**Date Submitted: 10/04/18**

**------------------------------------------------------------------------------------**

**Task 01:**

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

**Modified Code:**

**// Insert code here**

**int** **main**(**void**)

{

.

.

.

ui32Period = (**SysCtlClockGet**() / 6) // Changed Hz from 10 to 2

//ui32Period = (SysCtlClockGet() / 10) / 2;

.

}

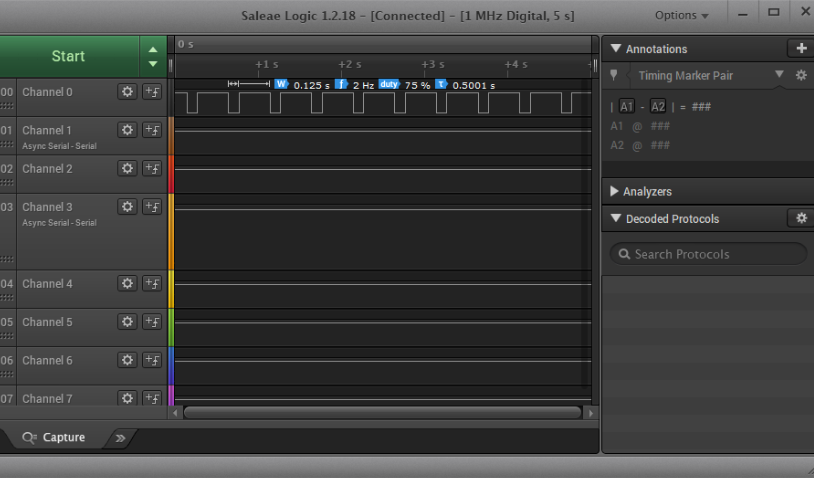
**void** **Timer0IntHandler**(**void**)

{

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 4);

**SysCtlDelay**(5000000);

}



**------------------------------------------------------------------------------------**

**Task 02:**

Youtube Link: https://www.youtube.com/watch?v=bAh3QGPBISw

**Modified Code:**

**int** **main**(**void**)

{

uint32\_t ui32Period, delay; // timer setup variables

**SysCtlClockSet**(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHZ|SYSCTL\_OSC\_MAIN);

/\* Switch and LED Interrupt SETUP \*/

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

// Unlock GPIO F pin 0 (PF0)

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = GPIO\_LOCK\_KEY;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_CR) = 0x1;

// buttons and led GPIO setup

**GPIOPinTypeGPIOInput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_0);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3);

// sets button interrupt to rising edge and enable interrupt

**GPIOIntTypeSet**(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_0, GPIO\_RISING\_EDGE);

**GPIOIntEnable**(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_0);

**IntEnable**(INT\_GPIOF);

/\* Timer 0 Interrupt SETUP \*/

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_TIMER0);

**TimerConfigure**(TIMER0\_BASE, TIMER\_CFG\_PERIODIC);

ui32Period = (**SysCtlClockGet**() / 6); Change for 2 Hz with 75% duty cycle

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, ui32Period-1);

**IntEnable**(INT\_TIMER0A);

**TimerIntEnable**(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

/\* Timer 1 Interrupt SETUP \*/

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_TIMER1);

**TimerConfigure**(TIMER1\_BASE, TIMER\_CFG\_PERIODIC);

delay = **SysCtlClockGet**() \* (3/2);

// multiply clock by 3/2 to get 1.5 second delay

**TimerLoadSet**(TIMER1\_BASE, TIMER\_A, delay-1);

**IntEnable**(INT\_TIMER1A);

**TimerIntEnable**(TIMER1\_BASE, TIMER\_TIMA\_TIMEOUT);

/\* Enable master interrupt and start timer 0 \*/

**IntMasterEnable**();

**TimerEnable**(TIMER0\_BASE, TIMER\_A);

**while**(1)

{

}

}

**void** **Timer0IntHandler**(**void**)

{

// Clear the timer 0 interrupt

**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);

}

**else**

{

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 4);

**SysCtlDelay**(5000000);

}

}

/\* Added two interrupts/ISRs to handle LED turn on for 1.5 seconds: \*/

**void** **PortFPin0IntHandler**(**void**)

{

// Turn on LED

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 4);

// Clear the GPIO interrupt

**GPIOIntClear**(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_0);

// Disable Timer 0 interrupt

**TimerIntDisable**(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

// Enable Timer 1 for LED's 1.5 second delay

**TimerEnable**(TIMER1\_BASE, TIMER\_A);

}

**void** **Timer1IntHandler**(**void**)

{

// Clear the timer 1 interrupt

**TimerIntClear**(TIMER1\_BASE, TIMER\_TIMA\_TIMEOUT);

// Disable Timer 1

**TimerDisable**(TIMER1\_BASE, TIMER\_A);

// Re-enable Timer 0 Interrupt

**TimerIntEnable**(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

}

**------------------------------------------------------------------------------------**