MISRA-2004 REPORT		
n Description 1 #1385 D (MISRA-C2004 8.2/R) Whenever an object or function is declared or defined, its type shall be explicitly stated	Resource ADC.c	Location line 112
2 #1386-D (MISRA-C2004.8.6/R) Functions shall be declared at file scope (function "sel") 3 #1397-D (MISRA-C2004.8.6/R) Functions shall be declared at file scope (function "sel") 4 #1397-D (MISRA-C2004.10.5/R) if the binvise operations " and <<	ADC.c ADC.c ADC.c	line 47 line 45 line 59
4 #1397-0 (MRSRA-C2004 10.5/lil if the bitwise operators "and or are applied to an operand of underhine; type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand #1466-0 (MRSRA-C2004 12.7/lil) Bitwise operators shall not be applied to operands whose underlying type is signed #1466-0 (MRSRA-C2004 12.7/lil) Bitwise operators shall not be applied to operands whose underlying type is signed	ADC.c ADC.c	line 54
7 if1406-0 [MISRA-C2004 127/R] Bitwise operators shall not be applied to operands whose underlying type is signed 8 if1406-0 [MISRA-C2004 127/R] Bitwise operators shall not be applied to operands whose underlying type is signed	ADC.c ADC.c	line 72
9 I1406 O (MISRA-C2004 12.7/R) Birvise operators shill not be applied to operands whose underlying type is signed 10 I1406 O (MISRA-C2001 12.7/R) Birvise operators shill not be applied to operands whose underlying type is signed	ADC.c	line 84 line 91
11 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying two is sixned 12 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 13 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators what not be applied to operands whose underlying type is signed	ADC.c ADC.c ADC.c	line 100 line 103 line 109
13 #1466 D MRSRA-C2004 12.7/R Bitwise coerators shall not be a collect to ocerands whose underhrine two is signed 14 #1421-0 (MRSRA-C2004 15.8/R) all exit paths from a function with non-void return type shall have an explicit return statement with an expression (function "SR") 15 ca het "File: (-RitCscs) Princips (-MRSRA-C2004 15.8/R) all exit paths from a function with non-void return type shall have an explicit return statement with an expression (function "SR") 16 ca het "File: (-RitCscs) Princips (-MRSRA-C2004 15.8/R) all exit paths (-MRSRA-C2004 15.8/R) a	ADC.c ADC.c	line 114
4 greets mit./cruccy/rooss/complain/min/2/cs/min/3/cs/cs/cs/s-microsin/second-minicity/s-cs/cs/s-microsin/second-minicity/s-cs/cs/s-microsin/second-minicity/s-cs/cs/s-microsin/second-minicity/s-cs/cs/s-microsin/second-minicity/s-cs/cs/s-microsin/second-minicity/s-cs/cs/s-minicity/s-cs/cs/s-minicity/s-cs/cs/s-minicity/s-cs/cs/s-minicity/s-cs/cs/s-minicity/s-cs/cs/s-minicity/s-cs/cs/s-minicity/s-cs/cs/s-minicity/s-cs/cs/s-minicity/s-cs/s-minici	ter "REG" is used as an operand of # or ##. common_macros.h	line 37
##430-D (MISRA-C-2004 19.10/R) In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of if or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of its or its order shall be enclosed in parentheses unless it is used as the operand of its order shall be enclosed in parentheses.	ter "REG" is used as an operand of # or ##. common_macros.h	line 32 line 29
20 #1430-D (MISRA-C:2004 19.10/R) in the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. The compiler cannot determine if the parameter shall be enclosed in parentheses.	ter "REG" is used as an operand of # or ##. common_macros.h	line 26 line 23
22 #1430 O (MISRA-C2004 19.10/R) in the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #III. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parenthese unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parenthese unless it is used as the operand of ill or #II. The compiler cannot determine if the parameter shall be enclosed in parenthese unless it is used as the operand of ill or #II. The compiler cannot shall be enclosed in parenthese unless it is used as the operand of ill or #II. The compiler cannot shall be enclosed in parenthese unless it is used as the operand of ill or #II	ter "REG" is used as an operand of # or ##. common_macros.h	line 20 line 17 line 29
