CSE3442 (Fall 2020)
Lab #3

1. Start with a file lab3a_*your_name*.c, where *your_name* is replaced with your name as it appears in MyMav.

2. In these steps, do not change the SYSCTL_RCC_R value from the class code (fcyc = 40 MHz as in the class examples).

3. Write a new assembly language function, void wait10Seconds(void) that waits 15 seconds and then returns. Write a program, adding a call in main to call this function. Your function should contain one loop (not a loop in a loop as in waitMicroseconds()). You cannot call or use waitMicroseconds().

   (Hint: it is not possible to use a large constant value with the MOV instruction, since the number of bits is quite small in the instruction. Use the LDR and .field methods used in class from the stop go asm example to store any large constants.)

4. Place a breakpoint and count event in the main() while loop. The grader will show you how to do this. You will need to create a breakpoint and add a count event once stopped.

5. Verify that the timing is within 100 clocks of 400,000,000 and the LED toggles every 10 seconds.

6. For the next steps, start with the serial.c file from class, renaming the file lab3b_*your_name*.c, where *your_name* is replaced with your name as it appears in MyMav. Modify the code from class to operate at 38400 baud, 8 data bits, no parity, and 1 stop bit. The divider is calculated as the system clock rate divided by (baud rate * 16). The divider is divided into an integer and 6-bit fractional value. The grader will work a short example since this lab corresponds with the Tuesday lecture material which is after the Tuesday lab. The function setBaudRate() sets the integer and fractional divider, but you should be able to calculate these values for the grader and for the upcoming test.

7. Demonstrate your code examples to the grader. Show your calculated values for the divider to the grader.

8. Send the two C files to the grader with your name in the header of the file.