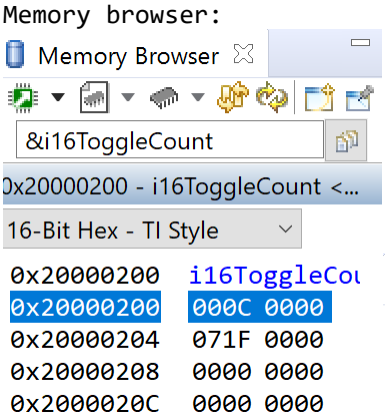
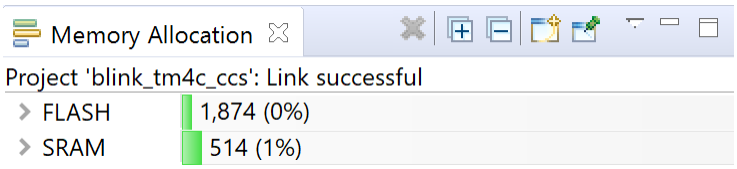
Tenniel Takenaka-Fuller
Github root directory: https://github.com/TennielTakenaka/psychic-invention/

Lab 2
Date Submitted: **11/23/19**

Task 01:

Memory browser:





Task 02:

Modified Code:

```
//---------------------------------------------------------------------------
// Project: Blink TM4C - CCS Lab - STARTER
//
// Author: Eric Wilbur
//
// Date: June 2014
//
//---------------------------------------------------------------------------


//-----------------------------------------
// TivaWare Header Files
//-----------------------------------------
#include <stdint.h>
#include <stdbool.h>

#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "inc/hw_ints.h"
#include "driverlib/interrupt.h"
#include "driverlib/timer.h"
#include <time.h>


//-----------------------------------------
// Prototypes
```

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

```
//---------------------------------------
void hardware_init(void);
void ledToggle(void);
void delay(void);


//---------------------------------------
// Globals
//---------------------------------------
volatile int16_t i16ToggleCount = 0;


//---------------------------------------------------------------------------
// main()
//---------------------------------------------------------------------------
void main(void)
{

   hardware_init();                                          // init hardware via Xware

   while(1)                                                        // forever loop
   {
          ledToggle();                                  // toggle LED

          delay();                                                 // create a delay
of ~1/2sec

          i16ToggleCount += 1;                                  // keep track of #toggles
   }

}


//---------------------------------------------------------------------------
// hardware_init()
//
// inits GPIO pins for toggling the LED
//---------------------------------------------------------------------------
void hardware_init(void)
{
       //Set CPU Clock to 40MHz. 400MHz PLL/2 = 200 DIV 5 = 40MHz
       SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAI
N);

       // ADD Tiva-C GPIO setup - enables port, sets pins 1-3 (RGB) pins for output
       SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
       GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);

       // Turn on the LED
       GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 4);

}
```

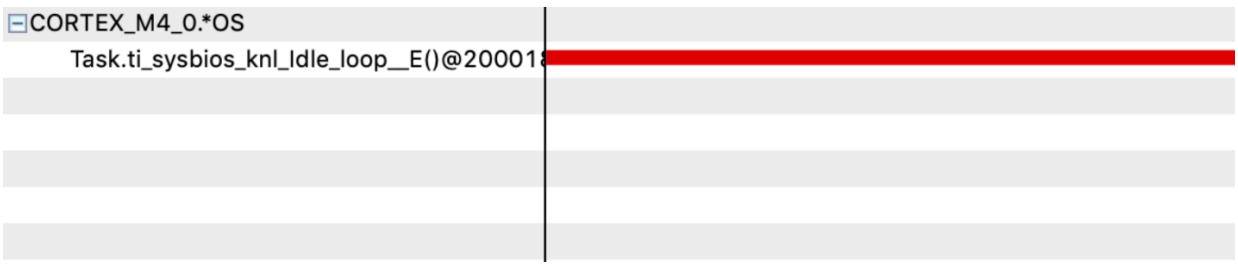
Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

```
//*****************************************************************************
// ledToggle()
//
// toggles LED on Tiva-C LaunchPad
//*****************************************************************************
void ledToggle(void)
{
	// LED values - 2=RED, 4=BLUE, 8=GREEN
	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);
	}
}



//*****************************************************************************
// delay()
//
// Creates a 500ms delay via TivaWare fxn
//*****************************************************************************
void delay(void)
{
	SysCtlDelay(13400000);		// makes delay 2x as slow!

}
```

