MISRA Compliance for XEN project

Dr David Ward

Senior Technical Manager
Functional Safety

November 2020

Agenda

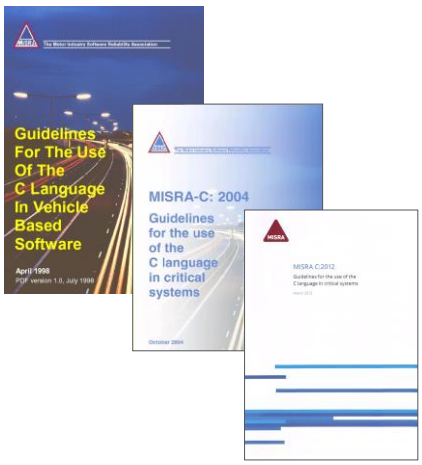
- Brief history of MISRA C
- What does it mean to “comply” with MISRA C?
- Compliance to language subsets – MISRA Compliance:2016 and :2020

A brief history of MISRA C

November 2020

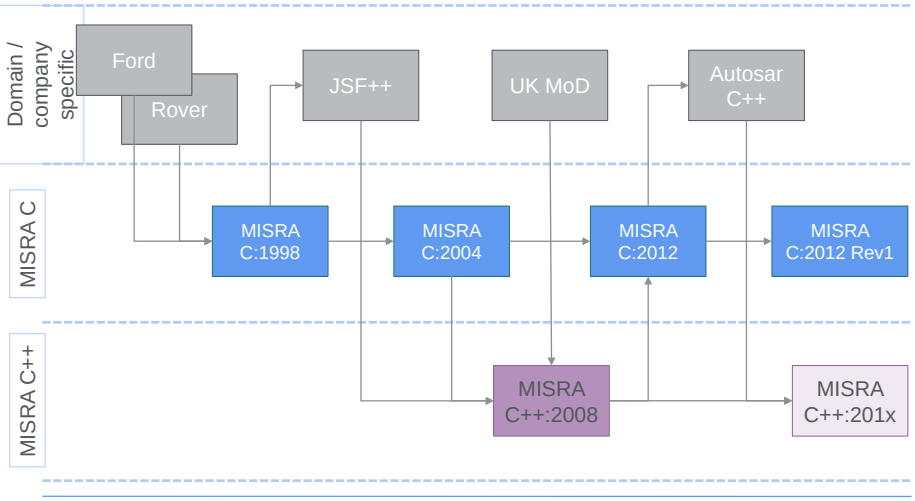
A brief history of MISRA C

- MISRA C:1998
  - Developed by Ford and Land Rover pooling effort from their own in-house C guidelines
  - Targeting automotive need for “a restricted subset of a standardized structured language” in MISRA 1994 “Development guidelines for vehicle based software”
- MISRA C:2004
  - Reworked based on user experiences including wider industry use
  - Exemplar Suite
- MISRA C:2012



November 2020

History of MISRA C/C++ guidelines



November 2020

Guideline types in MISRA C:2012

- Rules
  - Provide a clear and unambiguous specification
  - Compliance can be checked by performing static analysis on the code
  - Deal with issues related to C code
- Directives
  - Specification may be open to different interpretations
  - Static analysis may be of some assistance when checking compliance but other checks are likely to be required
  - May deal with non-code issues such as
    - Software development process requirements
    - Software design guidance

November 2020

Guideline types: examples

Directives	
Dir 4.2:	All usage of assembly language should be documented
Dir 4.11:	The validity of values passed to library functions shall be checked

Rules	
Rule 8.2:	Function types shall be in prototype form with named parameters
Rule 11.1:	Conversions shall not be performed between a pointer to a function and any other type

Guideline categories in MISRA C:2012

- Mandatory
  - Must always be followed and may never be deviated
- Required
  - Must be followed unless a formal deviation is in place
- Advisory
  - An organization, project or user may choose not to comply
  - No formal deviation is required although documentation is recommended nonetheless
- Readability
  - Applicable only to automatically-generated code
  - Compliance only required if code is to be read by humans

Guideline categories: examples

Mandatory	
Rule 9.1:	The value of an object with automatic storage duration shall not be read before it has been set

Required	
Rule 14.1:	A loop counter shall not have essentially floating type

Advisory	
Rule 12.3:	The comma operator should not be used

Readability (classification applicable to auto-generated code)	
Rule 7.3:	The lowercase character “l” shall not be used in a literal suffix

November 2020

Rule decidability in MISRA C:2012

- A rule is “decidable” if it is theoretically possible for a static analysis tool to identify **every** non-compliance in **every** program; otherwise it is “undecidable”

