Exercise 3 – Due Thursday 10/3 before noon

You may have noticed from the previous exercises that the UART speed in TinyOS has been 115200 baud but only 9600 baud in C. In this week's exercise we will upgrade the baud rate in the C applications to 115200 baud by running the UART module of the high frequency DCO instead of the low frequency external clock. You can use the **cPrintf** application in the cs450 directory as your starting point.

A) In the **MSP430F1611 User Guide**, read the following pages well enough to complete the rest of the exercise:

   4-1 to 4-14
   4-15 to 4-17 (cursorily)
   11-1 to 11-18
   11-19 to 11-23 (cursorily)
   12.1.1
   12-20 to 12-25 (cursorily)

B) In order to run anything at a known frequency from the DCO it has to be calibrated. We will use the external 32 kHz crystal to calibrate the DCO to run at 2 MHz by comparing TimerA (sourced from the DCO) with TimerB (sourced from the external crystal) and increase/decrease the DCO frequency based on the difference.

   1. Assign initial values to DCOCTL, BCSCTL1, and BCSCTL2 as follows:
      i. DCOx = 3, MODx = 16, RSELx = 6
         This initial setting corresponds to the 1.7-2.3 MHz range.
      ii. The 32 kHz crystal is connected to LFXT1. Set LFXT1 to low-frequency mode and pass it straight to ACLK (divide by 1).
      iii. XT2 has nothing connected, so switch it off.
      iv. Use the DCO as the source for MCLK without scaling (divide by 1).
      v. Use the DCO as the source for SMCLK, at half the frequency.
      vi. Use the internal resistor for the DCO.

   2. Assign values to TACTL so:
      i. TimerA is sourced from SMCLK without scaling.
      ii. TimerA runs in continuous mode, with interrupts enabled.
      iii. TimerA is cleared.

   3. Assign values to TBCTL so:
      i. TimerB is sourced from ACLK without scaling.
      ii. TimerB runs in continuous mode, with interrupts disabled.
      iii. TimerB is cleared.
      iv. Keep the remaining options at their default values.

4. Implement the interrupt service routine for TimerA (**TIMERA1_VECTOR**):
    i. When TimerA overflows, exactly 0xFFFF counts have passed. Since the goal is to have TimerA/SMCLK to run at 1 MHz, if TimerB/ACLK is running at 32 kHz, how many counts have passed on TimerB when TimerA has counted to 0xFFFF, i.e., the interrupt fires?

    ii. Compare this value with the actual counters on TimerA and TimerB to determine whether the DCO should be sped up or slowed down. This can be done by increasing/decreasing DCOCTL and BCSCTL1, where DCOCTL is the fine grained control and BCSCTL1 is the coarse grained control. So, each time the interrupt fires then increase, decrease, or leave the DCO unchanged so that the SMCLK is running as close to 1 MHz as possible.

    iii. Use the UART settings from last week's assignment to make it run at 115200 baud when sourced from SMCLK.

C) Upload your C file to Blackboard with all the above implemented and the answer to the following question: What is the frequency of MCLK?