24 str480-0 (MRSRA-22004 13.0/R) in the definition of a function-like marro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used as the operand of 8 or #fl. The compiler cannot determine if the parameter shall be enclosed in parentheses unless it is used	ter "num" is used as an operand of # or ##. common_macros.h	line 26
27 #1397-D (MISRA-C2004.10.5/R) if the bitwise operators "and << are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand 28 #1397-D (MISRA-C2004.10.5/R) if the bitwise operators "and << are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand	GPIO.c	line 38 line 48
29 #1397-D (MSSA-C2004 135/fill if the bitwise operators " and cs are spelled to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand and provided of the perand of the p	GPIO.c GPIO.c	line 58 line 68
31 #1397-D (MRSAC-2004-10.5/fill if the bitwise operators " and << are applied to an operand of underlying type unsigned shar or unsigned short, the result shall be immediately cast to the underlying type of the operand 32 #1397-D (MRSAC-2001-10.5/fill if the bitwise operators " and << are applied to an operand of underlying type unsigned shar or unsigned short, the result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share. The operand share the result shall be immediately cast to the underlying type of the operand share. The result shall be immediately cast to the underlying type of the operand share the result shall be immediately cast to the underlying type of the operand share the result shall be immediately cast to the underlying type of the operand share the result shall be immediately cast to the underlying type of the operand share the result shall be immediately cast to the underlying type of the operand share the result shall be immediately cast to the underlying type of the operand share the result shall be immediately cast to the underlying type of the operand share the result shall be immediately cast to the underlying type of the operand share the result shall be immediately	GPIO.c GPIO.c	line 102 line 112
33 #1397-0 (MISRA-C2004 10.5/R) if the bitwise operators " and << are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand #1397-0 (MISRA-C2004 10.5/R) if the bitwise operators " and << are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand for underlying type unsigned char or unsigned short.	GPIO.c GPIO.c	line 122 line 132
35 #1406-D (MRSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 36 #1406-D (MRSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 37 #1406-D (MRSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying to signed	GPIO.c GPIO.c GPIO.c	line 34 line 38 line 44
37   #1406-D (MRSRA-22004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 38   #1406-D (MRSRA-22004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 39   #1406-D (MRSRA-22004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	GPIO.c GPIO.c	line 48
40 #1406 D (MISRA-C:2004 12.7/R) Bitwise operators shall not be apolied to operands whose underlyine two is siened 41 #1406 D (MISRA-C:2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	GPIO.c GPIO.c	line 58
42 #1406-D (MSRA-C2004 13.77/8) Bitwise operators shall not be applied to operands whose underlying type is signed 43 #1406-D (MSRA-C2004 13.77/8) Bitwise operators shall not be applied to operands whose underlying type is signed	GPIO.c GPIO.c	line 68 line 98
44 #1405-D (MRSA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 45 #1405-D (MRSA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	GPIO.c GPIO.c	line 102 line 108
46 #1406-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 47 #1406-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 48 #1406-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 48 #1406-D (MSRA-C2004 12.7/R) Bitwise operators while only be applied to operands whose underlying type is signed	GPIO.c GPIO.c GPIO.c	line 112 line 118 line 122
48 #106.D (MBRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 49 #106.D (MBRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 50 #106.D (MBRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	GPIO.c GPIO.c GPIO.c	line 122 line 128 line 132