Lab 4
Date Submitted: 11/17/19

Task 01:

Execution:



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

Tenniel Takenaka-Fuller
Github root directory: https://github.com/TennielTakenaka/psychic-invention/

Task 02:

Log:

	Type	Time	Error	Master	Message	Event	EventClass	Data1	Data2	SeqNo	Logger	Module	D
5	i	757949725		CORTEX_...	[../main.c:116] TOGGLED LED [3] t...	Log_L_info	Info			2	Main...	xdc.r...	x
7	i	1010596575		CORTEX_...	[../main.c:116] TOGGLED LED [4] t...	Log_L_info	Info			3	Main...	xdc.r...	x
8		1010604212		CORTEX_...	LS_cpuLoad: 0%	Load	CPU	CPU	0.00	1	Load...	ti.sys...	t
9	i	1263248625		CORTEX_...	[../main.c:116] TOGGLED LED [5] t...	Log_L_info	Info			4	Main...	xdc.r...	x
10	i	1515889975		CORTEX_...	[../main.c:116] TOGGLED LED [6] t...	Log_L_info	Info			5	Main...	xdc.r...	x
11		1515897612		CORTEX_...	LS_cpuLoad: 0%	Load	CPU	CPU	0.00	2	Load...	ti.sys...	t
12	i	1768542037		CORTEX_...	[../main.c:116] TOGGLED LED [7] t...	Log_L_info	Info			6	Main...	xdc.r...	x
13	i	2021188875		CORTEX_...	[../main.c:116] TOGGLED LED [8] t...	Log_L_info	Info			7	Main...	xdc.r...	x

---

Lab 5
Date Submitted: 12/3/19

Task 01:

Modified Code:

```
//---------------------------------------
// BIOS header files
//---------------------------------------
#include <xdc/std.h>                        //mandatory - have to include first, for
BIOS types
#include <ti/sysbios/BIOS.h>                //mandatory - if you call APIs like
BIOS_start()
#include <xdc/runtime/Log.h>                //needed for any Log_info() call
#include <xdc/cfg/global.h>                 //header file for statically defined
objects/handles


//----------------------------------------
// TivaWare Header Files
//----------------------------------------
#include <stdint.h>
#include <stdbool.h>

#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "inc/hw_ints.h"
#include "driverlib/interrupt.h"
#include "driverlib/timer.h"


//---------------------------------------
// Prototypes
//---------------------------------------
void hardware_init(void);
void ledToggle(void);
void delay(void);
```

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

```
//---------------------------------------
// Globals
//---------------------------------------
volatile int16_t i16ToggleCount = 0;


//---------------------------------------------------------------------------
// main()
//---------------------------------------------------------------------------
void main(void)
{

    hardware_init();                            // init hardware via Xware

    BIOS_start();

}


//---------------------------------------------------------------------------
// hardware_init()
//
// inits GPIO pins for toggling the LED
//---------------------------------------------------------------------------
void hardware_init(void)
{
    uint32_t ui32Period;

    //Set CPU Clock to 40MHz. 400MHz PLL/2 = 200 DIV 5 = 40MHz
    SysCtlClockSet(SYSCTL_SYSDIV_5 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    // ADD Tiva-C GPIO setup - enables port, sets pins 1-3 (RGB) pins for output
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3);

    // Turn on the LED
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3, 4);

    // Timer 2 setup code
    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER2);           // enable Timer 2 periph
clks
    TimerConfigure(TIMER2_BASE, TIMER_CFG_PERIODIC);        // cfg Timer 2 mode -
periodic

    ui32Period = (SysCtlClockGet() / 2);                        // period = CPU clk
div 2 (500ms)
    TimerLoadSet(TIMER2_BASE, TIMER_A, ui32Period);         // set Timer 2 period

    TimerIntEnable(TIMER2_BASE, TIMER_TIMA_TIMEOUT);        // enables Timer 2 to
interrupt CPU

    TimerEnable(TIMER2_BASE, TIMER_A);                      // enable Timer 2
```

