Polyspace Bug Finder

Detailed Report for Project: mis2012

Report Author: YangLiMin

Polyspace Bug Finder: Detailed Report for Project: mis2012

by Report Author: YangLiMin

Published 02-Dec-2024 14:23:48

Analysis Author(s): YangLiMin  
Polyspace Version(s): Polyspace Bug Finder 3.2 (R2020a)  
Project Version(s): 1.0  
  
  
  
  
  
  
Result Folder(s):  
C:\Users\yanglimin\Documents\Polyspace\_Workspace\mis2012\Module\_1\BF\_Result\_1

Table of Contents

[Chapter 1. Polyspace Bug Finder Summary 1](#_Toc184041881)

[Chapter 2. MISRA C:2012 Guidelines 14](#_Toc184041882)

[**MISRA C:2012 Guidelines Summary - Violations by File** 14](#_Toc184041883)

[**MISRA C:2012 Guidelines Violations** 21](#_Toc184041884)

[Chapter 3. Appendix 1 - Configuration Settings 121](#_Toc184041885)

[**Polyspace Settings** 121](#_Toc184041886)

[**Coding Standard Configuration** 121](#_Toc184041887)

[**Files with compilation errors (files partially analyzed)** 129](#_Toc184041888)

[Chapter 4. Appendix 2 - Definitions 130](#_Toc184041889)

Chapter 1. Polyspace Bug Finder Summary

Table 1.1. Project Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Count** | **Reviewed** | **Unreviewed** | **Pass/Fail** |
| MISRA C:2012 Guidelines | 1034 | 0 | 1034 | NA |
| **Total** | **1034** | **0** | **1034** | **NA** |

Table 1.2. Summary By File

|  |  |
| --- | --- |
| **File** | **MISRA C:2012 Guidelines (Reviewed)** |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_01.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_02\_01.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_02\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_02\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_03\_01.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_03\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_03\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_01.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_01.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_02.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_03.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_04.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_05.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_06.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_07.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_08.h | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_08\_1.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_08\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_09.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_10.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_10\_1.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_10\_2.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_10\_3.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_11.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_12.c | 11 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_13.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_13.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_14.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\MISRA\_Complete\_main.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_01\_01.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_01\_02.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_01\_03.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_01\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_01\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_01.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_01\_1.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_01\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_02.c | 11 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_02.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_03.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_03.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_04.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_04.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_05.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_05.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_06.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_07.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_01.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_02.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_01.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_02.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01\_1.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01\_2.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_02.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_03.c | 11 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_03.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_04.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_05.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_06.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_06.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_07.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_07.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_08.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_08\_1.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_08\_2.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_09.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_09\_1.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_09\_2.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_01.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_02.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_01.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_02.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_03.c | 9 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_04.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_04.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_01.c | 17 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_02.c | 13 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_03.c | 14 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_03.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_04.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_04.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_05.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_05\_1.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_05\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_1.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_2.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_07.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_07\_1.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_07\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_08.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_09.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_09.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_10.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_11.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_11\_1.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_11\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_12.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_13.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_13\_1.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_13\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_14.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_01.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_01.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_02.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_03.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_04.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_05.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_01.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_02.c | 12 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_03.c | 18 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_04.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_04.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_05.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_06.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_07.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_08.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_01.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_02.c | 10 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_03.c | 9 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_04.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_05.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_06.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_07.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_08.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_09.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_01.c | 13 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_01.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_02.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_02.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_03.c | 11 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_04.c | 10 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_05.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_01.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_01\_1.c | 10 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_01\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_02.c | 13 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_02.h | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_03.c | 16 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_04.c | 20 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_05.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_05\_1.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_05\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_06.c | 12 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_01.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_01.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_02.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_02.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_03.c | 16 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_03.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_04.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_01.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_02.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_03.c | 12 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_04.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_04.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_05.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_06.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_06.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_07.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_07.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_01.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_02.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_03.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_04.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_05.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_06.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_07.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_01.c | 9 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_02.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_02.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_03.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_04.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_05.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_05.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_06.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_07.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_08.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_08.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_01.c | 20 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_01.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_02.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_02.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_03.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_03.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_04.c | 19 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_05.c | 11 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_06.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_06\_1.c | 12 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_06\_2.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_07.c | 6 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_08.c | 15 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_01.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_01.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_02.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_01.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_01.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_02.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_03.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_03.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_04.c | 10 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_05.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_06.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_07.c | 10 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_08.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_09.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_10.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_11.c | 9 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_12.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_13.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14\_1.h | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14\_2.h | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_01.c | 9 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_02.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_03.c | 12 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_04.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_05.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_06.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_07.c | 1 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_08.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_09.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_10.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_11.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_12.c | 11 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_13.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_14.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_14.h | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_15.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_16.c | 12 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_17.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_18.c | 5 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_19.c | 16 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_20.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_01.c | 11 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_01.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_02.c | 21 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_02.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_03.c | 8 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_03.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_04.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_04.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_05.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_05.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_06.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_06.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_07.c | 7 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_08.c | 4 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_09.c | 12 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_10.c | 3 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_support.c | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_system.c | 2 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\mc3\_header.h | 0 (0) |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\mc3\_types.h | 1 (0) |

Table 1.3. Files with compilation errors (files partially analyzed)

|  |
| --- |
| **File** |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\MISRA\_Complete\_main.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_07.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_03.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_04.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_01.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_01.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_06.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_07.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_03.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_04.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_11.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_11.c |

Chapter 2. MISRA C:2012 Guidelines

MISRA C:2012 Guidelines Summary - Violations by File

|  |  |
| --- | --- |
| **File** | **Total** |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_02\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_03\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_01.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_03.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_04.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_05.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_06.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_07.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_08.h | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_08\_1.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_09.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_10\_3.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_12.c | 11 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_13.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_14.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_01\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_01\_1.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_02.c | 11 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_03.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_04.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_05.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_06.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_07.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_01.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_02.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_01.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_02.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01\_1.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01\_2.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_02.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_03.c | 11 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_05.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_06.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_07.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_08\_1.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_08\_2.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_09\_1.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_09\_2.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_01.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_02.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_01.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_02.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_03.c | 9 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_04.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_01.c | 17 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_02.c | 13 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_03.c | 14 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_04.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_05.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_05\_1.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_1.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_2.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_07\_1.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_08.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_09.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_10.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_11.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_12.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_13\_1.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_14.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_01.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_02.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_03.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_04.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_05.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_01.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_02.c | 12 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_03.c | 18 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_04.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_05.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_06.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_07.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_08.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_01.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_02.c | 10 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_03.c | 9 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_04.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_05.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_06.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_07.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_08.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_09.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_01.c | 13 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_01.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_02.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_03.c | 11 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_04.c | 10 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_05.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_01\_1.c | 10 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_02.c | 13 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_02.h | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_03.c | 16 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_04.c | 20 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_05\_1.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_06.c | 12 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_01.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_02.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_03.c | 16 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_04.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_01.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_02.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_03.c | 12 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_04.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_05.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_06.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_07.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_01.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_02.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_03.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_04.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_05.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_06.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_07.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_01.c | 9 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_02.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_03.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_04.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_05.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_06.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_07.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_08.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_01.c | 20 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_02.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_03.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_04.c | 19 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_05.c | 11 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_06\_1.c | 12 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_07.c | 6 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_08.c | 15 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_01.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_02.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_01.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_01.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_02.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_03.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_04.c | 10 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_05.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_06.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_07.c | 10 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_08.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_09.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_10.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_11.c | 9 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_12.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_13.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14\_1.h | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14\_2.h | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_01.c | 9 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_02.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_03.c | 12 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_04.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_05.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_06.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_07.c | 1 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_08.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_09.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_10.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_11.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_12.c | 11 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_13.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_14.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_14.h | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_15.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_16.c | 12 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_17.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_18.c | 5 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_19.c | 16 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_20.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_01.c | 11 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_02.c | 21 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_03.c | 8 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_04.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_05.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_06.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_07.c | 7 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_08.c | 4 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_09.c | 12 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_10.c | 3 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_system.c | 2 |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\mc3\_types.h | 1 |
| **Total** | **1034** |

MISRA C:2012 Guidelines Violations

Table 2.1. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 1 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 681 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'main' has external declarations in multiple files | File Scope | Unset | Unreviewed |  |

Table 2.2. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_02\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 2 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 683 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_02\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.3. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_03\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 3 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 711 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_03\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.4. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 809 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable f1 conflicts with the function name f1 (R\_07\_04.c line 20). | File Scope | Unset | Unreviewed |  |
| 884 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable f2 conflicts with the function name f2 (R\_07\_04.c line 22). | File Scope | Unset | Unreviewed |  |
| 730 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable f3 conflicts with the function name f3 (R\_11\_03.c line 28). | File Scope | Unset | Unreviewed |  |
| 908 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable f4 conflicts with the function name f4 (R\_09\_04.c line 23). | File Scope | Unset | Unreviewed |  |
| 980 | D4.1 | Run-time failures shall be minimized. Operation \* overflows. Valid range: [-3.4028E^+38 .. 3.4028E^+38] | D\_4\_1() | Unset | Unreviewed |  |
| 756 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable f5 conflicts with the function name f5 (R\_13\_05\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 987 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | D\_4\_1() | Unset | Unreviewed |  |

Table 2.5. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 4 | D4.3 | Assembly language shall be encapsulated and isolated. | D\_4\_3\_1() | Unset | Unreviewed |  |
| 5 | D4.3 | Assembly language shall be encapsulated and isolated. | D\_4\_3\_1() | Unset | Unreviewed |  |

Table 2.6. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 743 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |

Table 2.7. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 10 | D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. Identifiers "id1\_a\_b\_c" (D\_04\_05.c line 24) and "id1\_abc" (D\_04\_05.c line 25) differ by the presence/absence of the underscore character. | D\_4\_5() | Unset | Unreviewed |  |
| 7 | D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. Identifiers "id2\_abc" (D\_04\_05.c line 27) and "id2\_ABC" (D\_04\_05.c line 28) differ by a mixture of case. | D\_4\_5() | Unset | Unreviewed |  |
| 8 | D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. Identifiers "id3\_a\_bc" (D\_04\_05.c line 30) and "id3\_ab\_c" (D\_04\_05.c line 31) differ by the presence/absence of the underscore character. | D\_4\_5() | Unset | Unreviewed |  |
| 12 | D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. Identifiers "id4\_I" (D\_04\_05.c line 33) and "id4\_1" (D\_04\_05.c line 34) differ by the interchange of the letter 'I' with the number '1'. | D\_4\_5() | Unset | Unreviewed |  |
| 6 | D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. Identifiers "id5\_Z" (D\_04\_05.c line 36) and "id5\_2" (D\_04\_05.c line 37) differ by the interchange of the letter 'Z' with the number '2'. | D\_4\_5() | Unset | Unreviewed |  |
| 9 | D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. Identifiers "id6\_O" (D\_04\_05.c line 39) and "id6\_0" (D\_04\_05.c line 40) differ by the interchange of the letter 'O' with the number '0'. | D\_4\_5() | Unset | Unreviewed |  |
| 11 | D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. Identifiers "id7\_B" (D\_04\_05.c line 42) and "id7\_8" (D\_04\_05.c line 43) differ by the interchange of the letter 'B' with the number '8'. | D\_4\_5() | Unset | Unreviewed |  |
| 13 | D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. Identifiers "id8\_rn" (D\_04\_05.c line 45) and "id8\_m" (D\_04\_05.c line 46) differ by the interchange of the letter sequence 'rn' ('r' followed by 'n') with the letter 'm'. | D\_4\_5() | Unset | Unreviewed |  |

Table 2.8. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 14 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | D\_4\_6() | Unset | Unreviewed |  |
| 750 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable speed conflicts with the variable name speed (R\_05\_03.c line 45). | File Scope | Unset | Unreviewed |  |
| 901 | 5.6 | A typedef name shall be a unique identifier. variable speed conflicts with the typedef name speed (R\_05\_03.c line 49). | File Scope | Unset | Unreviewed |  |

Table 2.9. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 982 | 2.2 | There shall be no dead code. Variable 'error\_status' is rewritten later without an intermediate read. | D\_4\_7() | Unset | Unreviewed |  |

Table 2.10. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_08.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 18 | D4.8 | If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden. | File Scope | Unset | Unreviewed |  |
| 19 | D4.8 | If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden. | File Scope | Unset | Unreviewed |  |

Table 2.11. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_08\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 912 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 1018 | D4.14 | The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory. | D\_4\_8\_1() | Unset | Unreviewed |  |

Table 2.12. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_09.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 15 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 774 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function fn conflicts with the function name fn (R\_15\_06.h line 19). | File Scope | Unset | Unreviewed |  |
| 845 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 16 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |

Table 2.13. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_10\_3.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 17 | D4.10 | Precautions shall be taken in order to prevent the contents of a header file being included more than once. | File Scope | Unset | Unreviewed |  |

Table 2.14. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_12.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 885 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 21 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | f() | Unset | Unreviewed |  |
| 20 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | f() | Unset | Unreviewed |  |
| 25 | D4.12 | Dynamic memory allocation shall not be used. | f() | Unset | Unreviewed |  |
| 22 | D4.12 | Dynamic memory allocation shall not be used. | f() | Unset | Unreviewed |  |
| 23 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | f() | Unset | Unreviewed |  |
| 27 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | f() | Unset | Unreviewed |  |
| 24 | D4.12 | Dynamic memory allocation shall not be used. | f() | Unset | Unreviewed |  |
| 26 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | f() | Unset | Unreviewed |  |
| 28 | D4.12 | Dynamic memory allocation shall not be used. | f() | Unset | Unreviewed |  |
| 29 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | f() | Unset | Unreviewed |  |

Table 2.15. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_13.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 781 | 2.4 | A project should not contain unused tag declarations. Tag mutex\_t is not used. | File Scope | Unset | Unreviewed |  |

Table 2.16. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_14.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 30 | 21.6 | The Standard Library input/output functions shall not be used. | D\_4\_14() | Unset | Unreviewed |  |
| 31 | 21.6 | The Standard Library input/output functions shall not be used. | D\_4\_14() | Unset | Unreviewed |  |

Table 2.17. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 36 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 698 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_04\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.18. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_01\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 37 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 685 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_01\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.19. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_01\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 39 | 15.5 | A function should have a single point of exit at the end. | next\_light() | Unset | Unreviewed |  |
| 38 | 2.1 | A project shall not contain unreachable code. Statement is unreachable | next\_light() | Unset | Unreviewed |  |
| 970 | 2.2 | There shall be no dead code. Variable 'res' is never read after this point. | next\_light() | Unset | Unreviewed |  |

Table 2.20. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 830 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function g conflicts with the function name g (R\_08\_01.c line 28). | File Scope | Unset | Unreviewed |  |
| 744 | 2.2 | There shall be no dead code. The call to function g has no effect. | File Scope | Unset | Unreviewed |  |
| 849 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 40 | 2.2 | There shall be no dead code. | R\_2\_2() | Unset | Unreviewed |  |
| 41 | 2.2 | There shall be no dead code. | R\_2\_2() | Unset | Unreviewed |  |
| 989 | 2.2 | There shall be no dead code. Variable 'x' is never read after this point. | R\_2\_2() | Unset | Unreviewed |  |
| 42 | 2.2 | There shall be no dead code. | R\_2\_2() | Unset | Unreviewed |  |
| 995 | 2.2 | There shall be no dead code. Variable 'p1' is never read after this point. | R\_2\_2() | Unset | Unreviewed |  |
| 43 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_2\_2() | Unset | Unreviewed |  |
| 996 | D4.14 | The validity of values received from external sources shall be checked. Pointer used in arithmetic operation is from an unsecure source. Pointer may be NULL or point to unknown memory. | R\_2\_2() | Unset | Unreviewed |  |
| 1000 | D4.14 | The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory. | R\_2\_2() | Unset | Unreviewed |  |

Table 2.21. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 876 | 2.3 | A project should not contain unused type declarations. Type local\_Type is not used. | File Scope | Unset | Unreviewed |  |

Table 2.22. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 751 | 2.4 | A project should not contain unused tag declarations. Tag state is not used. | File Scope | Unset | Unreviewed |  |
| 878 | 2.3 | A project should not contain unused type declarations. Type state is not used. | File Scope | Unset | Unreviewed |  |
| 786 | 2.4 | A project should not contain unused tag declarations. Tag record\_t is not used. | File Scope | Unset | Unreviewed |  |
| 881 | 2.4 | A project should not contain unused tag declarations. Tag B is not used. | File Scope | Unset | Unreviewed |  |

Table 2.23. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 748 | 2.5 | A project should not contain unused macro declarations. Macro DATA is not used. | File Scope | Unset | Unreviewed |  |
| 44 | 20.5 | #undef should not be used. | File Scope | Unset | Unreviewed |  |

Table 2.24. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 778 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 45 | 2.6 | A function should not contain unused label declarations. Label label1 is not used. | R\_2\_6() | Unset | Unreviewed |  |

Table 2.25. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 46 | 2.7 | There should be no unused parameters in functions. Function withunusedpara has unused parameters. | withunusedpara() | Unset | Unreviewed |  |

Table 2.26. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 54 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 686 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_02\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.27. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 47 | 3.1 | The character sequences /\* and // shall not be used within a comment. /\* is used within a comment. | File Scope | Unset | Unreviewed |  |
| 48 | 3.1 | The character sequences /\* and // shall not be used within a comment. // is used within a comment. | File Scope | Unset | Unreviewed |  |
| 49 | 3.1 | The character sequences /\* and // shall not be used within a comment. /\* is used within a comment. | File Scope | Unset | Unreviewed |  |
| 749 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 50 | 3.1 | The character sequences /\* and // shall not be used within a comment. /\* is used within a comment. | File Scope | Unset | Unreviewed |  |

Table 2.28. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 51 | 3.1 | The character sequences /\* and // shall not be used within a comment. // is used within a comment. | File Scope | Unset | Unreviewed |  |
| 762 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 52 | 3.2 | Line-splicing shall not be used in // comments. | File Scope | Unset | Unreviewed |  |
| 53 | 3.2 | Line-splicing shall not be used in // comments. | File Scope | Unset | Unreviewed |  |