51 #1406-D [MISRA-C200412.7/R] Bitwise operators shall not be applied to operands whose underlying type is signed 52 #1406-D [MISRA-C200412.7/R] Bitwise operators shall not be applied to operands whose underlying type is signed	GPIO.c GPIO.c	line 160 line 162
53 B1406-0 (MISRA-C2004 127/R) Bitwise operators shall not be applied to operands whose underlying type is signed  54 B1406-0 (MISRA-C2004 127/R) Bitwise operators shall not be applied to operands whose underlying type is signed	GPIO.c GPIO.c	line 164 line 166
55 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 56 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	GPIO.c GPIO.c	line 177 line 179
57 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 58 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 59 #1.1 bit in ord file angle signed a new line.	GPIO.c GPIO.c 12C.c	line 181 line 183 line 192
59 II-D last line of file ends without a newline 393 To IMS&A-C2004 12.47 The right-hand operand of underlying type unsigned than or unsigned short, the result shall be immediately cast to the underlying type of the operand 51 II-D last line of file ends without a newline 52 II-D last line of file ends without a newline 53 II-D last line of file ends without a newline 54 II-D last line of file ends without a newline 55 II-D last line of file ends without a newline 56 II-D last line of file ends without a newline 57 II-D last line of file ends without a newline 58 II-D last line of file ends without a newline 59 II-D last line of file ends without a newline 59 II-D last line of file ends without a newline 59 II-D last line of file ends without a newline 59 II-D last line of file ends without a newline 59 II-D last line of file ends without a newline 59 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 50 II-D last line of file ends without a newline 51 II-D last line of file ends without a newline 51 II-D last line of file ends without a newline 51 II-D last line of file ends without a newline 51 II-D last line of file ends without a newline 52 II-D last line of file ends without a newline 52 II-D last line of file ends without a newline 53 II-D last line of file ends without a newline 54 II-D last line of file ends without a newline 54 II-D last line of file ends without a newline 55 II-D last line of file ends without a newline 55 II-D last line of file ends without a newline 55 II-D last line of file ends without a newline 55 II-D last line of file ends without a newline 55 II-D l	12Cc 12Cc	line 133 line 138
52 SEASON INSIGNAC 2004 12.4/11 The right-hand operation of a logical && or   1 operator shall not contain side effects 63 #1405 to MMSRAC 2004 12.4/11 The right-hand operator of a logical && or   1 operator shall not contain side effects 63 #1405 to MMSRAC 2004 12.5/11 The operator of a logical && or   1 operator shall not contain side effects	12Cc 12Cc	line 142 line 138
64 #1405 D (MISRA-C:2004 12.5/R) The operands of a logical && or    shall be primary-expressions 65 #1406 D (MISRA-C:2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	12C.c 12C.c	line 142 line 11
66 #1406-D (MISBA-C-2004 12.7/R) Elitwise operators shall not be applied to operands whose underlying type is signed  7 #1406-D (MISBA-C-2004 12.7/R) Elitwise operators shall not be applied to operands whose underlying type is signed	12C.c 12C.c	line 15 line 19
68 #1406-D (MISRA-C200412.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed  69 #1406-D (MISRA-C200412.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	12Cc 12Cc	line 23 line 27 line 35
70 #1406-D (MSSA-2-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 71 #1406-D (MSSA-2-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 72 #1406-D (MSSA-2-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	12C.c 12C.c 12C.c	line 37 line 42
7 state (INSEA-C.2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed  74 state() (INSEA-C.2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed  74 state() (INSEA-C.2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	12C.c 12C.c	line 44
