

STATIC ANALYSIS

Tasks: 203 total, 0 suppressed

All Tasks by Category

- [1] **Rule 8.6 (Required) An identifier with external linkage shall have exactly one external definition** (MISRA2012-RULE_8_6)
 - [1] An identifier with external linkage shall have external definition (MISRA2012-RULE_8_6-b-2)
- [1] **Rule 11.5 (Advisory) A conversion should not be performed from pointer to void into pointer to object** (MISRA2012-RULE_11_5)
 - [1] A conversion should not be performed from pointer to void into pointer to object (MISRA2012-RULE_11_5-a-4)
- [4] **Rule 8.4 (Required) A compatible declaration shall be visible when an object or function with external linkage is defined** (MISRA2012-RULE_8_4)
 - [4] A declaration shall be visible when an object or function with external linkage is defined (MISRA2012-RULE_8_4-a-2)
- [2] **Rule 11.9 (Required) The macro NULL shall be the only permitted form of integer null pointer constant** (MISRA2012-RULE_11_9)
 - [1] Literal zero (0) shall not be used as the null-pointer-constant (MISRA2012-RULE_11_9-a-2)
 - [1] Use NULL instead of literal zero (0) as the null-pointer-constant (MISRA2012-RULE_11_9-b-2)
- [6] **Rule 8.7 (Advisory) Functions and objects should not be defined with external linkage if they are referenced in only one translation unit** (MISRA2012-RULE_8_7)
 - [6] Functions and objects should not be defined with external linkage if they are referenced in only one translation unit (MISRA2012-RULE_8_7-a-4)
- [2] **Dir 4.12 (Required) Dynamic memory allocation shall not be used** (MISRA2012-DIR_4_12)
 - [2] Dynamic heap memory allocation shall not be used (MISRA2012-DIR_4_12-a-2)
- [22] **Rule 8.2 (Required) Function types shall be in prototype form with named parameters** (MISRA2012-RULE_8_2)
 - [5] Identifiers shall be given for all of the parameters in a function prototype declaration (MISRA2012-RULE_8_2-a-2)
 - [17] Function types shall be in prototype form (MISRA2012-RULE_8_2-c-2)
- [1] **Rule 10.4 (Required) Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category** (MISRA2012-RULE_10_4)
 - [1] Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category (MISRA2012-RULE_10_4-a-2)
- [26] **Rule 7.4 (Required) A string literal shall not be assigned to an object unless the object's type is "pointer to const-qualified char"** (MISRA2012-RULE_7_4)
 - [26] A string literal shall not be modified (MISRA2012-RULE_7_4-a-2)
- [1] **Rule 18.1 (Required) A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand** (MISRA2012-RULE_18_1)
 - [1] Avoid accessing arrays out of bounds (MISRA2012-RULE_18_1-a-2)
- [1] **Rule 10.3 (Required) The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category** (MISRA2012-RULE_10_3)
 - [1] The value of an expression shall not be assigned to an object of a different essential type category (MISRA2012-RULE_10_3-b-2)
- [3] **Rule 21.6 (Required) The Standard Library input/output functions shall not be used** (MISRA2012-RULE_21_6)
 - [3] The input/output functions from the 'cstdio' and 'cwchar' libraries should not be used (MISRA2012-RULE_21_6-a-2)
- [1] **Rule 21.7 (Required) The atof, atoi, atol and atoll functions of <stdlib.h> shall not be used** (MISRA2012-RULE_21_7)
 - [1] The 'atof', 'atoi', 'atol' and 'atoll' functions from the 'stdlib.h' or 'cstdlib' library should not be used (MISRA2012-RULE_21_7-a-2)
- [4] **Rule 21.1 (Required) #define and #undef shall not be used on a reserved identifier or reserved macro name** (MISRA2012-RULE_21_1)
 - [4] Do not #define or #undef identifiers with names which start with underscore (MISRA2012-RULE_21_1-a-2)
- [2] **Rule 21.3 (Required) The memory allocation and deallocation functions of <stdlib.h> shall not be used** (MISRA2012-RULE_21_3)
 - [2] Dynamic heap memory allocation shall not be used (MISRA2012-RULE_21_3-a-2)
- [1] **Rule 13.3 (Advisory) 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** (MISRA2012-RULE_13_3)
 - [1] A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects (MISRA2012-RULE_13_3-a-4)
- [1] **Rule 2.7 (Advisory) A function should not contain unused parameters** (MISRA2012-RULE_2_7)
 - [1] There should be no unused parameters in functions (MISRA2012-RULE_2_7-a-4)
- [5] **Rule 2.8 (Advisory) A project should not contain unused object definitions** (MISRA2012-RULE_2_8)
 - [5] A project should not contain unused local variables (MISRA2012-RULE_2_8-c-4)
- [1] **Dir 4.8 (Advisory) If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden** (MISRA2012-DIR_4_8)
 - [1] If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden (MISRA2012-DIR_4_8-a-4)
- [1] **Dir 4.7 (Required) If a function returns error information, then that error information shall be tested** (MISRA2012-DIR_4_7)
 - [1] Always check the returned value of non-void function (MISRA2012-DIR_4_7-b-2)
- [22] **Dir 4.6 (Advisory) typedefs that indicate size and signedness should be used in place of the basic numerical types** (MISRA2012-DIR_4_6)
 - [22] typedefs should be used in place of the basic types (MISRA2012-DIR_4_6-b-4)
- [2] **Dir 4.1 (Required) Run-time failures shall be minimized** (MISRA2012-DIR_4_1)
 - [1] Avoid accessing arrays out of bounds (MISRA2012-DIR_4_1-a-2)
 - [1] Avoid null pointer dereferencing (MISRA2012-DIR_4_1-b-2)
- [1] **Rule 8.13 (Advisory) A pointer should point to a const-qualified type whenever possible** (MISRA2012-RULE_8_13)
 - [1] A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object (MISRA2012-RULE_8_13-a-4)

