

■ Model-Based Design
for IEC 61508 (Excerpts)

Applying Model-Based Design to Safety-Critical Systems

Model-Based Design with Simulink® products enables the use of executable specifications, automated verification and validation, and production code generation—an approach that reduces software development time, improves quality, and enhances innovation.

The benefits of Model-Based Design apply to a variety of embedded systems, including safety-critical applications. Code generated using MathWorks tools has been certified on safety-critical systems, and developers report 6.3 sigma quality and 50% productivity improvement [1, 2].

To apply Model-Based Design successfully to a safety-critical system requires extra consideration and rigor because these systems must adhere to tightly defined safety standards.

IEC 61508, “Functional safety of electrical / electronic / programmable electronic safety-related systems” [3], is one of the most widely used of these standards. IEC 61508 was published between 1998 and 2000, when most software was hand-coded. As a result, it does not cover advanced software development technologies, and must be mapped onto the processes and tools used in Model-Based Design.

IEC 61508 Software Safety Integrity Tables and Model-Based Design

IEC 61508 considers software failures to be the result of systematic faults introduced during development. Part 3 of the standard restricts software development and quality assurance processes for embedded software in order to reduce the number of faults introduced and increase the number of faults revealed.

The degree of rigor required for software development and quality assurance depends on the criticality of the embedded software, or its safety integrity level (SIL). Depending on the SIL required, IEC 61508-3 recommends more than 100 specific techniques and measures for software development and quality assurance for each lifecycle phase [4]. These techniques and measures are presented in 19 software safety integrity tables, where they are mapped to four safety integrity levels, SIL 1 to SIL 4.

About this Guide

A comprehensive paper, *IEC 61508 Conformant Model-Based Design with Simulink Products*, augments these tables with recommendations on how to apply Simulink and third-party products for Model-Based Design to the IEC 61508 measures and techniques [5, 6].

What follows is an extract from one of the augmented safety integrity tables in *IEC 61508 Conformant Model-Based Design with Simulink Products*.

To obtain a copy of *IEC 61508 Conformant Model-Based Design with Simulink Products*, visit <http://www.mathworks.com/support/solutions/data/1-32COJP.html>

EXCERPT 1

IEC 61508-3 A Tables

Guide to the Selection of Techniques and Measures

Table A.3: Software design and development: Support tools and programming language

Technique / Measure	SIL1	SIL2	SIL3	SIL4	Applicable Tools and Processes for Model-Based Design
1 Suitable programming language (➔ See also Table C.1)	++*	++	++	++	<p>Simulink® uses time-based block diagrams as a modeling language.</p> <p>A defined subset of the modeling language can be achieved by adhering to the MAAB Style Guides and/or organization specific modeling guidelines.</p> <p>The Stateflow® language can be restricted to Stateflow charts that implement pure Mealy or Moore semantics.</p> <p>The Simulink Block Data Type Support table¹ lists the blocks that can be used for code generation with Real-Time Workshop® Embedded Coder.</p> <p>Real-Time Workshop Embedded Coder can be configured to support a MISRA®-C compliant subset of the C language.</p> <p>The MISRA compliance information for Real-Time Workshop® Embedded Coder³ documents the compliance considerations for the individual MISRA-C rules.</p>

* ++ = Highly recommended for this SIL

Technique / Measure	SIL1	SIL2	SIL3	SIL4	Applicable Tools and Processes for Model-Based Design
					<p>PolySpace™ – MISRA C Checker facilitates MISRA-C compliance checking of code generated by Real-Time Workshop Embedded Coder as well as of hand-written code portions.</p> <p>Simulink Model Advisor can be used to check certain restrictions on the languages used.</p> <p>➤ See [7] for information about using built-in model settings and developing custom model scripts to check for MISRA-C.</p> <p>Configuration Sets can be customized to enforce specific settings of the tools for Model-Based Design, e.g. diagnostics and optimization settings.</p> <p>Third-party products, such as the TASKING®, compiler tool chain (for Infineon, ARM, and other devices), supported by Link for TASKING®, facilitate MISRA-C compliance checking of generated code.</p> <p>➤ See the Link for TASKING product demos²</p>

¹ To launch the table, open a MATLAB Command window, and type `showblockdatatypetable`.

