CPE403 – Advanced Embedded Systems

Design Assignment 3

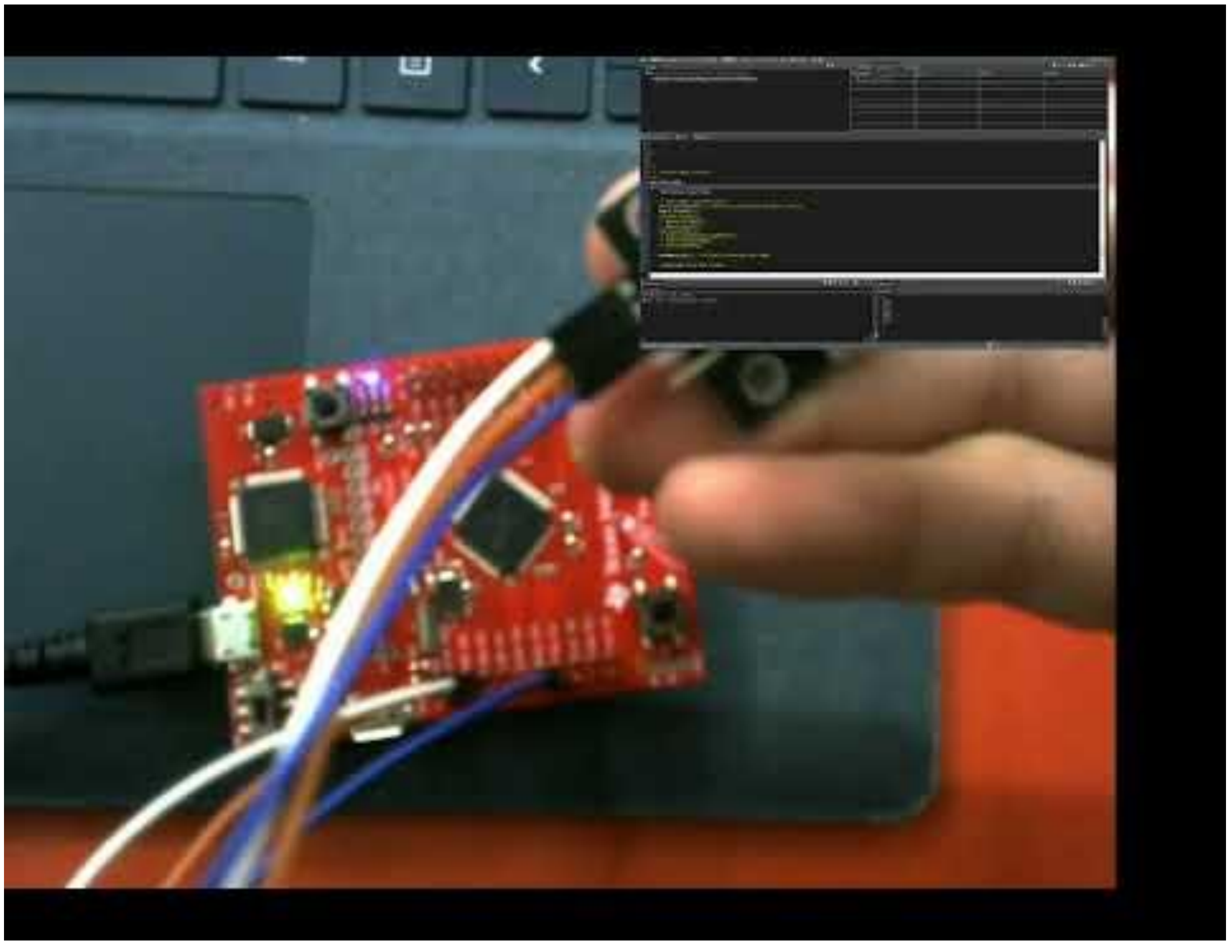
DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Angelo Nolasco

Email: Nolasco@unlv.nevada.edu

Github Repository link (root): https://github.com/AngeloNol/Design_Assignments

Youtube Playlist link (root): Assignment 3



1. Code for Tasks

```c
/*
*  ======== AS3_template.c ========
*  Angelo Nolasco, DA3, CPE403
*/
/* XDCtools Header files */
#define TARGET_IS_BLIZZARD_RB1
#include <xdc/std.h>
#include <xdc/runtime/System.h>
#include <xdc/cfg/global.h>
#include <xdc/runtime/Log.h>

/* BIOS Header files */
#include <ti/sysbios/BIOS.h>
#include <ti/sysbios/knl/Task.h>
#include <ti/sysbios/knl/Semaphore.h>
#include <ti/sysbios/knl/Task.h>


/* TI-RTOS Header files */
#include <ti/drivers/GPIO.h> //all real time compatible
#include <ti/drivers/UART.h>


/* Board Header file */
#include "Board.h" //configuration comes from here

//
#include "inc/hw_i2c.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
```