Decidable	
Rule 8.6:	An identifier with external linkage shall have exactly one external definition
Rule 20.8:	The controlling expression of a *#if* or *#elif* preprocessing directive shall evaluate to 0 or 1

Undecidable	
Rule 2.1:	A project shall not contain *unreachable* code
Rule 14.3:	Controlling expressions shall not be invariant

November 2020

Rule analysis scopes in MISRA C:2012

- A rule has “Single Translation Unit” scope if it is possible to check the entire program for compliance one unit at a time; otherwise it has “System” scope

Single Translation Unit	
Rule 9.2:	The initializer for an aggregate or union shall be enclosed in braces
Rule 20.4:	A macro shall not be defined with the same name as a keyword

System	
Rule 8.5:	An external object or function shall be declared once in one and only one file
Rule 17.2:	Functions shall not call themselves, either directly or indirectly

- ISO 26262 context: changes in Edition 2 to software verification activities

November 2020

What does it mean to “comply” with MISRA C?

November 2020

What does it mean to comply with MISRA C?

- Frequently as an assessor my experience is
  - "We've selected a tool that claims to check MISRA C"
  - "We've not got any unresolved tool messages"
  - So we comply … don't we??

November 2020

What does it mean to comply with MISRA C?

- The reality is
  - Not all guidelines are enforceable through static analysis
    - MISRA C contains 17 directives and 156 rules
      - Of those rules only 119 are "decidable"
  - Not all tools enforce all guidelines
  - It is necessary to understand how well tools enforce guidelines – vendor claims e.g. "we enforce 143 rules" need to be verified – "how good" is the checking? Remember not all of those 143 rules are "decidable"
    - Code examples are available that can be used to evaluate a tool but do not represent a conformance suite
    - The tool configuration is also significant
  - MISRA C does not have any sensitivity to e.g. ASIL in ISO 26262
  - Some guidelines may not be achievable in a particular project or context
  - Tools can be configured to suppress messages in various ways

November 2020

What does it mean to comply with MISRA C?

- As a minimum an assessor should expect to see …
  - That there is a plan for which guidelines are enforced and how
  - That tools have been selected considering their capability to enforce decidable rules (and support evaluation of other guidelines)
  - That any deviations are documented and formally approved
    - Including that any use of message suppression in tools is linked to a documented deviation

November 2020

The role of deviations

- There are cases where it is necessary to deviate from MISRA C requirements for example in I/O code
- A formalized deviation process is required to capture and approve necessary deviations
  - At a project level – where a deviation is required for a specific class of circumstances
  - As a specific deviation – for a single instance in a single source file, where a deviation is unavoidable, for example for a specific timing reason
- However deviations should not be an excuse for avoiding writing compliant code

November 2020

MISRA Compliance:2016
A more rigorous approach to deviations

- A framework for claiming compliance with MISRA coding guidelines
- Includes guidance on a robust and structured process for the use of deviations
- Includes a mechanism for establishing pre-approved "permits" to help streamline the deviation process
- Supersedes the compliance, deviation and process requirements previously published in various MISRA coding guidelines (e.g. Section 5 of MISRA C)
- MISRA Compliance:2020 makes its application mandatory
- MISRA C:2004 permits
  - Contains a number of deviation permits covering commonly-encountered use cases for use with the MISRA C:2004 guidelines
  - Intended for use alongside MISRA Compliance:2016
  - These are **not** "carte blanche" deviations …

November 2020

MISRA Compliance:2016
The context

- The development contract
  - Decision to apply MISRA Guidelines should be applied at the start of the project
  - ISO 26262 context: Development Interface Agreement (DIA)
- The software development process
  - Mature and effective software development process assumed e.g. according to ISO 26262
- New guideline classifications in MISRA C:2012 may impact approach to deviations
  - Directives vs Rules
  - Analysis scope
  - Decidability (this is a particularly important area)

November 2020

MISRA Compliance:2016
Key work products

- Inputs
  - MISRA C:2012 (or …)
  - Guideline enforcement plan
    - Including information associated with tool use (⇒ ISO 26262 Part 8 Clause 11) and any manual processes
  - Guideline re-categorization plan
  - Permits
- Outputs
  - Guideline compliance summary
  - Approved deviations
  - Permits

November 2020

Guideline enforcement plan (sample excerpt)