Table 2.29. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 58 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 688 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_03\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.30. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 59 | 4.1 | Octal and hexadecimal escape sequences shall be terminated. | R\_4\_1() | Unset | Unreviewed |  |
| 60 | 4.1 | Octal and hexadecimal escape sequences shall be terminated. | R\_4\_1() | Unset | Unreviewed |  |

Table 2.31. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 55 | 4.2 | Trigraphs should not be used. | File Scope | Unset | Unreviewed |  |
| 56 | 4.2 | Trigraphs should not be used. | File Scope | Unset | Unreviewed |  |
| 57 | 4.2 | Trigraphs should not be used. | File Scope | Unset | Unreviewed |  |

Table 2.32. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 61 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 708 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_04\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.33. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 929 | 5.1 | External identifiers shall be distinct. External variable engine\_exhaust\_gas\_temperature\_scaled conflicts with the external identifier engine\_exhaust\_gas\_temperature\_raw (R\_05\_01.h line 18). | File Scope | Unset | Unreviewed |  |

Table 2.34. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 865 | 5.1 | External identifiers shall be distinct. External variable engine\_exhaust\_gas\_temperature\_scaled conflicts with the external identifier engine\_exhaust\_gas\_temperature\_raw (R\_05\_01.h line 18). | File Scope | Unset | Unreviewed |  |
| 62 | 8.5 | An external object or function shall be declared once in one and only one file. Variable 'abc' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 724 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Variable 'abc' should have internal linkage | File Scope | Unset | Unreviewed |  |

Table 2.35. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_01\_2.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 715 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Variable 'ABC' should have internal linkage | File Scope | Unset | Unreviewed |  |

Table 2.36. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 65 | 8.5 | An external object or function shall be declared once in one and only one file. Variable 'engin2\_exhaust\_gas\_temperature\_raw' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 689 | 8.6 | An identifier with external linkage shall have exactly one external definition. undefined global variable 'engin2\_exhaust\_gas\_temperature\_raw' | File Scope | Unset | Unreviewed |  |
| 735 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |

Table 2.37. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 798 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 66 | 5.3 | An identifier declared in an inner scope shall not hide an identifier declared in an outer scope. | fn1() | Unset | Unreviewed |  |
| 898 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 68 | 5.3 | An identifier declared in an inner scope shall not hide an identifier declared in an outer scope. | fn2() | Unset | Unreviewed |  |
| 752 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable xyz conflicts with the variable name xyz (R\_05\_03.c line 36). | File Scope | Unset | Unreviewed |  |
| 916 | 2.2 | There shall be no dead code. The call to function g has no effect. | File Scope | Unset | Unreviewed |  |
| 67 | 17.8 | A function parameter should not be modified. Parameter xyz is changed. | fn2() | Unset | Unreviewed |  |
| 861 | 5.6 | A typedef name shall be a unique identifier. variable speed conflicts with the typedef name speed (R\_05\_03.c line 49). | File Scope | Unset | Unreviewed |  |
| 69 | 5.3 | An identifier declared in an inner scope shall not hide an identifier declared in an outer scope. | fn3() | Unset | Unreviewed |  |
| 770 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. type speed conflicts with the variable name speed (R\_05\_03.c line 45). | File Scope | Unset | Unreviewed |  |
| 877 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable sp conflicts with the variable name sp (R\_18\_06\_1.c line 28). | File Scope | Unset | Unreviewed |  |

Table 2.38. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 63 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 64 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 70 | 5.5 | Identifiers shall be distinct from macro names. Identifier Sum has same significant characters as macro Sum (R\_05\_05.c line 30). | R\_5\_5() | Unset | Unreviewed |  |
| 803 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |

Table 2.39. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 909 | 5.6 | A typedef name shall be a unique identifier. type u8\_t conflicts with the typedef name u8\_t (R\_05\_06.c line 24). | File Scope | Unset | Unreviewed |  |
| 793 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function func1 conflicts with the function name func1 (R\_08\_02.c line 30). | File Scope | Unset | Unreviewed |  |
| 71 | 5.3 | An identifier declared in an inner scope shall not hide an identifier declared in an outer scope. | func1() | Unset | Unreviewed |  |
| 731 | 5.6 | A typedef name shall be a unique identifier. variable mass32 conflicts with the typedef name mass32 (R\_05\_06.c line 38). | File Scope | Unset | Unreviewed |  |
| 917 | 5.6 | A typedef name shall be a unique identifier. type chain conflicts with the typedef name chain (R\_05\_06.c line 62). | File Scope | Unset | Unreviewed |  |
| 72 | 11.9 | The macro NULL shall be the only permitted form of integer null pointer constant. | R\_5\_6() | Unset | Unreviewed |  |

Table 2.40. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 73 | D2.1 | All source files shall compile without any compilation errors. "elk" has already been declared in the current scope | File Scope | Unset | Unreviewed |  |
| 76 | 8.3 | All declarations of an object or function shall use the same names and type qualifiers. Global declaration of 'elk' variable has incompatible type with its definition. | File Scope | Unset | Unreviewed |  |
| 74 | 19.2 | The union keyword should not be used. | R\_5\_7() | Unset | Unreviewed |  |
| 75 | D2.1 | All source files shall compile without any compilation errors. Tag kind of union is incompatible with declaration of struct "stag" (declared at line 21) | R\_5\_7() | Unset | Unreviewed |  |
| 77 | 5.7 | A tag name shall be a unique identifier. tag name 'stag' should not be reused. already used as tag name (R\_05\_07.c line 21). | R\_5\_7() | Unset | Unreviewed |  |

Table 2.41. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_08\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 79 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Global definition of 'count8' variable has no previous declaration. | File Scope | Unset | Unreviewed |  |
| 710 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Variable 'count8' should have internal linkage | File Scope | Unset | Unreviewed |  |
| 78 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Function 'foo8' has no visible prototype at definition. | File Scope | Unset | Unreviewed |  |
| 716 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Function 'foo8' should have internal linkage | File Scope | Unset | Unreviewed |  |
| 855 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |

Table 2.42. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_08\_2.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 842 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function foo8 conflicts with the function name foo8 (R\_05\_08\_1.c line 25). | File Scope | Unset | Unreviewed |  |
| 766 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable count8 conflicts with the variable name count8 (R\_05\_08\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 866 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |

Table 2.43. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_09\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 80 | 5.3 | An identifier declared in an inner scope shall not hide an identifier declared in an outer scope. | foo9() | Unset | Unreviewed |  |
| 882 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable count conflicts with the variable name count (R\_05\_09\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 775 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |
| 81 | 5.3 | An identifier declared in an inner scope shall not hide an identifier declared in an outer scope. | bar1() | Unset | Unreviewed |  |
| 765 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable count conflicts with the variable name count (R\_05\_09\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 829 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |

Table 2.44. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_09\_2.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 816 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable count conflicts with the variable name count (R\_05\_09\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 847 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function foo9 conflicts with the function name foo9 (R\_05\_09\_1.c line 24). | File Scope | Unset | Unreviewed |  |
| 838 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |

Table 2.45. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 82 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 706 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.46. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 83 | 6.1 | Bit-fields shall only be declared with an appropriate type. | File Scope | Unset | Unreviewed |  |
| 84 | 6.1 | Bit-fields shall only be declared with an appropriate type. | File Scope | Unset | Unreviewed |  |

Table 2.47. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 85 | 6.2 | Single-bit named bit fields shall not be of a signed type. | File Scope | Unset | Unreviewed |  |

Table 2.48. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 86 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 692 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_06\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.49. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 87 | 7.1 | Octal constants shall not be used. | R\_7\_1() | Unset | Unreviewed |  |
| 88 | 7.1 | Octal constants shall not be used. | R\_7\_1() | Unset | Unreviewed |  |

Table 2.50. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 89 | 7.2 | A "u" or "U" suffix shall be applied to all integer constants that are represented in an unsigned type. | R\_7\_2() | Unset | Unreviewed |  |

Table 2.51. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 747 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 96 | 7.3 | The lowercase character "l" shall not be used in a literal suffix. | R\_7\_3() | Unset | Unreviewed |  |
| 97 | 7.3 | The lowercase character "l" shall not be used in a literal suffix. | R\_7\_3() | Unset | Unreviewed |  |
| 818 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 98 | 7.3 | The lowercase character "l" shall not be used in a literal suffix. | R\_7\_3() | Unset | Unreviewed |  |
| 913 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable g conflicts with the function name g (R\_08\_01.c line 28). | File Scope | Unset | Unreviewed |  |
| 772 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable h conflicts with the function name h (R\_02\_02.c line 27). | File Scope | Unset | Unreviewed |  |
| 99 | 7.3 | The lowercase character "l" shall not be used in a literal suffix. | R\_7\_3() | Unset | Unreviewed |  |
| 100 | 7.3 | The lowercase character "l" shall not be used in a literal suffix. | R\_7\_3() | Unset | Unreviewed |  |

Table 2.52. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 90 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f1' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 92 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f2' has external declarations in multiple files Function 'f2' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 91 | 7.4 | A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char". | g2() | Unset | Unreviewed |  |
| 93 | 7.4 | A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char". | name1() | Unset | Unreviewed |  |
| 910 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 94 | 7.4 | A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char". | R\_7\_4() | Unset | Unreviewed |  |
| 95 | 7.4 | A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char". | R\_7\_4() | Unset | Unreviewed |  |

Table 2.53. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 101 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 697 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.54. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 102 | 8.1 | Types shall be explicitly specified. Explicit type is missing ("int" assumed) | File Scope | Unset | Unreviewed |  |
| 103 | 8.5 | An external object or function shall be declared once in one and only one file. Variable 'x' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 104 | 8.5 | An external object or function shall be declared once in one and only one file. Variable 'x\_ok' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 113 | 8.1 | Types shall be explicitly specified. Explicit type is missing ("int" assumed) | File Scope | Unset | Unreviewed |  |
| 114 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 107 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f\_ok' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 110 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'g' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 108 | 8.1 | Types shall be explicitly specified. Explicit type is missing ("int" assumed) | File Scope | Unset | Unreviewed |  |
| 109 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'g\_ok' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 111 | 8.1 | Types shall be explicitly specified. Explicit type is missing ("int" assumed) | File Scope | Unset | Unreviewed |  |
| 105 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 106 | 8.1 | Types shall be explicitly specified. Explicit type is missing ("int" assumed) | File Scope | Unset | Unreviewed |  |
| 115 | 8.1 | Types shall be explicitly specified. Explicit type is missing ("int" assumed) | R\_8\_1() | Unset | Unreviewed |  |
| 116 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 112 | 8.1 | Types shall be explicitly specified. Explicit type is missing ("int" assumed) | R\_8\_1() | Unset | Unreviewed |  |
| 896 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 117 | 11.1 | Conversions shall not be performed between a pointer to a function and any other type. | R\_8\_1() | Unset | Unreviewed |  |

Table 2.55. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 118 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'func1' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 121 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'func2' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 119 | 8.2 | Function types shall be in prototype form with named parameters. | File Scope | Unset | Unreviewed |  |
| 125 | 8.2 | Function types shall be in prototype form with named parameters. | File Scope | Unset | Unreviewed |  |
| 126 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'func4' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 721 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Function 'func1' should have internal linkage | File Scope | Unset | Unreviewed |  |
| 124 | 8.3 | All declarations of an object or function shall use the same names and type qualifiers. | File Scope | Unset | Unreviewed |  |
| 122 | 8.2 | Function types shall be in prototype form with named parameters. | File Scope | Unset | Unreviewed |  |
| 127 | 8.13 | A pointer should point to a const-qualified type whenever possible. | func3() | Unset | Unreviewed |  |
| 120 | 8.2 | Function types shall be in prototype form with named parameters. | File Scope | Unset | Unreviewed |  |
| 123 | 8.2 | Function types shall be in prototype form with named parameters. | R\_8\_2() | Unset | Unreviewed |  |
| 894 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable arr conflicts with the variable name arr (R\_19\_01.c line 38). | File Scope | Unset | Unreviewed |  |
| 128 | 11.1 | Conversions shall not be performed between a pointer to a function and any other type. | R\_8\_2() | Unset | Unreviewed |  |

Table 2.56. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 142 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f3' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 137 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 141 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 138 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'g3' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 144 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 145 | 8.3 | All declarations of an object or function shall use the same names and type qualifiers. Function 'g3' is not compatible with previous declaration. | File Scope | Unset | Unreviewed |  |
| 146 | D2.1 | All source files shall compile without any compilation errors. Declaration is incompatible with "void g3(const int \*)" (declared at line 29) | File Scope | Unset | Unreviewed |  |
| 139 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 143 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'func' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 140 | 8.3 | All declarations of an object or function shall use the same names and type qualifiers. Parameter number 1 has different names. Parameter number 2 has different names. | File Scope | Unset | Unreviewed |  |
| 147 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'area' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 149 | 8.3 | All declarations of an object or function shall use the same names and type qualifiers. Parameter number 2 has different types. | File Scope | Unset | Unreviewed |  |
| 148 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 150 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |

Table 2.57. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 129 | 8.5 | An external object or function shall be declared once in one and only one file. Variable 'count' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 130 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Global definition of 'speed' variable has no previous declaration. | File Scope | Unset | Unreviewed |  |
| 131 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Global definition of 'pressure' variable has no previous declaration. | File Scope | Unset | Unreviewed |  |
| 132 | 8.5 | An external object or function shall be declared once in one and only one file. Variable 'ext\_val1' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 133 | 8.3 | All declarations of an object or function shall use the same names and type qualifiers. Parameter number 2 has different types. Function 'func43' is not compatible with previous declaration. Function 'func43' has no visible compatible prototype at definition. | File Scope | Unset | Unreviewed |  |
| 134 | D2.1 | All source files shall compile without any compilation errors. Declaration is incompatible with "void func43(int16\_t, int16\_t)" (declared at line 28 of "D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_04.h") | File Scope | Unset | Unreviewed |  |
| 135 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Function 'func43' has no visible compatible prototype at definition. | File Scope | Unset | Unreviewed |  |
| 136 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Function 'func44' has no visible prototype at definition. | File Scope | Unset | Unreviewed |  |

Table 2.58. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_05.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 151 | 8.5 | An external object or function shall be declared once in one and only one file. | File Scope | Unset | Unreviewed |  |

Table 2.59. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_05\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 152 | 8.5 | An external object or function shall be declared once in one and only one file. Variable 'ex\_file\_head\_decl' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |

Table 2.60. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 720 | 8.6 | An identifier with external linkage shall have exactly one external definition. undefined global variable 'no\_definition' | File Scope | Unset | Unreviewed |  |

Table 2.61. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 713 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Variable 'j' should have internal linkage | File Scope | Unset | Unreviewed |  |
| 153 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Global definition of 'k' variable has no previous declaration. | File Scope | Unset | Unreviewed |  |

Table 2.62. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_2.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 687 | 8.6 | An identifier with external linkage shall have exactly one external definition. Global variable 'i' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_2.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_1.c | File Scope | Unset | Unreviewed |  |
| 154 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Global definition of 'k' variable has no previous declaration. | File Scope | Unset | Unreviewed |  |
| 691 | 8.6 | An identifier with external linkage shall have exactly one external definition. The global variable 'k' has multiple tentative definitions This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_2.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_06\_1.c | File Scope | Unset | Unreviewed |  |

Table 2.63. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_07\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 155 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Global definition of 'implicit\_extern' variable has no previous declaration. | File Scope | Unset | Unreviewed |  |
| 722 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Variable 'implicit\_extern' should have internal linkage | File Scope | Unset | Unreviewed |  |
| 712 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Variable 'explicit\_extern' should have internal linkage | File Scope | Unset | Unreviewed |  |
| 725 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Function 'explicit\_ext' should have internal linkage | File Scope | Unset | Unreviewed |  |

Table 2.64. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_08.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 156 | 8.8 | The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage. | File Scope | Unset | Unreviewed |  |
| 157 | 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. Global definition of 'y8' variable has no previous declaration. | File Scope | Unset | Unreviewed |  |
| 700 | 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. Variable 'y8' should have internal linkage | File Scope | Unset | Unreviewed |  |
| 158 | 8.8 | The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage. External/internal linkage conflict with previous declaration | File Scope | Unset | Unreviewed |  |
| 159 | 8.8 | The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage. | File Scope | Unset | Unreviewed |  |
| 161 | 8.8 | The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage. | File Scope | Unset | Unreviewed |  |
| 160 | 8.8 | The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage. External/internal linkage conflict with previous declaration | File Scope | Unset | Unreviewed |  |

Table 2.65. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_09.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 162 | 8.9 | An object should be defined at block scope if its identifier only appears in a single function. | File Scope | Unset | Unreviewed |  |
| 163 | 8.9 | An object should be defined at block scope if its identifier only appears in a single function. | File Scope | Unset | Unreviewed |  |

Table 2.66. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_10.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 164 | 8.10 | An inline function shall be declared with the static storage class. | max() | Unset | Unreviewed |  |

Table 2.67. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_11.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 165 | 8.11 | When an array with external linkage is declared, its size should be explicitly specified. Size of array 'array2' should be explicitly stated. | File Scope | Unset | Unreviewed |  |

Table 2.68. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_12.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 166 | 8.12 | Within an enumerator list, the value of an implicitly-specified enumeration constant shall be unique. The constant yellow has same value as the implicitly-specified constant green. | File Scope | Unset | Unreviewed |  |

Table 2.69. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_13\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 167 | 8.13 | A pointer should point to a const-qualified type whenever possible. | f13() | Unset | Unreviewed |  |
| 168 | 8.13 | A pointer should point to a const-qualified type whenever possible. | last\_char() | Unset | Unreviewed |  |
| 923 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 1025 | D4.14 | The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory. | last\_char() | Unset | Unreviewed |  |
| 1026 | D4.14 | The validity of values received from external sources shall be checked. Argument to 'strlen' is from an unsecure source. Argument may be NULL or not NULL-terminated. | last\_char() | Unset | Unreviewed |  |
| 926 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 169 | 8.13 | A pointer should point to a const-qualified type whenever possible. | first() | Unset | Unreviewed |  |

