have to be qualified. Qualification might be restricted to the identified use cases and to show the absence of critical errors. In ISO 26262, there are four possible qualification methods: increased confidence from use, process assessment, validation, and development according to a safety standard.

3. *Usage:* The tools must be used taking into consideration the known or found restrictions in the development process. Documentation must contain the constraints from the process that have been considered in the analysis phase and workarounds for all restrictions found during tool qualification.

2. Tool Qualification Processes and Roles

While the standards pose requirements on tool qualification, they do not describe the involved processes. The involved roles are the tool user (or a department responsible for the usage of tools) and the tool provider. We see the following three processes during tool qualification.

- *Tool chain analysis (tool user):* Determination of the required confidence in all used tools.
- *Creation of tool qualification kits (tool provider).*
- *Tool qualification (tool user and tool provider):* Application of qualification kits.

All processes can be supported with qualification service providers who can share their experiences with tool users and tool providers on this specific topic.

▲ **Figure 1** *– Tool Qualification Processes*

Figure 1 shows the tool qualification processes. The main process "Tool Qualification" (depicted on the right side) creates the required confidence needed for the tool by applying a qualification kit. The prerequisites of tool qualification are determined by the two required processes:

- Determination of the qualification needed during the tool chain analysis of all tools in the development process (depicted in the upper left part).
- Availability of a qualification kit that fits to the requirements (depicted in the lower left part).

Tool qualification requires a classification determine the qualification needs and a qualification kit that covers the needs. If the tool chain analysis classifies a tool without confidence requirements, the qualification process can be omitted. In all cases (even when no tool qualification is needed), the application of the tool requires guidelines (a tool safety manual) that specify the requirements determined during the classification of the tool to which the user must adhere for the tool to operate correctly (e.g., restricting the used features or requiring specific verification steps.)

The tool qualification process benefits from a comprehensive tool classification that provides detailed information on the used features and the applicable guidelines to detect or prevent potential tool errors. In this case, the tool user has to qualify only the "critical errors" of the tool. The critical errors are those errors that might occur during the application of the selected tool features and have no high detection or prevention probability in the applied process. A good qualification kit supports the tool user during the determination of the critical errors and supplies test cases that provide confidence in the absence of the critical errors.
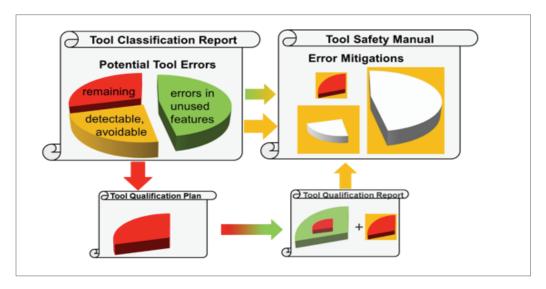
Of course the amount of critical errors can also be reduced without qualification by applying more mitigation measures. However, this requires introducing additional and sometimes expensive process steps. The goal of the application of a qualification kit is therefore to qualify as many features as possible with the fewest error mitigation restrictions on the tool application. On the other hand, the creation of qualification kits that test all features of the tool requires significant effort. Therefore, there is a prioritization of the test creation according to the mitigation measures that can be avoided by successfully executing the tests. Designing qualification kits with fixed amount of effort for an optimal set of selected tool features is a challenge that cannot be met without an appropriate model of the tool and an understanding of acceptable user restrictions. During the application of a qualification kit, the user can select which tool features he or she intends to use. For each feature, a "qualification state" characterizing the usability of the feature is identified:

- *Red:* The feature cannot be used, since there are neither tests nor mitigation measures for the potential errors of the feature.
- *Yellow:* The feature can be used with the restriction that a certain set of guidelines is applied to mitigate the potential errors of the feature.
- *Green:* The feature can be used without constraints, since there are tests available that provide evidence for the absence of the potential errors.

3. Model-Based Tool Qualification

The safety standards (ISO 26262, IEC 61508, DO-178/DO-330) require the user to analyze the tools used for the development of safety-critical products. The result of the analysis is a requirement on the reliability of the tool stated in the tool classification report.

The confidence is determined by an analysis of the use cases of the tool as used within the development process. If the tool has an impact on the safety of the product, all potential errors within the used features are analyzed to determine how they can be detected or avoided within the process. If there is no high probability for detecting or avoiding the errors, the tool has to be qualified to ensure the absence of these errors.

