

Bachelor of IT

CAB403 – Systems Programming

Assessment 2 Project

Group 88

Name: Dane Madsen

Student Number: n10983864

Email: n10983864@qut.edu.au

# Contributors

Dane Madsen – n10983864 (sole member)

# Statement of Completeness

Assignment submission is complete. Application fulfills all required functionality. The fire alarm system attempts to follows NASA The power of 10, ISO 26262-6:2018 and MISRA C to a standard practical for the assessment.

# Statement of contributions

Dane Madsen: Everything assessed (sole member)

# Fire Alarm Safety Assessment

## Safety-critical standards provided fire alarm fails

### Nasa the power of 10

1. **Avoid complex flow constructs, such as goto and recursion.**

Fail – firealarm.c uses goto on line 159 and deletenodes() is a recursive function.

1. **All loops must have fixed bounds. This prevents runaway code.**

Fail – firealarm.c uses unbound loops on lines 61, 134, 157 and 189.

1. **Avoid heap memory allocation.**

Fail – firealarm.c uses malloc on lines 67, 82, 91, 152 and 176.

1. **Restrict functions to a single printed page.**

Fail – functions tempmonitor() on line 56, and main() on line 147 go well over one page.

1. **Use a minimum of two runtime assertions per function.**

Fail – assertions are not used.

1. **Restrict the scope of data to the smallest possible.**

Fail – integers are used for values where uint8\_t, uint16\_t or bool could be used to save memory on lines 10, 13, 36, 59, 82, 169, 182, 192 and 204.

1. **Check the return value of all non-void functions or cast to void to indicate the return value is useless.**

Fail – values of deletenodes() is not checked or cast on lines 73 and 97.

1. **Use the pre-processor sparingly.**

Pass.

1. **Limit pointer use to a single dereference, and do not use function pointers.**

Fail – shm is dereferenced multiple times on line 64, 171, 179, 184 and 194.

1. **Compile with all possible warnings active; all warnings should then be addressed before release of the software.**

Fail – Warnings not addressed; code does not compile.

### ISO 26262-6:2018

firealarm.c undoubtably fails numerous ISO 26262-6:2018 rules but I don’t have time to go through them all manually and I can’t find an appropriate tool to scan for them. But through a preliminary analysis of the code, I can see that at the very least rules regarding implicit type conversion, recursion and unconditional jumps have been broken multiple times in the code.

### MISRA C

firealarm.c violates 70 MISRA C rules as reported by Cppchecks MISRA module

MISRA rules violated:

misra-c2012-7.1 (-): 2 misra-c2012-7.4 (-): 1 misra-c2012-8.2 (-): 1

misra-c2012-8.4 (-): 9 misra-c2012-10.4 (-): 5 misra-c2012-11.5 (-): 11

misra-c2012-11.6 (-): 1 misra-c2012-12.1 (-): 7 misra-c2012-12.3 (-): 7

misra-c2012-13.3 (-): 1 misra-c2012-14.4 (-): 2 misra-c2012-15.1 (-): 1

misra-c2012-15.5 (-): 1 misra-c2012-15.6 (-): 3 misra-c2012-17.2 (-): 1

misra-c2012-17.7 (-): 3 misra-c2012-18.4 (-): 5 misra-c2012-21.3 (-): 6

misra-c2012-21.6 (-): 1 misra-c2012-21.9 (-): 1 misra-c2012-21.10 (-): 1

## Approach to rectify issues

The approach to rectify these issues was to largely scrap the existing code and rewrite it in a safer manner.

In my improved implementation temperatures are no longer treated as a structure chain and are instead stored in an array local to the temperature monitor.

Instead of using the boomgate and parkingsign structures included in the original firealarm.c I have instead opted to use the respective structures defined in common.h.

I have changed tempmonitor() to return a void pointer so that its return type does not need to be cast in pthread\_create().

I have scrapped compare() as I don’t think it’s necessary. I have completely rewritten the method for calculating the median temperature to no longer use structs as I found using structures for this purpose to be too complex for this application.

I moved all the include statements into the firealarm.h and common.h header files to reduce pre-processor strain.

I've completely rewritten the function for opening all boom gates and have made it so it is no longer a threaded function.

I made emergency\_mode into its own function instead of using a goto.

I implemented functions for checking whether the alarm is still active, sending the evacuation message, checking if there is a fire from smoothed temperatures, getting the median temperature from raw temperatures and printing temperatures to the terminal.

## Safety-critical standards of new Implementation

### Nasa the power of 10

1. **Avoid complex flow constructs, such as goto and recursion.**

Pass – No goto or recursive functions are used.

1. **All loops must have fixed bounds. This prevents runaway code.**

Pass – All loops have fixed bounds.

1. **Avoid heap memory allocation.**

Pass – Heap memory allocation is not used.

1. **Restrict functions to a single printed page.**

Pass – All functions are shorter than a single page in length.

1. **Use a minimum of two runtime assertions per function.**

Pass – All functions use a minimum of two assertions.

1. **Restrict the scope of data to the smallest possible.**

Pass – The smallest possible data type is always used.

1. **Check the return value of all non-void functions or cast to void to indicate the return value is useless.**

Pass – All non-void returns are checked.

1. **Use the pre-processor sparingly.**

Pass.

1. **Limit pointer use to a single dereference, and do not use function pointers.**

Pass – Pointers are dereferenced sparingly.

1. **Compile with all possible warnings active; all warnings should then be addressed before release of the software.**

Pass – Code compiles with -Wall -Wextra -Werror –Wpedantic flags.

### ISO 26262-6:2018

Without going too far in-depth the code of the new firealarm.c attempts to mostly follows ISO 26262-6:2018 rules.

### MISRA C

The improved firealarm.c now violates only one MISRA rule which is rule 11.6 (A cast shall not be performed between pointer to void and an arithmetic type). It is necessary to break this rule as a level index must be passed to the temperature monitor for it to function correctly. An alternative would be to either directly pass the respective Level\_t structure or pass the uint16\_t value of the temperature but as the scope of these data type is much larger than the level index it would make the code less safe. Regardless, I have mitigated this violation by immediately asserting that the level index is in the range of levels.