Table 2.70. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_14.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 170 | 8.14 | The restrict type qualifier shall not be used. | File Scope | Unset | Unreviewed |  |
| 171 | 8.14 | The restrict type qualifier shall not be used. | File Scope | Unset | Unreviewed |  |
| 173 | 17.8 | A function parameter should not be modified. Parameter p is changed. | user\_copy() | Unset | Unreviewed |  |
| 172 | 11.8 | A cast shall not remove any const or volatile qualification from the type pointed to by a pointer. Cast from type 'const void \*' to type 'void \*' removes qualifiers. | user\_copy() | Unset | Unreviewed |  |

Table 2.71. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 174 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 690 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.72. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 924 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 806 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function g conflicts with the function name g (R\_08\_01.c line 28). | File Scope | Unset | Unreviewed |  |
| 1028 | 9.1 | The value of an object with automatic storage duration shall not be read before it has been set. Local variable 'u' is read before being initialized. | g() | Unset | Unreviewed |  |
| 176 | 15.1 | The goto statement should not be used. | jmp\_over\_init() | Unset | Unreviewed |  |
| 175 | 2.1 | A project shall not contain unreachable code. Dynamic initialization in unreachable code | jmp\_over\_init() | Unset | Unreviewed |  |
| 780 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 983 | 9.1 | The value of an object with automatic storage duration shall not be read before it has been set. Local variable 'x' is read before being initialized. | jmp\_over\_init() | Unset | Unreviewed |  |
| 911 | 2.2 | There shall be no dead code. The call to function g has no effect. | File Scope | Unset | Unreviewed |  |

Table 2.73. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 870 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable y1 conflicts with the function name y1 (mathcalls.h line 253). | File Scope | Unset | Unreviewed |  |
| 177 | 9.2 | The initializer for an aggregate or union shall be enclosed in braces. | R\_9\_2() | Unset | Unreviewed |  |
| 178 | 9.2 | The initializer for an aggregate or union shall be enclosed in braces. | R\_9\_2() | Unset | Unreviewed |  |
| 179 | 19.2 | The union keyword should not be used. | R\_9\_2() | Unset | Unreviewed |  |
| 928 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 180 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (character) | R\_9\_2() | Unset | Unreviewed |  |

Table 2.74. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 875 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 181 | 9.3 | Arrays shall not be partially initialized. | R\_9\_3() | Unset | Unreviewed |  |
| 873 | 5.7 | A tag name shall be a unique identifier. variable t conflicts with the tag name t (R\_11\_02.c line 25). | File Scope | Unset | Unreviewed |  |
| 182 | 9.3 | Arrays shall not be partially initialized. | R\_9\_3() | Unset | Unreviewed |  |
| 862 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable arr conflicts with the variable name arr (R\_19\_01.c line 38). | File Scope | Unset | Unreviewed |  |
| 920 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable h conflicts with the function name h (R\_02\_02.c line 27). | File Scope | Unset | Unreviewed |  |

Table 2.75. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 785 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 184 | 13.1 | Initializer lists shall not contain persistent side effects. | f4() | Unset | Unreviewed |  |
| 187 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | f4() | Unset | Unreviewed |  |
| 183 | 9.4 | An element of an object shall not be initialized more than once. | f4() | Unset | Unreviewed |  |
| 185 | 9.4 | An element of an object shall not be initialized more than once. | R\_9\_4() | Unset | Unreviewed |  |
| 186 | 9.4 | An element of an object shall not be initialized more than once. | R\_9\_4() | Unset | Unreviewed |  |

Table 2.76. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 188 | 9.5 | Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly. | R\_9\_5() | Unset | Unreviewed |  |

Table 2.77. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 189 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 701 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_09\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.78. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 190 | D2.1 | All source files shall compile without any compilation errors. Expression must have integral type | R\_10\_1() | Unset | Unreviewed |  |
| 191 | D2.1 | All source files shall compile without any compilation errors. Expression must have integral type | R\_10\_1() | Unset | Unreviewed |  |

Table 2.79. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 195 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category character) is assigned to an object with a different essential type category (signed) | R\_10\_2() | Unset | Unreviewed |  |
| 192 | 10.2 | Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations. The left operand of the - operator shall have essentially character type if the right operand has essentially character type. | R\_10\_2() | Unset | Unreviewed |  |
| 193 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the - operator has essentially signed type while the right operand has essentially character type. | R\_10\_2() | Unset | Unreviewed |  |
| 202 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category floating) is assigned to an object with a different essential type category (character) | R\_10\_2() | Unset | Unreviewed |  |
| 194 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases. | R\_10\_2() | Unset | Unreviewed |  |
| 200 | 10.2 | Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations. The right operand of the + operator applied to an expression of essentially character type shall have essentially signed or unsigned type. | R\_10\_2() | Unset | Unreviewed |  |
| 201 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the + operator has essentially character type while the right operand has essentially floating type. | R\_10\_2() | Unset | Unreviewed |  |
| 198 | 10.2 | Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations. The right operand of the + operator applied to an expression of essentially character type shall have essentially signed or unsigned type. | R\_10\_2() | Unset | Unreviewed |  |
| 197 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (character) | R\_10\_2() | Unset | Unreviewed |  |
| 196 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the - operator has essentially character type while the right operand has essentially enum type. | R\_10\_2() | Unset | Unreviewed |  |
| 199 | 10.1 | Operands shall not be of an inappropriate essential type. The right operand of the - operator is of an inappropriate essential type category enum. | R\_10\_2() | Unset | Unreviewed |  |
| 203 | 10.2 | Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations. The right operand of the - operator applied to an expression of essentially character type shall have essentially signed or unsigned or character type. | R\_10\_2() | Unset | Unreviewed |  |

Table 2.80. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 886 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 206 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits) | foo1() | Unset | Unreviewed |  |
| 1023 | D4.1 | Run-time failures shall be minimized. Conversion from unsigned int16 to unsigned int8 overflows. Valid range: [0 .. 255] | foo1() | Unset | Unreviewed |  |
| 217 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category floating) is assigned to an object with a different essential type category (unsigned) | R\_10\_3() | Unset | Unreviewed |  |
| 207 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (Boolean) | R\_10\_3() | Unset | Unreviewed |  |
| 209 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (character) | R\_10\_3() | Unset | Unreviewed |  |
| 218 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category character) is assigned to an object with a different essential type category (unsigned) | R\_10\_3() | Unset | Unreviewed |  |
| 205 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | R\_10\_3() | Unset | Unreviewed |  |
| 204 | 12.4 | Evaluation of constant expressions should not lead to unsigned integer wrap-around. Integer operation result is out of range. | R\_10\_3() | Unset | Unreviewed |  |
| 210 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category character) is assigned to an object with a different essential type category (unsigned) | R\_10\_3() | Unset | Unreviewed |  |
| 212 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (unsigned) | R\_10\_3() | Unset | Unreviewed |  |
| 215 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type signed on 16 bits) is assigned to an object with a narrower essential type (signed on 8 bits) | R\_10\_3() | Unset | Unreviewed |  |
| 208 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 32 bits) is assigned to an object with a narrower essential type (unsigned on 16 bits) | R\_10\_3() | Unset | Unreviewed |  |
| 213 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 32 bits) is assigned to an object with a narrower essential type (unsigned on 16 bits) | R\_10\_3() | Unset | Unreviewed |  |
| 214 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type signed on 32 bits) is assigned to an object with a narrower essential type (signed on 8 bits) | R\_10\_3() | Unset | Unreviewed |  |
| 216 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits) | R\_10\_3() | Unset | Unreviewed |  |
| 211 | 12.4 | Evaluation of constant expressions should not lead to unsigned integer wrap-around. Integer operation result is out of range. | R\_10\_3() | Unset | Unreviewed |  |
| 219 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type unsigned on 16 bits) is assigned to an object with a narrower essential type (unsigned on 8 bits) | R\_10\_3() | Unset | Unreviewed |  |

Table 2.81. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 220 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. The left operand of the += operator has essentially signed type while the right operand has essentially unsigned type. | R\_10\_4() | Unset | Unreviewed |  |
| 221 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category unsigned) is assigned to an object with a different essential type category (signed) | R\_10\_4() | Unset | Unreviewed |  |
| 223 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. Operands of the > operator have different essentially enum types. | R\_10\_4() | Unset | Unreviewed |  |
| 222 | 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. Operands of the == operator have different essentially enum types. | R\_10\_4() | Unset | Unreviewed |  |
| 224 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category character) is assigned to an object with a different essential type category (unsigned) | R\_10\_4() | Unset | Unreviewed |  |

Table 2.82. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 225 | 10.5 | The value of an expression should not be cast to an inappropriate essential type. | R\_10\_5() | Unset | Unreviewed |  |
| 226 | 10.5 | The value of an expression should not be cast to an inappropriate essential type. | R\_10\_5() | Unset | Unreviewed |  |
| 227 | 10.5 | The value of an expression should not be cast to an inappropriate essential type. | R\_10\_5() | Unset | Unreviewed |  |

Table 2.83. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 231 | 10.6 | The value of a composite expression shall not be assigned to an object with wider essential type. The composite expression (of essential type unsigned on 16 bits) is assigned to an object with a wider essential type (unsigned on 32 bits) | R\_10\_6() | Unset | Unreviewed |  |
| 229 | 10.6 | The value of a composite expression shall not be assigned to an object with wider essential type. The composite expression (of essential type unsigned on 16 bits) is assigned to an object with a wider essential type (unsigned on 32 bits) | R\_10\_6() | Unset | Unreviewed |  |
| 230 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_10\_6() | Unset | Unreviewed |  |
| 232 | 10.6 | The value of a composite expression shall not be assigned to an object with wider essential type. The composite expression (of essential type unsigned on 8 bits) is assigned to an object with a wider essential type (unsigned on 16 bits) | R\_10\_6() | Unset | Unreviewed |  |
| 975 | D4.1 | Run-time failures shall be minimized. Conversion from int32 to unsigned int32 overflows. Valid range: [0 .. 2^32-1] | R\_10\_6() | Unset | Unreviewed |  |
| 228 | 10.6 | The value of a composite expression shall not be assigned to an object with wider essential type. The composite expression (of essential type unsigned on 32 bits) is assigned to an object with a wider essential type (unsigned on 64 bits) | R\_10\_6() | Unset | Unreviewed |  |
| 233 | 10.6 | The value of a composite expression shall not be assigned to an object with wider essential type. The composite expression (of essential type unsigned on 32 bits) is assigned to an object with a wider essential type (unsigned on 64 bits) | R\_10\_6() | Unset | Unreviewed |  |

Table 2.84. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 234 | 12.1 | The precedence of operators within expressions should be made explicit. | R\_10\_7() | Unset | Unreviewed |  |
| 235 | 10.7 | If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The left operand of the \* operator shall not have wider essential type (unsigned on 32 bits) than the right operand (unsigned on 16 bits) which is a composite expression | R\_10\_7() | Unset | Unreviewed |  |
| 236 | 10.7 | If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. The left operand of the += operator shall not have wider essential type (unsigned on 32 bits) than the right operand (unsigned on 16 bits) which is a composite expression | R\_10\_7() | Unset | Unreviewed |  |

Table 2.85. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_08.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 237 | 10.8 | The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type category signed shall not be cast to the different essential type category unsigned. | R\_10\_8() | Unset | Unreviewed |  |
| 238 | 10.8 | The value of a composite expression shall not be cast to a different essential type category or a wider essential type. The value of the composite expression of essential type uint16\_t shall not be cast to the wider essential type uint32\_t. | R\_10\_8() | Unset | Unreviewed |  |

Table 2.86. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 239 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 717 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.87. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 248 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f1' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 249 | D2.1 | All source files shall compile without any compilation errors. Invalid type conversion | R\_11\_1() | Unset | Unreviewed |  |
| 250 | 11.1 | Conversions shall not be performed between a pointer to a function and any other type. | R\_11\_1() | Unset | Unreviewed |  |

Table 2.88. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 246 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'use\_structs\_ptr' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 240 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'use\_structt\_ptr' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 241 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f2' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 726 | 8.3 | All declarations of an object or function shall use the same names and type qualifiers. Global declaration of 'f2' function has a type incompatible with its definition. This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_02.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_07\_04.c | File Scope | Unset | Unreviewed |  |
| 820 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable sp conflicts with the variable name sp (R\_18\_06\_1.c line 28). | File Scope | Unset | Unreviewed |  |
| 242 | 11.2 | Conversions shall not be performed between a pointer to an incomplete type and any other type. | R\_11\_2() | Unset | Unreviewed |  |
| 244 | 11.2 | Conversions shall not be performed between a pointer to an incomplete type and any other type. | R\_11\_2() | Unset | Unreviewed |  |
| 243 | 11.3 | A cast shall not be performed between a pointer to object type and a pointer to a different object type. | R\_11\_2() | Unset | Unreviewed |  |
| 247 | 11.2 | Conversions shall not be performed between a pointer to an incomplete type and any other type. | R\_11\_2() | Unset | Unreviewed |  |
| 245 | 11.3 | A cast shall not be performed between a pointer to object type and a pointer to a different object type. | R\_11\_2() | Unset | Unreviewed |  |

Table 2.89. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 251 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'read\_value' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 252 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'print' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 253 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'get\_int32\_ptr\_ptr' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 255 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'use\_int16\_cvptr' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 257 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'use\_int32\_ccptr' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 256 | 11.3 | A cast shall not be performed between a pointer to object type and a pointer to a different object type. | f3() | Unset | Unreviewed |  |
| 254 | 11.3 | A cast shall not be performed between a pointer to object type and a pointer to a different object type. | R\_11\_3() | Unset | Unreviewed |  |
| 932 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 258 | 11.3 | A cast shall not be performed between a pointer to object type and a pointer to a different object type. | R\_11\_3() | Unset | Unreviewed |  |

Table 2.90. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 259 | 11.4 | A conversion should not be performed between a pointer to object and an integer type. | R\_11\_4() | Unset | Unreviewed |  |
| 261 | 11.4 | A conversion should not be performed between a pointer to object and an integer type. | R\_11\_4() | Unset | Unreviewed |  |
| 260 | 11.4 | A conversion should not be performed between a pointer to object and an integer type. | R\_11\_4() | Unset | Unreviewed |  |
| 262 | 11.4 | A conversion should not be performed between a pointer to object and an integer type. | R\_11\_4() | Unset | Unreviewed |  |
| 771 | 5.7 | A tag name shall be a unique identifier. enum constant A conflicts with the tag name A (R\_02\_04.c line 51). | File Scope | Unset | Unreviewed |  |
| 868 | 5.7 | A tag name shall be a unique identifier. enum constant B conflicts with the tag name B (R\_02\_04.c line 59). | File Scope | Unset | Unreviewed |  |
| 263 | 11.7 | A cast shall not be performed between pointer to object and a non-integer arithmetic type. | R\_11\_4() | Unset | Unreviewed |  |
| 264 | 11.4 | A conversion should not be performed between a pointer to object and an integer type. | R\_11\_4() | Unset | Unreviewed |  |

Table 2.91. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 265 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | R\_11\_5() | Unset | Unreviewed |  |
| 266 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | R\_11\_5() | Unset | Unreviewed |  |

Table 2.92. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 267 | D2.1 | All source files shall compile without any compilation errors. Invalid type conversion | R\_11\_6() | Unset | Unreviewed |  |
| 268 | 11.6 | A cast shall not be performed between pointer to void and an arithmetic type. | R\_11\_6() | Unset | Unreviewed |  |

Table 2.93. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 269 | D2.1 | All source files shall compile without any compilation errors. Invalid type conversion | R\_11\_7() | Unset | Unreviewed |  |
| 270 | 11.7 | A cast shall not be performed between pointer to object and a non-integer arithmetic type. | R\_11\_7() | Unset | Unreviewed |  |
| 271 | D2.1 | All source files shall compile without any compilation errors. Invalid type conversion | R\_11\_7() | Unset | Unreviewed |  |
| 272 | 11.7 | A cast shall not be performed between pointer to object and a non-integer arithmetic type. | R\_11\_7() | Unset | Unreviewed |  |

Table 2.94. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_08.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 858 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 273 | 11.8 | A cast shall not remove any const or volatile qualification from the type pointed to by a pointer. Cast from type 'const uint16\_t \*' to type 'uint16\_t \*' removes qualifiers. | R\_11\_8() | Unset | Unreviewed |  |
| 274 | 11.8 | A cast shall not remove any const or volatile qualification from the type pointed to by a pointer. Cast from type 'volatile uint16\_t \*' to type 'uint16\_t \*' removes qualifiers. | R\_11\_8() | Unset | Unreviewed |  |
| 275 | 11.8 | A cast shall not remove any const or volatile qualification from the type pointed to by a pointer. Cast from type 'uint16\_t \*const \*' to type 'uint16\_t \*\*' removes qualifiers. | R\_11\_8() | Unset | Unreviewed |  |

Table 2.95. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_09.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 280 | 11.9 | The macro NULL shall be the only permitted form of integer null pointer constant. | File Scope | Unset | Unreviewed |  |
| 279 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | File Scope | Unset | Unreviewed |  |
| 276 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f9' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 277 | 11.9 | The macro NULL shall be the only permitted form of integer null pointer constant. | R\_11\_9() | Unset | Unreviewed |  |
| 278 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | R\_11\_9() | Unset | Unreviewed |  |
| 957 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_11\_9() | Unset | Unreviewed |  |
| 962 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_11\_9() | Unset | Unreviewed |  |

