

MISRA Compliance in Freescale Automotive C Language Header Files

Contents

1 Introduction

This document, developed in response to customer inquiries, describes Motor Industry Software Reliability Association (MISRA™) compliance in the C Language header files provided by Freescale for the automotive market.

MISRA (www.misra.org.uk) created software development standards in 1996, 2004, and 2012 for the C and C++ programming languages. These standards are widely used in automotive and other application domains.

Freescale delivers C language header files to enable the development of software that interfaces to Freescale silicon products. These files primarily consist of structure, union, and pointer declarations that enable software to read and write the hardware registers.

2 Statement of MISRA compliance

Freescale Automotive C Language header files are implemented with a goal of using good coding practices in order to make the files readable and maintainable. Because of this many, but not all, of the MISRA standards are implemented in the header files.

| | | |
|---|--|---|
| 1 | Introduction..... | 1 |
| 2 | Statement of MISRA compliance..... | 1 |
| 3 | Common MISRA issues in Freescale Automotive C Language header files..... | 2 |

If the Freescale Automotive C Language header file causes any MISRA standards violations in your project, the best approach may be to use the MISRA project deviation procedure to authorize a deviation from the MISRA standard.

3 Common MISRA issues in Freescale Automotive C Language header files

This section describes the most common MISRA issues found in the C language header files and also provides the justification for their continued usage.

- Use of unions is a violation of the MISRA standard. But unions are used to provide bit-level access to the fields in a hardware register.
- Use of unnamed unions is a violation of the MISRA standard. But unnamed unions are used to allow compatibility with previous generation devices, and to allow multiple access paths to the registers. The need for multiple access paths most commonly occurs when there is an array of registers. The array of registers is used because there are multiple usage models to access the registers.
- Use of a cast between a pointer type and an integral type is a violation of the MISRA standard. But the main purpose of the header files is to enable application software by providing pointers to the hardware registers. Constructs similar to the example below are commonly used throughout the header files where a hardware pointer to the oscillator module OSC is declared, and is initialized to the integral address 0xC3FE0000.

```
#define OSC    (*(volatile OSC_tag *) 0xC3FE0000UL)
```

The MISRA rules state that this construct may be unavoidable when addressing memory mapped registers. To avoid this issue, some header files may support the MISRA rule by inserting the following **boldface text**:

```
#define OSC    (*(volatile OSC_tag *) (volatile void*) ((vuint8_t*)0UL + 0xC3FE0000UL))
```

This does not generate a MISRA violation with many MISRA rules checkers but adds additional complexity to the pointer declaration. Thus this workaround is not found in all header files.

How to Reach Us:**Home Page:**freescale.com**Web Support:**freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2013 Freescale Semiconductor, Inc.