CMPE2250 – ICA #1 RTI (Real Time Interrupt)

Supporting Materials

- RTI Notes
- Interrupts Introductory Notes
- CMPE2250 Course Notes: Pages 5-7

Using the header provided (*'rti.h'*) Implement the following functions for the RTI in the source file you will create (*'rti.c'*). Also, have the proper ISR (interrupt service routine), ideally coded in *'rti.c'* so it does not have to be added to every project that uses the RTI.

- ***RTI_Init*** – Enables the RTI Module to generate a 1[ms] tick interrupt.
- ***RTI_Delay_ms*** – Generates a blocking delay based on the RTI 1[ms] tick interrupt. The number of milliseconds delayed will be passed to the function as a parameter.
- ***RTI_InitCallback*** - Enables the RTI Module to generate a 1[ms] tick interrupt and attaches a callback function defined in '*main.*c' using a function pointer, as discussed in class.

To test these functions, create a project that does the following:

<u>Part A</u>

Using the *RTI_Delay_ms* function, create a counter that increments every 1[s] and displays on the top 7-segment display row in decimal. The counter should start at 0 when the program starts, and it should roll from 9999 to 0000. Please note that to use this delay feature, your RTI module should be initialized, and the interrupts enabled.

<u>Part B</u>

Display this counter on the first row of the lcd display, in decimal, using the format: "DEC: XXXX". Display it also on the second row of the lcd display, in HEX, using the format: "HEX:XXXX".

<u>Part C</u>

For part A and B, you could have just used the *RTI_Init* function implemented. If you did so, now update your code to initialize the RTI so it can call a function defined in '*main.*c' every 1[ms] (use *RTI_InitCallback).*

To test this functionality, have the RED LED to toggle every 500[ms], the YELLOW LED to do the same every 250[ms], and the GREEN LED every 125[ms]. Implement this functionality inside the callback function defined. If you have done this correctly, the leds should be blinking with the pattern defined despite the delay in the main infinite loop.

Lastly, have another counter to increment every 1[s], also done in the callback function and display it on the second row of the 7-segment display.

After running your program for a while (a few minutes) are the 2 counters showing on the 7-segment displays in sync? Explain your findings.

Explain why the LEDs blink faster even when the main infinite loop is blocking the code for 1[s] every iteration.