Table 2.96. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 281 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 695 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.97. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 753 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 819 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 745 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 927 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable arr conflicts with the variable name arr (R\_19\_01.c line 38). | File Scope | Unset | Unreviewed |  |
| 976 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_12\_1() | Unset | Unreviewed |  |
| 979 | D4.14 | The validity of values received from external sources shall be checked. Array index is from an unsecure source. Index may be negative or too big. | R\_12\_1() | Unset | Unreviewed |  |
| 282 | 2.2 | There shall be no dead code. | R\_12\_1() | Unset | Unreviewed |  |
| 283 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_12\_1() | Unset | Unreviewed |  |
| 284 | 12.1 | The precedence of operators within expressions should be made explicit. The operand of the sizeof operator should be enclosed in parentheses. | R\_12\_1() | Unset | Unreviewed |  |
| 285 | 12.1 | The precedence of operators within expressions should be made explicit. | R\_12\_1() | Unset | Unreviewed |  |
| 286 | 12.1 | The precedence of operators within expressions should be made explicit. | R\_12\_1() | Unset | Unreviewed |  |
| 288 | 2.2 | There shall be no dead code. | R\_12\_1() | Unset | Unreviewed |  |
| 287 | 12.3 | The comma operator should not be used | R\_12\_1() | Unset | Unreviewed |  |

Table 2.98. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_01.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 707 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'f' has external declarations in multiple files | File Scope | Unset | Unreviewed |  |

Table 2.99. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 291 | 12.2 | The right hand operand of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand. | R\_12\_2() | Unset | Unreviewed |  |
| 292 | 12.2 | The right hand operand of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand. | R\_12\_2() | Unset | Unreviewed |  |

Table 2.100. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 930 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f3 conflicts with the function name f3 (R\_11\_03.c line 28). | File Scope | Unset | Unreviewed |  |
| 299 | 2.2 | There shall be no dead code. Expression has no effect | R\_12\_3() | Unset | Unreviewed |  |
| 301 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type signed on 32 bits) is assigned to an object with a narrower essential type (signed on 16 bits) | R\_12\_3() | Unset | Unreviewed |  |
| 298 | 12.3 | The comma operator should not be used | R\_12\_3() | Unset | Unreviewed |  |
| 864 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 833 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 300 | 12.3 | The comma operator should not be used | R\_12\_3() | Unset | Unreviewed |  |
| 297 | 14.2 | A for loop shall be well-formed. The second expression should contain a comparison with loop counter (p) | R\_12\_3() | Unset | Unreviewed |  |
| 304 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_12\_3() | Unset | Unreviewed |  |
| 302 | 12.3 | The comma operator should not be used | R\_12\_3() | Unset | Unreviewed |  |
| 303 | 14.2 | A for loop shall be well-formed. Third expression: There should be no persistent side effect other than modifying the loop counter (p) | R\_12\_3() | Unset | Unreviewed |  |

Table 2.101. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 293 | 12.4 | Evaluation of constant expressions should not lead to unsigned integer wrap-around. Integer operation result is out of range. | fixed\_pulse() | Unset | Unreviewed |  |
| 918 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f4 conflicts with the function name f4 (R\_09\_04.c line 23). | File Scope | Unset | Unreviewed |  |
| 960 | D4.1 | Run-time failures shall be minimized. Operation + overflows. Valid range: [0 .. 2^32-1] | f4() | Unset | Unreviewed |  |
| 863 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 296 | 14.3 | Controlling expressions shall not be invariant. | g4() | Unset | Unreviewed |  |
| 294 | 12.4 | Evaluation of constant expressions should not lead to unsigned integer wrap-around. Integer operation result is out of range. | g4() | Unset | Unreviewed |  |
| 773 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 295 | 12.4 | Evaluation of constant expressions should not lead to unsigned integer wrap-around. Integer operation result is out of range. | R\_12\_4() | Unset | Unreviewed |  |
| 290 | 20.8 | The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1. | File Scope | Unset | Unreviewed |  |
| 289 | 12.4 | Evaluation of constant expressions should not lead to unsigned integer wrap-around. Integer operation result is out of range. | File Scope | Unset | Unreviewed |  |

Table 2.102. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 306 | 8.13 | A pointer should point to a const-qualified type whenever possible. | fn\_5() | Unset | Unreviewed |  |
| 305 | 12.5 | The sizeof operator shall not have an operand which is a function parameter declared as "array of type". | fn\_5() | Unset | Unreviewed |  |

Table 2.103. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 307 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 714 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.104. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_01\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 902 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f1 conflicts with the function name f1 (R\_07\_04.c line 20). | File Scope | Unset | Unreviewed |  |
| 880 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 309 | 13.1 | Initializer lists shall not contain persistent side effects. | f1() | Unset | Unreviewed |  |
| 802 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 754 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 1027 | D4.1 | Run-time failures shall be minimized. Conversion from unsigned int32 to unsigned int16 overflows. Valid range: [0 .. 65535] | g1() | Unset | Unreviewed |  |
| 791 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 308 | 13.1 | Initializer lists shall not contain persistent side effects. | h1() | Unset | Unreviewed |  |
| 311 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | h1() | Unset | Unreviewed |  |
| 310 | 13.1 | Initializer lists shall not contain persistent side effects. | h1() | Unset | Unreviewed |  |

Table 2.105. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 779 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function g2 conflicts with the function name g2 (R\_07\_04.c line 24). | File Scope | Unset | Unreviewed |  |
| 312 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 313 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of 'i' depends on the order of evaluation | File Scope | Unset | Unreviewed |  |
| 316 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | File Scope | Unset | Unreviewed |  |
| 835 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 964 | D4.14 | The validity of values received from external sources shall be checked. Array index is from an unsecure source. Index may be negative or too big. | R\_13\_2() | Unset | Unreviewed |  |
| 782 | 5.7 | A tag name shall be a unique identifier. variable t conflicts with the tag name t (R\_11\_02.c line 25). | File Scope | Unset | Unreviewed |  |
| 317 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of the expression depends on the order of evaluation because of multiple read accesses to objects of volatile-qualified type | R\_13\_2() | Unset | Unreviewed |  |
| 314 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of 'i' depends on the order of evaluation | R\_13\_2() | Unset | Unreviewed |  |
| 319 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_13\_2() | Unset | Unreviewed |  |
| 318 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of 'i' depends on the order of evaluation | R\_13\_2() | Unset | Unreviewed |  |
| 315 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of 'p' depends on the order of evaluation | R\_13\_2() | Unset | Unreviewed |  |
| 999 | D4.14 | The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory. | R\_13\_2() | Unset | Unreviewed |  |

Table 2.106. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_02.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 723 | 8.3 | All declarations of an object or function shall use the same names and type qualifiers. Global declaration of 'f' function has a type incompatible with its definition. This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_02.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_12\_01.c | File Scope | Unset | Unreviewed |  |
| 684 | 8.6 | An identifier with external linkage shall have exactly one external definition. undefined global variable 'v1' | File Scope | Unset | Unreviewed |  |
| 703 | 8.6 | An identifier with external linkage shall have exactly one external definition. undefined global variable 'v2' | File Scope | Unset | Unreviewed |  |

Table 2.107. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 808 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f3 conflicts with the function name f3 (R\_11\_03.c line 28). | File Scope | Unset | Unreviewed |  |
| 323 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_13\_3() | Unset | Unreviewed |  |
| 320 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_13\_3() | Unset | Unreviewed |  |
| 787 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 737 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 851 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 1003 | D4.14 | The validity of values received from external sources shall be checked. Array index is from an unsecure source. Index may be negative or too big. | R\_13\_3() | Unset | Unreviewed |  |
| 1012 | D4.14 | The validity of values received from external sources shall be checked. Dereferenced pointer is from an unsecure source. Pointer may be NULL or may point to unknown memory. | R\_13\_3() | Unset | Unreviewed |  |
| 321 | 2.2 | There shall be no dead code. | R\_13\_3() | Unset | Unreviewed |  |
| 325 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_13\_3() | Unset | Unreviewed |  |
| 322 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_13\_3() | Unset | Unreviewed |  |
| 326 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_13\_3() | Unset | Unreviewed |  |
| 327 | 14.3 | Controlling expressions shall not be invariant. | R\_13\_3() | Unset | Unreviewed |  |
| 324 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_13\_3() | Unset | Unreviewed |  |
| 329 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | R\_13\_3() | Unset | Unreviewed |  |
| 328 | 14.3 | Controlling expressions shall not be invariant. | R\_13\_3() | Unset | Unreviewed |  |

Table 2.108. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 899 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f4 conflicts with the function name f4 (R\_09\_04.c line 23). | File Scope | Unset | Unreviewed |  |
| 790 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 759 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 955 | D4.14 | The validity of values received from external sources shall be checked. Array index is from an unsecure source. Index may be negative or too big. | R\_13\_4() | Unset | Unreviewed |  |
| 331 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of 'x' depends on the order of evaluation | R\_13\_4() | Unset | Unreviewed |  |
| 959 | D4.14 | The validity of values received from external sources shall be checked. Array index is from an unsecure source. Index may be negative or too big. | R\_13\_4() | Unset | Unreviewed |  |
| 330 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of 'x' depends on the order of evaluation | R\_13\_4() | Unset | Unreviewed |  |
| 333 | 13.4 | The result of an assignment operator should not be used. | R\_13\_4() | Unset | Unreviewed |  |
| 967 | 14.3 | Controlling expressions shall not be invariant. If condition is always false. | R\_13\_4() | Unset | Unreviewed |  |
| 968 | 2.1 | A project shall not contain unreachable code. If-condition always evaluates to false. Dead branch at line 42. | R\_13\_4() | Unset | Unreviewed |  |
| 334 | 13.4 | The result of an assignment operator should not be used. | R\_13\_4() | Unset | Unreviewed |  |
| 978 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_13\_4() | Unset | Unreviewed |  |
| 335 | 13.5 | The right hand operand of a logical && or || operator shall not contain persistent side effects. | R\_13\_4() | Unset | Unreviewed |  |
| 340 | 13.4 | The result of an assignment operator should not be used. | R\_13\_4() | Unset | Unreviewed |  |
| 336 | 13.4 | The result of an assignment operator should not be used. | R\_13\_4() | Unset | Unreviewed |  |
| 338 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of 'b' depends on the order of evaluation | R\_13\_4() | Unset | Unreviewed |  |
| 337 | 13.4 | The result of an assignment operator should not be used. | R\_13\_4() | Unset | Unreviewed |  |
| 332 | 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. The value of 'b' depends on the order of evaluation | R\_13\_4() | Unset | Unreviewed |  |
| 339 | 13.4 | The result of an assignment operator should not be used. | R\_13\_4() | Unset | Unreviewed |  |
| 341 | 13.4 | The result of an assignment operator should not be used. | R\_13\_4() | Unset | Unreviewed |  |

Table 2.109. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_05\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 732 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 760 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 342 | 13.5 | The right hand operand of a logical && or || operator shall not contain persistent side effects. | g5() | Unset | Unreviewed |  |
| 343 | 13.5 | The right hand operand of a logical && or || operator shall not contain persistent side effects. | g5() | Unset | Unreviewed |  |
| 344 | 13.5 | The right hand operand of a logical && or || operator shall not contain persistent side effects. | g5() | Unset | Unreviewed |  |
| 758 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |

Table 2.110. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 822 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 345 | 18.8 | Variable-length array types shall not be used. | f6() | Unset | Unreviewed |  |
| 347 | 13.6 | The operand of the sizeof operator shall not contain any expression which has potential side effects. | f6() | Unset | Unreviewed |  |
| 351 | 18.8 | Variable-length array types shall not be used. | f6() | Unset | Unreviewed |  |
| 346 | 17.8 | A function parameter should not be modified. Parameter n is changed. | f6() | Unset | Unreviewed |  |
| 348 | 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | f6() | Unset | Unreviewed |  |
| 349 | 18.8 | Variable-length array types shall not be used. | f6() | Unset | Unreviewed |  |
| 350 | 18.8 | Variable-length array types shall not be used. | f6() | Unset | Unreviewed |  |
| 814 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 828 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable j conflicts with the variable name j (R\_08\_06\_1.c line 25). | File Scope | Unset | Unreviewed |  |
| 801 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 352 | 13.6 | The operand of the sizeof operator shall not contain any expression which has potential side effects. | R\_13\_6() | Unset | Unreviewed |  |

Table 2.111. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 353 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 693 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_13\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.112. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 354 | 14.1 | A loop counter shall not have essentially floating type. The loop-counter 'f1' shall not have floating type. | R\_14\_1() | Unset | Unreviewed |  |
| 741 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable f1 conflicts with the function name f1 (R\_07\_04.c line 20). | File Scope | Unset | Unreviewed |  |
| 907 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 355 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases. | R\_14\_1() | Unset | Unreviewed |  |
| 356 | 14.1 | A loop counter shall not have essentially floating type. The loop-counter 'f' shall not have floating type. | R\_14\_1() | Unset | Unreviewed |  |
| 357 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Conversion of integer to floating-point number uses an implementation-defined direction of rounding in some cases. | R\_14\_1() | Unset | Unreviewed |  |

Table 2.113. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 358 | 14.2 | A for loop shall be well-formed. Loop counter (i) should not be modified in the body of the loop | R\_14\_2() | Unset | Unreviewed |  |
| 836 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 840 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |

Table 2.114. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 733 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f3 conflicts with the function name f3 (R\_11\_03.c line 28). | File Scope | Unset | Unreviewed |  |
| 844 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 951 | 14.3 | Controlling expressions shall not be invariant. If condition is always false. | f3() | Unset | Unreviewed |  |
| 952 | 2.1 | A project shall not contain unreachable code. If-condition always evaluates to false. Dead branch from line 34 to line 41. | f3() | Unset | Unreviewed |  |
| 867 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function g3 conflicts with the function name g3 (R\_13\_03.c line 27). | File Scope | Unset | Unreviewed |  |
| 359 | 14.3 | Controlling expressions shall not be invariant. Expression is always true. | g3() | Unset | Unreviewed |  |
| 1032 | D4.1 | Run-time failures shall be minimized. Conversion from unsigned int32 to unsigned int8 may overflow. Valid range: [0 .. 255] | g3() | Unset | Unreviewed |  |
| 364 | 14.3 | Controlling expressions shall not be invariant. Expression is always false. | R\_14\_3() | Unset | Unreviewed |  |
| 361 | 14.3 | Controlling expressions shall not be invariant. Expression is always true. | R\_14\_3() | Unset | Unreviewed |  |
| 360 | 14.3 | Controlling expressions shall not be invariant. | R\_14\_3() | Unset | Unreviewed |  |
| 362 | 14.3 | Controlling expressions shall not be invariant. Expression is always false. | R\_14\_3() | Unset | Unreviewed |  |
| 1014 | 2.1 | A project shall not contain unreachable code. If-condition always evaluates to false. Dead branch at line 83. | R\_14\_3() | Unset | Unreviewed |  |
| 363 | 14.3 | Controlling expressions shall not be invariant. Expression is always true. | R\_14\_3() | Unset | Unreviewed |  |
| 1015 | D4.14 | The validity of values received from external sources shall be checked. Loop is controlled by a value from an unsecure source. Loop may be infinite. | R\_14\_3() | Unset | Unreviewed |  |
| 1016 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_14\_3() | Unset | Unreviewed |  |
| 365 | 14.3 | Controlling expressions shall not be invariant. Expression is always true. | R\_14\_3() | Unset | Unreviewed |  |

Table 2.115. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 837 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 366 | 14.4 | The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type. | R\_14\_4() | Unset | Unreviewed |  |
| 367 | 14.4 | The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type. | R\_14\_4() | Unset | Unreviewed |  |

Table 2.116. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 368 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 682 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_14\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.117. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 369 | 15.1 | The goto statement should not be used. | R\_15\_1() | Unset | Unreviewed |  |

Table 2.118. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 854 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable j conflicts with the variable name j (R\_08\_06\_1.c line 25). | File Scope | Unset | Unreviewed |  |
| 370 | 15.1 | The goto statement should not be used. | R\_15\_2() | Unset | Unreviewed |  |
| 371 | 15.1 | The goto statement should not be used. | R\_15\_2() | Unset | Unreviewed |  |
| 372 | 15.2 | The goto statement shall jump to a label declared later in the same function. | R\_15\_2() | Unset | Unreviewed |  |

Table 2.119. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 789 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f1 conflicts with the function name f1 (R\_07\_04.c line 20). | File Scope | Unset | Unreviewed |  |
| 763 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 373 | 15.1 | The goto statement should not be used. | f1() | Unset | Unreviewed |  |
| 375 | 15.3 | Any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement. | f1() | Unset | Unreviewed |  |
| 378 | 15.1 | The goto statement should not be used. | f1() | Unset | Unreviewed |  |
| 374 | 2.1 | A project shall not contain unreachable code. Statement is unreachable | f1() | Unset | Unreviewed |  |
| 377 | 15.1 | The goto statement should not be used. | f1() | Unset | Unreviewed |  |
| 376 | 15.1 | The goto statement should not be used. | f1() | Unset | Unreviewed |  |
| 379 | 15.3 | Any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement. | f1() | Unset | Unreviewed |  |
| 794 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 380 | 15.1 | The goto statement should not be used. | R\_15\_3() | Unset | Unreviewed |  |
| 381 | 15.3 | Any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement. | R\_15\_3() | Unset | Unreviewed |  |

Table 2.120. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 768 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 382 | 15.1 | The goto statement should not be used. | R\_15\_4() | Unset | Unreviewed |  |
| 384 | 15.4 | There should be no more than one break or goto statement used to terminate any iteration statement. | R\_15\_4() | Unset | Unreviewed |  |
| 1007 | D4.14 | The validity of values received from external sources shall be checked. Loop is controlled by a value from an unsecure source. Loop may be infinite. | R\_15\_4() | Unset | Unreviewed |  |
| 1013 | D4.14 | The validity of values received from external sources shall be checked. Loop is controlled by a value from an unsecure source. Loop may be infinite. | R\_15\_4() | Unset | Unreviewed |  |
| 383 | 15.1 | The goto statement should not be used. | R\_15\_4() | Unset | Unreviewed |  |
| 385 | 15.4 | There should be no more than one break or goto statement used to terminate any iteration statement. | R\_15\_4() | Unset | Unreviewed |  |

Table 2.121. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 386 | 15.5 | A function should have a single point of exit at the end. | f() | Unset | Unreviewed |  |
| 915 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |

