


Assignment 4 – RTOS Part 2 MISRA Report Justification

This document will explain any remaining MISRA violations from our MISRA report (misra_report.pdf).

Check	File	Line	Entity	Violation
MISRA23_21.6	bbbio.c	68	fgets	stdio.h input/output function fgets used in file bbbio.c
MISRA23_8.7	bbbio.c	80	write_gpio_value	write_gpio_value has external linkage but is only used in the translation unit built from bbbio.c, so the static keyword should be used.
MISRA23_8.7	bbbio.c	193	set_pwm_duty_cycle	set_pwm_duty_cycle has external linkage but is only used in the translation unit built from bbbio.c, so the static keyword should be used.
MISRA23_8.7	bbbio.c	239	set_pwm_frequency	set_pwm_frequency has external linkage but is only used in the translation unit built from bbbio.c, so the static keyword should be used.
MISRA23_8.7	bbbio.c	161	set_pwm_enable	set_pwm_enable has external linkage but is only used in the translation unit built from bbbio.c, so the static keyword should be used.
MISRA23_8.7	bbbio.c	282	setup_pwm	setup_pwm has external linkage but is only used in the translation unit built from bbbio.c, so the static keyword should be used.

MISRA23_2.5	bbbio.h	139	PWM_OFF	Unused macro
www.scitools.com				Page
<hr/>				
MISRA C 2023 2025-04-16 8:49 AM				
Results Table				
				declaration PWM_OFF

For these first few violations, they are all related to functions and macros that are unused. Our custom BeagleBone library has functions that are related to PWM, but since PWM is not used, these methods are not called for this assignment. Since this assignment only required using the GPIO code (and not even all of the GPIO code we have), some functions from our library go unused, which is why we get this external linkage violation. We feel that this is acceptable since they are just functions that are not relevant to this assignment, and thus we shouldn't have to use them if we don't need to. We also don't want to remove these functions because this is a general library for us to use with BeagleBone I/O, so it does not make sense to remove these functions just for this assignment.

There is also one violation for using stdio.h functions, but we feel these are acceptable since we need that library to do file I/O which is how we control our GPIO pins. To get rid of that would require making our own I/O library.

MISRA23_15.5	stopwatch.c	247	main	Multiple exit points from function
MISRA23_19.2	stopwatch.c	9	mutex	Violation: union used for "mutex".
MISRA23_19.2	stopwatch.c	256	button_attr	Violation: union used for "button_attr".
MISRA23_19.2	stopwatch.c	256	display_attr	Violation: union used for "display_attr".
MISRA23_19.2	stopwatch.c	256	timer_attr	Violation: union used for "timer_attr".
MISRA23_19.2	stopwatch.c	311	mutex_attr	Violation: union used for "mutex_attr".
MISRA23_2.7	stopwatch.c	234	signum	Parameter signum is unused
MISRA23_21.10	stopwatch.c	2	/usr/include/x86_64-linux-gnu/sys/time.h	<time.h> was used
MISRA23_21.5	stopwatch.c	3	./stopwatch.c	header file signal.h used.

We feel that the union violations are acceptable because we don't actually use any unions ourselves, but these are a part of the sched.h and pthread.h libraries that we use in order to do Rate Monotonic Scheduling. As a result, there is no way to avoid these violations since we need those libraries to do that part of the assignment.

The violation for signum being unused is actually not even true, but MISRA doesn't catch that. This is a parameter to our cleanup function, or the function that is called when a signal is sent to our process. We have a signal handler that detects input like CTRL + C and various other signals to make sure that we can gracefully exit our application when those signals are received. So this variable is actually used, and even if it isn't, we feel that this is acceptable since we need it when we create a function that will be registered as a signal handler.

Lastly, we have violations for using time.h and signal.h. Signal.h is needed because of reasons mentioned previously, mainly for gracefully exiting our application. Time.h is required in order to get the elapsed time and add it to the current time so that our timing thread is able to update the counter without any problems and it has an accurate way of adding time. We feel these are both acceptable header files to import for this assignment.