CPE403 – Advanced Embedded Systems

Design Assignment

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Sai Balaji Jai Kumar

Email: jaikumar@unlv.nevada.edu

Github Repository link (root):
https://github.com/saibalaji1997/githubfiles/tree/main/CC1352/Assignment%204

Youtube Playlist link (root): https://www.youtube.com/watch?v=Qulj8gxByrg

Follow the submission guideline to be awarded points for this Assignment.

Submit the following for all Assignments:

1. In the document, for each task submit the modified or included code (from the base code) with highlights and justifications of the modifications. Also include the comments. If no base code is provided, submit the base code for the first task only.
2. Create a private Github repository with a random name (no CPE/403, Lastname, Firstname). Place all labs under the root folder TIVAC, sub-folder named Assignment1, with one document and one video link file for each lab, place modified c files named as asng_taskxx.c.
3. If multiple c files or other libraries are used, create a folder asng1_t01 and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) with startup_ccs.c and other include files, c) text file with youtube video links (see template).
5. Submit the doc file in canvas before the due date. The root folder of the github assignment directory should have the documentation and the text file with youtube video links.
6. Organize your youtube videos as playlist under the name “cpe403”. The playlist should have the video sequence arranged as submission or due dates.
7. Only submit pdf documents. Do not forget to upload this document in the github repository and in the canvas submission portal.

1. Code for Tasks. for each task submit the modified or included code (from the base code) with highlights and justifications of the modifications. Also include the comments. If no base code is provided, submit the base code for the first task only. Use separate page for each task.

Mutex Code:

```
/* XDC module Headers */
#include <xdc/std.h>
#include <xdc/runtime/System.h>

/* BIOS module Headers */
#include <ti/sysbios/BIOS.h>
#include <ti/sysbios/knl/Clock.h>
#include <ti/sysbios/knl/Task.h>
#include <ti/sysbios/knl/Semaphore.h>

#include <ti/drivers/Board.h>

#define TASKSTACKSIZE   512

Void task1Fxn(UArg arg0, UArg arg1);
Void task2Fxn(UArg arg0, UArg arg1);

Int resource = 0;
Int finishCount = 0;
UInt32 sleepTickCount;

Task_Struct task1Struct, task2Struct;
Char task1Stack[TASKSTACKSIZE], task2Stack[TASKSTACKSIZE];
Semaphore_Struct semStruct;
Semaphore_Handle semHandle;
```