Table 2.122. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 387 | 15.6 | The body of an iteration-statement or a selection-statement shall be a compound-statement. The statement forming the body of a while statement shall be a compound statement. | R\_15\_6() | Unset | Unreviewed |  |
| 389 | 15.6 | The body of an iteration-statement or a selection-statement shall be a compound-statement. An if (expression) construct shall be followed by a compound statement. | R\_15\_6() | Unset | Unreviewed |  |
| 388 | 15.6 | The body of an iteration-statement or a selection-statement shall be a compound-statement. An if (expression) construct shall be followed by a compound statement. | R\_15\_6() | Unset | Unreviewed |  |
| 390 | 15.6 | The body of an iteration-statement or a selection-statement shall be a compound-statement. The else keyword shall be followed by either a compound statement, or another if statement. | R\_15\_6() | Unset | Unreviewed |  |
| 391 | 15.6 | The body of an iteration-statement or a selection-statement shall be a compound-statement. The statement forming the body of a while statement shall be a compound statement. | R\_15\_6() | Unset | Unreviewed |  |

Table 2.123. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 392 | 15.7 | All if ... else if constructs shall be terminated with an else statement. | R\_15\_7() | Unset | Unreviewed |  |

Table 2.124. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 393 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 696 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_15\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.125. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 394 | 16.1 | All switch statements shall be well-formed. A switch clause shall only contain switch labels and switch clauses, and no other code. | R\_16\_1() | Unset | Unreviewed |  |

Table 2.126. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 879 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 396 | 16.1 | All switch statements shall be well-formed. Switch labels shall appear at the outermost level. An unconditional break statement shall terminate every switch-clause. | R\_16\_2() | Unset | Unreviewed |  |
| 397 | 16.3 | An unconditional break statement shall terminate every switch-clause. | R\_16\_2() | Unset | Unreviewed |  |
| 395 | 16.2 | A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement. | R\_16\_2() | Unset | Unreviewed |  |

Table 2.127. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 795 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 919 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 399 | 16.1 | All switch statements shall be well-formed. An unconditional break statement shall terminate every switch-clause. | R\_16\_3() | Unset | Unreviewed |  |
| 398 | 16.3 | An unconditional break statement shall terminate every switch-clause. | R\_16\_3() | Unset | Unreviewed |  |
| 400 | 16.3 | An unconditional break statement shall terminate every switch-clause. | R\_16\_3() | Unset | Unreviewed |  |
| 401 | 16.3 | An unconditional break statement shall terminate every switch-clause. | R\_16\_3() | Unset | Unreviewed |  |

Table 2.128. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 402 | 10.5 | The value of an expression should not be cast to an inappropriate essential type. | get\_colour() | Unset | Unreviewed |  |
| 823 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 403 | 16.4 | Every switch statement shall have a default label. | R\_16\_4() | Unset | Unreviewed |  |
| 404 | 16.1 | All switch statements shall be well-formed. Every switch statement shall have a default label. | R\_16\_4() | Unset | Unreviewed |  |
| 821 | 5.7 | A tag name shall be a unique identifier. variable colour conflicts with the tag name colour (R\_08\_12.c line 23). | File Scope | Unset | Unreviewed |  |
| 405 | 16.4 | Every switch statement shall have a default label. | R\_16\_4() | Unset | Unreviewed |  |
| 406 | 16.1 | All switch statements shall be well-formed. Every switch statement shall have a default label. | R\_16\_4() | Unset | Unreviewed |  |

Table 2.129. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 797 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 408 | 16.1 | All switch statements shall be well-formed. A default label shall appear as either the first or the last switch label of a switch statement. | R\_16\_5() | Unset | Unreviewed |  |
| 407 | 16.5 | A default label shall appear as either the first or the last switch label of a switch statement. | R\_16\_5() | Unset | Unreviewed |  |

Table 2.130. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 734 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 409 | 16.6 | Every switch statement shall have at least two switch-clauses. Switch is redundant | R\_16\_6() | Unset | Unreviewed |  |
| 410 | 16.1 | All switch statements shall be well-formed. Every switch statement shall have at least two switch-clauses. | R\_16\_6() | Unset | Unreviewed |  |
| 411 | 16.6 | Every switch statement shall have at least two switch-clauses. Switch is redundant | R\_16\_6() | Unset | Unreviewed |  |

Table 2.131. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 729 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 412 | 16.7 | A switch-expression shall not have essentially Boolean type. | R\_16\_7() | Unset | Unreviewed |  |

Table 2.132. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 413 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 699 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_16\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.133. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 777 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function h conflicts with the function name h (R\_02\_02.c line 27). | File Scope | Unset | Unreviewed |  |
| 418 | 17.1 | The features of <stdarg.h> shall not be used. | h() | Unset | Unreviewed |  |
| 414 | 17.1 | The features of <stdarg.h> shall not be used. The macro 'va\_arg' shall not be used. | File Scope | Unset | Unreviewed |  |
| 848 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 799 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 419 | 17.1 | The features of <stdarg.h> shall not be used. | f() | Unset | Unreviewed |  |
| 415 | 17.1 | The features of <stdarg.h> shall not be used. The macro 'va\_start' shall not be used. | File Scope | Unset | Unreviewed |  |
| 416 | 17.1 | The features of <stdarg.h> shall not be used. The macro 'va\_arg' shall not be used. | File Scope | Unset | Unreviewed |  |
| 417 | 17.1 | The features of <stdarg.h> shall not be used. The macro 'va\_arg' shall not be used. | File Scope | Unset | Unreviewed |  |

Table 2.134. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 420 | 17.2 | Functions shall not call themselves, either directly or indirectly. Function fn\_a shall not call itself, either directly or indirectly. | fn\_a() | Unset | Unreviewed |  |

Table 2.135. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 421 | 17.3 | A function shall not be declared implicitly. Function 'power' has no visible prototype when called. | R\_17\_3() | Unset | Unreviewed |  |
| 422 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category signed) is assigned to an object with a different essential type category (floating) | R\_17\_3() | Unset | Unreviewed |  |

Table 2.136. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 425 | 15.5 | A function should have a single point of exit at the end. | absolute() | Unset | Unreviewed |  |
| 423 | 17.4 | All exit paths from a function with non-void return type shall have an explicit return statement with an expression. Missing return statement in non-void function 'absolute'. | absolute() | Unset | Unreviewed |  |
| 424 | 15.5 | A function should have a single point of exit at the end. | lookup() | Unset | Unreviewed |  |
| 426 | 17.4 | All exit paths from a function with non-void return type shall have an explicit return statement with an expression. Missing expression for return statement in non-void function 'lookup'. | lookup() | Unset | Unreviewed |  |
| 853 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |

Table 2.137. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 738 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function fn1 conflicts with the function name fn1 (R\_05\_03.c line 21). | File Scope | Unset | Unreviewed |  |
| 841 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable array1 conflicts with the variable name array1 (R\_08\_11\_2.c line 20). | File Scope | Unset | Unreviewed |  |
| 1019 | D4.1 | Run-time failures shall be minimized. Attempt to dereference pointer outside its bounds. | fn1() | Unset | Unreviewed |  |
| 1020 | 18.1 | A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand. Attempt to dereference pointer outside its bounds. | fn1() | Unset | Unreviewed |  |
| 739 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function fn2 conflicts with the function name fn2 (R\_05\_03.c line 38). | File Scope | Unset | Unreviewed |  |
| 807 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable array2 conflicts with the variable name array2 (R\_08\_11\_2.c line 21). | File Scope | Unset | Unreviewed |  |
| 776 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function fn conflicts with the function name fn (R\_15\_06.h line 19). | File Scope | Unset | Unreviewed |  |
| 427 | 17.5 | The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements. The argument type has 3 elements whereas the parameter type expects 4 elements. | fn() | Unset | Unreviewed |  |

Table 2.138. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 921 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 428 | 17.6 | The declaration of an array parameter shall not contain the static keyword between the [ ]. | File Scope | Unset | Unreviewed |  |
| 817 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 883 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 897 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable v1 conflicts with the variable name v1 (R\_13\_02.h line 31). | File Scope | Unset | Unreviewed |  |
| 857 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable v2 conflicts with the variable name v2 (R\_13\_02.h line 31). | File Scope | Unset | Unreviewed |  |
| 429 | 17.5 | The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements. The argument type has 10 elements whereas the parameter type expects 20 elements. | R\_17\_6() | Unset | Unreviewed |  |

Table 2.139. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 742 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function func conflicts with the function name func (R\_05\_06.c line 21). | File Scope | Unset | Unreviewed |  |
| 889 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 430 | 17.7 | The value returned by a function having non-void return type shall be used. | discarded() | Unset | Unreviewed |  |

Table 2.140. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_08.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 431 | 17.8 | A function parameter should not be modified. Parameter para is changed. | proc() | Unset | Unreviewed |  |
| 796 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f8 conflicts with the function name f8 (R\_08\_08.c line 30). | File Scope | Unset | Unreviewed |  |
| 432 | 17.8 | A function parameter should not be modified. Parameter p is changed. | f8() | Unset | Unreviewed |  |

Table 2.141. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 433 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 705 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_17\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.142. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 931 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f1 conflicts with the function name f1 (R\_07\_04.c line 20). | File Scope | Unset | Unreviewed |  |
| 1021 | D4.14 | The validity of values received from external sources shall be checked. Pointer used in arithmetic operation is from an unsecure source. Pointer may be NULL or point to unknown memory. | f1() | Unset | Unreviewed |  |
| 434 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | f1() | Unset | Unreviewed |  |
| 740 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f2 conflicts with the function name f2 (R\_07\_04.c line 22). | File Scope | Unset | Unreviewed |  |
| 914 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |
| 972 | D4.1 | Run-time failures shall be minimized. Attempt to dereference pointer at index 10. Valid range: [0 .. 9] | f2() | Unset | Unreviewed |  |
| 973 | 18.1 | A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand. Attempt to dereference pointer at index 10. Valid range: [0 .. 9] | f2() | Unset | Unreviewed |  |
| 993 | D4.1 | Run-time failures shall be minimized. Attempt to access to array element -1.  Valid index range: [0 .. 9]. | f2() | Unset | Unreviewed |  |
| 994 | 18.1 | A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand. Attempt to access to array element -1.  Valid index range: [0 .. 9]. | f2() | Unset | Unreviewed |  |
| 997 | D4.1 | Run-time failures shall be minimized. Attempt to access to array element 10.  Valid index range: [0 .. 9]. | f2() | Unset | Unreviewed |  |
| 998 | 18.1 | A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand. Attempt to access to array element 10.  Valid index range: [0 .. 9]. | f2() | Unset | Unreviewed |  |
| 435 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | f2() | Unset | Unreviewed |  |
| 436 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | f2() | Unset | Unreviewed |  |
| 437 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | f2() | Unset | Unreviewed |  |
| 1004 | D4.1 | Run-time failures shall be minimized. Attempt to access to array element 3.  Valid index range: [0 .. 1]. | f2() | Unset | Unreviewed |  |
| 1005 | 18.1 | A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand. Attempt to access to array element 3.  Valid index range: [0 .. 1]. | f2() | Unset | Unreviewed |  |
| 438 | 12.1 | The precedence of operators within expressions should be made explicit. | f2() | Unset | Unreviewed |  |
| 834 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f3 conflicts with the function name f3 (R\_11\_03.c line 28). | File Scope | Unset | Unreviewed |  |
| 439 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | f3() | Unset | Unreviewed |  |
| 440 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | f3() | Unset | Unreviewed |  |

Table 2.143. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 441 | 18.2 | Subtraction between pointers shall only be applied to pointers that address elements of the same array. | R\_18\_2() | Unset | Unreviewed |  |

Table 2.144. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 942 | 14.3 | Controlling expressions shall not be invariant. If condition is always false. | f3\_1() | Unset | Unreviewed |  |
| 943 | 2.1 | A project shall not contain unreachable code. If-condition always evaluates to false. Dead branch at line 28. | f3\_1() | Unset | Unreviewed |  |
| 442 | 18.3 | The relational operators >, >=, < and <= shall not be applied to objects of pointer type except where they point into the same object. | f3\_1() | Unset | Unreviewed |  |
| 988 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | f3\_2() | Unset | Unreviewed |  |
| 443 | 18.3 | The relational operators >, >=, < and <= shall not be applied to objects of pointer type except where they point into the same object. | f3\_2() | Unset | Unreviewed |  |

Table 2.145. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 900 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function fn1 conflicts with the function name fn1 (R\_05\_03.c line 21). | File Scope | Unset | Unreviewed |  |
| 767 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 860 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable index conflicts with the function name index (string.h line 484). | File Scope | Unset | Unreviewed |  |
| 452 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | fn1() | Unset | Unreviewed |  |
| 811 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function fn2 conflicts with the function name fn2 (R\_05\_03.c line 38). | File Scope | Unset | Unreviewed |  |
| 871 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 891 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable j conflicts with the variable name j (R\_08\_06\_1.c line 25). | File Scope | Unset | Unreviewed |  |
| 922 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function fn3 conflicts with the function name fn3 (R\_05\_03.c line 47). | File Scope | Unset | Unreviewed |  |
| 446 | 17.8 | A function parameter should not be modified. Parameter p1 is changed. | fn3() | Unset | Unreviewed |  |
| 447 | 17.8 | A function parameter should not be modified. Parameter p1 is changed. | fn3() | Unset | Unreviewed |  |
| 445 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | fn3() | Unset | Unreviewed |  |
| 1033 | D4.1 | Run-time failures shall be minimized. Attempt to dereference pointer outside of the pointed object at offset 11. | fn3() | Unset | Unreviewed |  |
| 1034 | 18.1 | A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand. Attempt to dereference pointer outside of the pointed object at offset 11. | fn3() | Unset | Unreviewed |  |
| 449 | 17.8 | A function parameter should not be modified. Parameter p2 is changed. | fn3() | Unset | Unreviewed |  |
| 451 | 17.8 | A function parameter should not be modified. Parameter p2 is changed. | fn3() | Unset | Unreviewed |  |
| 450 | 17.8 | A function parameter should not be modified. Parameter p2 is changed. | fn3() | Unset | Unreviewed |  |
| 448 | 17.8 | A function parameter should not be modified. Parameter p2 is changed. | fn3() | Unset | Unreviewed |  |
| 444 | 17.8 | A function parameter should not be modified. Parameter p2 is changed. | fn3() | Unset | Unreviewed |  |
| 453 | 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | fn3() | Unset | Unreviewed |  |

Table 2.146. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 458 | 18.5 | Declarations should contain no more than two levels of pointer nesting. | File Scope | Unset | Unreviewed |  |
| 459 | 18.5 | Declarations should contain no more than two levels of pointer nesting. | function() | Unset | Unreviewed |  |
| 460 | 18.5 | Declarations should contain no more than two levels of pointer nesting. | function() | Unset | Unreviewed |  |
| 850 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable arr conflicts with the variable name arr (R\_19\_01.c line 38). | File Scope | Unset | Unreviewed |  |
| 456 | 18.5 | Declarations should contain no more than two levels of pointer nesting. | File Scope | Unset | Unreviewed |  |
| 457 | 18.5 | Declarations should contain no more than two levels of pointer nesting. | File Scope | Unset | Unreviewed |  |
| 454 | D4.8 | If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden. | File Scope | Unset | Unreviewed |  |
| 843 | 5.7 | A tag name shall be a unique identifier. type s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 455 | 18.5 | Declarations should contain no more than two levels of pointer nesting. | File Scope | Unset | Unreviewed |  |
| 461 | 18.5 | Declarations should contain no more than two levels of pointer nesting. | R\_18\_5() | Unset | Unreviewed |  |
| 462 | 11.9 | The macro NULL shall be the only permitted form of integer null pointer constant. | R\_18\_5() | Unset | Unreviewed |  |

Table 2.147. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_06\_1.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 893 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function func conflicts with the function name func (R\_05\_06.c line 21). | File Scope | Unset | Unreviewed |  |
| 933 | D4.1 | Run-time failures shall be minimized. Address of local memory local\_auto escapes from its scope through its return | func() | Unset | Unreviewed |  |
| 463 | 18.6 | The address of an object with automatic storage shall not be copied to another object that persists after the first object has ceased to exist. | func() | Unset | Unreviewed |  |
| 761 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function g conflicts with the function name g (R\_08\_01.c line 28). | File Scope | Unset | Unreviewed |  |
| 895 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 935 | D4.1 | Run-time failures shall be minimized. Address of local memory u escapes from its scope through sp | f() | Unset | Unreviewed |  |
| 887 | 2.2 | There shall be no dead code. The call to function g has no effect. | File Scope | Unset | Unreviewed |  |
| 464 | 17.8 | A function parameter should not be modified. Parameter u is changed. | f() | Unset | Unreviewed |  |
| 826 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function h conflicts with the function name h (R\_02\_02.c line 27). | File Scope | Unset | Unreviewed |  |
| 800 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 936 | D4.1 | Run-time failures shall be minimized. Address of local memory x escapes from its scope through q | h() | Unset | Unreviewed |  |
| 465 | 18.6 | The address of an object with automatic storage shall not be copied to another object that persists after the first object has ceased to exist. | h() | Unset | Unreviewed |  |

Table 2.148. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 474 | 18.7 | Flexible array members shall not be declared. | File Scope | Unset | Unreviewed |  |
| 477 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | copy() | Unset | Unreviewed |  |
| 475 | D4.12 | Dynamic memory allocation shall not be used. | copy() | Unset | Unreviewed |  |
| 476 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | copy() | Unset | Unreviewed |  |
| 1024 | 9.1 | The value of an object with automatic storage duration shall not be read before it has been set. Field of structure is read before being initialized. | copy() | Unset | Unreviewed |  |
| 940 | 22.1 | All resources obtained dynamically by means of Standard Library functions shall be explicitly released. Pointer 'ss2' points to dynamically allocated memory. It has not been freed before the end of its scope. | R\_18\_7() | Unset | Unreviewed |  |