Guideline	Compilers		Analysis tools		Manual review
	A	B	C	D	
Dir 1.1					Procedure x
Dir 2.1	No errors	No errors			
…					
Rule 4.1			Message 42		
Rule 4.2				Warning 178	
Rule 5.1	Warning 123				
…					
Rule 12.1				Message 26	
Rule 12.2			Message 262		Procedure y
Rule 12.3			Message 508		
Rule 12.4				Message 61	

November 2020

Guideline re-categorization plan

- The organizations involved in the development can re-categorize MISRA Guidelines depending on project and context needs
  - Some *Required* Guidelines could be made *Mandatory*
  - Some *Advisory* Guidelines could be made *Mandatory*, *Required* or *Disapplied*
- However
  - *Mandatory* Guidelines cannot be re-categorized
  - A *Required* Guideline cannot be downgraded to *Advisory* or *Disapplied*
- ISO 26262 Part 6 context
  - See Note to clause 5.4.3 “Existing coding guidelines and modelling guidelines can be modified for a specific item development”
    - ASIL dependency and rationale for methods could be an input to decisions
  - See also work product 5.5.1 “Documentation of the software development environment”

November 2020

Example re-categorization plan

Guideline	MISRA category	Revised* category
Dir 1.1	Required	Mandatory
Dir 2.1	Required	Required
…		
Rule 4.1	Required	Required
Rule 4.2	Advisory	Disapplied
Rule 5.1	Required	Mandatory
…		
Rule 12.1	Advisory	Mandatory
Rule 12.2	Required	Required
Rule 12.3	Advisory	Advisory
Rule 12.4	Advisory	Required

* Could be company-wide or project specific e.g. specified in a DIA

November 2020

Investigating tool messages

- General principle is that all messages reported by tools must be addressed
- Messages reported by tools or compilers could be
  - Correct diagnosis of a violation
    - Correction (ideally) or deviation needed
  - Diagnosis of a possible violation
    - Investigate and document why the code is correct despite the diagnosis
  - False diagnosis of a violation
    - Investigate and explain why diagnosis is incorrect
    - Ideally with support of tool developer ⇒ ISO 26262 context: Confidence in the use of software tools
  - Diagnosis of another issue that is not a violation
    - Rationale needed for why the message can be ignored

The (correct) role of deviations

- Claims of compliance with MISRA C in the presence of unavoidable rule violations must be authorized through a defined process and supported by deviation records
- Deviation record contains
  - The guideline(s) being violated
  - Description of circumstances where violation is acceptable
  - The reason why the deviation is required
  - Background information to explain context and language issues
  - Requirements to include risk assessment procedures and precautions to be observed
- Context of deviation – single instance or common use case
- The role of deviation permits (see later)

When is a deviation not permitted?

- When it is simply for the convenience of the developer
- Where there is a reasonable alternative coding strategy that would remove the violation
- When impact on other Guidelines has not been considered
- When there is not a suitable process in place for managing deviations
  - Including sign-off by a designated technical authority

Use cases for deviations

- Deviations should only be approved if justified on the basis of one or more of
  - Code quality
  - Access to hardware
  - Integration of adopted code
  - Non-compliant adopted code

Adopted code

- Term used to refer to code that has not been developed within the scope of the current project
  - ISO 26262 context: a software component that is “reused”, see Part 6 Clause 7.4.7
- Sources of adopted code include
  - The Standard Library
  - Device drivers
  - Middleware
  - Third party libraries
  - Automatically generated code
  - Legacy code

November 2020

Adopted code

- Key questions associated with adopted code (and that must both be addressed)
  - Has the code been developed to an adequate level of safety and security?
    - ISO 26262: Part 8 Clause 12 “Qualification of software components”
  - Has the code been developed to be compliant with MISRA Guidelines?
- Particular procedures may need to be applied concerning
  - Guidelines with system-wide analysis scope
  - Adopted binary (object) code
  - Adopted source code
    - Specific Guideline Re-categorization Plans may be needed (remember a Mandatory guideline cannot be re-categorized)
  - Adopted header files
  - The Standard Library

November 2020

Claiming compliance

- Process aspects
  - Staff competence
  - Management process
- Evidence of compliance
  - Guideline compliance summary
- Permitted levels of claimed compliance

MISRA category	Compliance levels that may be claimed			
Mandatory	Compliant			
Required	Compliant	Deviations		
Advisory	Compliant	Deviations	Violations	Disapplied

November 2020

Example Guideline Compliance Summary

Guideline	MISRA category	Compliance
Dir 1.1	Required	Compliant
Dir 2.1	Required	Deviations
…		
Rule 4.1	Required	Deviations
Rule 4.2	Advisory	Disapplied
Rule 5.1	Required	Compliant
…		
Rule 12.1	Advisory	Compliant
Rule 12.2	Required	Deviations
Rule 12.3	Advisory	Violations
Rule 12.4	Advisory	Deviations