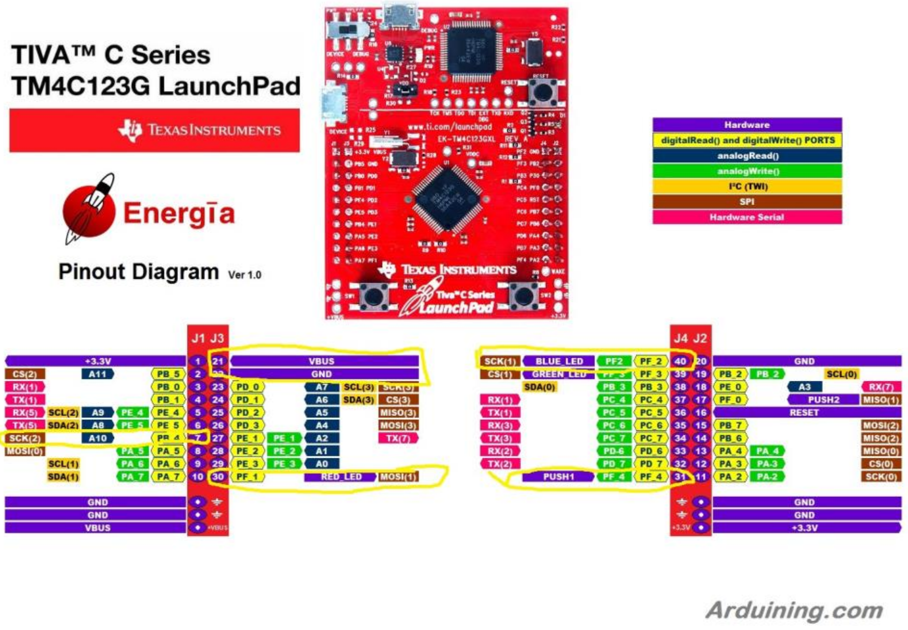
```
#include "inc/hw_ints.h"
#include "inc/hw_gpio.h"
#include <stdint.h>
#include <stdarg.h>
#include <stdbool.h>
#include <string.h>
#include <time.h>
#include "driverlib/i2c.h"
#include "driverlib/debug.h"
#include "driverlib/gpio.h"
#include "driverlib/gpio.c"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"
#include "driverlib/pin_map.h"
#include "driverlib/adc.h"
#include "driverlib/adc.c"
#include "driverlib/sysctl.c"
#include "driverlib/interrupt.h"
#include "driverlib/timer.h"
#include "driverlib/pwm.h"
#include "driverlib/i2c.h"
#include "driverlib/rom.h"
#include "driverlib/rom_map.h"
#include "utils/uartstdio.h"
#include "inc/tm4c123gh6pm.h"

//
//definitions and global variables
#define TASKSTACKSIZE   512
#define PWM_FREQUENCY 100
```