Table 2.149. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_08.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 783 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f8 conflicts with the function name f8 (R\_08\_08.c line 30). | File Scope | Unset | Unreviewed |  |
| 466 | 18.8 | Variable-length array types shall not be used. | f8() | Unset | Unreviewed |  |
| 905 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function g8 conflicts with the function name g8 (R\_08\_08.c line 36). | File Scope | Unset | Unreviewed |  |
| 471 | 2.7 | There should be no unused parameters in functions. Function h8 has unused parameters. | h8() | Unset | Unreviewed |  |
| 856 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function h8 conflicts with the function name h8 (R\_08\_08.c line 44). | File Scope | Unset | Unreviewed |  |
| 906 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 467 | 18.8 | Variable-length array types shall not be used. | File Scope | Unset | Unreviewed |  |
| 468 | 11.3 | A cast shall not be performed between a pointer to object type and a pointer to a different object type. | h8() | Unset | Unreviewed |  |
| 1030 | D4.1 | Run-time failures shall be minimized. Attempt to dereference pointer at index 180. Valid range: [0 .. 99] | h8() | Unset | Unreviewed |  |
| 1031 | 18.1 | A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand. Attempt to dereference pointer at index 180. Valid range: [0 .. 99] | h8() | Unset | Unreviewed |  |
| 472 | 2.2 | There shall be no dead code. | R\_18\_8() | Unset | Unreviewed |  |
| 469 | 18.8 | Variable-length array types shall not be used. | R\_18\_8() | Unset | Unreviewed |  |
| 470 | 18.8 | Variable-length array types shall not be used. | R\_18\_8() | Unset | Unreviewed |  |
| 888 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable arr conflicts with the variable name arr (R\_19\_01.c line 38). | File Scope | Unset | Unreviewed |  |
| 473 | 11.3 | A cast shall not be performed between a pointer to object type and a pointer to a different object type. | R\_18\_8() | Unset | Unreviewed |  |

Table 2.150. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 478 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 709 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_18\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.151. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 792 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function fn conflicts with the function name fn (R\_15\_06.h line 19). | File Scope | Unset | Unreviewed |  |
| 479 | 19.2 | The union keyword should not be used. | fn() | Unset | Unreviewed |  |
| 846 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 480 | 19.1 | An object shall not be assigned or copied to an overlapping object. | fn() | Unset | Unreviewed |  |
| 824 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 481 | 19.1 | An object shall not be assigned or copied to an overlapping object. Overlap between source and destination of 'memcpy' causes undefined behavior. | f() | Unset | Unreviewed |  |
| 831 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function g conflicts with the function name g (R\_08\_01.c line 28). | File Scope | Unset | Unreviewed |  |
| 872 | 2.2 | There shall be no dead code. The call to function g has no effect. | File Scope | Unset | Unreviewed |  |

Table 2.152. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 903 | 5.7 | A tag name shall be a unique identifier. variable s conflicts with the tag name s (R\_06\_01.c line 24). | File Scope | Unset | Unreviewed |  |
| 482 | 19.2 | The union keyword should not be used. | zext() | Unset | Unreviewed |  |
| 859 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |

Table 2.153. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 490 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 702 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_19\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.154. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 869 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable i conflicts with the variable name i (R\_08\_06\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 486 | 20.1 | #include directives should only be preceded by preprocessor directives or comments. | File Scope | Unset | Unreviewed |  |
| 487 | 20.1 | #include directives should only be preceded by preprocessor directives or comments. | File Scope | Unset | Unreviewed |  |

Table 2.155. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_01.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 813 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable xyz conflicts with the variable name xyz (R\_05\_03.c line 36). | File Scope | Unset | Unreviewed |  |

Table 2.156. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 483 | 3.1 | The character sequences /\* and // shall not be used within a comment. /\* is used within a comment. | File Scope | Unset | Unreviewed |  |
| 484 | 3.1 | The character sequences /\* and // shall not be used within a comment. // is used within a comment. | File Scope | Unset | Unreviewed |  |
| 485 | 20.2 | The ', " or \ characters and the /\* or // character sequences shall not occur in a header file name. | File Scope | Unset | Unreviewed |  |

Table 2.157. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 488 | 20.3 | The #include directive shall be followed by either a or "filename"sequence. '#include' expects "FILENAME" or . | File Scope | Unset | Unreviewed |  |
| 489 | D2.1 | All source files shall compile without any compilation errors. Expected a file name | File Scope | Unset | Unreviewed |  |

Table 2.158. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 513 | D2.1 | All source files shall compile without any compilation errors. Invalid combination of type specifiers | File Scope | Unset | Unreviewed |  |
| 514 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "long" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 492 | 20.4 | A macro shall not be defined with the same name as a keyword. The macro 'int' shall not be redefined. | File Scope | Unset | Unreviewed |  |
| 494 | 20.4 | A macro shall not be defined with the same name as a keyword. The macro 'while' shall not be redefined. | File Scope | Unset | Unreviewed |  |
| 496 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 495 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 493 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 498 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 497 | 20.4 | A macro shall not be defined with the same name as a keyword. The macro 'inline' shall not be redefined. | File Scope | Unset | Unreviewed |  |
| 499 | 20.7 | Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses. Expanded macro parameter 'S1' shall be enclosed in parentheses. | File Scope | Unset | Unreviewed |  |

Table 2.159. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 491 | 20.5 | #undef should not be used. | File Scope | Unset | Unreviewed |  |
| 728 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function f5 conflicts with the function name f5 (R\_13\_05\_1.c line 22). | File Scope | Unset | Unreviewed |  |
| 512 | 17.8 | A function parameter should not be modified. Parameter p is changed. | f5() | Unset | Unreviewed |  |

Table 2.160. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 500 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 516 | 21.6 | The Standard Library input/output functions shall not be used. | File Scope | Unset | Unreviewed |  |
| 502 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 501 | 20.6 | Tokens that look like a preprocessing directive shall not occur within a macro argument. Macro argument shall not look like a preprocessing directive. | File Scope | Unset | Unreviewed |  |
| 503 | 20.6 | Tokens that look like a preprocessing directive shall not occur within a macro argument. Macro argument shall not look like a preprocessing directive. | File Scope | Unset | Unreviewed |  |

Table 2.161. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 504 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 515 | 12.1 | The precedence of operators within expressions should be made explicit. | File Scope | Unset | Unreviewed |  |
| 506 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 507 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 508 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 505 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 509 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 510 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 511 | 20.7 | Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses. Expanded macro parameter 'x' shall be enclosed in parentheses. Expanded macro parameter 'y' shall be enclosed in parentheses. | File Scope | Unset | Unreviewed |  |
| 810 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |

Table 2.162. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_08.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 517 | 20.8 | The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1. | File Scope | Unset | Unreviewed |  |

Table 2.163. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_09.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 518 | 20.9 | All identifiers used in the controlling expression of #if or #elif preprocessing directives shall be #define'd before evaluation. 'M' is not defined. | File Scope | Unset | Unreviewed |  |

Table 2.164. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_10.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 519 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 520 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 521 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 522 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |

Table 2.165. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_11.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 529 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 524 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 523 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 525 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 526 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 531 | D2.1 | All source files shall compile without any compilation errors. Expected a ";" | File Scope | Unset | Unreviewed |  |
| 528 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 527 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 530 | 20.11 | A macro parameter immediately following a # operator shall not immediately be followed by a ## operator. The ## preprocessor operator shall not follow a macro parameter following a # preprocessor operator. | File Scope | Unset | Unreviewed |  |

Table 2.166. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_12.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 532 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 533 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 534 | 20.12 | A macro parameter used as an operand to the # or ## operators, which is itself subject to further macro replacement, shall only be used as an operand to these operators. Expanded macro parameter 'x' is also an operand of '##' operator. | File Scope | Unset | Unreviewed |  |
| 535 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |
| 536 | 20.10 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | File Scope | Unset | Unreviewed |  |
| 827 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable speed conflicts with the variable name speed (R\_05\_03.c line 45). | File Scope | Unset | Unreviewed |  |
| 904 | 5.6 | A typedef name shall be a unique identifier. variable speed conflicts with the typedef name speed (R\_05\_03.c line 49). | File Scope | Unset | Unreviewed |  |

Table 2.167. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_13.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 812 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 537 | 20.13 | A line whose first token is # shall be a valid preprocessing directive. Directive is not syntactically meaningful. | File Scope | Unset | Unreviewed |  |
| 804 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |

Table 2.168. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 539 | 20.14 | All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if, #ifdef or #ifndef directive to which they are related. unterminated conditional directive. | File Scope | Unset | Unreviewed |  |
| 538 | D2.1 | All source files shall compile without any compilation errors. The #endif for this directive is missing. If a token uses non-ASCII characters, use -sources-encoding to specify source encoding. | File Scope | Unset | Unreviewed |  |

Table 2.169. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14\_1.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 32 | D4.10 | Precautions shall be taken in order to prevent the contents of a header file being included more than once. | File Scope | Unset | Unreviewed |  |

Table 2.170. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14\_2.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 35 | D4.10 | Precautions shall be taken in order to prevent the contents of a header file being included more than once. | File Scope | Unset | Unreviewed |  |
| 34 | 20.14 | All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if, #ifdef or #ifndef directive to which they are related. '#endif' not within a conditional. | File Scope | Unset | Unreviewed |  |
| 33 | D2.1 | All source files shall compile without any compilation errors. The #if for this directive is missing. If a token uses non-ASCII characters, use -sources-encoding to specify source encoding. | File Scope | Unset | Unreviewed |  |

Table 2.171. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 549 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 694 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.172. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 540 | 21.1 | #define and #undef shall not be used on a reserved identifier or reserved macro name. The macro '\_\_LINE\_\_' shall not be undefined. | File Scope | Unset | Unreviewed |  |
| 544 | 20.5 | #undef should not be used. | File Scope | Unset | Unreviewed |  |
| 543 | 21.1 | #define and #undef shall not be used on a reserved identifier or reserved macro name. The macro '\_GUARD\_H 1 /\* Non-compliant \*/' shall not be defined. | File Scope | Unset | Unreviewed |  |
| 541 | 20.5 | #undef should not be used. | File Scope | Unset | Unreviewed |  |
| 545 | 21.1 | #define and #undef shall not be used on a reserved identifier or reserved macro name. The macro '\_BUILTIN\_sqrt /\* Non-compliant, also breaks R.20.5 \*/' shall not be undefined. | File Scope | Unset | Unreviewed |  |
| 547 | 21.1 | #define and #undef shall not be used on a reserved identifier or reserved macro name. The macro 'defined' shall not be redefined. | File Scope | Unset | Unreviewed |  |
| 890 | 2.5 | A project should not contain unused macro declarations. Macro defined is not used. | File Scope | Unset | Unreviewed |  |
| 542 | 21.1 | #define and #undef shall not be used on a reserved identifier or reserved macro name. The macro 'errno' shall not be redefined. | File Scope | Unset | Unreviewed |  |
| 546 | D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | File Scope | Unset | Unreviewed |  |

Table 2.173. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 567 | 21.2 | A reserved identifier or macro name shall not be declared. The identifier '\_BUILTIN\_sqrt' shall not be declared. | File Scope | Unset | Unreviewed |  |
| 757 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 548 | 20.1 | #include directives should only be preceded by preprocessor directives or comments. | File Scope | Unset | Unreviewed |  |
| 566 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'memcpy' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 568 | 21.2 | A reserved identifier or macro name shall not be declared. Reserved identifier 'memcpy' shall not be declared. | File Scope | Unset | Unreviewed |  |
| 564 | 8.14 | The restrict type qualifier shall not be used. | File Scope | Unset | Unreviewed |  |
| 565 | 8.14 | The restrict type qualifier shall not be used. | File Scope | Unset | Unreviewed |  |
| 815 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |

Table 2.174. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 552 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | R\_21\_3() | Unset | Unreviewed |  |
| 556 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 559 | D4.12 | Dynamic memory allocation shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 553 | D4.12 | Dynamic memory allocation shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 554 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 558 | D4.12 | Dynamic memory allocation shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 560 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 563 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | R\_21\_3() | Unset | Unreviewed |  |
| 555 | D4.12 | Dynamic memory allocation shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 557 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 561 | D4.12 | Dynamic memory allocation shall not be used. | R\_21\_3() | Unset | Unreviewed |  |
| 562 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | R\_21\_3() | Unset | Unreviewed |  |

Table 2.175. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 550 | 21.4 | The standard header file <setjmp.h> shall not be used. | File Scope | Unset | Unreviewed |  |
| 570 | 21.4 | The standard header file <setjmp.h> shall not be used. | jmpfunc() | Unset | Unreviewed |  |
| 1022 | D4.11 | The validity of values passed to library functions shall be checked. Standard function 'longjmp' may be called with an invalid argument.  first argument (context/environment buffer) may not be a readable pointer. | jmpfunc() | Unset | Unreviewed |  |
| 551 | 21.4 | The standard header file <setjmp.h> shall not be used. The macro 'setjmp' shall not be used. | File Scope | Unset | Unreviewed |  |

Table 2.176. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 569 | 21.5 | The standard header file <signal.h> shall not be used. | File Scope | Unset | Unreviewed |  |

Table 2.177. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 571 | 21.6 | The Standard Library input/output functions shall not be used. | R\_21\_6() | Unset | Unreviewed |  |

Table 2.178. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 572 | 21.7 | The atof, atoi, atol, and atoll functions of <stdlib.h> shall not be used. | R\_21\_7() | Unset | Unreviewed |  |

Table 2.179. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_08.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 573 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. The getenv function uses an implementation-defined list of environment names and method for altering the list. | R\_21\_8() | Unset | Unreviewed |  |
| 575 | 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type. | R\_21\_8() | Unset | Unreviewed |  |
| 576 | 21.8 | The library functions abort, exit and system of <stdlib.h> shall not be used. system invokes a command processor. This can result in exploitable vulnerabilities. | R\_21\_8() | Unset | Unreviewed |  |
| 949 | D4.14 | The validity of values received from external sources shall be checked. Path to the command argument of 'system' is from an unsecure source. | R\_21\_8() | Unset | Unreviewed |  |
| 574 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. The abort function returns an implementation-defined termination status to the host environment. | R\_21\_8() | Unset | Unreviewed |  |
| 578 | 21.8 | The library functions abort, exit and system of <stdlib.h> shall not be used. | R\_21\_8() | Unset | Unreviewed |  |
| 577 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. The exit function returns an implementation-defined termination status to the host environment. | R\_21\_8() | Unset | Unreviewed |  |
| 579 | 21.8 | The library functions abort, exit and system of <stdlib.h> shall not be used. | R\_21\_8() | Unset | Unreviewed |  |

Table 2.180. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_09.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 580 | 8.5 | An external object or function shall be declared once in one and only one file. Function 'compare' that has external linkage should be declared in a header file. | File Scope | Unset | Unreviewed |  |
| 582 | 21.9 | The library functions bsearch and qsort of <stdlib.h> shall not be used. | R\_21\_9() | Unset | Unreviewed |  |
| 581 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | R\_21\_9() | Unset | Unreviewed |  |
| 583 | 21.9 | The library functions bsearch and qsort of <stdlib.h> shall not be used. | R\_21\_9() | Unset | Unreviewed |  |

Table 2.181. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_10.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 589 | 21.10 | The Standard Library time and date functions shall not be used. | R\_21\_10() | Unset | Unreviewed |  |
| 590 | 21.10 | The Standard Library time and date functions shall not be used. | R\_21\_10() | Unset | Unreviewed |  |
| 591 | 21.10 | The Standard Library time and date functions shall not be used. | R\_21\_10() | Unset | Unreviewed |  |

Table 2.182. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_11.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 587 | 21.11 | The standard header file <tgmath.h> shall not be used. | File Scope | Unset | Unreviewed |  |
| 588 | 21.11 | The standard header file <tgmath.h> shall not be used. The macro 'sqrt' shall not be used. | File Scope | Unset | Unreviewed |  |

Table 2.183. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_12.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 825 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 874 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |
| 594 | 21.12 | The exception handling features of <fenv.h> should not be used. | f() | Unset | Unreviewed |  |
| 584 | 21.12 | The exception handling features of <fenv.h> should not be used. The macro 'FE\_DIVBYZERO' shall not be used. | File Scope | Unset | Unreviewed |  |
| 595 | 21.12 | The exception handling features of <fenv.h> should not be used. | f() | Unset | Unreviewed |  |
| 585 | 21.12 | The exception handling features of <fenv.h> should not be used. The macro 'FE\_DIVBYZERO' shall not be used. | File Scope | Unset | Unreviewed |  |
| 592 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Expected "ON", "OFF", or "DEFAULT" | f() | Unset | Unreviewed |  |
| 593 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. Expected "ON", "OFF", or "DEFAULT" | f() | Unset | Unreviewed |  |
| 596 | 21.12 | The exception handling features of <fenv.h> should not be used. | f() | Unset | Unreviewed |  |
| 586 | 21.12 | The exception handling features of <fenv.h> should not be used. The macro 'FE\_OVERFLOW' shall not be used. | File Scope | Unset | Unreviewed |  |
| 805 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable x conflicts with the variable name x (R\_08\_01.c line 22). | File Scope | Unset | Unreviewed |  |

Table 2.184. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_13.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 755 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 947 | 21.13 | Any value passed to a function in <ctype.h> shall be representable as an unsigned char or be the value EOF. Standard function 'isalpha' is called with an invalid argument.  argument is not within either expected range: [0..UCHAR\_MAX] or EOF.  value returned may not fit in range of returned type. | f\_13() | Unset | Unreviewed |  |

Table 2.185. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_14.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 597 | 21.16 | The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type. The first argument to the function memcmp points to type 'char'. | f\_14\_1() | Unset | Unreviewed |  |
| 944 | 21.14 | The Standard Library function memcmp shall not be used to compare null terminated strings. Third argument of 'memcmp' exceeds the length of strings compared and cause irrelevant comparison of bytes after the null-terminator. Use 'strncmp' instead of 'memcmp' or use 'strlen' to determine string lengths for the third argument. | f\_14\_1() | Unset | Unreviewed |  |
| 598 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category character) is assigned to an object with a different essential type category (unsigned) | f\_14\_2() | Unset | Unreviewed |  |
| 599 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category character) is assigned to an object with a different essential type category (unsigned) | f\_14\_2() | Unset | Unreviewed |  |
| 600 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category character) is assigned to an object with a different essential type category (unsigned) | f\_14\_2() | Unset | Unreviewed |  |
| 601 | 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. The expression (of essential type category character) is assigned to an object with a different essential type category (unsigned) | f\_14\_2() | Unset | Unreviewed |  |
| 974 | 21.14 | The Standard Library function memcmp shall not be used to compare null terminated strings. Third argument of 'memcmp' exceeds the length of strings compared and cause irrelevant comparison of bytes after the null-terminator. Use 'strncmp' instead of 'memcmp' or use 'strlen' to determine string lengths for the third argument. | f\_14\_2() | Unset | Unreviewed |  |

