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

**Date Submitted:** **10/12/18**

--------------------------------------------------------------------------------

## Task 01:

Youtube Link: https://www.youtube.com/watch?v=BlXOg6rI5XM

**Modified Code:**

- **Only modification is to add a variable (direction) that dictates which direction the servo is moving. Once the end is reached, it will change to allow the servo to start going the other direction.**

```
volatile uint32_t direction = 1;  // used to direct which way the servo is sweeping

    while(1)
    {


        if (direction == 1)  // move from left to right
        {
            ui8Adjust++;
            if (ui8Adjust >= 125)
                direction = 0;  // change direction when end is reached

            ROM_PWMPulseWidthSet(PWM1_BASE, PWM_OUT_0, ui8Adjust * ui32Load / 1000);
        }
        else if (direction == 0)  // move from right to left
        {
            ui8Adjust--;
            if (ui8Adjust <= 50)
                direction = 1;   // change direction.

            ROM_PWMPulseWidthSet(PWM1_BASE, PWM_OUT_0, ui8Adjust * ui32Load / 1000);
        }
        ROM_SysCtlDelay(100000);
    }
// Insert code here
```

--------------------------------------------------------------------------------

## Task 02:

Youtube Link: https://www.youtube.com/watch?v=eHB2RQWQ6ls

**Modified Code:**

- **I needed to enable PORTF and configure it as a PWM port. Other than that, the while loop is the same, just output to the LED's PWM instead of the servo**

```
    SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM1);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);

    GPIOPinTypePWM(GPIO_PORTF_BASE, GPIO_PIN_1); // enable LED
    GPIOPinConfigure(GPIO_PF1_M1PWM5); // set LED as PWM output

    ui32PWMClock = SysCtlClockGet() / 64;
    ui32Load = (ui32PWMClock / PWM_FREQUENCY) - 1;
```

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

```
    PWMGenConfigure(PWM1_BASE, PWM_GEN_2, PWM_GEN_MODE_DOWN);
    PWMGenPeriodSet(PWM1_BASE, PWM_GEN_2, ui32Load);

    PWMPulseWidthSet(PWM1_BASE, PWM_OUT_5, 50 * ui32Load / 1000);  // start at leftmost pos
    PWMOutputState(PWM1_BASE, PWM_OUT_5_BIT, true);
    PWMGenEnable(PWM1_BASE, PWM_GEN_2);

    volatile uint32_t direction = 1;  // used to direct which way the servo is sweeping

    while(1)
    {


        if (direction == 1)  // go from dim to bright
        {
            ui8Adjust++;
            if (ui8Adjust >= 250)
                direction = 0;  // change direction when end is reached

            PWMPulseWidthSet(PWM1_BASE, PWM_OUT_5, ui8Adjust * ui32Load / 1000);
        }
        else if (direction == 0)  // go from bright to dim
        {
            ui8Adjust--;
            if (ui8Adjust <= 10)
                direction = 1;   // change direction.

            PWMPulseWidthSet(PWM1_BASE, PWM_OUT_5, ui8Adjust * ui32Load / 1000);
        }
        SysCtlDelay(100000);
    }
-------------------------------------------------------------------------------------
```

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

## Task 03:

Youtube Link: https://www.youtube.com/watch?v=PYgmu2UeXFY

**Modified Schematic (if applicable):**

**Modified Code:**

- **Configure the rest of the LED's for PWM:**

```
GPIOPinTypePWM(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3);
GPIOPinConfigure(GPIO_PF1_M1PWM5);    // for PF1
GPIOPinConfigure(GPIO_PF2_M1PWM6);    // for PF2
GPIOPinConfigure(GPIO_PF3_M1PWM7);    // for PF3

ui32PWMClock = SysCtlClockGet() / 64;
ui32Load = (ui32PWMClock / PWM_FREQUENCY) - 1;

// configure for PWM5 and 6
PWMGenConfigure(PWM1_BASE, PWM_GEN_2, PWM_GEN_MODE_DOWN);
PWMGenPeriodSet(PWM1_BASE, PWM_GEN_2, ui32Load);
// configure for PWM7
PWMGenConfigure(PWM1_BASE, PWM_GEN_3, PWM_GEN_MODE_DOWN);
PWMGenPeriodSet(PWM1_BASE, PWM_GEN_3, ui32Load);
PWMOutputState(PWM1_BASE, PWM_OUT_5_BIT | PWM_OUT_6_BIT | PWM_OUT_7_BIT, true);

PWMGenEnable(PWM1_BASE, PWM_GEN_2);  // turn on both PWMs
PWMGenEnable(PWM1_BASE, PWM_GEN_3);
```

- **For loop for working all 3 LED's:**

```
uint32_t i, j, k;

while(1)
{
    for (i = 1500; i >= 10; i--) {
        PWMPulseWidthSet(PWM1_BASE, PWM_OUT_5, i * ui32Load / 1000);
        SysCtlDelay(50000);
        for (j = 1500; j >= 10; j--) {
            PWMPulseWidthSet(PWM1_BASE, PWM_OUT_6, j * ui32Load / 1000);
            SysCtlDelay(50000);
            for (k = 1500; k >= 10; k--) {
                PWMPulseWidthSet(PWM1_BASE, PWM_OUT_7, k * ui32Load / 1000);
                SysCtlDelay(50000);
            }

        }
    }
}
```

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