75 #1406-D [MISRA-C:2004 12:7/R] Bitwise operators shall not be applied to operands whose underlying type is signed  76 #1406-D [MISRA-C:2004 12:7/R] Bitwise operators shall not be applied to operands whose underlying type is signed	12C.c 12C.c	line 62 line 64
77 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed  78 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	12C.c 12C.c	line 69 line 71
79 #3406-D (MSRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 80 #3406-D (MSRA-C-2004 12.7/R) Bitwise operators shall not be asoleted to operands whose underlying type is signed 81 #3406-D (MSRA-C-2004 12.7/R) Bitwise operators shall not be asoleted to operands whose underlying type is signed 81 #3406-D (MSRA-C-2004 12.7/R) Bitwise operators shall not be asoleted to operands whose underlying type is signed	12C.c 12C.c	line 80 line 113
81 #1406-D (MRSRA-22004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 92 #1406-D (MRSRA-22004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 83 #1406-D (MRSRA-22004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	12C.c 12C.c 12C.c	line 124 line 138 line 142
** # ## ## ## ## ## ## ## ## ## ## ## ##	12Cc 12Cc	line 156
86 #1406 D (MISRA-C-2004 12.7/R) Bitwise operators shall not be apolied to operands whose underlyine two is siened 87 #1406 D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	12C.c 12C.c	line 174 line 178
88 #1406-D (MSRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 89 #1-D last line of file ends without a newline	12C.c 12C.h	line 191 line 97
90 #1405-D (MISRA-C.2004 12.5/R) The operands of a logical && or    shall be primary-expressions 91 #1406-D (MISRA-C.2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	led.c	line 167
92 #1406-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 93 #1406-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 94 #1406-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 95 #1406-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	led.e led.e led.c	line 50 line 51 line 52
94 #1406-D (MISRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed  95 #1406-D (MISRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed  96 #1406-D (MISRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	led.c led.c led.c	line 60 line 61
97 #1406-D (MiSRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 98 #1406-D (MiSRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	lcd.c lcd.c	line 62 line 63
99 #1406-D (MRSA-C-2004 12.7/fl) Bitwise operators shall not be applied to operands whose underlying type is signed 100 #1406-D (MRSA-C-2004 12.7/fl) Bitwise operators shall not be applied to operands whose underlying type is signed	led.c led.c	line 80 line 81
101 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed  102 #1406-D (MISRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	led.c	line 82 line 83
103 #1406-D (MSRA-C-2004 12.7/R) Bitwise operators shall not be asolied to operands whose underlying type is signed 104 #1406-D (MSRA-C-2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	led.c led.c led.c	line 91 line 92 line 93
105 #1060-DIMBRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying two is sinned 106 #1060-DIMBRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 107 #1040-DIMBRA-C2004 17.7/R) Array indexine shall be the only allowed form of pointer arithmetic	Ica.c Ica.c Ica.c	line 93 line 94 line 104
2474-0 IMMSRA-C2004 17 AFK Array indexing shall be the only allowed form of pointer arithmetic  109 #1474-0 IMMSRA-C2004 17 AFK Array indexing shall be the only allowed form of pointer arithmetic  109 #1484-0 IMMSRA-C2004 6 J/R) The olain char two shall be used only for the storace and use of character values	led.c	line 106 line 132
110 #1505-D (MSRAC-2004-13.5/B) The three expressions of a for statement shall be concerned only with loop control 111 #1505-D (MSRAC-2004-13.5/B) The three expressions of a for statement shall be concerned only with loop control 111 #1505-D (MSRAC-2004-13.5/B) In the three to price of the form of the function of the statement of the s	lcd.c lcd.c	line 124 line 117