- [17] **Rule 1.5 (Required) Obsolescent language features shall not be used** (MISRAC2012-RULE_1_5)
- [17] Function types shall be in prototype form (MISRAC2012-RULE_1_5-c-2)
- [3] **Rule 15.5 (Advisory) A function should have a single point of exit at the end** (MISRAC2012-RULE_15_5)
- [3] A function shall have a single point of exit at the end of the function (MISRAC2012-RULE_15_5-a-4)
- [1] **Rule 14.3 (Required) Controlling expressions shall not be invariant** (MISRAC2012-RULE_14_3)
- [1] Avoid conditions that always evaluate to the same value (MISRAC2012-RULE_14_3-ac-2)
- [2] **Rule 14.4 (Required) The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type** (MISRAC2012-RULE_14_4)
- [2] Tests of a value against zero should be made explicit, unless the operand is effectively Boolean (MISRAC2012-RULE_14_4-a-2)
- [19] **Rule 21.10 (Required) The Standard Library time and date functions shall not be used** (MISRAC2012-RULE_21_10)
- [2] The standard header files <time.h> or <ctime> shall not be used (MISRAC2012-RULE_21_10-a-2)
- [10] The time handling functions and macros of the library <time.h> shall not be used (MISRAC2012-RULE_21_10-b-2)
- [7] The types defined in the library <time.h> shall not be used (MISRAC2012-RULE_21_10-c-2)
- [15] **Rule 17.3 (Mandatory) A function shall not be declared implicitly** (MISRAC2012-RULE_17_3)
- [15] Functions shall always have visible prototype at the function call (MISRAC2012-RULE_17_3-a-1)
- [35] **Rule 17.7 (Required) The value returned by a function having non-void return type shall be used** (MISRAC2012-RULE_17_7)
- [35] The value returned by a function having non-void return type shall be used (MISRAC2012-RULE_17_7-a-2)

syounng Total Tasks : 203

/Timer/Timer/clock.c

- | | |
|---|---------------------------|
| 13: Function 'display_time' is not in prototype form | MISRAC2012-RULE_1_5-c-2 |
| 13: Function 'display_time' is not in prototype form | MISRAC2012-RULE_8_2-c-2 |
| 16: The 'time_t' identifier should not be used | MISRAC2012-RULE_21_10-c-2 |
| 16: Usage of 'time' function is not allowed | MISRAC2012-RULE_21_10-b-2 |
| 17: Usage of 'ctime' function is not allowed | MISRAC2012-RULE_21_10-b-2 |
| 19: The 'sprintf' library function should not be used | MISRAC2012-RULE_21_6-a-2 |
| 19: Unused function's "sprintf" return value | MISRAC2012-RULE_17_7-a-2 |
| 20: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 27: Parameter 'new_time' is not used in function 'set_time' | MISRAC2012-RULE_2_7-a-4 |
| 27: The 'time_t' identifier should not be used | MISRAC2012-RULE_21_10-c-2 |