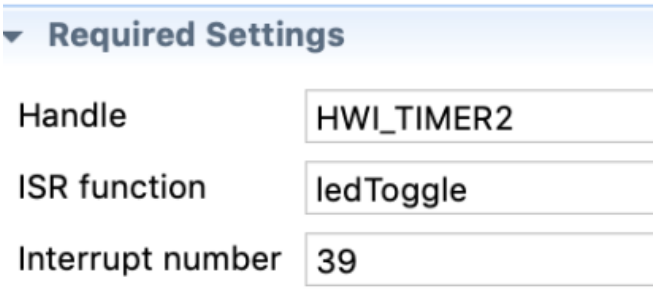
```
}


//*****************************************************************************
// ledToggle()
//
// toggles LED on Tiva-C LaunchPad
//*****************************************************************************
void ledToggle(void)
{
    TimerIntClear(TIMER2_BASE, TIMER_TIMA_TIMEOUT);         // must clear timer flag
FROM timer

    // LED values - 2=RED, 4=BLUE, 8=GREEN
    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);
    }

    i16ToggleCount += 1;                                    // keep track of #toggles
    Log_info1("LED TOGGLED [%u] TIMES", i16ToggleCount);        // send toggle count
to UIA

}
```

Task 02:

	Type	Time	Error	Master	Message	Event	EventClass	Data1	Data2	SeqNo	Logger	Module	D
2		18775		CORTEX_...	LM_switch: oldtsk: 0x0, oldfunc: 0x...	CtxChg	TSK	ti_s...		1	SYS...	_ti.ui...	ti
3	i	250014137		CORTEX_...	[../main.c:135] LED TOGGLED [1]...	Log_L_info	Info			0	Main...	xdc.r...	x
4		500004462		CORTEX_...	LS_cpuLoad: 1%	Load	CPU	CPU	1.00	0	Load...	ti.sys...	ti
5	i	500013637		CORTEX_...	[../main.c:135] LED TOGGLED [2]...	Log_L_info	Info			1	Main...	xdc.r...	x
6	i	750013650		CORTEX_...	[../main.c:135] LED TOGGLED [3]...	Log_L_info	Info			2	Main...	xdc.r...	x
7		1000008112		CORTEX_...	LS_cpuLoad: 1%	Load	CPU	CPU	1.00	1	Load...	ti.sys...	ti
8	i	1000013662		CORTEX_...	[../main.c:135] LED TOGGLED [4]...	Log_L_info	Info			3	Main...	xdc.r...	x

ROV:

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



Lab 6
Date Submitted: **12/3/19**

Task 01:

Modified Code:

```
//---------------------------------------
// BIOS header files
//---------------------------------------
#include <xdc/std.h>        //mandatory - have to include first, for BIOS types
#include <ti/sysbios/BIOS.h>    //mandatory - if you call APIs like BIOS_start()
#include <xdc/runtime/Log.h>    //needed for any Log_info() call
#include <xdc/cfg/global.h>     //header file for statically defined objects/handles


//---------------------------------------
// TivaWare Header Files
//---------------------------------------
#include <stdint.h>
#include <stdbool.h>

#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "inc/hw_ints.h"
#include "driverlib/interrupt.h"
#include "driverlib/timer.h"


//---------------------------------------
// Prototypes
//---------------------------------------
void hardware_init(void);
void ledToggle(void);
void Timer_ISR(void);

//---------------------------------------
// Globals
//---------------------------------------
volatile int16_t i16ToggleCount = 0;
```