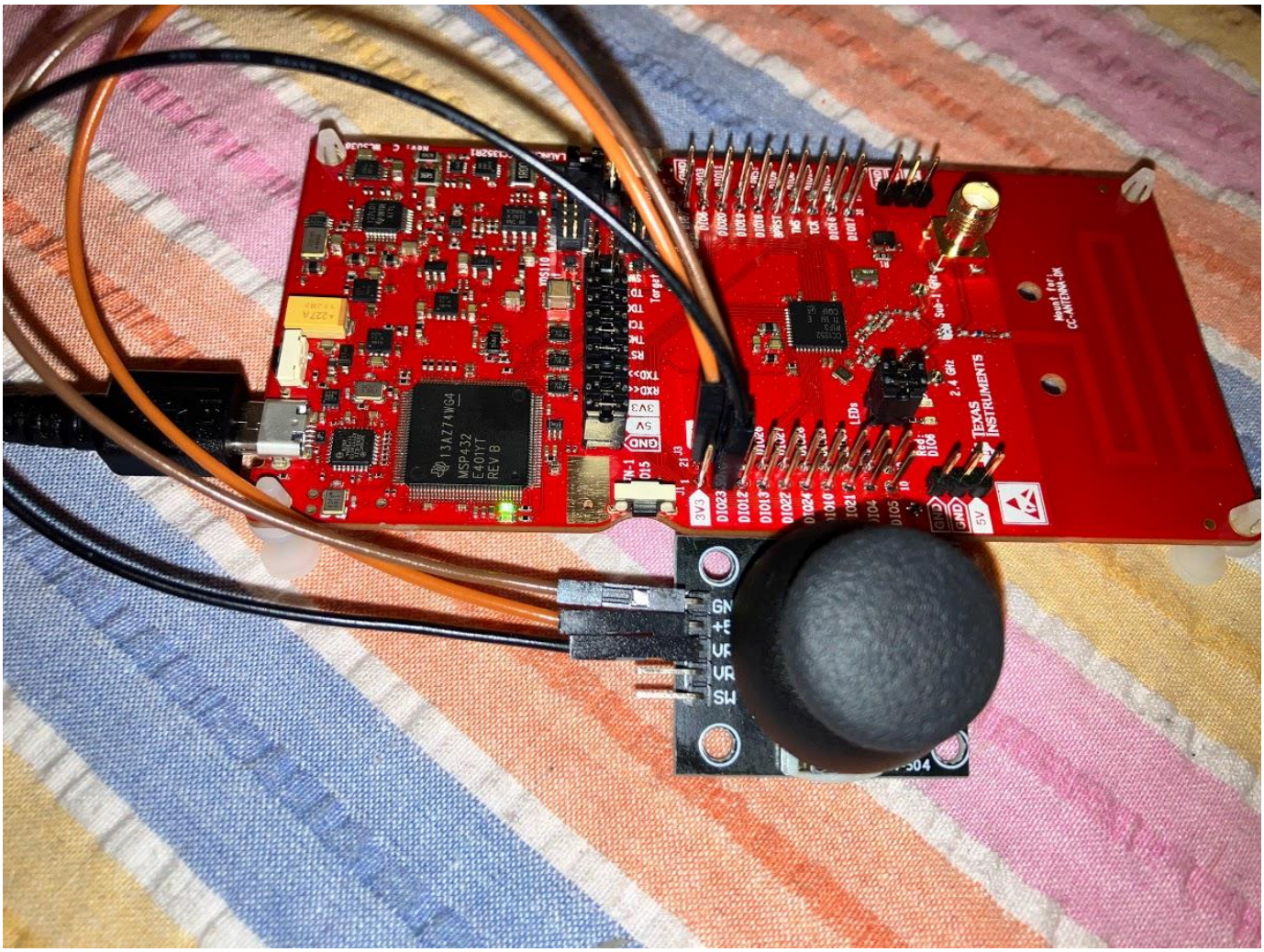
```
/*
 *  ======== main ========
 */
int main()
{
    /* Construct BIOS objects */
    Task_Params taskParams;
    Semaphore_Params semParams;

    /* Call driver init functions */
    Board_init();

    /* Construct writer/reader Task threads */
    Task_Params_init(&taskParams);
    taskParams.stackSize = TASKSTACKSIZE;
    taskParams.stack = &task1Stack;
    taskParams.priority = 1;
    Task_construct(&task1Struct, (Task_FuncPtr)task1Fxn, &taskParams, NULL);

    taskParams.stack = &task2Stack;
    taskParams.priority = 2;
    Task_construct(&task2Struct, (Task_FuncPtr)task2Fxn, &taskParams, NULL);

    /* Construct a Semaphore object to be use as a resource lock, inital count 1 */
    Semaphore_Params_init(&semParams);
    Semaphore_construct(&semStruct, 1, &semParams);

    /* Obtain instance handle */
    semHandle = Semaphore_handle(&semStruct);
```

```
    /* We want to sleep for 10000 microseconds */
    sleepTickCount = 10000 / Clock_tickPeriod;

    BIOS_start();    /* Does not return */
    return(0);
}

/*
 *  ======== task1Fxn ========
 */
Void task1Fxn(UArg arg0, UArg arg1)
{
    UInt32 time;

    for (;;) {
        System_printf("Running task1 function\n");

        if (Semaphore_getCount(semHandle) == 0) {
            System_printf("Sem blocked in task1\n");
        }

        /* Get access to resource */
        Semaphore_pend(semHandle, BIOS_WAIT_FOREVER);

        /* Do work by waiting for 2 system ticks to pass */
        time = Clock_getTicks();
        while (Clock_getTicks() <= (time + 1)) {
            ;
        }

        /* Do work on locked resource */
```

