Guillermo Gálvez

Github root directory:(https://github.com/galveg1/VMs_House-of-Fun-Or-Pain.git)

Date Submitted: **10.06.2018**

Task 00: *No submission*

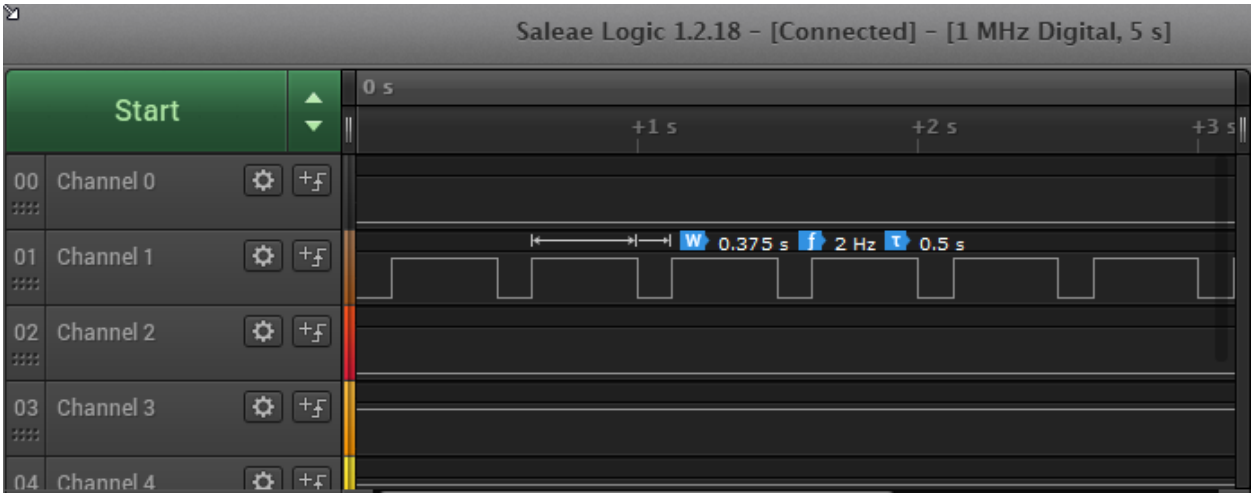
Youtube Link: N/A

Task 01: Change the toggle of the GPIO at 2 Hz using Timer0 with 75% duty cycle and verify the waveform generated.

Youtube Link: https://youtu.be/WXc8OrG1GSk

Task 01 asks to modify the given example to generate a 2Hz signal with a 75% duty cycle. Using SysCtlDelay varied delays were used to toggle the LED on-off and achieve the desired signal.

```
TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);

if(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_2))
{
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3, 0);
    SysCtlDelay(1250000);//ADDED code to create requested Duty Cycle
}
else
{
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 4);
    SysCtlDelay(5000000);//ADDED code to create requested Duty Cycle
}
```



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

Github root directory:(https://github.com/galveg1/VMs_House-of-Fun-Or-Pain.git)

Task 02: Include a GPIO Interrupt to Task 02 from switch SW2 to turn ON and the LED for 1.5 sec. Use a Timer1 to calculate the 1.5 sec delay. The toggle of the GPIO is suspended when executing the interrupt.

Youtube Link: https://youtu.be/WXc8OrG1GSk

```
void Mytimer1A_Delay uint32_t ttime1)
{
    // Dis-enable Timer 0 Interrupt
    TimerIntDisable TIMER0_BASE, TIMER_TIMA_TIMEOUT);

    //Following code configures Timer1
    //for use in Task 02
    int i = 0;

    SYSCTL_RCGCTIMER_R |= 2;
    TIMER1_CTL_R = 0;
    TIMER1_CFG_R = 0x04;
    TIMER1_TAMR_R = 0X02;
    TIMER1_TAILR_R = 64000-1;
    TIMER1_TAPR_R = 250-1;
    TIMER1_CTL_R |= 0X01;


    for(i=0;i < ttime1;i++)
        while (TIMER1_RIS_R & 0x1) == 0;
            TIMER1_ICR_R = 0x1;

}//end timer1A_delaySec
```

Additional handler added for Timer1

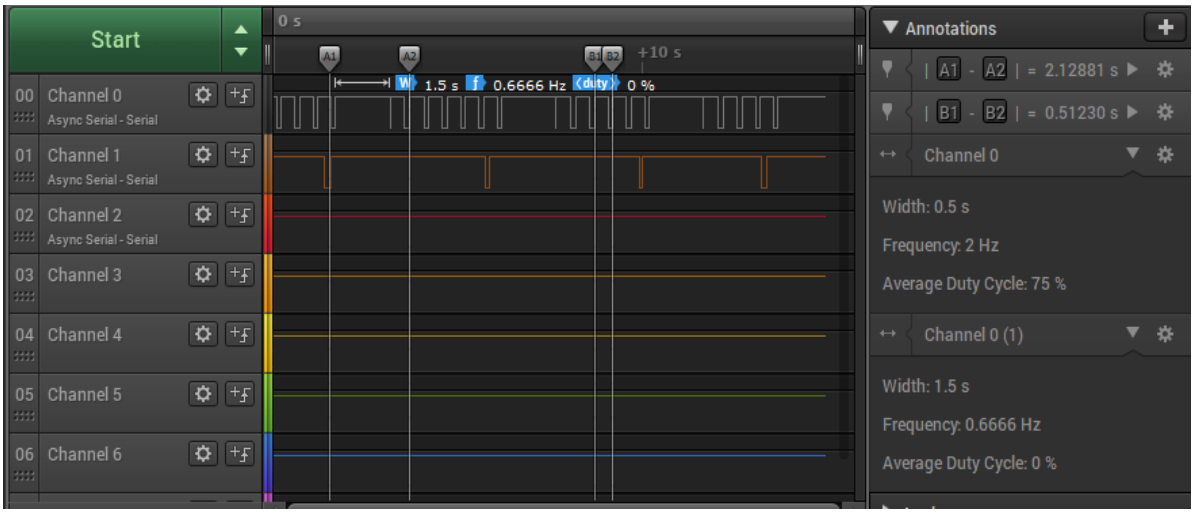
```
void Timer1IntHandler void)
{
    //Clear GPIO Int
    GPIOIntClear GPIO_PORTF_BASE, GPIO_INT_PIN_0);
    //LED ON
    //Blue LED PIN2
    GPIOPinWrite GPIO_PORTF_BASE, GPIO_PIN_2, 4);
    Mytimer1A_Delay 3529412);
    //LED OFF
    GPIOPinWrite GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3, 0);

    //Re-Enable(INT_TIMER0A);
    TimerIntEnable TIMER0_BASE, TIMER_TIMA_TIMEOUT);

}//end Timer1Handler
```

Below is a sample a 15s sample of the output wave. On the right you can see the desired f and duty cycle from Task 01. From both wave forms the desired delay for Task 02 of 1.5s is obtained when SW2 is pressed which can also been seen at the start of the 1.5 pulse.

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

Guillermo Gálvez

Github root directory:(https://github.com/galveg1/VMs_House-of-Fun-Or-Pain.git)



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