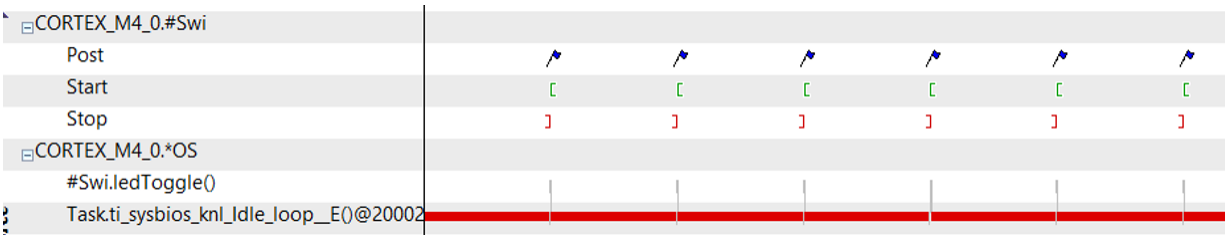
Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

```
//-----------------------------------------------------------------------------
// main()
//-----------------------------------------------------------------------------
void main(void)
{

    hardware_init();    // init hardware via Xware

    BIOS_start();

}


//-----------------------------------------------------------------------------
// hardware_init()
//
// inits GPIO pins for toggling the LED
//-----------------------------------------------------------------------------
void hardware_init(void)
{
    uint32_t ui32Period;

    //Set CPU Clock to 40MHz. 400MHz PLL/2 = 200 DIV 5 = 40MHz
    SysCtlClockSet(SYSCTL_SYSDIV_5 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    // ADD Tiva-C GPIO setup - enables port, sets pins 1-3 (RGB) pins for output
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3);

    // Turn on the LED
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_3, 4);

    // Timer 2 setup code
    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER2);
    TimerConfigure(TIMER2_BASE, TIMER_CFG_PERIODIC);

    ui32Period = (SysCtlClockGet() / 2);    // period = CPU clk div 2 (500ms)
    TimerLoadSet(TIMER2_BASE, TIMER_A, ui32Period);     // set Timer 2 period

    TimerIntEnable(TIMER2_BASE, TIMER_TIMA_TIMEOUT);        // enables Timer 2
    TimerEnable(TIMER2_BASE, TIMER_A);              // enable Timer 2

}


//-----------------------------------------------------------------------------
// ledToggle()
//
// toggles LED on Tiva-C LaunchPad
//-----------------------------------------------------------------------------
void ledToggle(void)
```

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

```
{
    TimerIntClear(TIMER2_BASE, TIMER_TIMA_TIMEOUT);

    // LED values - 2=RED, 4=BLUE, 8=GREEN
    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);
    }

    i16ToggleCount += 1;        // keep track of #toggles

    Log_info1("LED TOGGLED [%u] TIMES", i16ToggleCount); // send toggle count to UIA

}

void Timer_ISR(void)
{
    TimerIntClear(TIMER2_BASE, TIMER_TIMA_TIMEOUT);
    Swi_post(LEDSwi);
}
```

Task 02:



1		1500052450	CORT...	LD_end: swi: 0x20002264	Stop	SWI	ledToggle()	
1		2000026125	CORT...	LM_post: swi: 0x20002264, ...	Post	SWI	ledToggle()	
1		2000037625	CORT...	LM_begin: swi: 0x2000226...	Start	SWI	ledToggle()	
2	i	2000044775	CORT...	[../main.c:132] LED TOGGL...	Log_L_info	Info		
2		2000050425	CORT...	LD_end: swi: 0x20002264	Stop	SWI	ledToggle()	
2		2000058775	CORT...	LS_cpuLoad: 1%	Load	CPU	CPU	1.00
2		2500026150	CORT...	LM_post: swi: 0x20002264, ...	Post	SWI	ledToggle()	

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

Tenniel Takenaka-Fuller
Github root directory: https://github.com/TennielTakenaka/psychic-invention/

Required Settings

Handle	LEDSwi
Function	ledToggle
Interrupt priority	1
Initial trigger	0x0

Required Settings

Handle	HWI_TIMER2
ISR function	Timer_ISR
Interrupt number	39

Lab 7
Date Submitted: 12/3/19

Task 01:

