Nathan Hanuscin
Github root directory: https://github.com/nhanuscin/HappyFunStuff

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

Task 01:

Youtube Link: N/A

Modified Schematic (if applicable): N/A

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);


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


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

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

```
{

   hardware_init();                                    // init hardware via Xware

   BIOS_start();

}


//-------------------------------------------------------------------------
// 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);


}


//-------------------------------------------------------------------------
// 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;

      // 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
```

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

```
            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

}
-------------------------------------------------------------------------------
```