112 #189-D pointless comparison of unsigned integer with zero	led.c	line 151 line 159
114 #188-D pointless comparison of unsigned integer with zero 115 #188-D pointless comparison of unsigned integer with zero	led.c led.c Potentiometer.c	line 167 line 175 line 37
116 #1390-D (MSRA-C-2004 9.1/R) All automatic variables shall have been assigned a value before being used (variable "ADC Data") 117 #1420-D (MSRA-C-2001 16.5/R) Functions with no parameters shall be declared and defined with the parameter ist void 118 #1420-D (MSRA-C-2001 16.5/R) Functions with no parameters shall be declared and defined with the parameter ist void	Potentiometer.c  Potentiometer.c  Potentiometer.c	line 37 line 13 line 33
118   sta20-0 (MSRA-C2004 16.5/R) Functions with no parameter shall be declared and defined with the parameter list void 119   sta35-0 (MSRA-C2004 16.5/R) Functions with no parameter shall be declared and defined with the parameter list void 120   sta20-0 (MSRA-C2004 16.5/R) Functions with no parameter shall be declared and defined defined, redefined or undefined ("NULL") 120   sta20-0 (MSRA-C2004 16.5/R) Functions with no parameter shall be declared and defined with the parameter list void	Potentiometer.c Potentiometer.c	line 11
121 #1420-0 (MSRA-22004 ISS/R) Functions with no parameter shall be declared and defined with the parameter list void 122 #1329-0 (MSRA-22004 ISS/R) destrifiers in an inner scope shall not use the same name as an identifier in an outer scope, and therefore hide that identifier ["speed .limit"]	Potentiometer.h SpeedLimit.c	line 16 line 158
123 #1405-D (MISRA-C-2004 12.5/R) The operands of a logical && or    shall be primary-expressions 124 #1405-D (MISRA-C-2004 12.5/R) The operands of a logical && or    shall be primary-expressions	SpeedLimit.c SpeedLimit.c	line 108 line 113
125 #70-D integer conversion resulted in truncation 126 #70-D integer conversion resulted in truncation 127 #70-D integer conversion resulted in truncation	SpeedLimit.c	line 267 line 264
127 #1468-D (MISRA-C-200414.7/R) A function shall have a single point of exit at the end of the function 128 #1467-D (MISRA-C-200414.6/R) For any iteration statement there shall be at most one break statement used for loop termination	Speedlimit.c Speedlimit.c spi.c	line 78 line 67 line 46
120   1146 / D IMDIKAC-(2004 14,0/K) For any iteration statement there shall be at most one press statement used for 1000 termination		line 46
129 #1405-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed 310 #1405-D (MSRA-C2004 12.7/R) Bitwise operators shall not be applied to operands whose underlying type is signed	spi.c	
129 ;1406-D (MRSRA-2004 12.7/li) Bitwise operators shall not be applied to operands whose underlyinic type is signed 310 ;1406-D (MRSRA-2004 12.7/li) Bitwise operators shall not be applied to operands whose underlyinic two is sliend 311 ;1406-D (MRSRA-2004 12.7/li) Bitwise operators shall not be applied to operands whose underlyinic two is sliend 312 ;1406-D (MRSRA-2004 12.7/li) Bitwise operators shall not be applied to operands whose underlyinic two is sliend 313 ;1406-D (MRSRA-2004 12.7/li) Bitwise operators shall not be applied to operands whose underlyinic two is sliend		line 77 line 80 line 94
129   1466-D IMSRA-C2001 12.7/ll Bitwise operators shall not be aspliced to operands whose underlyink type is signed  130   1456-D IMSRA-C2001 12.7/ll Bitwise operators shall not be assled to operands whose underlyink two is signed  131   1450-D IMSRA-C2001 12.7/ll Bitwise operators shall not be assled to operands whose underlyink two is signed  131   1450-D IMSRA-C2001 12.7/ll Bitwise operators shall not be assled to operands whose underlyink two is signed  131   1450-D IMSRA-C2001 12.7/ll Bitwise operators shall not be assled to operands whose underlyink two is stend  132   1450-D IMSRA-C2001 12.7/ll Source code shall only use? ** "." "style comments.  134   1450-D IMSRA-C2001 12.7/ll Agrazy decade, and but the operands underlyink type is slowed  135   1415-D IMSRA-C2001 12.7/ll Agrazy decade, and but the only allowed destroyed.	spi.c spi.c spi.c	line 77 line 80 line 94 line 12 line 114