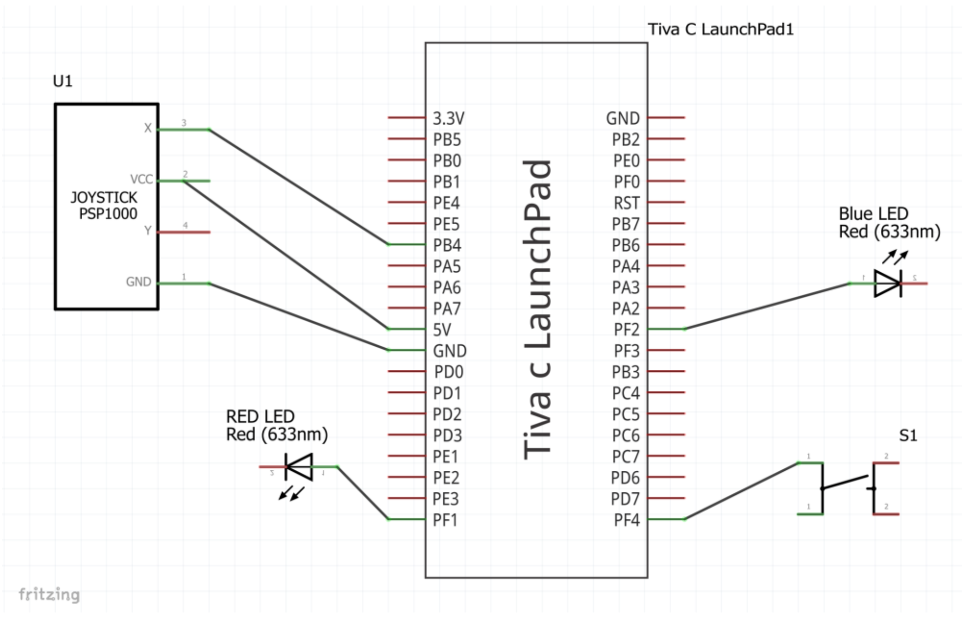
```
uint32_t count = 0; //gets current count of timer interrupts
char check; //gets input from user
uint32_t ui32PinStatus = 0x00000000; //variable to store the pin status of GPIO PortF
uint32_t ui32Period;

//PWM variables
volatile uint32_t ui32Load;
volatile uint32_t ui32PWMClock;
volatile uint8_t ui8Adjust = 1;

//ADC variables
uint32_t xValue[6];                 // Array to store the ADC values of X
volatile uint32_t xValueAvg;              // Variable to store the Average of ADC values of X
char    buffer[4];

Task_Struct task1Struct,task2Struct,task3Struct;

Char task1Stack[TASKSTACKSIZE];
Char task2Stack[TASKSTACKSIZE];
Char task3Stack[TASKSTACKSIZE];

/*
*  ======== heartBeatFxn ========
*  Toggle the Board_LED0. The Task_sleep is determined by arg0 which
*  is configured for the heartBeat Task instance.
*/
void heartBeatFxn(UArg arg0, UArg arg1)
{
// Fill heartbeat function
   while(1){
        Task_sleep(1000);
        GPIO_toggle(Board_LED2);
    }
}

//ALL TASKS

//ADC function
void task1(void)
{
    while(1){
        Semaphore_pend(taskSem1, BIOS_WAIT_FOREVER); //wait for Semaphore from ISR

            //ADC performed, store ADC value
            ADCIntClear(ADC0_BASE, 0);
          ADCProcessorTrigger(ADC0_BASE, 0);
          ADCSequenceDataGet(ADC0_BASE, 0, xValue);

           xValueAvg = (xValue[0] + xValue[1] + xValue[2] + xValue[3]
                             + xValue[4] + xValue[5]+ 4) / 6;



}


}

//UART function
void task2(void)
{
    while(1){
        Semaphore_pend(taskSem2, BIOS_WAIT_FOREVER); //wait for Semaphore from ISR
```

```
            //UART performed, display stored ADC value
        UARTprintf("X: %d\t",xValueAvg );
        UARTprintf("\n");
}

}

//PWM Update function
void task3(void)
{
     while(1){
        Semaphore_pend(taskSem3, BIOS_WAIT_FOREVER); //wait for Semaphore from ISR

            //Update PWM DC, from the value captured in
            if(xValueAvg < 1570){
                ui8Adjust = xValueAvg;

                if (ui8Adjust < 1)
                {
                            ui8Adjust = 1;

                }
            }
            if(xValueAvg > 1570){
                ui8Adjust = xValueAvg;

                if (ui8Adjust > 100)
                {
                            ui8Adjust = 100;
```