Modified Code:

```
//---------------------------------------
// BIOS header files
//---------------------------------------
#include <xdc/std.h>                        //mandatory - have to include first, for
BIOS types
#include <ti/sysbios/BIOS.h>                //mandatory - if you call APIs like
BIOS_start()
#include <xdc/runtime/Log.h>                //needed for any Log_info() call
#include <xdc/cfg/global.h>                 //header file for statically defined
objects/handles
#include <xdc/runtime/Timestamp.h>          // used for Timestamp() calls


//---------------------------------------
// TivaWare Header Files
//---------------------------------------
#include <stdint.h>
#include <stdbool.h>

#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "inc/hw_ints.h"
#include "driverlib/interrupt.h"
#include "driverlib/timer.h"


//---------------------------------------
// Prototypes
//---------------------------------------
void hardware_init(void);
void ledToggle(void);
//void Timer_ISR(void);


//---------------------------------------
```

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

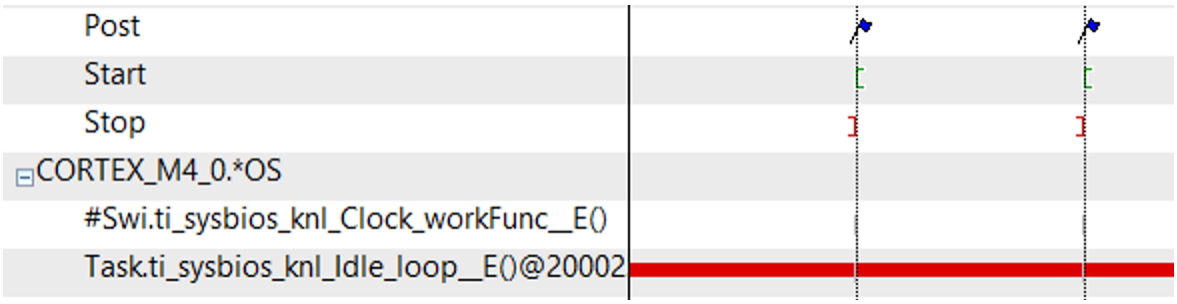
```
// Globals
//--------------------------------------
volatile int16_t i16ToggleCount = 0;


//---------------------------------------------------------------------------
// main()
//---------------------------------------------------------------------------
void main(void)
{

   hardware_init();                          // init hardware via Xware

   BIOS_start();

}


//---------------------------------------------------------------------------
// hardware_init()
//
// inits GPIO pins for toggling the LED
//---------------------------------------------------------------------------
void hardware_init(void)
{
//  uint32_t ui32Period;

    //Set CPU Clock to 40MHz. 400MHz PLL/2 = 200 DIV 5 = 40MHz
    SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

    // ADD Tiva-C GPIO setup - enables port, sets pins 1-3 (RGB) pins for output
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);

    // Turn on the LED
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 4);

}


//---------------------------------------------------------------------------
// ledToggle()
//
// toggles LED on Tiva-C LaunchPad
//---------------------------------------------------------------------------
void ledToggle(void)
{
    static uint32_t ui32_t0, ui32_t1, ui32_t2, ui32start, ui32stop, ui32delta;  //
used for Timestamp calculations

    ui32_t0 = Timestamp_get32();                                    // calculate
Timestamp() overhead (ui32_t2)
    ui32_t1 = Timestamp_get32();
    ui32_t2 = ui32_t1 - ui32_t0;
```

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

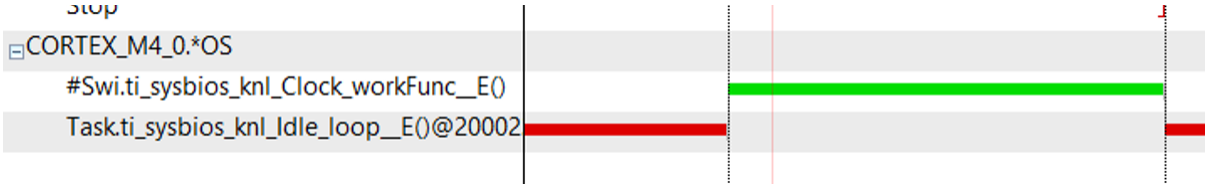
```
    // LED values - 2=RED, 4=BLUE, 8=GREEN
    if(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_2))
    {
        ui32start = Timestamp_get32();                              // get starting
Timer snapshot for LED benchmark

        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0);   //
toggle GPIO/LED

        ui32stop = Timestamp_get32();                               // get ending
Timer snapshot for LED benchmark

        ui32delta = ui32stop - ui32start - ui32_t2;                 // calculate LED
toggle benchmark

        Log_info1("LED BENCHMARK = [%u] TM4C CYCLES", ui32delta);   // send LED
benchmark to Log display
    }
    else
    {
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 4);
    }

    i16ToggleCount += 1;                                            // keep track of
#toggles

    Log_info1("LED TOGGLED [%u] TIMES",i16ToggleCount);             // send toggle
count to UIA

}
```