/Timer/Timer/clock.h

- | | |
|--|---------------------------|
| 3: Do not use reserved identifiers : _clock_h_ | MISRAC2012-RULE_21_1-a-2 |
| 5: Do not include the <time.h> header file | MISRAC2012-RULE_21_10-a-2 |
| 8: Function 'display_time' is not in prototype form | MISRAC2012-RULE_1_5-c-2 |
| 8: Function 'display_time' is not in prototype form | MISRAC2012-RULE_8_2-c-2 |
| 11: Names shall be given for all the parameters in function 'set_time' | MISRAC2012-RULE_8_2-a-2 |
| 11: The 'time_t' identifier should not be used | MISRAC2012-RULE_21_10-c-2 |

/Timer/Timer/consts.h

- | | |
|---|--------------------------|
| 3: Do not use reserved identifiers : _consts_h_ | MISRAC2012-RULE_21_1-a-2 |
|---|--------------------------|

/Timer/Timer/driver.c

- | | |
|---|--------------------------|
| 13: Function 'print_menu_get_action' is not in prototype form | MISRAC2012-RULE_1_5-c-2 |
| 13: Function 'print_menu_get_action' is not in prototype form | MISRAC2012-RULE_8_2-c-2 |
| 13: Prototype of function 'print_menu_get_action' does not precede function definition | MISRAC2012-RULE_8_4-a-2 |
| 13: The basic numerical type 'int' should not be used | MISRAC2012-DIR_4_6-b-4 |
| 13: The function 'print_menu_get_action' is referenced only in the translation unit where it is defined | MISRAC2012-RULE_8_7-a-4 |
| 15: String literal "*****" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 15: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 16: String literal "Timer Menu" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 16: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 17: String literal "*****" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 17: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 18: String literal "*****" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 18: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 19: String literal "1) Add a timer" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 19: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 20: String literal "2) Remove a timer" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 20: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 21: String literal "3) List all timers" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 21: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 22: String literal "4) Show time" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 22: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 23: String literal "*****" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 23: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 24: String literal "9) Exit" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 24: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 25: String literal "*****" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 25: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |
| 26: String literal "*****" is passed to function 'print_string' as pointer to non-const object | MISRAC2012-RULE_7_4-a-2 |
| 26: Unused function's "print_string" return value | MISRAC2012-RULE_17_7-a-2 |