▲ **Figure 2** – *Tool Safety Manual Derivation*

The tool safety manual for a tool has to contain the mitigations against all potential tool errors that are considered during tool classification. The errors can be grouped into the three classes (see Figure 2):

1. *Potential errors in unused features* (green in Figure 2): Using these features is prohibited in the tool safety manual (upper right chart in Figure 2). The user has to work around these potential errors. The tool safety manual represents these restrictions by the white pies in that chart.

2. *Potential errors with mitigations* (yellow in Figure 2): These include potential errors for which detection and restriction mechanisms are described in the tool safety manual

3. *Remaining potential errors* (red in Figure 2): Demonstrating the absence of this category of potential errors is the goal of the tool qualification (tool qualification plan).

The tool qualification report (lower right chart in Figure 2) shows some specific examples of errors in the potential error classes. The qualification report contains proposed workarounds for these specific errors which information would be included as part of the safety tool manual (upper right chart in Figure 2).

The tool safety manual therefore must contain the following information:

- Allowed features and configurations of the tool.
- Requirements to apply checks and restrictions to mitigate potential tool errors:
  for potential errors that might occur in required features and that are not excluded by tool qualification.
- Workarounds for known bugs and errors found during qualification.
- Other information required by the standards to precisely identify the tool (e.g., version, configuration).

The tool qualification plan must identify potential errors of the TI C/C++ compiler which are not "detectable, avoidable" and show that they cannot occur. This is done by applying a validation suite in a systematic way that shows the absence of these potential errors that are not detectable, avoidable.

[1.] Note that the analysis of potential errors in unused functions is not required, but the features need to be identified.

Since the TI C/C++ compiler will be qualified using validation according to this qualification plan, the following documents are provided: (the documents that depend on the model are typed using italic font.)
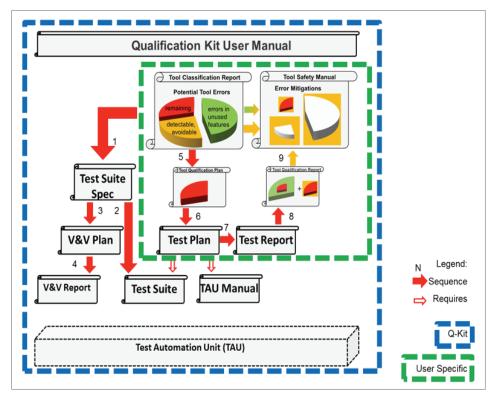
- *Test Plan:* (specifies the required test cases for execution.)
- *Test Report:* (contains the test results.)
- *Test Automation Unit Manual:* (contains instructions to execute the planned test cases correctly.)
- *Test suite validation and verification documents (plan and report):* confirms that the test plan. shows the absence of the potential errors, if the test plan passed successfully.

In the case where the model and the validation suite need to be extended and new test cases need to be produced and validated, the following documents are required or need to be extended:

- *Test specifications:* including a test strategy to show the absence of the potential errors.
- *Test suite* Validation and Verification plan and report.

The test suite validated against the implementation by performing a review and verification that the test suite runs correctly on the selected target. This quality process creates confidence in the effectiveness of the test suite. The V&V documents for the test suite are contained in the qualification kit to demonstrate the level of confidence to the user. If the test suite is extended, these documents will also be extended.

Figure 3 shows the relation between the documents and their variability (i.e., which are constant and which depend on the use case).
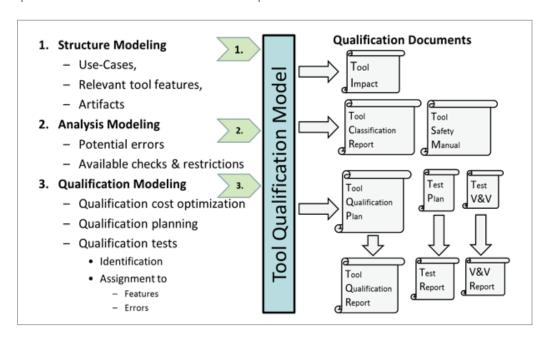
▲ *Figure 3 –* *Documentation Plan*

There are many documents in Figure 3 that are required and that need to be adapted depending on the user's process. The process is captured in the qualification model by selecting the required tool features and the executed mitigations. The use-case-specific parts in the user-specific documents are generated from the qualification support tool.