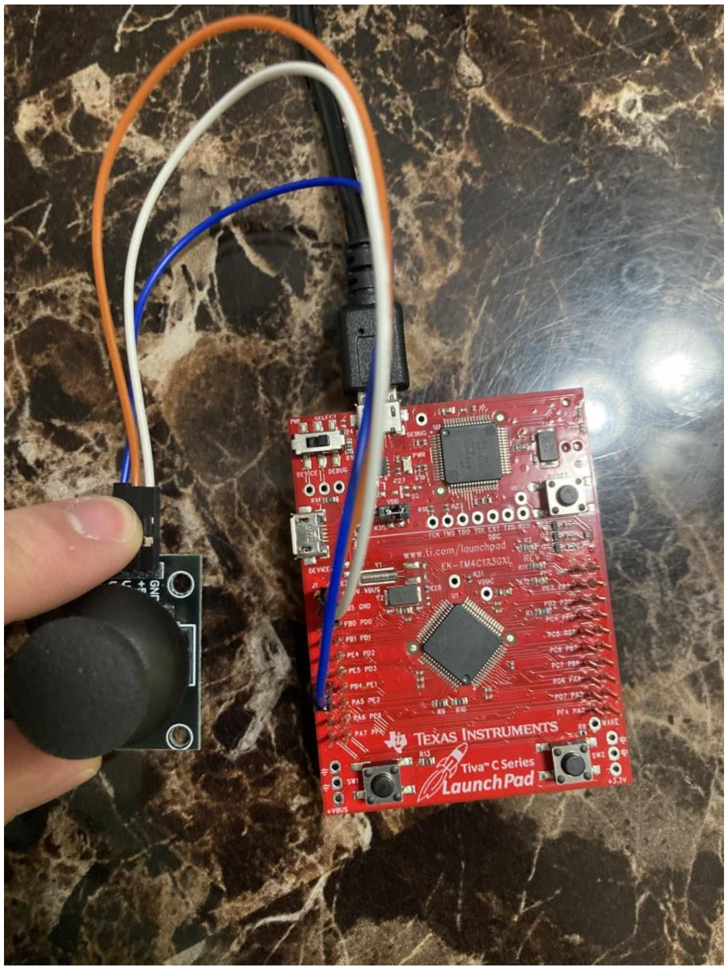
```
                    }
                  }
                }
            }
```

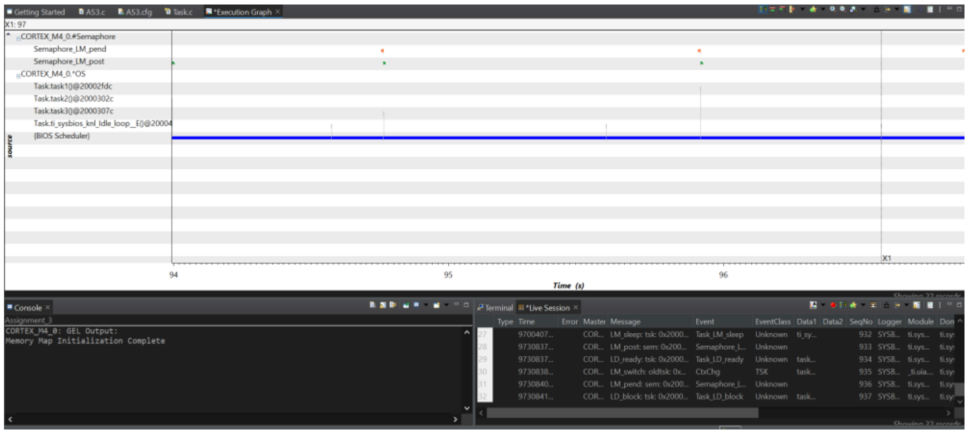
2. Block diagram and/or Schematics showing the components, pins used, and interface.





3. Screenshots of the IDE, physical setup, debugging process





Assignment_3

```
CORTEX_M4_0: GEL Output:
Memory Map Initialization Complete
```

	Type	Time	Error	Master	Message	Event	EventClass	Data1	Data2	SeqNo	Logger	Module	Dom
27		9700407...		COR...	LM_sleep: tsk: 0x2000...	Task_LM_sleep	Unknown	ti_sy...		932	SYSB...	ti.sys...	ti.sy
28		9730837...		COR...	LM_post: sem: 0x200...	Semaphore_L...	Unknown			933	SYSB...	ti.sys...	ti.sy
29		9730837...		COR...	LD_ready: tsk: 0x2000...	Task_LD_ready	Unknown	task...		934	SYSB...	ti.sys...	ti.sy
30		9730838...		COR...	LM_switch: oldtsk: 0x...	CtxChg	TSK	task...		935	SYSB...	_ti.uia...	ti.sy
31		9730840...		COR...	LM_pend: sem: 0x200...	Semaphore_L...	Unknown			936	SYSB...	ti.sys...	ti.sy
32		9730841...		COR...	LD_block: tsk: 0x2000...	Task_LD_block	Unknown	task...		937	SYSB...	ti.sys...	ti.sy

4. Declaration
   I understand the Student Academic Misconduct Policy -
   http://studentconduct.unlv.edu/misconduct/policy.html

“This assignment submission is my own, original work”.
Angelo Nolasco