28: String literal "Enter selection, when done press <enter> > " is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
28: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
29: Prototype of function 'get_input_digit' does not precede function call	MISRAC2012-RULE_17_3-a-1
32: Function 'main_loop' is not in prototype form	MISRAC2012-RULE_1_5-c-2
32: Function 'main_loop' is not in prototype form	MISRAC2012-RULE_8_2-c-2
32: Prototype of function 'main_loop' does not precede function definition	MISRAC2012-RULE_8_4-a-2
32: The function 'main_loop' is referenced only in the translation unit where it is defined	MISRAC2012-RULE_8_7-a-4
35: Prototype of function 'print_menu_get_action' does not precede function call	MISRAC2012-RULE_17_3-a-1
35: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
40: Prototype of function 'add_timer' does not precede function call	MISRAC2012-RULE_17_3-a-1
42: String literal "Error ... timer not added!" is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
42: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
44: String literal "Timer added!" is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
44: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
48: String literal "Which timer should I nuke? > " is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
48: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
49: Prototype of function 'get_input_digit' does not precede function call	MISRAC2012-RULE_17_3-a-1
52: Prototype of function 'list_timers' does not precede function call	MISRAC2012-RULE_17_3-a-1
55: Prototype of function 'display_time' does not precede function call	MISRAC2012-RULE_17_3-a-1
59: String literal "Goodbye" is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
59: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
60: 'return' statement should be at the end of the 'main_loop' function	MISRAC2012-RULE_15_5-a-4
63: String literal "Invalid Action ... try again!" is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
63: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
69: Function 'main' is not in prototype form	MISRAC2012-RULE_1_5-c-2
69: Function 'main' is not in prototype form	MISRAC2012-RULE_8_2-c-2
71: Prototype of function 'init_timer' does not precede function call	MISRAC2012-RULE_17_3-a-1
72: Prototype of function 'main_loop' does not precede function call	MISRAC2012-RULE_17_3-a-1
73: Prototype of function 'uninit_timer' does not precede function call	MISRAC2012-RULE_17_3-a-1
/Timer/Timer/inout.h	
3: Do not use reserved identifiers : _input_h_	MISRAC2012-RULE_21_1-a-2
6: Function 'get_input_digit' is not in prototype form	MISRAC2012-RULE_1_5-c-2
6: Function 'get_input_digit' is not in prototype form	MISRAC2012-RULE_8_2-c-2
6: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
9: Names shall be given for all the parameters in function 'print_string'	MISRAC2012-RULE_8_2-a-2
9: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
12: Function 'print_string_array' is not in prototype form	MISRAC2012-RULE_1_5-c-2
12: Function 'print_string_array' is not in prototype form	MISRAC2012-RULE_8_2-c-2
12: The 'print_string_array' function has no definition	MISRAC2012-RULE_8_6-b-2
12: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
/Timer/Timer/stdinout.c	
16: Function 'get_input_digit' is not in prototype form	MISRAC2012-RULE_1_5-c-2
16: Function 'get_input_digit' is not in prototype form	MISRAC2012-RULE_8_2-c-2
16: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
18: Local variable 'c' declared but not used	MISRAC2012-RULE_2_8-c-4
19: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
35: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
46: Do not use the 'atoi' function	MISRAC2012-RULE_21_7-a-2
52: Pass parameter "str" with const specifier	MISRAC2012-RULE_8_13-a-4
52: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
54: The 'printf' library function should not be used	MISRAC2012-RULE_21_6-a-2
/Timer/Timer/timer.c	
17: Declaration of object 'max_records' does not precede its definition	MISRAC2012-RULE_8_4-a-2
17: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
17: The object 'max_records' is referenced only in the translation unit where it is defined	MISRAC2012-RULE_8_7-a-4
18: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
20: Function 'init_timer' is not in prototype form	MISRAC2012-RULE_1_5-c-2
20: Function 'init_timer' is not in prototype form	MISRAC2012-RULE_8_2-c-2
22: Operands of the '*' operator have different essential types: 'unsigned int' and 'signed char'	MISRAC2012-RULE_10_4-a-2
22: Unused function's "memset" return value	MISRAC2012-RULE_17_7-a-2
25: Function 'uninit_timer' is not in prototype form	MISRAC2012-RULE_1_5-c-2
25: Function 'uninit_timer' is not in prototype form	MISRAC2012-RULE_8_2-c-2
28: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
38: Function 'query_user' is not in prototype form	MISRAC2012-RULE_1_5-c-2
38: Function 'query_user' is not in prototype form	MISRAC2012-RULE_8_2-c-2
38: Prototype of function 'query_user' does not precede function definition	MISRAC2012-RULE_8_4-a-2
38: The function 'query_user' is referenced only in the translation unit where it is defined	MISRAC2012-RULE_8_7-a-4
40: Local variable 'end_h' declared but not used	MISRAC2012-RULE_2_8-c-4
40: Local variable 'end_m' declared but not used	MISRAC2012-RULE_2_8-c-4

