TEXAS INSTRUMENTS THE WORLD LEADER IN DSP AND ANALOG

STARTERWARE AND BSP
MISRA C COMPLIANCE POLICY

Version 1.7

TABLE OF CONTENTS

1. Scope

The scope of this document is to identify the MISRA-C compliance of the BSP and Starterware package.
It is assumed that the readers of this document are familiar with MISRA-C:2004 and Klocwork (Static Code Analyzer and MISRA-C rule checker tool).

2. Purpose

This document identifies the formal set of policies and formal set of deviations that will be used across the BSP and Starterware.

3. Rule Checking Tool

Following tool shall be used for checking the MISRA-C:2004 compliance.

Klocwork Version 10.0

3.1. Klocwork Limitation and Known Issue

List of known limitation in the 10.0 version of Klocwork is at
http://www.klocwork.com/products/documentation/current/Release_Notes

4. Deviations

The Starterware and BSP related waiver reason is part of the Klocwork report present in *docs* folder of Starterware and BSP packages.
Following are the list of waiver types that are moved to different state's given below in the Klocwork server.

Waiver Type	State used in STW/BSP
Case-by-Case Waiver	"Ignore"
Blanket Waiver	"No state. Not present in report"
KW issue (false positive)	"Not a problem"
No plan to fix	"Defer"

Following are the list of deviations that shall be taken:

1. All examples of Starterware and BSP and FATLIB module of Starterware will not be MISRA-C compliant. All the violations will be moved to "Defer" state in the tool
2. Dependent and other Third Party components

   Dependent and other Third party components like BIOS, XDC, tool generated files etc., shall not be MISRA-C compliant.

Rule	Klocwork Code	Category	Description
8.1	MISRA.FUNC.NOPROT.CALL	Required	Function is called but has no prototype
11.1	MISRA.CAST.FUNC_PTR	Required	Cast between a function pointer and a non-integral type
12.7	MISRA.BITS.NOT_UNSIGNED	Required	Operand of bitwise operation is not unsigned integer

3. HW files

   Auto generated register layer files (HW files) shall not be MISRA-C compliant

4. No supported in the rule checker tool

   Following rules are not checked because of “no support” in the checker tool

   - Rule# 1.1 - All code shall conform to ISO 9899:1990 "Programming languages -C"
   - Rule# 1.2 - No reliance shall be placed on undefined or unspecified behavior
   - Rule# 1.3 - Multiple compilers and/or languages shall only be used if there is a common defined interface standard for object code to which the languages/compilers/assemblers conform
   - Rule# 1.4 - The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers
   - Rule# 1.5 - Floating point implementation should comply with a defined floating-point standard
   - Rule# 2.4 - Sections of code should not be "commented out"
   - Rule# 3.1 - All usage of implementation-defined behavior shall be documented
   - Rule# 3.2 - The character set and the corresponding encoding shall be documented
   - Rule# 3.3 - The implementation of integer division in the chosen compiler should be determined
   - Rule# 3.4 – All uses of #pragma directive shall be documented and explained
   - Rule# 3.5 - The implementation-defined behavior and packing of bit fields shall be documented if being relied upon
   - Rule# 3.6 - All libraries used in production code shall be written to comply with the provisions of this document, and shall have been subject to appropriate validation
   - Rule# 8.8 – All external object or function shall be declared in one and only file
   - Rule# 8.9

- Rule# 8.10 – All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required
- Rule# 16.10 – If a function returns error information, then that error information shall be tested
- Rule# 17.2 – Pointer subtraction shall only be applied to pointers that address elements of the same array
- Rule# 17.3 – >, >=, <, <= shall not be applied to pointer types except where they point to the same array
- Rule# 18.3 - An area of memory shall not be reused for unrelated purposes
- Rule# 19.16 - Preprocessing directives shall be syntactically meaningful even when excluded by the preprocessor
- Rule# 20.3 - The validity of values passed to library functions shall be checked
- Rule# 21.1 - Minimization of runtime failures shall be ensured by the use of at least one of:
    - static analysis tools/techniques
    - dynamic analysis tools/techniques
    - explicit coding of checks to handle run-time faults"

5. Known Issue in the rule checker tool

   BSP and Starterware package shall not be checked for compliance, for rules which are associated with known issue in Klocwork.