² To view the demos, see www.mathworks.com/products/tasking/demos.html or open a MATLAB Command window and type `demo('links and targets')`.

³ See “Generating Code Within MISRA-C Guidelines” section of the Real-Time Workshop® Embedded Coder User’s Guide.

EXCERPT 2

Adapting the Tables for Individual Projects

The Applicable Tools and Processes for Model-Based Design column offers multiple ways to comply with a particular measure or technique. These can be adapted or tailored to meet individual project requirements.

The tailored software safety integrity tables are intended to be used within the compliance demonstration process. During system certification they provide evidence that the objectives of IEC 61508-3 were met.

To document the chosen measures and techniques and demonstrate IEC 61508 compliance, do the following:

- Remove the columns related to SIL levels that do not apply.
- Label the measures and techniques recommended (+) or highly recommended (++) for the selected SIL, as 'Used', 'Used to a limited degree,' or 'Not used' in the project.
- Identify the selected techniques and measures used in the project and adapt them as necessary.

The following table is an example of project-specific tailoring.

Technique / Measure	SIL3	Interpretation in this application
1 Suitable programming language (↗ See also Table C.1)	++	<p>Used:</p> <p>Simulink and Stateflow with subset, modeling standard and static checker are used as modeling languages:</p> <ul style="list-style-type: none">• The MAAB Style Guides are used as modeling standard.• The Stateflow language is restricted to Stateflow charts that implement pure Moore semantics.• The MAAB related modeling standards checks provided by Simulink Model Advisor are used to check compliance with the MAAB Style Guides. <p>C code generated by Real-Time Workshop Embedded Coder with subset, coding standard and static checker is used as implementation language:</p> <ul style="list-style-type: none">• MISRA-C:2004 is used as implementation language subset and coding standard.• PolySpace™ - MISRA C Checker is used to verify MISRA-C: 2004 compliance. All reported deviations are manually reviewed.
...		...

References

- [1] "Achieving Six Sigma Software Quality Using Automatic Code Generation." www.mathworks.com/industries/aerospace/miadc05/presentations/potter.pdf
- [2] "Alstom Generates Production Code for Safety-Critical Power Converter Control Systems." www.mathworks.com/products/rtwembedded/userstories.html?file=10591
- [3] "International Standard IEC 61508 Functional safety of electrical/electronic/ programmable electronic safety-related systems." First edition, 1998-2000
- [4] J. McDermid, T. Kelly, "Software in Safety Critical Systems: Achievement and Prediction." *Nuclear Future*, Volume 02, No. 03, 2006, pp. 140-145
- [5] M. Conrad, "Using Simulink and Real-Time Workshop Embedded Coder for Safety-Critical Automotive Applications." *Proc. Workshop Modellbasierte Entwicklung Eingebetteter Systeme III (MBEES'07)*, Dagstuhl, Germany, Jan. 2007, pp. 41-50
- [6] T. Erkkinen, M. Conrad, "Safety-Critical Software Development Using Automatic Production Code Generation." *SAE World Congress 2007, Detroit, MI, US, Apr 2007 (SAE technical paper #2007-01-1493)*
- [7] T. Erkkinen, D. Hachmeister, "Checking Code and Models in Production Environments." *MATLAB Digest*, July 2003

Resources

VISIT

www.mathworks.com

TECHNICAL SUPPORT

www.mathworks.com/support

ONLINE USER COMMUNITY

www.mathworks.com/matlabcentral

DEMOS

www.mathworks.com/demos

TRAINING SERVICES

www.mathworks.com/training

THIRD-PARTY PRODUCTS AND SERVICES

www.mathworks.com/connections

WORLDWIDE CONTACTS

www.mathworks.com/contact

E-MAIL

info@mathworks.com