40: Local variable 'start_h' declared but not used	MISRAC2012-RULE_2_8-c-4
40: Local variable 'start_m' declared but not used	MISRAC2012-RULE_2_8-c-4
40: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
42: The 'time_t' identifier should not be used	MISRAC2012-RULE_21_10-c-2
43: The 'tm' identifier should not be used	MISRAC2012-RULE_21_10-c-2
45: Usage of 'time' function is not allowed	MISRAC2012-RULE_21_10-b-2
46: Usage of 'localtime' function is not allowed	MISRAC2012-RULE_21_10-b-2
48: 'malloc' function should not be used	MISRAC2012-RULE_21_3-a-2
48: 'malloc' function should not be used	MISRAC2012-DIR_4_12-a-2
48: Pointer to void type should not be converted to pointer to object type 'struct timer_record **	MISRAC2012-RULE_11_5-a-4
49: "the_record" may possibly be null	MISRAC2012-DIR_4_1-b-2
timer.c (40): int start_h, start_m, end_h, end_m; timer.c (41): struct timer_record* the_record; timer.c (42): time_t timer; timer.c (43): struct tm* tm_tmp; timer.c (45): timer = time(NULL); timer.c (46): tm_tmp = localtime(&timer); Source of null value timer.c (48): malloc(...) *** Null value carrier: malloc(...) timer.c (48): the_record = (struct timer_record*)malloc(sizeof(struct timer_record)); *** Null value carrier: malloc(...) -> the_record Null pointer dereferencing point timer.c (49): memset(the_record, 0, sizeof(struct timer_record)); *** Null value carrier: the_record	
49: Unused function's "memset" return value	MISRAC2012-RULE_17_7-a-2
49: Value returned from method "malloc" should be checked	MISRAC2012-DIR_4_7-b-2
timer.c (40): int start_h, start_m, end_h, end_m; timer.c (41): struct timer_record* the_record; timer.c (42): time_t timer; timer.c (43): struct tm* tm_tmp; timer.c (45): timer = time(NULL); timer.c (46): tm_tmp = localtime(&timer); Method call timer.c (48): malloc(...) timer.c (48): the_record = (struct timer_record*)malloc(sizeof(struct timer_record)); Point by which returned value should be checked timer.c (49): memset(the_record, 0, sizeof(struct timer_record));	
52: String literal "Please enter the start hour [0-23] > " is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
52: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
53: Prototype of function 'get_input_digit' does not precede function call	MISRAC2012-RULE_17_3-a-1
54: String literal "Please enter the start minute [0-59] > " is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
54: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
55: Prototype of function 'get_input_digit' does not precede function call	MISRAC2012-RULE_17_3-a-1
57: Usage of 'mktime' function is not allowed	MISRAC2012-RULE_21_10-b-2
60: String literal "Please enter the end hour [0-23] > " is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
60: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
61: Prototype of function 'get_input_digit' does not precede function call	MISRAC2012-RULE_17_3-a-1
62: String literal "Please enter the end minute [0-59] > " is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
62: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
63: Prototype of function 'get_input_digit' does not precede function call	MISRAC2012-RULE_17_3-a-1
65: Usage of 'mktime' function is not allowed	MISRAC2012-RULE_21_10-b-2
68: String literal "Please enter the channel to record > " is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
68: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
69: Prototype of function 'get_input_digit' does not precede function call	MISRAC2012-RULE_17_3-a-1
69: The expression of the 'essentially signed' type is assigned to an object with the 'essentially unsigned' type	MISRAC2012-RULE_10_3-b-2
74: Function 'add_timer' is not in prototype form	MISRAC2012-RULE_1_5-c-2
74: Function 'add_timer' is not in prototype form	MISRAC2012-RULE_8_2-c-2
74: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
78: Prototype of function 'query_user' does not precede function call	MISRAC2012-RULE_17_3-a-1
80: Test of a value against zero should be made explicit	MISRAC2012-RULE_14_4-a-2
83: Provide only one exit point in 'add_timer' function	MISRAC2012-RULE_15_5-a-4
85: Provide only one exit point in 'add_timer' function	MISRAC2012-RULE_15_5-a-4
88: The function 'add_timer_record' is referenced only in the translation unit where it is defined	MISRAC2012-RULE_8_7-a-4
98: The '++' operator should cause the only side effect in an expression	MISRAC2012-RULE_13_3-a-4
100: String literal "All timers used ... timer not added" is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
100: Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
108: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
111: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
116: Expression of pointer type is compared with '0'	MISRAC2012-RULE_11_9-a-2

