Why functional safety in embedded systems?
Rafael Taubinger – Global FAE Manager
Agenda

• Motivation
• Functional safety?
• Relevant standards
• What is tool qualification and why do I need it?
• Available solution
• Using code analysis for certification efforts
• Summary
Motivation

• More human-machine interactions than ever
• Autonomous technology
• Maintain a safe state in case of:
  • Random or systematic failures
  • Environmental circumstances
  • Human error
  • Loss of power supply
  • ...
  “without” risks to human life
What is functional safety?

Definitions from IEC (http://www.iec.ch/functionalsafety/explained/)

- **Safety**
  Freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment

- **Functional safety**
  The *detection* of a potentially dangerous condition resulting in the *activation* of a protective or corrective device or mechanism to *prevent* hazardous events arising or providing *mitigation* to reduce the consequence of the hazardous event
Relevant standards for software in a Functional Safety context

- IEC 61508
  Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems

- IEC 62304
  Medical device software – Software life cycle processes

- EN 50128
  Railway applications - Communication, signaling and processing systems - Software for railway control and protection systems

- ISO 26262
  Road vehicles – Functional safety

- (DO178C)
  Software Considerations in Airborne Systems and Equipment Certification
How am I affected by these standards?

- All these Functional Safety standards pose requirements on the development process!
  - Requirements gathering, implementation, testing, verification and validation of relevant safety functionality, programming language selection, etc, etc, etc...

- All these Functional Safety standards, independent from the Safety Integrity Level pose requirements on how to select development tools!
Benefits by following the standards

- Reduce liability risks associated with your application
- Reduce odds of product recall
- Reduce number of firmware updates
- Ensure compliance with international standards and requirements
- Protects your company’s reputation
- Also protects your company’s bottom line
Tool certification is important

- It means that your development tool has gone through a rigorous qualification process to ensure that it produces reliable and repeatable results when compiling your code.

- Additionally, it means:
  - Development processes are in place to manage how the tool works with specific requirements put forth by different functional safety standards
  - There are test and quality measures of the tool show validation of compliance with different language standards
Ensuring integrity

• Certifications make you consider carefully your device’s design
  • Defines failure modes and how to recover
  • Makes you think about other failure modes for new device types
• You must outline carefully these modes for review by a certification-granting entity to show you understand the risks and have them mitigated
How can I cope with tools validation and justification?

Essentially two choices:

• Do your own validation
  – Time and resource consuming
  – Compiler validation is outside the core competence for the absolute majority of tools users (If done fully in accordance with functional safety best practices...)
  – Project specific!
  Projects must be very similar to be able to reuse previous validation:
    – Hardware
    – Functional Safety requirements
    – Tool(s) options
    – Project staffing, etc...
How can I cope with tools validation and justification?

Essentially two choices:
• Rely on third-party validation/qualification evidence
  – Certification by e.g. TÜV SÜD
  – (Or testing by independent testing house)
Solutions for safety-critical applications

Certified toolchain
• A special functional safety edition of IAR Embedded Workbench

Simplified validation
• Functional Safety certificate from TÜV SÜD
• Safety report from TÜV SÜD
• Safety guide

Guaranteed support through the product life cycle
• Prioritized support
• Validated service packs
• Regular reports of known problems

Validated according to:
IEC 61508
ISO 26262
EN 50128 (Arm and RH850)
IEC 62304 (RX)
Certifications

IAR Embedded Workbench for RL78 V3.10.2
IAR Embedded Workbench for Arm V8.22.3
IAR Embedded Workbench for RX V3.10.5
IAR Embedded Workbench for RH850 V1.40.3

Certified for safety-related software development for each Safety Integrity Level (SIL) according to IEC 61508 and each Automotive Safety Integrity Level (ASIL) of ISO 26262 without further tool qualification.

The tools are also certified according to the European railway standard EN 50128 and according to IEC 62304 for medical device software.

The certification validates the quality of IAR Systems’ entire development processes, as well as the delivered software.
Simplified validation

- Functional safety certificate from TÜV SÜD
- Safety report from TÜV SÜD
- Safety Guide
  - Complement to the IAR Embedded Workbench user guides
  - Highlights issues to be considered when using the build toolchain for projects with functional-safety requirements
  - Includes system considerations, implementation and coding considerations, etc.
Validated product versions

- For a certified product, a new certified version is released approximately every 12-18 months.
- A certified version is considered a "frozen" version, on which bug fixes are applied in terms of validated service packs.
- No new product features are added to a certified version or the corresponding service packs.
Support and updates

Functional Safety Support and Update Agreement (SUA)

• Guaranteed support for the sold version for the longevity of the contract
• Prioritized technical support
• Validated service packs
• Regular reports of known deviations and problems
• Included for the first year

Extensive technical support when and where you need it provided by support offices worldwide
Ensuring safety

• Certifications are easier to achieve when you can prove that your code conforms to a coding standard (such as MISRA).
• Testing reports show that the overall number of defects in the software is low, despite many hours of testing and proves maturity of your development organization.
• Code analysis also shows that your results are repeatable because you have a process in place to find and fix defects.
Integrated analysis tools

We enable developers to take full control of their development and gain efficient, adaptable workflows delivering dependable products.
Take full control of your development

Implement your design in code

Build and debug the application

Let C-RUN analyze your project

Let C-STAT analyze your code

Review potential issues

Investigate runtime errors

Requirements

Design

Implementation

Verification

Maintenance
C-STAT static analysis

Complete static analysis tool fully integrated in IAR Embedded Workbench

- Intuitive and easy-to-use settings with flexible rule selection
- Support for export/import of selected checks
- Support for command line execution
- Extensive and detailed documentation
- List of messages and data base file available
- Includes ~250 checks mapping to hundreds of issues covered by CWE and CERT C/C++

CWE (the Common Weakness Enumeration): http://cwe.mitre.org/
CERT (Computer Emergency Response Team): http://www.cert.org/
C-RUN runtime analysis

Complete runtime analysis tool fully integrated in IAR Embedded Workbench for Arm and RX

- Find actual errors at runtime
- Bounds checking to ensure accesses to arrays and other objects are within boundaries
- Arithmetic checking
- Heap and memory leaks checking

- Intuitive and easy-to-use settings with flexible rule selection
- Code correlation and graphical feedback in editor
- Comprehensive and detailed feedback
- Very efficient instrumentation of compiled code
Summary

• Tools certified for functional safety
  – are exceedingly well-tested.
  – help you to focus on just your application.
  – help speed the path to your certification.

• Code analysis tools help prove your design’s safety and integrity.