4. The Tool Qualification Model

The tool qualification model is designed to capture a complete tool chain, consisting of tools with features and use cases that are connected via artifacts which they exchange. It is based on a simple "Structure Model" that captures the dataflow within a tool chain. To determine the qualification needs, this model is enriched by a so-called "Analysis Model" that consists of potential errors and mitigation possibilities (checks and restrictions) that are assigned to features and use cases of the tool. The third corresponding model is the "Qualification Model". The qualification model includes test case descriptions and their assignment to potential errors. During the development of qualification kits, the qualification model can be used for a systematic derivation of tests to the potential errors. A more detailed description of this approach can be found in [WPJSZ12] or in [TCA].

The construction and the usage of the tool qualification model is depicted in Figure 4. While the logical sequence starts with the creation of the structure model and continues with the analysis and finishes with qualification modeling, the qualification kit already contains the analysis and the qualification model. These models are based on predefined structures which can be selected by the user. This simplifies the qualification process, since the user does not need to analyze the features and find appropriate tests. The generated qualification documents are those described in the previous section.

▲ **Figure 4 –** Tool Qualification Model

5. Modeling Tools

The model-based approach has many benefits as described in the previous sections. To work with the model, two kinds of tools are required:

1. One tool to edit the model (i.e., to add new elements).

2. A second tool to use the model for qualification.

In the model-based tool qualification approach, we use the qualification support tool to work with the model (see Section 5.1) and the Tool Chain Analyzer to edit the model (see Section 5.2).

5.1   Qualification Support Tool: Application of the Qualification Kit

The Qualification Support Tool guides the user through the qualification and installs all required artifacts for the qualification in a chosen target directory. The main purpose of the tool is to determine the qualification need based on the desired structure, select the tests to be executed and generate the qualification documents based on the user selections.

A tool requires qualification if it has potential errors that cannot be mitigated with a high probability. The potential errors depend on the features from the tool that are used. The error mitigation probability depends on the applied process, especially which error mitigations (checks and restrictions) are applied in the use case of the tool or within other tools.
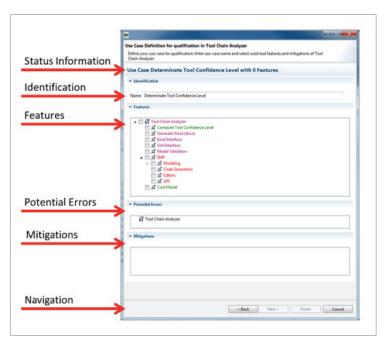
The qualification needs of the TI C/C++ compiler is determined by selecting the used features of the tool and by selecting the applied mitigations. While the list of features of the TI C/C++ compiler is constant, the list of mitigations depends on the selected features. For example, if no features are selected, then no mitigations are required.

Whether the tool is qualified successfully depends on:

• The commitment to the required mitigations (if required by the selected features). The mitigations are part of the tool safety manual that is generated.

• The successful execution of the test cases (if required by the selected features).

Accordingly, the determination of the qualification needs can show either that test cases have to be executed or that usage constraints to mitigate potential tool errors have to be integrated into the processes (or a combination of both situations). The specified test cases are documented in the tool qualification plan and the error mitigations are described in the tool safety manual.

The determination of the required "qualification need" is done in the so-called "Use Case Definition page" (see Figure 5).

▲ **Figure 5 –** *Use Case Definition in the Qualification Support Tool*

The page contains six parts:

- Status information on the top of the page with the use case name and the number of selected features and status information. If no tests are listed there the tool does not need qualification (if the errors are mitigated and the "Next" Button is enabled).
- Identification: Defines the name of the use case of the TI C/C++ Compiler.
- Features: Selects the features of the TI C/C++ Compiler that are used.
- Potential Errors: Shows the potential errors of the selected features. The errors have the following color encoding (similar to the qualification status in Section 2):
  - *Red:* The error cannot be qualified nor mitigated. In this case also the feature is red text and cannot be used.
  - *Pink:* (for easier read-ability, pink text was chosen rather than the color yellow used in Section 2): The error can be mitigated (but not qualified by tests). If pink errors are selected, the corresponding mitigations are highlighted in the Mitigation part.
    Pink errors have a stop symbol that is either green (if the mitigations are selected) or yellow if no mitigations are selected for this error.
  - *Green:* There is at least one test that shows the absence of this error.
- *Mitigations:* Provide a selection possibility for the applicable selections.
- *Navigation:* At the bottom there is a navigation line that allows the user to go to the next page ("Next >") or finish ("Finish") the qualification preparation. These buttons are only enabled if sufficient mitigations are selected.