116:	Expression of pointer type is compared with '0'	MISRAC2012-RULE_11_9-b-2
116:	Possibly accessing array "timer_records" out of bounds at index -1. Correct index(es): [0...99]	MISRAC2012-DIR_4_1-a-2
Start of the path timer.c (25): void uninit_timer() timer.c (28): int i; timer.c (30): for (i = 0; i < max_records; i++) { *** Loop condition evaluation: (i < 100) (true) *** Entering the loop timer.c (31): delete_timer_record(i); timer.c (110): struct timer_record* tr = timer_records[idx]; timer.c (111): int i; timer.c (114): for (i = idx-1; i < curr_index; i++) *** Loop condition evaluation: (i < curr_index) (assuming true) *** Entering the loop Out of bounds access point timer.c (116): if (0 == timer_records[i]) {		
116:	Possibly accessing array "timer_records" out of bounds at index -1. Correct index(es): [0...99]	MISRAC2012-RULE_18_1-a-2
Start of the path timer.c (25): void uninit_timer() timer.c (28): int i; timer.c (30): for (i = 0; i < max_records; i++) { *** Loop condition evaluation: (i < 100) (true) *** Entering the loop timer.c (31): delete_timer_record(i); timer.c (110): struct timer_record* tr = timer_records[idx]; timer.c (111): int i; timer.c (114): for (i = idx-1; i < curr_index; i++) *** Loop condition evaluation: (i < curr_index) (assuming true) *** Entering the loop Out of bounds access point timer.c (116): if (0 == timer_records[i]) {		
123:	'free' function should not be used	MISRAC2012-RULE_21_3-a-2
123:	'free' function should not be used	MISRAC2012-DIR_4_12-a-2
126:	The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
126:	The function 'format_timer_record' is referenced only in the translation unit where it is defined	MISRAC2012-RULE_8_7-a-4
133:	Unused function's "strftime" return value	MISRAC2012-RULE_17_7-a-2
133:	Usage of 'localtime' function is not allowed	MISRAC2012-RULE_21_10-b-2
133:	Usage of 'strftime' function is not allowed	MISRAC2012-RULE_21_10-b-2
134:	Unused function's "strftime" return value	MISRAC2012-RULE_17_7-a-2
134:	Usage of 'localtime' function is not allowed	MISRAC2012-RULE_21_10-b-2
134:	Usage of 'strftime' function is not allowed	MISRAC2012-RULE_21_10-b-2
136:	Condition "tr" always evaluates to true	MISRAC2012-RULE_14_3-ac-2
Start of the path timer.c (126): void format_timer_record(int idx, char* buf) timer.c (128): char start[BUF_SIZE]; timer.c (129): char end[BUF_SIZE]; timer.c (131): struct timer_record* tr = timer_records[idx]; timer.c (133): strftime(start, BUF_SIZE, "%l:%M %p", localtime(&tr->starttime)); timer.c (134): strftime(end, BUF_SIZE, "%l:%M %p", localtime(&tr->endtime)); Point where senseless condition is used timer.c (136): if (tr) {		
136:	Test of a value against zero should be made explicit	MISRAC2012-RULE_14_4-a-2
137:	The 'sprintf' library function should not be used	MISRAC2012-RULE_21_6-a-2
137:	Unused function's "sprintf" return value	MISRAC2012-RULE_17_7-a-2
142:	Function 'list_timers' is not in prototype form	MISRAC2012-RULE_1_5-c-2
142:	Function 'list_timers' is not in prototype form	MISRAC2012-RULE_8_2-c-2
145:	The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
146:	String literal "Current Set Timers" is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
146:	Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
147:	String literal "Record# Start Time End Time Channel" is passed to function 'print_string' as pointer to non-const object	MISRAC2012-RULE_7_4-a-2
147:	Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
151:	Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
153:	Unused function's "print_string" return value	MISRAC2012-RULE_17_7-a-2
/Timer/Timer/timer.h		
3:	Do not use reserved identifiers : _timer_h_	MISRAC2012-RULE_21_1-a-2
5:	Do not include the <time.h> header file	MISRAC2012-RULE_21_10-a-2
9:	Consider hiding the implementation of the structure 'timer_record' in a translation unit 'driver.c'	MISRAC2012-DIR_4_8-a-4
11:	The 'time_t' identifier should not be used	MISRAC2012-RULE_21_10-c-2
12:	The 'time_t' identifier should not be used	MISRAC2012-RULE_21_10-c-2
13:	The basic numerical type 'unsigned' should not be used	MISRAC2012-DIR_4_6-b-4
17:	Function 'init_timer' is not in prototype form	MISRAC2012-RULE_1_5-c-2
17:	Function 'init_timer' is not in prototype form	MISRAC2012-RULE_8_2-c-2
18:	Function 'uninit_timer' is not in prototype form	MISRAC2012-RULE_1_5-c-2
18:	Function 'uninit_timer' is not in prototype form	MISRAC2012-RULE_8_2-c-2
21:	Function 'add_timer' is not in prototype form	MISRAC2012-RULE_1_5-c-2
21:	Function 'add_timer' is not in prototype form	MISRAC2012-RULE_8_2-c-2
21:	The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
24:	Names shall be given for all the parameters in function 'add_timer_record'	MISRAC2012-RULE_8_2-a-2

27: Names shall be given for all the parameters in function 'delete_timer_record'	MISRAC2012-RULE_8_2-a-2
27: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
30: Names shall be given for all the parameters in function 'format_timer_record'	MISRAC2012-RULE_8_2-a-2
30: The basic numerical type 'int' should not be used	MISRAC2012-DIR_4_6-b-4
33: Function 'list_timers' is not in prototype form	MISRAC2012-RULE_1_5-c-2
33: Function 'list_timers' is not in prototype form	MISRAC2012-RULE_8_2-c-2
Flow Analysis Legend:	
Message - violation cause	Message - violation point
	Message - path element annotation
©Parasoft Corp. - C++test 2024.2.0.20241115B2628 Reporting System	