Prep part 2

Analysis and discussion

1. The ISR toggles PF2 three times. Is this debugging intrusive, nonintrusive or minimally intrusive? Justify your answer.

This debugging method is minimally intrusive because in implementing it, there will be a running time of a few cycles to run the code, which MAY impact results but is not probably. The effect on the system is almost negligible in this case.

1. In this lab we dumped strategic information into arrays and processed the arrays later. Notice this approach gives us similar information we could have generated with a printf statement. In ways are printf statements better than dumps? In what ways are dumps better than printf statements?

In some cases, data dumps would be better than implementing printf statements because data dumps allow us to implement various data processing methods into our program so that we could extract key pieces of information such as the jitter and maximum frequency of appearance. On the other hand, printf statements may be better in some cases where we need to check the specific values being outputted, in which case we could just view the outputs of the program vs having to go in the debugger and look through the array. A third case could be implemented such that a data dump would be created and this data dump is printed out with a series of printf statements in which case, this would contain the advantages of both types of debugging but would alternatively require too much processing time.

1. What are the necessary conditions for a critical section to occur? In other words, what type of software activities might result in a critical section?

The necessary conditions for a critical section to occur is that a multi-process program to not allow concurrent execution by more than one of the programs processes. Some software activities that may result in a critical section are pieces of programs that require exclusion of access to a particular part, such as using a shared resource in the program, whether it be a shared data structure or shared memory.

1. Define “minimally intrusive”.

Minimally intrusiveness is a debugging categorization that measures how efficient a debugging process is. Specifically, a minimally intrusive debugging method is one that the process does have an impact on the system but this effect is negligible. Generally, the existence of a minimally intrusive process will have minimal effect on the parameters being measured (ie. Execution time of the debugging portion is small compared to the execution time of the process being measured). Another classification of minimally intrusiveness is if the debugging time ratio is 1/1000 to the original program.

1. The PMF results should show hardware averaging is less noisy than not averaging. If it is so good why don’t we always use it?

The reason that we do not always use hardware averaging is as it states; it implements an average. Generally hardware averaging is to obtain a mass number of ADC values more efficiently and with less noise than generally can be done. The problem is that in some cases, we will need to view the noise and individual values of ADC outputs to properly process some data. Therefore, we cannot use hardware averaging options in these cases. In addition, the use of Hardware averaging leads to much more time consumption in the interrupt call.