   Note: These shall be considered as "Case-by-Case" deviations (across BSP and Starterware package)

6. Following rules shall be blanket deviations and are not checked by the tool and doesn't appear in the report

Rule	Klocwork Code	Category	Description
5.1	MISRA.IDENT.LONG, MISRA.DEFINE.LONGNAME	Required	Identifier is longer than 31 characters
5.6	MISRA.TYPE.NAMECLASH	Advisory	No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure and union member names
5.7	MISRA.VAR.UNIQUE	Advisory	No identifier name should be reused
13.2	MISRA.ZERO_EQ.IMPLICIT	Advisory	Non-boolean expression is implicitly tested against zero

7. Following rules shall be flagged and deviations recorded on a per-instance basis. These will be moved to “Ignore” state in the tool.

Rule	Klocwork Code	Category	Description
6.3	MISRA.BUILTIN_NUMERIC	Advisory	Built-in numeric type is used
8.5	MISRA.ONEDEFRULE.FUNC	Required	Global function definition in a header file
8.7	MISRA.VAR.MIN.VIS	Required	Name visibility is too wide
11.3	MISRA.CAST.PTR_TO_INT	Advisory	A cast shall not be performed between a pointer type and an integral type
11.4	MISRA.CAST.PTR	Advisory	Cast between a pointer to object type and a different pointer to object type
14.1	UNREACH.GEN, UNREACH.RETURN	Required	Unreachable code
16.1	MISRA.FUNC.VARARG	Required	Function with variable number of arguments
17.1, 17.4	MISRA.PTR.ARITH	Required	Pointer is used in arithmetic or array index expression
18.4	MISRA.UNION	Required	Union is used
19.7	MISRA.DEFINE.FUNC	Advisory	Function-like macro definition
19.12	MISRA.DEFINE.SHARP.MANY	Required	Several # or ## operators in a macro definition
19.13	MISRA.DEFINE.SHARP	Advisory	# or ## operator in a macro definition
19.15	MISRA.INCGUARD	Required	Include guard is not provided
20.1	MISRA.DEFINE.WRONGNAME , MISRA.DEFINE. WRONGNAME. UNDERSCORE	Required	Usage of a reserved name for naming a macro. **Note:** This is there only in header file protection macro
20.9	MISRA.STDLIB.STDIO	Required	Use of input/output library stdio.h in production code
KW Critical Issue	NPD.FUNC.MUST NPD.FUNC.MIGHT	Required	Previously checked null pointer is dereferenced. Previously checked null pointer may be dereferenced.

KW Critical Issue	NPD.CONST.CALL	Required	Null-pointer constant value may be dereferenced through a function call
KW Error Issue	INFINITE_LOOP.LOCAL	Required	Infinite loop with local variable

8. Following rules shall be flagged and deviations recorded due to KW Tool Issue. These will be moved to “Not a Problem” state in the tool.

Rule	Klocwork Code	Category	Description
5.2	MISRA.VAR.HIDDEN	Required	Identifier declared in an inner scope hides identifier in outer scope
6.4	MISRA.BITFIELD.TYPE	Required	Type of bit-field is not signed/unsigned integer
8.7	MISRA.ONEDEFRULE.VAR	Required	Global variable definition in a header file
11.5	MISRA.CAST.CONST	Required	Cast operation removes const or volatile modifier from a pointer or reference
16.7	MISRA.PPARAM.NEEDS.CONST	Advisory	Pointer parameter is not used to modify the addressed object but is not declared as a pointer to const

5. Deviation Documenting Process

1. Reports shall be generated in Excel (.xls) format
2. All blanket deviations shall be ignored at the tool level

6. Revision History

Version	Date	Status	Author	Revision History
1.0	28-Sep-12	Draft	Sivaraj R	Initial draft
1.1	20-Mar-13	Draft	Sivaraj R	Added few more deviation based on further assessment
1.2	17-Apr-13	Draft	Sivaraj R	Reworked based on review meeting with other internal team
1.3	7-Nov-13	Approved	Sivaraj R	Synced with latest MISRA-C policy aligned across components

1.4	2-Jun-14	Approved	Sivaraj R	Made is generic to Starterware and BSP. Also made it platform independent
1.5	6-Jul-15	Approved	Sunil MS Sivaraj R	Added few more MISRA-C deviations approved by stakeholders
1.6	8-Aug-15	Approved	Sunil MS	Updating Waiver types and Issue State's followed in STW/BSP
1.7	13-Oct-15	Approved	Sivaraj R	Added scope: Examples and fatlib not MISRAC compliant