Texas Instruments

PLATFORM DEVELOPMENT KIT (PDK)
MISRA C COMPLIANCE POLICY

Version 1.1

TABLE OF CONTENTS

1. Scope

The scope of this document is to identify the MISRA-C compliance of the Platform Development Kit (PDK) driver 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 for PDK package.

3. Rule Checking Tool

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

Klocwork Version 11.x

3.1. Klocwork Limitation and Known Issue

List of known limitation in the 11.x version of Klocwork is at
http://www.klocwork.com/products/documentation/current/Release_Notes

4. Deviations

The PDK related waiver reason is part of the Klocwork report present in *docs* folder of PDK 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 PDK
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 PDK and FATLIB module will not be MISRA-C compliant. All the violations will be moved to "Defer" state in the tool
2. PDK security add-on package will not be MISRA-C compliant. All the violations will be moved to "Defer" state in the tool.
3. MISRA-C violations present in dependent and other third party components like BIOS, XDC, tool generated files etc., shall not be fixed. They shall be

moved to “Ignore” state in tool. Below are the list of MISRA-C violations that falls under this category.

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
12.9	MISRA.UMINUS.UNSIGNED	Required	Operand of unary minus is unsigned
18.1	MISRA.INCOMPLETE.STRUCT	Required	Incomplete struct type is used

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

   PDK 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 and moved to “Not a Problem” state in the tool.
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
12.1	MISRA.EXPR.PARENS.REDUNDANT	Advisory	Limited dependence

			required for operator precedence rules in expressions
13.2	MISRA.ZERO_EQ.IMPLICIT	Advisory	Non-boolean expression is implicitly tested against zero

7. The “Case-by-Case” waiver rules shall be flagged and deviations recorded on a per-instance basis. These will be moved to “Ignore” state in the tool. Refer the KW report present in PDK docs folder for the list of rules under this category.

8. The “Not a Problem” waiver rules shall be flagged and deviations recorded due to KW Tool Issue. These will be moved to “Not a Problem” state in the tool. Refer the KW report present in PDK docs folder for the list of rules under this category.

5. Deviation Documenting Process

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

6. Revision History

Version	Date	Status	Author	Revision History
1.0	14-JUN-17	Draft	Sivaraj R	Initial draft derived from Starterware and BSP MISRAC policy
1.0	30-JUN-17	Approved	Sivaraj R	Sync'd the document with the final policy