The background color of the page shows the overall qualification state (see Section 2).

5.2   Tool Chain Analyzer: Extending the Qualification Kit

The Tool Chain Analyzer tool is not part of the qualification kit and therefore it is not included in the qualification kit. The Tool Chain Analyzer tool is available from www.validas.de/TCA.html. It can be used to analyze tool chains (see the first process in section 2) and to extend the qualification kit.

The tool qualification kit for the TI C/C++ Compiler can be extended by the user with the following elements:

- New feature elements for functions that are either new or currently not considered in the qualification kit.
- New safety guidelines to mitigate potential errors.
- New test elements for new test cases.

Every extension of the qualification kit consists of maximal four steps that need to be executed and are described in the user manual of the qualification kit:

1. Extending the model.
2. Validating the model.
3. Extending the test suite.
4. Validating the test suite.

Further details are described in the user manual of the qualification kit.

6.   SafeTI™ Compiler Qualification Kit

The SafeTI™ Compiler Qualification Kit is a model-based qualification kit as described in the previous sections. The TI C/C++ compiler has the following types of features that are modeled and can be used:

- C language as defined by ISO/IEC 9899:1990 (C++ ANSI/ISO/IEC 14882:1998).
- Component features like Parser, Optimizer, Code Generator, Assembler and Linker.
- Options features like Thumb instruction set, floating point hardware, or optimization level.

The SafeTI Compiler Qualification Kit allows the user to create specific personal use case(s) by selecting the features he or she intends to use.

The features are treated independently, even if they may impact each other. The test strategy is to analyze the features independently and provide tests and argumentations for each of the features selected. This is justified by analyzing the code coverage of the TI C/C++ compiler. If the test coverage during the qualification can be compared with the test coverage during the application, this gap in the argumentation can be closed by the following steps:

1. Measure the coverage of the TI C/C++ Compiler during the qualification tests.
2. Measure the coverage of the TI C/C++ Compiler during the application of the tool.
3. Compare the coverage in the source code of the TI C/C++ Compiler between the two scenarios and analyze the differences.

All functions in the compiler that are used in the application must be considered during qualification. In the case that the code coverage is sufficient, this is an indication that the tested features are sufficiently independent or the test cases already implicitly test the combinations.

6.1   The TI C/C++ Compiler

TI develops C/C++ compilers for many TI embedded processor platforms, including ARM. TI has been developing compilers for nearly 30 years and has extensive experience in compiling for high-performance architectures (VLIW, DSP, SIMD, µC, RISC, CISC). The TI compiler generates highly optimized code for each target, including state-of-the-art whole program optimization. Each compiler release undergoes an intensive and comprehensive validation and benchmarking process.

6.2   Contents of the SafeTI™ Compiler Qualification Kit

The SafeTI Compiler Qualification Kit contains the following elements:

- Documentation:
  - User Manual of the qualification kit.
  - User Manual of the Test Automation Unit (TAU).
  - Templates for documents.
- Tool Classification Report.
- Tool Qualification Plan/Report.
- Tool Safety Manual.
- Tests (see Section 6.2.4):
  - SuperTest qualification suite to test C language conformance of the compiler.
  - TI specific validation test cases.
- Test Automation Unit (see Section 6.2.3).
- An instrumented version of the compiler to measure the coverage.
- Scripts to install and collect code coverage measurement[2].
- Qualification Support Tool (including the qualification model, see Section 5.1).
- Up to 24 hours of qualification support from Validas AG.

6.2.1   The Qualification Model

The qualification model for the TI C/C++ Compiler contains a list of features that have been analyzed to derive potential errors. Every feature has different potential errors in the model. According to the qualification state (see Section 2) there are three kinds of features:

- *Testable:* A testable feature is a feature which has test cases assigned that can show the absence of all potential errors of the feature. For example, the C language feature "loops" is assigned to a comprehensive set of test cases within the SuperTest qualification suite that contain a variety of loops and multiple combinations. See Section 6.2.4.
- *Mitigatable:* A mitigatable error is an error that can be detected/avoided with a high probability. For example, if using optimization would cause an error, this could be detected by running the tests twice - once with and once without the optimization.