November 2020

Project delivery

- Supporting information for a claim of compliance
  - Guideline enforcement plan; the following may be requested by the acquirer (“customer” in ISO 26262 DIA)
    - Supporting information on tools and processes
    - Evidence of tool checks
    - Evidence of detected violations
  - Guideline compliance summary
  - Details of any approved deviation permits that have been used
  - Deviation records for all *Required* guidelines
    - Whether *Required* in MISRA C itself or re-categorized

Example deviation record

Project	F10_BCM		
Deviation ID	D_00102	Status	Approved
Permit	Permit / Example / C:2012 / R.10.6.A.1		
Rule 10.6	The value of a composite expression shall not be assigned to an object with wider essential type		
Use-case	The value of a composite expression is assigned to an object of wider essential type to avoid sub-optimal compiler code generation		
Reason	Code Quality (Time behaviour)	Scope	Project
Tracing tags	D_00102_1 to D_00102_10		

Raised by	E C Unwin	Approved by	D B Stevens
	Signature		*Signature*
Position	Software Team Leader	Position	Engineering Director
Date	14-Mar-2015	Date	12-Apr-2015

See Appendix A of MISRA Compliance:2016 for full details including supporting rationales

Deviation permits

- A deviation permit defines a use case under which a violation may be justified and specifies the documentation and process requirements which must be supplied in the deviation record
- The motivation for permits is to capture common use cases and provide common information that can be the basis of a deviation record
- **Note: a deviation record is still required, a permit is not a pre-defined deviation!**

MISRA C:2004 Permits

- A set of pre-written Permits (which can form the basis of Deviations) for certain MISRA C:**2004** rules
  - Rule 1.1 – code compliance with ISO/IEC C90 (mostly around language extensions in later language versions)
  - Rule 5.1 – length of external identifiers
  - Rule 6.4 – bit-fields of type other than *unsigned int* or *signed int*
  - Rule 8.5 – in-line function defined in header file
  - Rule 12.7 – bitwise operators applied to signed operands
  - Rule 13.7 – Boolean operations with invariant results
  - Rule 14.1 – unreachable code
  - Rule 17.1 – pointer operations performed on a memory region that is not an array object
  - Rule 17.4 – array indexing applied to a pointer addressing a specific hardware memory space

Mapping MISRA Compliance ↔ ISO 26262

MISRA Compliance topic	ISO 26262
The development contract	Part 8 Clause 5 “Interfaces in distributed development”
The software development process	Part 6 plus supporting and management processes
Guideline enforcement plan Guideline reclassification plan	Safety plan (software specific content) Part 6 requirements for coding guidelines (including adaptation for specific item / element development) Part 8 Clause 11 “Confidence in the use of software tools”
Guideline compliance summary Deviations	Part 6 requirements for coding guidelines (including adaptation for specific item / element development) Safety case
Adopted code	Part 8 Clause 12 “Qualification of software components”
Competence and management process	ISO 26262 Part 2 Clauses 5 and 6

Future directions

- MISRA Compliance:2016 provides an essential rigorous framework for managing deviations from coding guidelines
  - But can be aligned to and implemented within the framework of a standard such as ISO 26262
- MISRA Compliance:2020 makes it application mandatory
  - Effectively replaces the existing guidelines on deviations and claiming compliance (e.g. MISRA C:2012 Sections 5.4 and 5.5)
- Under discussion: whether approach to MISRA compliance in automatically-generated code becomes a set of “permits”
  - MISRA C:2004 – separate document provided: MISRA AC AGC
  - MISRA C:2012 – reclassification provided in Appendix E
- Permits also under development for MISRA C:2012

Conclusions and recommendations for XEN

- Establish a guideline enforcement plan
  - Carefully consider tool capability and coverage of rules
- Establish a guideline recategorization plan
- Developed permits for identified XEN use cases
- Ensure all deviations have an approved deviation record
  - “Permits” are not an automatic approval of deviations but streamline the process

November 2020

Contact details

Dr David Ward
MA (Cantab), PhD, CEng, CPhys, MInstP, MIEEE, MSAE
Senior Technical Manager, Functional Safety

Direct T: +44 24 7635 5430
E: david.ward@horiba-mira.com

HORIBA MIRA Ltd
Watling Street,
Nuneaton, Warwickshire,
CV10 0TU, UK

T: +44 (0)24 7635 5000
F: +44 (0)24 7635 8000

www.horiba-mira.com

November 2020