129   1406-D. IMMSRA-C2001 12.7/II Bitwise operators shall not be asplied to operands whose underlyinic type is signed 130   1406-D. IMMSRA-C2001 12.7/II Bitwise operators shall not be asplied to operands whose underlyinic two is idented 131   1406-D. IMMSRA-C2001 12.7/II Bitwise operators shall not be asplied to operands whose underlyinic two is idented 131   1406-D. IMMSRA-C2001 12.7/II Bitwise operators shall not be asplied to operands whose underlyinic two is idented 132   1406-D. IMMSRA-C2001 12.7/II Bitwise operators shall not be asplied to operands whose underlyinic two is idented 133   1406-D. IMMSRA-C2001 12.7/II Bitwise operators shall not be asplied to operands whose underlyinic type is slowed 145-D. IMMSRA-C2001 12.7/II Source code shall not be asplied to operands whose underlyinic type is slowed 145-D. IMMSRA-C2001 12.7/II Agray shall not be shall not be used to be applied to the operands whose underlyinic type is slowed 145-D. IMMSRA-C2001 12.7/II Agray shall not be applied to the operand of the operands whose underlyinic type is slowed 145-D. IMMSRA-C2001 12.7/II Agray shall not be applied to the operand of the operand operands whose underlyinic type is slowed	\$96.00 \$9	line 77 line 80 line 94 line 12 line 114 line 120 line 111
129 #1406-D (MRSA-2204 12.7/8) Bitwise operation shall not be assigned to operands whose underlying type is signed 131 #1406-D (MRSA-2204 12.7/8) Bitwise operation shall not be assigned to operands whose underlying type is signed 131 #1406-D (MRSA-2204 12.7/8) Bitwise operation shall not be assigned to operands whose underlying type is signed 131 #1406-D (MRSA-2204 12.7/8) Bitwise operation shall not be assigned to operands whose underlying type is signed 132 #1406-D (MRSA-2204 12.7/8) Bitwise operation shall not be assigned to operands whose underlying type is signed 133 #1406-D (MRSA-2204 12.7/8) Bitwise operation shall not be assigned to operands whose underlying type is signed 134 #1406-D (MRSA-2204 12.7/8) Bitwise operation shall not be assigned to operands whose underlying type is signed 135 #1476-D (MRSA-2204 12.4/8) Areasy indensity in a second shall not be assigned to operands whose underlying type is signed 136 #1476-D (MRSA-2204 12.4/8) Areasy indensity all be the only allowed form of pointer arithmetic 137 #1476-D (MRSA-2204 12.4/8) Areasy indensity shall be the only allowed form of pointer arithmetic 138 #1476-D (MRSA-2204 12.4/8) Areasy indensity shall be the only allowed form of pointer arithmetic	\$96.6 \$96.6 \$96.6 \$96.6 \$96.6 \$96.6 \$96.6 \$96.6 \$96.6 \$96.6 \$96.6	line 77 line 80 line 94 line 12 line 114 line 120 line 111 line 134
129   13406-C   MMSRA-2001 12.7(I) Bitwise operators shall not be assiglied to operands whose underlyinic type is signed 131   13406-C   MMSRA-2001 12.7(I) Bitwise operators shall not be assiglied to operands whose underlyinic two is sistend 132   13406-C   MMSRA-2001 12.7(I) Bitwise operators shall not be assiglied to operands whose underlyinic two is signed 133   13406-C   MMSRA-2001 12.7(I) Bitwise operators shall not be assiglied to operands whose underlyinic two is signed 13406-C   MMSRA-2001 12.7(I) Bitwise operators shall not be assiglied to operands whose underlyinic two is signed 135   13406-C   MMSRA-2001 12.7(I) Bitwise operators shall not be assiglied to operands whose underlyinic two is signed 136   13476-C   MMSRA-2001 12.7(I) Bitwise operators shall not be assiglied to operands whose underlyinic two is signed 137   13476-C   MMSRA-2001 12.7(I) Bitwise operators shall not be assiglied to operands whose underlyinic two is signed 138   13476-C   MMSRA-2001 12.7(I) Array indicein, shall be the only allowed form of pointer arithmetic 137   1352-C   13476-C   MMSRA-2001 12.7(I) Array indicein, shall be the only allowed form of pointer arithmetic 138   13476-C   MMSRA-2001 12.7(I) Array indicein, shall be the only allowed form of pointer arithmetic	\$94.0 \$94.0 \$94.0 \$94.0 \$94.0 \$94.0 \$94.0 \$94.0 \$94.0 \$94.0 \$94.0	line 77 line 80 line 94 line 12 line 114 line 120 line 111 line 134