ECE2049 E22: Homework 5 Solutions

Problem 2

```
void runtimerA2(void)
{
    TA2CTL = TASSEL_2 | MC_1 | ID_2;  // Clock source: SMCLK / 4, Up mode
    TA2CCR0 = 1309;                   // MAX_CNT = 1309
    TA2CCTL0 = CCIE;                  // Enable timer A2 interrupt
}
```

Part a

Based on the timer configuration, we can find t_{INT}:

$t_{INT} = \frac{MAX\_CNT+1}{f_{CLK}} = \frac{1309+1}{\frac{1048576}{4}}$

$t_{INT} = 0.0049972534\ seconds$

The exact time between interrupts, t_{Actual} is 0.0049972534 seconds

Part b

In this case, the actual time is less than the time reported by the system (0.005 seconds), so the device is fast. For every 0.0049972534 seconds that have elapsed, the device counts 0.005 seconds—therefore, the device will count faster than intended.

$0.005 = (x\ interrupts) * (0.0049972534\ sec - 0.005\ sec)$

x = 1820.444 or $\simeq$ 1820 interrupts until there is an error of 0.005sec.

$(1820\ interrupts) * (\frac{0.005sec}{1\ interrupt}) = 9.10\ seconds$

Part c

To implement the leap count, we need to know after how many interrupts to skip a count, since the system is fast. Because the interrupts happen at an interval of 0.005 seconds, we need to compensate by adding an extra count after 1820 interrupts, when we know the timer would otherwise by off by 0.005 seconds.

```
volatile unsigned long g_ulTimer = 0;     // variable to store count
volatile unsigned long g_ulLeapCount = 0; // variable to track leap counting

#pragma vector = TIMER2_A0_VECTOR
```

```
__interrupt void TIMER_A2_ISR()
{
  if (leap_cnt < 1820) {
    g_ulLeapCount++;
    g_ulTimer++;
  } else {
    // Don't increment timer here so we skip a count
    g_ulLeapCount = 0;
  }
}
```