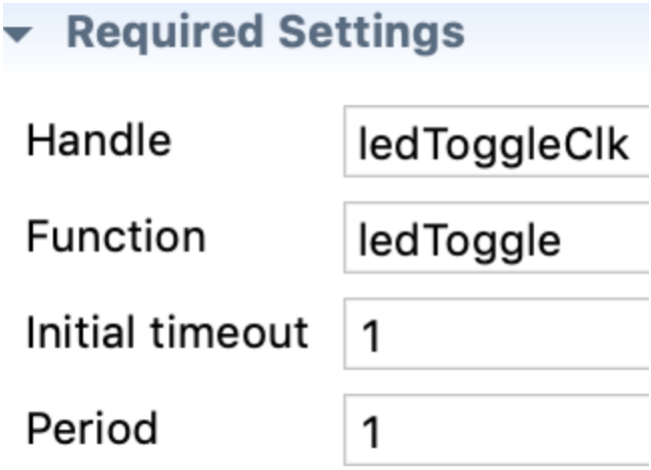
Task 02:

	Type	Time	Error	Master	Message	Event
4		3500031500		CORT...	LM_post: swi: 0x200023c0, func: 0x18c9, pri: 15	Post
4		3500043175		CORT...	LM_begin: swi: 0x200023c0, func: 0x18c9, preThread: 2	Start
4	i	3500058275		CORT...	[../main.c:140] LED BENCHMARK = [12] TM4C CYCLES	Log_L_info
4	i	3500065900		CORT...	[../main.c:153] LED TOGGLED [7] TIMES	Log_L_info
4	i	3500071925		CORT...	[../main.c:155] LED BENCHMARK = [311] CYCLES	Log_L_info



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





Lab 8
Date Submitted: **12/3/19**

Task 01:

Modified Code:

```
//---------------------------------------
// BIOS header files
//---------------------------------------
#include <xdc/std.h>                        //mandatory - have to include first, for
BIOS types
#include <ti/sysbios/BIOS.h>                //mandatory - if you call APIs like
BIOS_start()
#include <xdc/runtime/Log.h>                //needed for any Log_info() call
#include <xdc/cfg/global.h>                 //header file for statically defined
objects/handles


//---------------------------------------
// TivaWare Header Files
//---------------------------------------
#include <stdint.h>
#include <stdbool.h>

#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
#include "inc/hw_ints.h"
```

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

```
#include "driverlib/interrupt.h"
#include "driverlib/timer.h"


//---------------------------------------
// Prototypes
//---------------------------------------
void hardware_init(void);
void ledToggle(void);
void Timer_ISR(void);


//---------------------------------------
// Globals
//---------------------------------------
volatile int16_t i16ToggleCount = 0;


//---------------------------------------------------------------------------
// main()
//---------------------------------------------------------------------------
void main(void)
{

   hardware_init();                        // init hardware via Xware

   BIOS_start();

}


//---------------------------------------------------------------------------
// hardware_init()
//
// inits GPIO pins for toggling the LED
//---------------------------------------------------------------------------
void hardware_init(void)
{
    uint32_t ui32Period;

    //Set CPU Clock to 40MHz. 400MHz PLL/2 = 200 DIV 5 = 40MHz
    SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

    // ADD Tiva-C GPIO setup - enables port, sets pins 1-3 (RGB) pins for output
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);

    // Turn on the LED
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 4);

    // Timer 2 setup code
    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER2);           // enable Timer 2 periph
clks
```