---

[2] Note that the coverage is measured within .cov files. The comparing script requires .xml coverage files that can be produced from the used tool BullsEyeCoverage which is not part of the qualification kit. Coverage comparison can be provided from Validas I if the use does not want to perform these measurements himself. For evaluation purposes there are evaluation licenses available from BullsEye.

- *Others:* Other features include those that are not qualifiable. (for example, using C++ Templates). Currently there are neither tests nor mitigations defined for these features, hence they cannot be used safely.

6.2.2   Qualification Support Tool

The qualification support tool has been configured with the model for the TI C/C+ compiler and includes a reference use case that can be changed or directly qualified. A description of the qualification support tool is in Section 5.1.

6.2.3   Test Automation Unit

The test automation unit automates the execution of the test cases on the target.

The SafeTI Compiler Qualification Kit provides a framework for running test cases generated by the Qualification Support Tool. The test framework will compile and flash the generated executables to an evaluation board and return the test results. The test framework is available for Windows and Linux and provides command line options. It utilizes Perl and Code Composer Studio v5 integrated development environment and executes test cases on a hardware emulation environment.

This kit has been tested on a Hercules TMS570LS3137 MCU hardware development board [TMDX570LS31HDK] using the on-board USB XDS100v2 emulator and an XDS560v2 emulator [XDS560v2]. Other Hercules™ ARM® Cortex™-R4 based MCUs and emulation options are supportable.

6.2.4   SuperTest and TI Tests

The SuperTest qualification suite from Associated Computer Experts, (ACE) is integrated as part of the SafeTI Compiler Qualification Kit. This qualification suite ensures strict compliance to C standards and was specifically derived from ACE's SuperTest compiler test and validation suite to support compiler qualification with respect to functional safety standards.

SuperTest provides the most thorough testing of C/C++ compilers. ACE, being developers of compilers and compiler developer tools, uses SuperTest as one of its primary tools for compiler testing and validation. The development of SuperTest started over three decades ago and it has been a product of ACE that is used by external compiler developers for almost as long. Based on internal use feedback, external feedback, new versions of the C standard, new analysis and optimization techniques and new compiler use cases, SuperTest has been continuously augmented and updated and has stayed at the fore-front of compiler testing and validation.

The tests in SuperTest check every part of the compiler, identifying issues in language conformance, as well as internal functions such as optimizations and transformations. Compiler coverage testing is also used to identify new areas for extension of SuperTest. The test programs in SuperTest are ordered according to the C standard for ease of access and to ensure that it has complete language coverage. For specific C features,

there are so called "Utility Sets" that contain lists of special test cases. As an example, the Utility Set for the "loops" feature contains the following test case specifying two nested loops with some non-trivial handling of pointers and address arithmetic:

```
i = b[0];
p = &a[i];
for( j = *p; i < AR_SIZE; ){
      q = &i;
      for( ; j >= i ; ){
            c[(*q)++] = *p;
      }
      p += (a[*q-1]==b[j++]);
}
CVAL_VERIFY( i == AR_SIZE );
CVAL_VERIFY( j == AR_SIZE );
CVAL_VERIFY( p == &a[AR_SIZE] );
```

As the example demonstrates, test cases are not necessarily "cleanly coded" according to proper coding style because a compiler must be able to handle those "dirty" cases too. In general, test cases are focused on a specific feature so that when an error occurs, the origin of the error is clear.

Note that the version of SuperTest included in the SafeTI Compiler Qualification Kit is specifically for use in the TI Kit for the qualification of the TI ARM C/C++ compiler.

TI Tests are included for all key features of the compiler including for example: ARM target specific code generation, EABI, optimization, link time optimization, debug information generation, pragmas/intrinsics, linking and more.

6.2.5   Generated Documentation

The following documents for the qualification of the C/C++ compiler are generated, based on the selections of the model (in the below list "<Target>" stands for the directory that has been selected for qualification and "<UseCase>" the name of the specified use case):

- *Tool Classification Report:* Contains the classification of the tool according to its usage. It is created in <Target>\Validation\Documentation\TCR.docx.
- *Tool Qualification Plan:* Plans the validation according to its usage. It is created in <Target>\Validation\Documentation\TQP.docx.
- *Test Plan:* The list of directories in the test suite to be executed for qualification. It is created in <Target>\Validation\TestExecution.txt.
- *Test Report:* The test results are created in the directory <Target>\Validation\TestReport\.
- *Tool Qualification Report:* The extension of the qualification plan by the validation results. Shall be placed into <Target>\Validation\Documentation\TQR.docx.
- *Tool Safety Manual:* Contains the tool usage guidelines according to classification and qualification. It is created in <Target>\Validation\Documentation\TSM.docx.
- *Tool Chain Model:* Describes the specified use case. It is created in <Target>\Validation\Documentation\<UseCase>.tca.