Table 2.186. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_14.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 718 | 8.6 | An identifier with external linkage shall have exactly one external definition. undefined global variable 'buffer1' | File Scope | Unset | Unreviewed |  |
| 727 | 8.6 | An identifier with external linkage shall have exactly one external definition. undefined global variable 'buffer2' | File Scope | Unset | Unreviewed |  |

Table 2.187. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_15.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 603 | 8.13 | A pointer should point to a const-qualified type whenever possible. | f\_15() | Unset | Unreviewed |  |
| 602 | 21.15 | The pointer arguments to the Standard Library functions memcpy, memmove and memcmp shall be pointers to qualified or unqualified versions of compatible types. | f\_15() | Unset | Unreviewed |  |

Table 2.188. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_16.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 604 | 19.2 | The union keyword should not be used. | File Scope | Unset | Unreviewed |  |
| 608 | 21.16 | The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type. The first argument to the function memcmp points to type 'const struct S16'. | f\_16() | Unset | Unreviewed |  |
| 611 | 19.2 | The union keyword should not be used. | File Scope | Unset | Unreviewed |  |
| 605 | 19.2 | The union keyword should not be used. | File Scope | Unset | Unreviewed |  |
| 610 | 21.16 | The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type. The first argument to the function memcmp points to type 'const union U16'. | g\_16() | Unset | Unreviewed |  |
| 606 | 19.2 | The union keyword should not be used. | g\_16() | Unset | Unreviewed |  |
| 736 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 612 | 21.16 | The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type. The first argument to the function memcmp points to type 'const char'. | h\_16() | Unset | Unreviewed |  |
| 1010 | 21.14 | The Standard Library function memcmp shall not be used to compare null terminated strings. Third argument of 'memcmp' exceeds the length of strings compared and cause irrelevant comparison of bytes after the null-terminator. Use 'strncmp' instead of 'memcmp' or use 'strlen' to determine string lengths for the third argument. | h\_16() | Unset | Unreviewed |  |
| 609 | 19.2 | The union keyword should not be used. | R\_21\_16() | Unset | Unreviewed |  |
| 607 | 19.2 | The union keyword should not be used. | R\_21\_16() | Unset | Unreviewed |  |
| 746 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable arr conflicts with the variable name arr (R\_19\_01.c line 38). | File Scope | Unset | Unreviewed |  |

Table 2.189. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_17.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 950 | 21.17 | Use of the string handling functions from <string.h> shall not result in accesses beyond the bounds of the objects referenced by their pointer parameters. Standard function 'strcpy' is called with an invalid argument.  first argument (destination) is not the correct size. | f\_17() | Unset | Unreviewed |  |
| 986 | 21.17 | Use of the string handling functions from <string.h> shall not result in accesses beyond the bounds of the objects referenced by their pointer parameters. Standard function 'strlen' is called with an invalid argument.  argument is not a valid string. | g\_17() | Unset | Unreviewed |  |

Table 2.190. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_18.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 613 | 21.16 | The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type. The first argument to the function memcmp points to type 'const char'. | R\_21\_18() | Unset | Unreviewed |  |
| 948 | 21.14 | The Standard Library function memcmp shall not be used to compare null terminated strings. Third argument of 'memcmp' exceeds the length of strings compared and cause irrelevant comparison of bytes after the null-terminator. Use 'strncmp' instead of 'memcmp' or use 'strlen' to determine string lengths for the third argument. | R\_21\_18() | Unset | Unreviewed |  |
| 614 | 21.16 | The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type. The first argument to the function memcmp points to type 'const char'. | R\_21\_18() | Unset | Unreviewed |  |
| 956 | 21.18 | The size\_t argument passed to any function in <string.h> shall have an appropriate value. Standard function 'memcmp' is called with an invalid argument.  first argument is not a memory area that is accessible within the boundary given by the third argument.  second argument may not be a memory area that is accessible within the boundary given by the third argument.  value returned may not fit in range of returned type. | R\_21\_18() | Unset | Unreviewed |  |
| 958 | 21.14 | The Standard Library function memcmp shall not be used to compare null terminated strings. Third argument of 'memcmp' exceeds the length of strings compared and cause irrelevant comparison of bytes after the null-terminator. Use 'strncmp' instead of 'memcmp' or use 'strlen' to determine string lengths for the third argument. | R\_21\_18() | Unset | Unreviewed |  |

Table 2.191. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_19.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 616 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. The setlocale function has an implementation-defined effect when its second argument is not "" or "C". | f\_19() | Unset | Unreviewed |  |
| 617 | 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type. | f\_19() | Unset | Unreviewed |  |
| 615 | 11.9 | The macro NULL shall be the only permitted form of integer null pointer constant. | f\_19() | Unset | Unreviewed |  |
| 618 | 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type. | f\_19() | Unset | Unreviewed |  |
| 953 | 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type. Attempt to modify the internal buffer returned from a non-reentrant standard function. Modification attempt can fail, or the modified object can be overwritten by a subsequent call to the same standard function. | f\_19() | Unset | Unreviewed |  |
| 961 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | f\_19() | Unset | Unreviewed |  |
| 963 | 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type. Attempt to modify the internal buffer returned from a non-reentrant standard function. Modification attempt can fail, or the modified object can be overwritten by a subsequent call to the same standard function. | f\_19() | Unset | Unreviewed |  |
| 621 | 7.4 | A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char". | f\_19() | Unset | Unreviewed |  |
| 622 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. The setlocale function has an implementation-defined effect when its second argument is not "" or "C". | g\_19() | Unset | Unreviewed |  |
| 619 | 11.9 | The macro NULL shall be the only permitted form of integer null pointer constant. | g\_19() | Unset | Unreviewed |  |
| 990 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | g\_19() | Unset | Unreviewed |  |
| 1001 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | g\_19() | Unset | Unreviewed |  |
| 1002 | 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type. Attempt to modify the internal buffer returned from a non-reentrant standard function. Modification attempt can fail, or the modified object can be overwritten by a subsequent call to the same standard function. | g\_19() | Unset | Unreviewed |  |
| 620 | 7.4 | A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char". | g\_19() | Unset | Unreviewed |  |
| 1008 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | h\_19() | Unset | Unreviewed |  |
| 1011 | 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type. Attempt to modify the internal buffer returned from a non-reentrant standard function. Modification attempt can fail, or the modified object can be overwritten by a subsequent call to the same standard function. | h\_19() | Unset | Unreviewed |  |

Table 2.192. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_20.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 832 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable copy conflicts with the function name copy (R\_18\_07.c line 28). | File Scope | Unset | Unreviewed |  |
| 623 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. The setlocale function has an implementation-defined effect when its second argument is not "" or "C". | R\_21\_20() | Unset | Unreviewed |  |
| 625 | 11.9 | The macro NULL shall be the only permitted form of integer null pointer constant. | R\_21\_20() | Unset | Unreviewed |  |
| 624 | D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. The setlocale function has an implementation-defined effect when its second argument is not "" or "C". | R\_21\_20() | Unset | Unreviewed |  |
| 626 | 21.6 | The Standard Library input/output functions shall not be used. | R\_21\_20() | Unset | Unreviewed |  |
| 981 | 21.20 | The pointer returned by the Standard Library functions asctime, ctime, gmtime, localtime, localeconv, getenv, setlocale or strerror shall not be used following a subsequent call to the same function. Argument of 'printf' points to a buffer returned from 'setlocale' or 'localeconv'. The buffer is static and has been overwritten in a subsequent call to 'setlocale'. To fix, copy the static buffer into a local variable and use this copy as argument. | R\_21\_20() | Unset | Unreviewed |  |
| 627 | 21.6 | The Standard Library input/output functions shall not be used. | R\_21\_20() | Unset | Unreviewed |  |
| 628 | 21.6 | The Standard Library input/output functions shall not be used. | R\_21\_20() | Unset | Unreviewed |  |

Table 2.193. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 629 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 704 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.194. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_01.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 630 | D4.12 | Dynamic memory allocation shall not be used. | R\_22\_1\_heap() | Unset | Unreviewed |  |
| 631 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | R\_22\_1\_heap() | Unset | Unreviewed |  |
| 635 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_1\_file1() | Unset | Unreviewed |  |
| 632 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_1\_file1() | Unset | Unreviewed |  |
| 938 | 22.1 | All resources obtained dynamically by means of Standard Library functions shall be explicitly released. Stream 'fp1' has not been closed before the end of its scope. | R\_22\_1\_file1() | Unset | Unreviewed |  |
| 633 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_1\_file2() | Unset | Unreviewed |  |
| 637 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_1\_file2() | Unset | Unreviewed |  |
| 634 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_1\_file2() | Unset | Unreviewed |  |
| 941 | 22.1 | All resources obtained dynamically by means of Standard Library functions shall be explicitly released. Stream 'fp' has not been closed before new assignment. | R\_22\_1\_file2() | Unset | Unreviewed |  |
| 636 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_1\_file2() | Unset | Unreviewed |  |
| 638 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_1\_file2() | Unset | Unreviewed |  |

Table 2.195. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_02.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 892 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function fn conflicts with the function name fn (R\_15\_06.h line 19). | File Scope | Unset | Unreviewed |  |
| 769 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable a conflicts with the variable name a (R\_08\_05\_1.c line 19). | File Scope | Unset | Unreviewed |  |
| 646 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | fn() | Unset | Unreviewed |  |
| 651 | D4.12 | Dynamic memory allocation shall not be used. | fn() | Unset | Unreviewed |  |
| 945 | 22.2 | A block of memory shall only be freed if it was allocated by means of a Standard Library function. Freed pointer does not come from an adapted allocation. | fn() | Unset | Unreviewed |  |
| 784 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. function g conflicts with the function name g (R\_08\_01.c line 28). | File Scope | Unset | Unreviewed |  |
| 657 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | g() | Unset | Unreviewed |  |
| 645 | D4.12 | Dynamic memory allocation shall not be used. | g() | Unset | Unreviewed |  |
| 647 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | g() | Unset | Unreviewed |  |
| 648 | D4.12 | Dynamic memory allocation shall not be used. | g() | Unset | Unreviewed |  |
| 649 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | g() | Unset | Unreviewed |  |
| 652 | D4.12 | Dynamic memory allocation shall not be used. | g() | Unset | Unreviewed |  |
| 653 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | g() | Unset | Unreviewed |  |
| 937 | 22.2 | A block of memory shall only be freed if it was allocated by means of a Standard Library function. Pointer is already deallocated. | g() | Unset | Unreviewed |  |
| 658 | 11.5 | A conversion should not be performed from pointer to void into pointer to object. | g() | Unset | Unreviewed |  |
| 655 | D4.12 | Dynamic memory allocation shall not be used. | g() | Unset | Unreviewed |  |
| 656 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | g() | Unset | Unreviewed |  |
| 939 | 22.2 | A block of memory shall only be freed if it was allocated by means of a Standard Library function. Pointer is already deallocated. | g() | Unset | Unreviewed |  |
| 650 | D4.12 | Dynamic memory allocation shall not be used. | g() | Unset | Unreviewed |  |
| 654 | 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | g() | Unset | Unreviewed |  |
| 852 | 2.2 | There shall be no dead code. The call to function g has no effect. | File Scope | Unset | Unreviewed |  |

Table 2.196. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_03.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 925 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. function fn3 conflicts with the function name fn3 (R\_05\_03.c line 47). | File Scope | Unset | Unreviewed |  |
| 639 | 21.6 | The Standard Library input/output functions shall not be used. | fn3() | Unset | Unreviewed |  |
| 640 | 21.6 | The Standard Library input/output functions shall not be used. | fn3() | Unset | Unreviewed |  |
| 934 | 22.3 | The same file shall not be open for read and write access at the same time on different streams. Stream '\_\_tmp\_string\_119' has not been closed before the end of its scope. | fn3() | Unset | Unreviewed |  |
| 641 | 21.6 | The Standard Library input/output functions shall not be used. | fn3() | Unset | Unreviewed |  |
| 642 | 21.6 | The Standard Library input/output functions shall not be used. | fn3() | Unset | Unreviewed |  |
| 643 | 21.6 | The Standard Library input/output functions shall not be used. | fn3() | Unset | Unreviewed |  |
| 644 | 21.6 | The Standard Library input/output functions shall not be used. | fn3() | Unset | Unreviewed |  |

Table 2.197. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_04.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 659 | 21.6 | The Standard Library input/output functions shall not be used. | fn4() | Unset | Unreviewed |  |
| 660 | 21.6 | The Standard Library input/output functions shall not be used. | fn4() | Unset | Unreviewed |  |
| 946 | 22.4 | There shall be no attempt to write to a stream which has been opened as read-only. First argument of 'fprintf' expects a writeable stream whereas the stream has been opened as read-only. | fn4() | Unset | Unreviewed |  |
| 661 | 21.6 | The Standard Library input/output functions shall not be used. | fn4() | Unset | Unreviewed |  |

Table 2.198. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_05.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 662 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_5() | Unset | Unreviewed |  |
| 839 | 5.9 | Identifiers that define objects or functions with internal linkage should be unique. variable f3 conflicts with the function name f3 (R\_11\_03.c line 28). | File Scope | Unset | Unreviewed |  |
| 663 | 22.5 | A pointer to a FILE object shall not be dereferenced. | R\_22\_5() | Unset | Unreviewed |  |
| 664 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_5() | Unset | Unreviewed |  |

Table 2.199. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_06.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 665 | 21.6 | The Standard Library input/output functions shall not be used. | fn6() | Unset | Unreviewed |  |
| 666 | 21.6 | The Standard Library input/output functions shall not be used. | fn6() | Unset | Unreviewed |  |
| 667 | 21.6 | The Standard Library input/output functions shall not be used. | fn6() | Unset | Unreviewed |  |
| 969 | 22.6 | The value of a pointer to a FILE shall not be used after the associated stream has been closed. First argument of 'fprintf' expects an opened stream whereas the stream has been previously closed. | fn6() | Unset | Unreviewed |  |

Table 2.200. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_07.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 668 | 21.6 | The Standard Library input/output functions shall not be used. | f\_7() | Unset | Unreviewed |  |
| 954 | 22.7 | The macro EOF shall only be compared with the unmodified return value from any Standard Library function capable of returning EOF. Comparison with EOF can cause unexpected results. The return value from a fgetc-family function has undergone conversions that make the character UCHAR\_MAX same as EOF. To fix, perform the comparison before conversion. | f\_7() | Unset | Unreviewed |  |
| 670 | 21.6 | The Standard Library input/output functions shall not be used. | g\_7() | Unset | Unreviewed |  |
| 669 | 21.6 | The Standard Library input/output functions shall not be used. | g\_7() | Unset | Unreviewed |  |
| 984 | D4.7 | If a function returns error information, then that error information shall be tested. Return value of a sensitive function saved to 'ch', but not checked. Check the value of 'ch' before using it. | g\_7() | Unset | Unreviewed |  |
| 985 | D4.7 | If a function returns error information, then that error information shall be tested. Return value of a sensitive function saved to 'ch', but not checked in the scope. Before return, check 'ch' for errors or other state information. | g\_7() | Unset | Unreviewed |  |
| 671 | 21.6 | The Standard Library input/output functions shall not be used. | h\_7() | Unset | Unreviewed |  |

Table 2.201. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_08.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 1029 | 22.8 | The value of errno shall be set to zero prior to a call to an errno-setting-function. 'errno' is not initialized to zero before call to 'strtoul' and can contain values from a previous call. | R\_22\_8() | Unset | Unreviewed |  |
| 971 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_22\_8() | Unset | Unreviewed |  |
| 991 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_22\_8() | Unset | Unreviewed |  |
| 1006 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_22\_8() | Unset | Unreviewed |  |

Table 2.202. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_09.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 673 | 21.6 | The Standard Library input/output functions shall not be used. | handleError() | Unset | Unreviewed |  |
| 977 | 22.9 | The value of errno shall be tested against zero after calling an errno-setting-function. 'errno' is not checked for error conditions. For more secure code, \* Set 'errno' to zero before call to 'strtoul'. \* Test return value of 'strtoul' for ULONG\_MAX and test 'errno' for ERANGE after the call. | f\_9() | Unset | Unreviewed |  |
| 992 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | f\_9() | Unset | Unreviewed |  |
| 1009 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | f\_9() | Unset | Unreviewed |  |
| 764 | 5.8 | Identifiers that define objects or functions with external linkage shall be unique. variable f conflicts with the function name f (R\_08\_01.c line 25). | File Scope | Unset | Unreviewed |  |
| 672 | 21.6 | The Standard Library input/output functions shall not be used. | g\_9() | Unset | Unreviewed |  |
| 674 | 11.8 | A cast shall not remove any const or volatile qualification from the type pointed to by a pointer. Cast from type 'const FILE \*' to type 'FILE \*' removes qualifiers. | g\_9() | Unset | Unreviewed |  |
| 677 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_9() | Unset | Unreviewed |  |
| 675 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_9() | Unset | Unreviewed |  |
| 1017 | 22.10 | The value of errno shall only be tested when the last function to be called was an errno-setting-function. 'fgetpos' sets error indicators (nonzero) along with errno. Error indicators are not checked for errors. | R\_22\_9() | Unset | Unreviewed |  |
| 678 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_9() | Unset | Unreviewed |  |
| 676 | 21.6 | The Standard Library input/output functions shall not be used. | R\_22\_9() | Unset | Unreviewed |  |