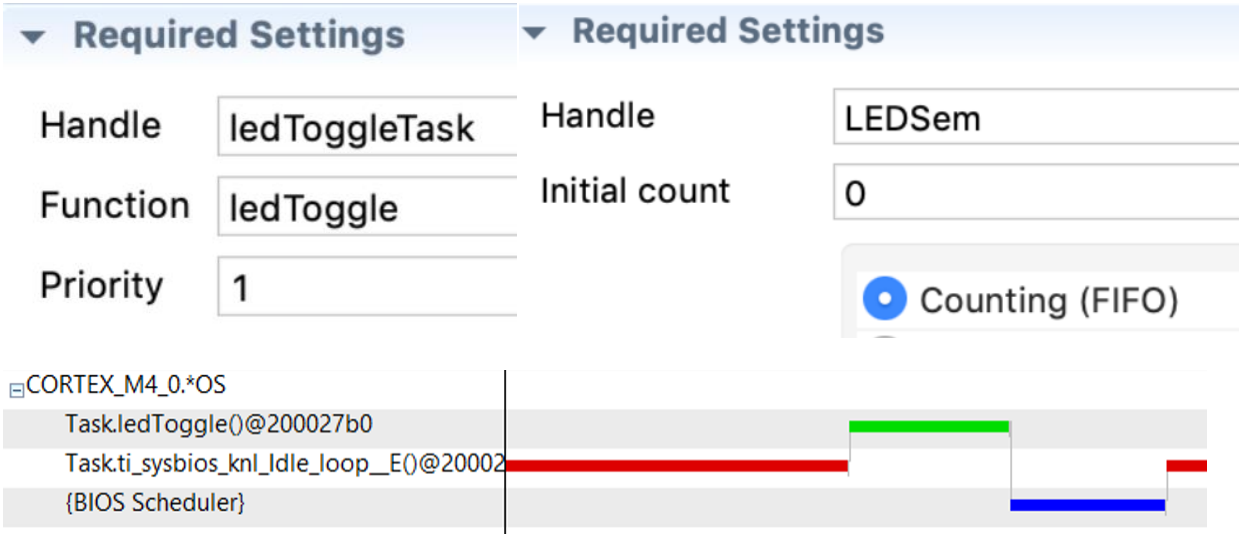
```
    TimerConfigure(TIMER2_BASE, TIMER_CFG_PERIODIC);        // cfg Timer 2 mode -
periodic

    ui32Period = (SysCtlClockGet() /2);                     // period = CPU clk div 2
(500ms)
    TimerLoadSet(TIMER2_BASE, TIMER_A, ui32Period);         // set Timer 2 period

    TimerIntEnable(TIMER2_BASE, TIMER_TIMA_TIMEOUT);        // enables Timer 2 to
interrupt CPU

    TimerEnable(TIMER2_BASE, TIMER_A);                      // enable Timer 2

}


//*****************************************************************************
// ledToggle()
//
// toggles LED on Tiva-C LaunchPad
//*****************************************************************************
void ledToggle(void)
{

    while(1)
    {
        Semaphore_pend(LEDSem, BIOS_WAIT_FOREVER);              // wait for Sem from
ISR

        // LED values - 2=RED, 4=BLUE, 8=GREEN
        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);
        }

        i16ToggleCount += 1;                                    // keep track of
#toggles

        Log_info1("LED TOGGLED [%u] TIMES",i16ToggleCount);     // send toggle count
to UIA

    }

}


//*****************************************************************************
// Timer ISR - called by BIOS Hwi (see app.cfg)
//
// Posts Swi (or later a Semaphore) to toggle the LED
//*****************************************************************************
```

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

```
void Timer_ISR(void)
{
    TimerIntClear(TIMER2_BASE, TIMER_TIMA_TIMEOUT);          // must clear timer flag
FROM timer

    Semaphore_post(LEDSem);
}
```

Task 02:

5		4000089550	CORT...	LM_switch: oldtsk: 0x200027b0, oldfunc...
5		4000158650	CORT...	LS_taskLoad: 0x200027b0,1543,200003...
5		4000164725	CORT...	LS_taskLoad: 0x20002800,19998839,200...
5		4000172200	CORT...	LS_cpuLoad: 0%



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