6.3 Standards Compliance

The requirements of the safety standards relevant for tool qualification and safety guidelines which are considered here are derived primarily from ISO 26262, IEC 61508 and DO-330. Many other safety standards (EN50128, DO-178-C, etc.) have similar requirements.

The relevant requirements are listed within the user manual of the qualification kit, which also includes demonstrations showing how the requirements are satisfied.

6.4 Projects Experiences

The qualification process has been applied several times. The qualification project lasts approximately six weeks from the definition of the configuration until the qualification reports have been created. This involves the planning, the execution of the tests, the code coverage measurements and their comparison, the analysis of the tests, the generation of reports and the review of the documents.

7. References

[61508]International Electrotechnical Commission, IEC 61508, Functional safety of electrical/electronic/ programmable electronic safety-related systmes, Edition 2.0, Apr 2010.

[BB98] P. Bishop and R. Bloomfield. A methodology for safety case development. In Safety-Critical Systems Symposium, Birmingham, UK, Feb. 1998. Available at www.adelard.com/resources/papers/pdf/sss98web.pdf.

[DO330] RTCA. DO-330: Software Tool Qualification Considerations 1st Edition 2011-12-13.

[HRMSP11] Joachim Hillebrand, Peter Reichenpfader, Irenka Mandic, Hannes Siegl, Christian Peer: Establishing confidence in the usage of software tools in context of ISO 26262 In SAFECOMP 2011.

[ISO26262] International Organization for Standardization. ISO 26262 Road Vehicles –Functional safety–. 1st Edition, 2011-11-15.

[Rus11] Rushby, J. 2011, 'New Challenges In Certification For Aircraft Software' EMSOFT'11, October 9–14, 2011, Taipei, Taiwan.

[TCA] Tool Chain Analyzer Tool, can be downloaded from www.validas.de/TCA.html

[WPJSZ12] Martin Wildmoser, Jan Philipps, Reinhard Jeschull, Oscar Slotosch Rafael Zalman. ISO 26262 - Tool Chain Analysis Reduces Tool Qualification Costs. In SAFECOMP 2012.

TI Worldwide Technical Support

Internet

TI Semiconductor Product Information Center Home Page
support.ti.com

TI E2E™ Community Home Page
e2e.ti.com

Product Information Centers

Americas	Phone	+1(512) 434-1560
Brazil	Phone	0800-891-2616
Mexico	Phone	0800-670-7544
	Fax	+1(972) 927-6377
	Internet/Email	support.ti.com/sc/pic/americas.htm

Europe, Middle East, and Africa
Phone

European Free Call	00800-ASK-TEXAS (00800 275 83927)
International	+49 (0) 8161 80 2121
Russian Support	+7 (4) 95 98 10 701

Note: The European Free Call (Toll Free) number is not active in all countries. If you have technical difficulty calling the free call number, please use the international number above.

Fax	+(49) (0) 8161 80 2045
Internet	www.ti.com/asktexas
Direct Email	asktexas@ti.com

Japan

Phone	Domestic	0120-92-3326
Fax	International	+81-3-3344-5317
	Domestic	0120-81-0036
Internet/Email	International	support.ti.com/sc/pic/japan.htm
	Domestic	www.tij.co.jp/pic

Asia
Phone

International	+91-80-41381665
Domestic	Toll-Free Number

Note: Toll-free numbers do not support mobile and IP phones.

Australia	1-800-999-084
China	800-820-8682
Hong Kong	800-96-5941
India	1-800-425-7888
Indonesia	001-803-8861-1006
Korea	080-551-2804
Malaysia	1-800-80-3973
New Zealand	0800-446-934
Philippines	1-800-765-7404
Singapore	800-886-1028
Taiwan	0800-006800
Thailand	001-800-886-0010

Fax	+8621-23073686
Email	tiasia@ti.com or ti-china@ti.com
Internet	support.ti.com/sc/pic/asia.htm

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

B090712

The platform bar and E2E are trademarks of Texas Instruments.

All other trademarks are the property of their respective owners.

TEXAS INSTRUMENTS

SPNY004

IMPORTANT NOTICE

Products		Applications	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		