Table 2.203. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_10.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 679 | 21.7 | The atof, atoi, atol, and atoll functions of <stdlib.h> shall not be used. | R\_22\_10() | Unset | Unreviewed |  |
| 965 | 14.3 | Controlling expressions shall not be invariant. If condition is always true. | R\_22\_10() | Unset | Unreviewed |  |
| 966 | 22.10 | The value of errno shall only be tested when the last function to be called was an errno-setting-function. 'atof' does not set 'errno' on errors, if ISO standard is used. | R\_22\_10() | Unset | Unreviewed |  |

Table 2.204. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_system.c

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 680 | D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. "int" doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | File Scope | Unset | Unreviewed |  |
| 719 | 8.6 | An identifier with external linkage shall have exactly one external definition. procedure 'main' multiply defined: This defect occurs when linking the 2 translation units: D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_22\_system.c D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\D\_01\_system.c | File Scope | Unset | Unreviewed |  |

Table 2.205. D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\mc3\_types.h

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Guideline** | **Message** | **Function** | **Severity** | **Status** | **Comment** |
| 788 | 2.4 | A project should not contain unused tag declarations. Tag enum\_tag is not used. | File Scope | Unset | Unreviewed |  |

Chapter 3. Appendix 1 - Configuration Settings

Polyspace Settings

|  |  |
| --- | --- |
| **Option** | **Value** |
| -author | YangLiMin |
| -bug-finder | true |
| -checkers |  |
| -compiler | generic |
| -date | 02/12/2024 |
| -disable-checkers | all |
| -dos | true |
| -I | D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012 |
| -import-comments | C:\Users\yanglimin\Documents\Polyspace\_Workspace\mis2012\Module\_1\BF\_Result\_1 |
| -lang | C |
| -misra3 | all |
| -prog | mis2012 |
| -report-output-format | Word |
| -report-template | Polyspace-Doc\BugFinder.rpt |
| -results-dir | C:\Users\yanglimin\Documents\Polyspace\_Workspace\mis2012\Module\_1\BF\_Result\_1 |
| -target | i386 |
| -verif-version | 1.0 |

Coding Standard Configuration

Table 3.1. MISRA C:2012 Guidelines Configuration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Guideline** | **Description** | **Mode** | **Comment** | **Enabled** |
| D1.1 | Any implementation-defined behaviour on which the output of the program depends shall be documented and understood. | required | - | yes |
| D2.1 | All source files shall compile without any compilation errors. | required | - | yes |
| D3.1 | All code shall be traceable to documented requirements. | required | Not enforceable | no |
| D4.1 | Run-time failures shall be minimized. | required | - | yes |
| D4.2 | All usage of assembly language should be documented. | advisory | Not enforceable | no |
| D4.3 | Assembly language shall be encapsulated and isolated. | required | - | yes |
| D4.4 | Sections of code should not be "commented out". | advisory | Not implemented | no |
| D4.5 | Identifiers in the same name space with overlapping visibility should be typographically unambiguous. | advisory | - | yes |
| D4.6 | typedefs that indicate size and signedness should be used in place of the basic numerical types. | advisory | - | yes |
| D4.7 | If a function returns error information, then that error information shall be tested. | required | - | yes |
| D4.8 | If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden. | advisory | - | yes |
| D4.9 | A function should be used in preference to a function-like macro where they are interchangeable. | advisory | - | yes |
| D4.10 | Precautions shall be taken in order to prevent the contents of a header file being included more than once. | required | - | yes |
| D4.11 | The validity of values passed to library functions shall be checked. | required | - | yes |
| D4.12 | Dynamic memory allocation shall not be used. | required | - | yes |
| D4.13 | Functions which are designed to provide operations on a resource should be called in an appropriate sequence. | advisory | - | yes |
| D4.14 | The validity of values received from external sources shall be checked. | required | - | yes |
| 1.1 | The program shall contain no violations of the standard C syntax and constraints, and shall not exceed the implementation's translation limits. | required | - | yes |
| 1.2 | Language extensions should not be used. | advisory | - | yes |
| 1.3 | There shall be no occurrence of undefined or critical unspecified behaviour. | required | - | yes |
| 2.1 | A project shall not contain unreachable code. | required | - | yes |
| 2.2 | There shall be no dead code. | required | - | yes |
| 2.3 | A project should not contain unused type declarations. | advisory | - | yes |
| 2.4 | A project should not contain unused tag declarations. | advisory | - | yes |
| 2.5 | A project should not contain unused macro declarations. | advisory | - | yes |
| 2.6 | A function should not contain unused label declarations. | advisory | - | yes |
| 2.7 | There should be no unused parameters in functions. | advisory | - | yes |
| 3.1 | The character sequences /\* and // shall not be used within a comment. | required | - | yes |
| 3.2 | Line-splicing shall not be used in // comments. | required | - | yes |
| 4.1 | Octal and hexadecimal escape sequences shall be terminated. | required | - | yes |
| 4.2 | Trigraphs should not be used. | advisory | - | yes |
| 5.1 | External identifiers shall be distinct. | required | - | yes |
| 5.2 | Identifiers declared in the same scope and name space shall be distinct. | required | - | yes |
| 5.3 | An identifier declared in an inner scope shall not hide an identifier declared in an outer scope. | required | - | yes |
| 5.4 | Macro identifiers shall be distinct. | required | - | yes |
| 5.5 | Identifiers shall be distinct from macro names. | required | - | yes |
| 5.6 | A typedef name shall be a unique identifier. | required | - | yes |
| 5.7 | A tag name shall be a unique identifier. | required | - | yes |
| 5.8 | Identifiers that define objects or functions with external linkage shall be unique. | required | - | yes |
| 5.9 | Identifiers that define objects or functions with internal linkage should be unique. | advisory | - | yes |
| 6.1 | Bit-fields shall only be declared with an appropriate type. | required | - | yes |
| 6.2 | Single-bit named bit fields shall not be of a signed type. | required | - | yes |
| 7.1 | Octal constants shall not be used. | required | - | yes |
| 7.2 | A "u" or "U" suffix shall be applied to all integer constants that are represented in an unsigned type. | required | - | yes |
| 7.3 | The lowercase character "l" shall not be used in a literal suffix. | required | - | yes |
| 7.4 | A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char". | required | - | yes |
| 8.1 | Types shall be explicitly specified. | required | - | yes |
| 8.2 | Function types shall be in prototype form with named parameters. | required | - | yes |
| 8.3 | All declarations of an object or function shall use the same names and type qualifiers. | required | - | yes |
| 8.4 | A compatible declaration shall be visible when an object or function with external linkage is defined. | required | - | yes |
| 8.5 | An external object or function shall be declared once in one and only one file. | required | - | yes |
| 8.6 | An identifier with external linkage shall have exactly one external definition. | required | - | yes |
| 8.7 | Functions and objects should not be defined with external linkage if they are referenced in only one translation unit. | advisory | - | yes |
| 8.8 | The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage. | required | - | yes |
| 8.9 | An object should be defined at block scope if its identifier only appears in a single function. | advisory | - | yes |
| 8.10 | An inline function shall be declared with the static storage class. | required | - | yes |
| 8.11 | When an array with external linkage is declared, its size should be explicitly specified. | advisory | - | yes |
| 8.12 | Within an enumerator list, the value of an implicitly-specified enumeration constant shall be unique. | required | - | yes |
| 8.13 | A pointer should point to a const-qualified type whenever possible. | advisory | - | yes |
| 8.14 | The restrict type qualifier shall not be used. | required | - | yes |
| 9.1 | The value of an object with automatic storage duration shall not be read before it has been set. | mandatory | - | yes |
| 9.2 | The initializer for an aggregate or union shall be enclosed in braces. | required | - | yes |
| 9.3 | Arrays shall not be partially initialized. | required | - | yes |
| 9.4 | An element of an object shall not be initialized more than once. | required | - | yes |
| 9.5 | Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly. | required | - | yes |
| 10.1 | Operands shall not be of an inappropriate essential type. | required | - | yes |
| 10.2 | Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations. | required | - | yes |
| 10.3 | The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category. | required | - | yes |
| 10.4 | Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category. | required | - | yes |
| 10.5 | The value of an expression should not be cast to an inappropriate essential type. | advisory | - | yes |
| 10.6 | The value of a composite expression shall not be assigned to an object with wider essential type. | required | - | yes |
| 10.7 | If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type. | required | - | yes |
| 10.8 | The value of a composite expression shall not be cast to a different essential type category or a wider essential type. | required | - | yes |
| 11.1 | Conversions shall not be performed between a pointer to a function and any other type. | required | - | yes |
| 11.2 | Conversions shall not be performed between a pointer to an incomplete type and any other type. | required | - | yes |
| 11.3 | A cast shall not be performed between a pointer to object type and a pointer to a different object type. | required | - | yes |
| 11.4 | A conversion should not be performed between a pointer to object and an integer type. | advisory | - | yes |
| 11.5 | A conversion should not be performed from pointer to void into pointer to object. | advisory | - | yes |
| 11.6 | A cast shall not be performed between pointer to void and an arithmetic type. | required | - | yes |
| 11.7 | A cast shall not be performed between pointer to object and a non-integer arithmetic type. | required | - | yes |
| 11.8 | A cast shall not remove any const or volatile qualification from the type pointed to by a pointer. | required | - | yes |
| 11.9 | The macro NULL shall be the only permitted form of integer null pointer constant. | required | - | yes |
| 12.1 | The precedence of operators within expressions should be made explicit. | advisory | - | yes |
| 12.2 | The right hand operand of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand. | required | - | yes |
| 12.3 | The comma operator should not be used | advisory | - | yes |
| 12.4 | Evaluation of constant expressions should not lead to unsigned integer wrap-around. | advisory | - | yes |
| 12.5 | The sizeof operator shall not have an operand which is a function parameter declared as "array of type". | mandatory | - | yes |
| 13.1 | Initializer lists shall not contain persistent side effects. | required | - | yes |
| 13.2 | The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders. | required | - | yes |
| 13.3 | A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator. | advisory | - | yes |
| 13.4 | The result of an assignment operator should not be used. | advisory | - | yes |
| 13.5 | The right hand operand of a logical && or || operator shall not contain persistent side effects. | required | - | yes |
| 13.6 | The operand of the sizeof operator shall not contain any expression which has potential side effects. | mandatory | - | yes |
| 14.1 | A loop counter shall not have essentially floating type. | required | - | yes |
| 14.2 | A for loop shall be well-formed. | required | - | yes |
| 14.3 | Controlling expressions shall not be invariant. | required | - | yes |
| 14.4 | The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type. | required | - | yes |
| 15.1 | The goto statement should not be used. | advisory | - | yes |
| 15.2 | The goto statement shall jump to a label declared later in the same function. | required | - | yes |
| 15.3 | Any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement. | required | - | yes |
| 15.4 | There should be no more than one break or goto statement used to terminate any iteration statement. | advisory | - | yes |
| 15.5 | A function should have a single point of exit at the end. | advisory | - | yes |
| 15.6 | The body of an iteration-statement or a selection-statement shall be a compound-statement. | required | - | yes |
| 15.7 | All if ... else if constructs shall be terminated with an else statement. | required | - | yes |
| 16.1 | All switch statements shall be well-formed. | required | - | yes |
| 16.2 | A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement. | required | - | yes |
| 16.3 | An unconditional break statement shall terminate every switch-clause. | required | - | yes |
| 16.4 | Every switch statement shall have a default label. | required | - | yes |
| 16.5 | A default label shall appear as either the first or the last switch label of a switch statement. | required | - | yes |
| 16.6 | Every switch statement shall have at least two switch-clauses. | required | - | yes |
| 16.7 | A switch-expression shall not have essentially Boolean type. | required | - | yes |
| 17.1 | The features of <stdarg.h> shall not be used. | required | - | yes |
| 17.2 | Functions shall not call themselves, either directly or indirectly. | required | - | yes |
| 17.3 | A function shall not be declared implicitly. | mandatory | - | yes |
| 17.4 | All exit paths from a function with non-void return type shall have an explicit return statement with an expression. | mandatory | - | yes |
| 17.5 | The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements. | advisory | - | yes |
| 17.6 | The declaration of an array parameter shall not contain the static keyword between the [ ]. | mandatory | - | yes |
| 17.7 | The value returned by a function having non-void return type shall be used. | required | - | yes |
| 17.8 | A function parameter should not be modified. | advisory | - | yes |
| 18.1 | A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand. | required | - | yes |
| 18.2 | Subtraction between pointers shall only be applied to pointers that address elements of the same array. | required | - | yes |
| 18.3 | The relational operators >, >=, < and <= shall not be applied to objects of pointer type except where they point into the same object. | required | - | yes |
| 18.4 | The +, -, += and -= operators should not be applied to an expression of pointer type. | advisory | - | yes |
| 18.5 | Declarations should contain no more than two levels of pointer nesting. | advisory | - | yes |
| 18.6 | The address of an object with automatic storage shall not be copied to another object that persists after the first object has ceased to exist. | required | - | yes |
| 18.7 | Flexible array members shall not be declared. | required | - | yes |
| 18.8 | Variable-length array types shall not be used. | required | - | yes |
| 19.1 | An object shall not be assigned or copied to an overlapping object. | mandatory | - | yes |
| 19.2 | The union keyword should not be used. | advisory | - | yes |
| 20.1 | #include directives should only be preceded by preprocessor directives or comments. | advisory | - | yes |
| 20.2 | The ', " or \ characters and the /\* or // character sequences shall not occur in a header file name. | required | - | yes |
| 20.3 | The #include directive shall be followed by either a <filename> or "filename"sequence. | required | - | yes |
| 20.4 | A macro shall not be defined with the same name as a keyword. | required | - | yes |
| 20.5 | #undef should not be used. | advisory | - | yes |
| 20.6 | Tokens that look like a preprocessing directive shall not occur within a macro argument. | required | - | yes |
| 20.7 | Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses. | required | - | yes |
| 20.8 | The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1. | required | - | yes |
| 20.9 | All identifiers used in the controlling expression of #if or #elif preprocessing directives shall be #define'd before evaluation. | required | - | yes |
| 20.10 | The # and ## preprocessor operators should not be used. | advisory | - | yes |
| 20.11 | A macro parameter immediately following a # operator shall not immediately be followed by a ## operator. | required | - | yes |
| 20.12 | A macro parameter used as an operand to the # or ## operators, which is itself subject to further macro replacement, shall only be used as an operand to these operators. | required | - | yes |
| 20.13 | A line whose first token is # shall be a valid preprocessing directive. | required | - | yes |
| 20.14 | All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if, #ifdef or #ifndef directive to which they are related. | required | - | yes |
| 21.1 | #define and #undef shall not be used on a reserved identifier or reserved macro name. | required | - | yes |
| 21.2 | A reserved identifier or macro name shall not be declared. | required | - | yes |
| 21.3 | The memory allocation and deallocation functions of <stdlib.h> shall not be used. | required | - | yes |
| 21.4 | The standard header file <setjmp.h> shall not be used. | required | - | yes |
| 21.5 | The standard header file <signal.h> shall not be used. | required | - | yes |
| 21.6 | The Standard Library input/output functions shall not be used. | required | - | yes |
| 21.7 | The atof, atoi, atol, and atoll functions of <stdlib.h> shall not be used. | required | - | yes |
| 21.8 | The library functions abort, exit and system of <stdlib.h> shall not be used. | required | - | yes |
| 21.9 | The library functions bsearch and qsort of <stdlib.h> shall not be used. | required | - | yes |
| 21.10 | The Standard Library time and date functions shall not be used. | required | - | yes |
| 21.11 | The standard header file <tgmath.h> shall not be used. | required | - | yes |
| 21.12 | The exception handling features of <fenv.h> should not be used. | advisory | - | yes |
| 21.13 | Any value passed to a function in <ctype.h> shall be representable as an unsigned char or be the value EOF. | mandatory | - | yes |
| 21.14 | The Standard Library function memcmp shall not be used to compare null terminated strings. | required | - | yes |
| 21.15 | The pointer arguments to the Standard Library functions memcpy, memmove and memcmp shall be pointers to qualified or unqualified versions of compatible types. | required | - | yes |
| 21.16 | The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type. | required | - | yes |
| 21.17 | Use of the string handling functions from <string.h> shall not result in accesses beyond the bounds of the objects referenced by their pointer parameters. | mandatory | - | yes |
| 21.18 | The size\_t argument passed to any function in <string.h> shall have an appropriate value. | mandatory | - | yes |
| 21.19 | The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type. | mandatory | - | yes |
| 21.20 | The pointer returned by the Standard Library functions asctime, ctime, gmtime, localtime, localeconv, getenv, setlocale or strerror shall not be used following a subsequent call to the same function. | mandatory | - | yes |
| 22.1 | All resources obtained dynamically by means of Standard Library functions shall be explicitly released. | required | - | yes |
| 22.2 | A block of memory shall only be freed if it was allocated by means of a Standard Library function. | mandatory | - | yes |
| 22.3 | The same file shall not be open for read and write access at the same time on different streams. | required | - | yes |
| 22.4 | There shall be no attempt to write to a stream which has been opened as read-only. | mandatory | - | yes |
| 22.5 | A pointer to a FILE object shall not be dereferenced. | mandatory | - | yes |
| 22.6 | The value of a pointer to a FILE shall not be used after the associated stream has been closed. | mandatory | - | yes |
| 22.7 | The macro EOF shall only be compared with the unmodified return value from any Standard Library function capable of returning EOF. | required | - | yes |
| 22.8 | The value of errno shall be set to zero prior to a call to an errno-setting-function. | required | - | yes |
| 22.9 | The value of errno shall be tested against zero after calling an errno-setting-function. | required | - | yes |
| 22.10 | The value of errno shall only be tested when the last function to be called was an errno-setting-function. | required | - | yes |

Files with compilation errors (files partially analyzed)

Table 3.2. Files with compilation errors (files partially analyzed)

|  |
| --- |
| **File** |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\MISRA\_Complete\_main.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_05\_07.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_03.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_08\_04.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_10\_01.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_01.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_06.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_11\_07.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_03.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_04.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_11.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_20\_14.c |
| D:\polyspace\_yanglimin\194 C Code\hou\MISRA-C-2012\R\_21\_11.c |

Chapter 4. Appendix 2 - Definitions

Table 4.1. Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Definition** |
| NA | Not Available |