**Date Submitted: 11/28/2019**

**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);

//---------------------------------------

// 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);

}

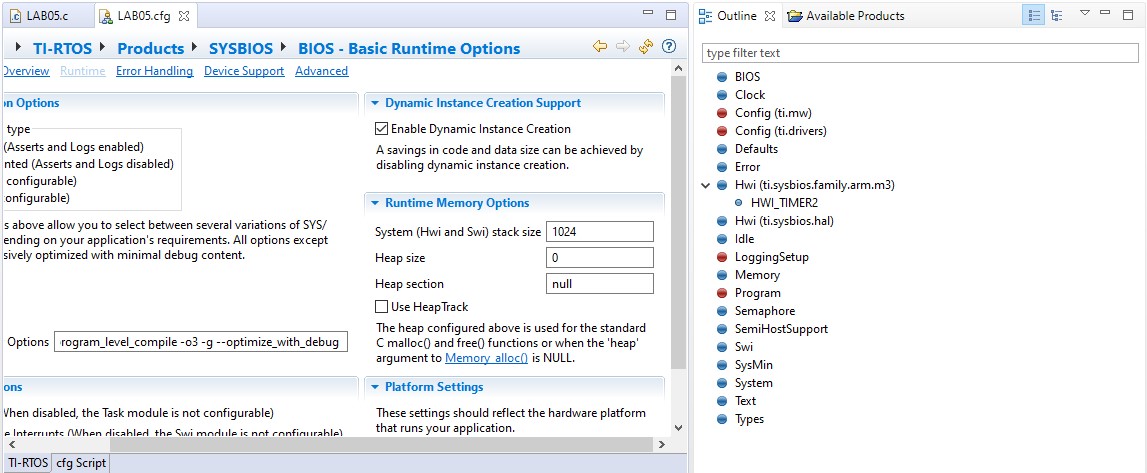
i16ToggleCount += 1; // keep track of #toggles

Log\_info1("LED TOGGLED [%u] TIMES", i16ToggleCount); // send toggle count to UIA

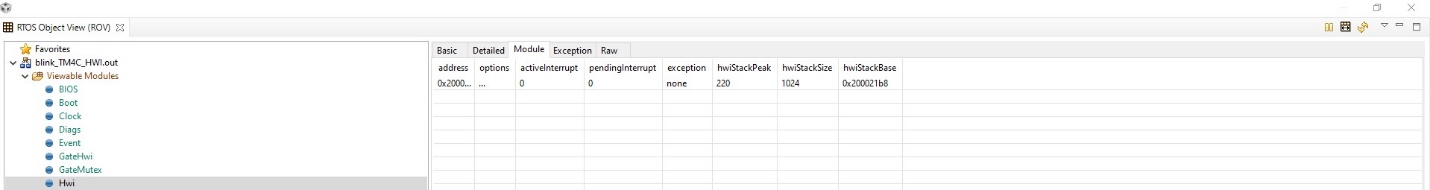
}

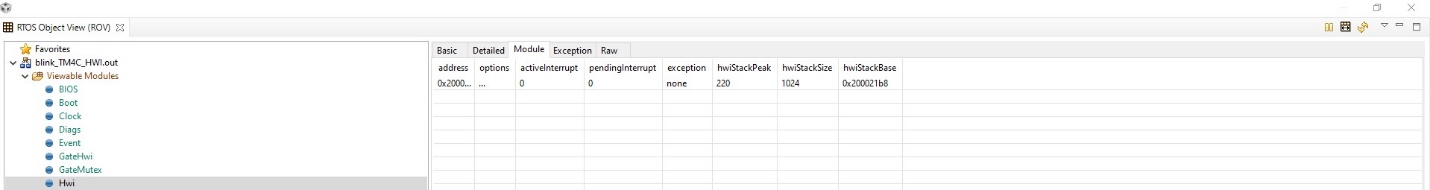
**Screenshots:**

**Cfg:**

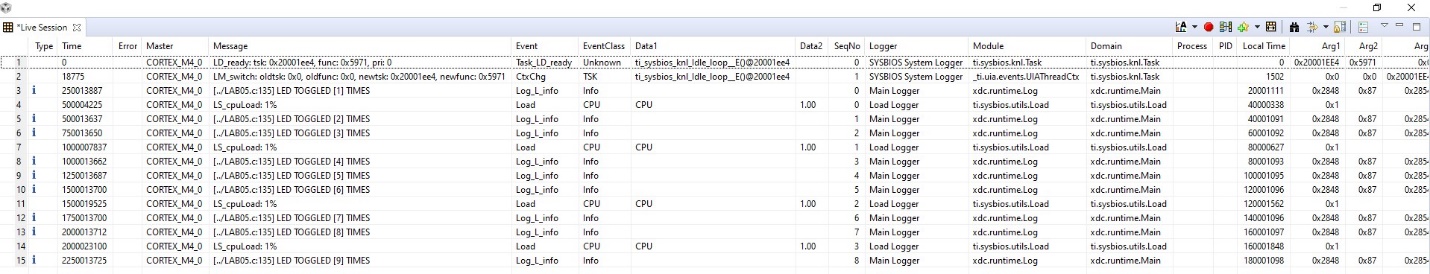
****

**ROV:**

****

****

**Log Outputs:**

****