Brian Lopez
Github root directory: github.com/brian4280/Bed_Assign

Date Submitted: 10/5/18

Task 02:

Youtube Link: https://www.youtube.com/watch?v=kJJU31Otyo8

Modified Code:

```
    ui32Period = (SysCtlClockGet() / 2) / 1.333;    // 2 Hz with 75% duty cycle
    TimerLoadSet(TIMER0_BASE, TIMER_A, ui32Period - 1);   // load period into timer0
```

// Insert code here

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Task 03:

Youtube Link: https://www.youtube.com/watch?v=0Zd24tRaez8

Modified Schematic (if applicable):

Modified Code:

```
int main(void)
{

    uint32_t ui32Period;


    SysCtlClockSet(SYSCTL_SYSDIV_5 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);   // enable port F
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3); //
set LEDs as output
    GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_4);           // set SW1 as input

    GPIOPadConfigSet(GPIO_PORTF_BASE, GPIO_PIN_4 , GPIO_STRENGTH_2MA,
GPIO_PIN_TYPE_STD_WPU); // turn weak pull up on
    GPIOIntEnable(GPIO_PORTF_BASE, GPIO_INT_PIN_4);     // enable SW1 interrupt
    GPIOIntTypeSet(GPIO_PORTF_BASE, GPIO_INT_PIN_4, GPIO_FALLING_EDGE); // Interrupt
happens on rising edge of button
    IntEnable(INT_GPIOF);      // turn on GPIOF interrupt

    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);   // configure timer 0
    TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC);
    ui32Period = (SysCtlClockGet() / 2) / 1.333;    // 2 Hz with 75% duty cycle
    TimerLoadSet(TIMER0_BASE, TIMER_A, ui32Period - 1);   // load period into timer0

    IntEnable(INT_TIMER0A);  // enable interrupts for timer 0A
    TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
    TimerEnable(TIMER0_BASE, TIMER_A);  // turn on timer 0A

    IntMasterEnable();         // turn on global interrupt

    while(1)
    {

    }
}

void PortFPin0IntHandler(void)
{
    int i;
    uint32_t t1period = 60000000;  // at 40Mhz, 60Mhz = 1.5 sec.
    GPIOIntClear(GPIO_PORTF_BASE, GPIO_INT_PIN_4);  // clear switch interrupt

    TimerDisable(TIMER0_BASE, TIMER_A);      // turn off timer 0
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3, 8);  // turn
on LED

    SYSCTL_RCGCTIMER_R |= 2;
    TIMER1_CTL_R = 0;
    TIMER1_CFG_R = 0x04; // 16 bit
    TIMER1_TAMR_R = 0x02;  // periomic mode and down counter
    TIMER1_TAILR_R = 160000-1;  // 60M / 250prescalar for 1 mSec
    TIMER1_TAPR_R = 250-1;  // prescalar of 250
    TIMER1_ICR_R = 0x1;     // remove timeout flag
    TIMER1_CTL_R |= 0x01;   // enable timer 1 A

    for (i = 0; i < 9; i++) {
        while((TIMER1_RIS_R & 0x1) == 0)
        {

        }
        TIMER1_ICR_R = 0x1;
    }

    TIMER1_CTL_R = 0;      // turn off timer
    TIMER1_ICR_R = 0x1;    // remove timeout flag

    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3, 0);
    TimerEnable(TIMER0_BASE, TIMER_A);
}
```

- Timer0 interrupt is omitted here since no changes was done.

```
----------------------------------------------------------------------------------
```

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.