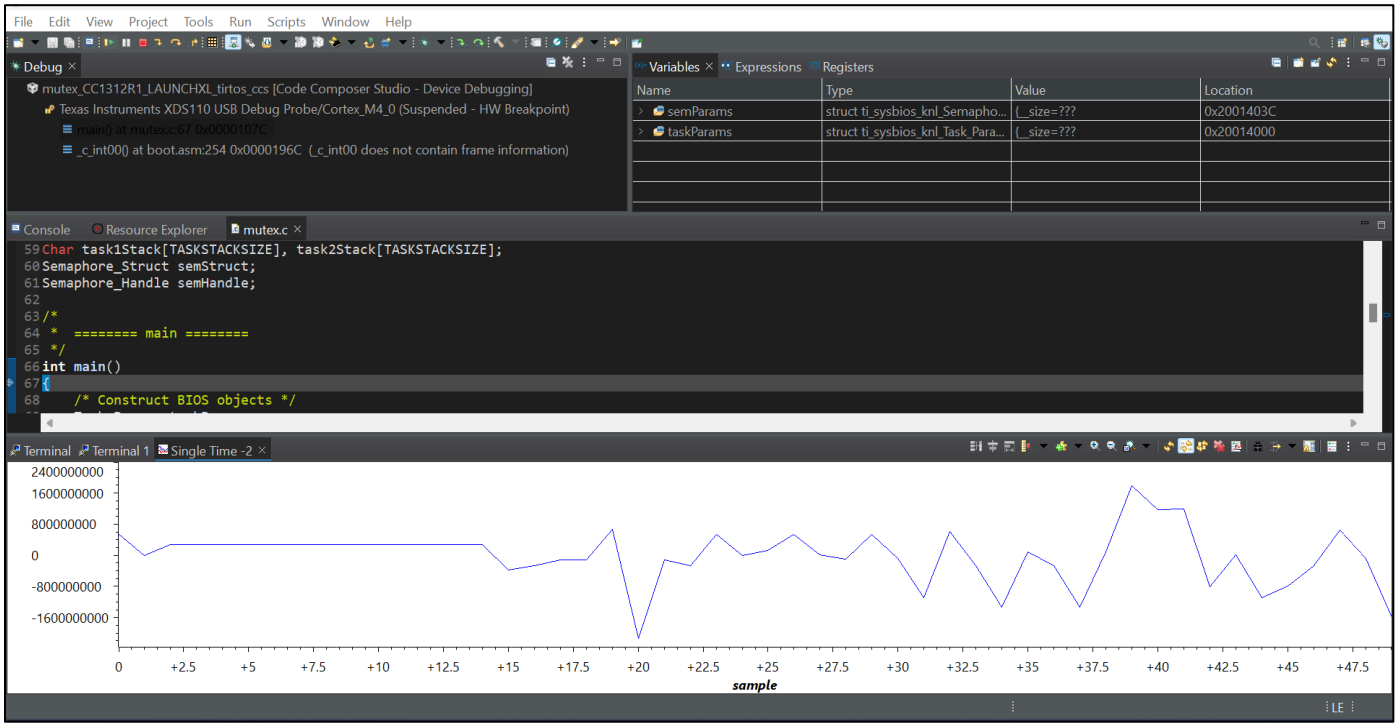
```
        resource += 1;
        /* Unlock resource */

        Semaphore_post(semHandle);

        Task_sleep(sleepTickCount);
    }
}

/*
 *  ======== task2Fxn ========
 */
Void task2Fxn(UArg arg0, UArg arg1)
{
    for (;;) {
        System_printf("Running task2 function\n");

        if (Semaphore_getCount(semHandle) == 0) {
            System_printf("Sem blocked in task2\n");
        }

        /* Get access to resource */
        Semaphore_pend(semHandle, BIOS_WAIT_FOREVER);

        /* Do work on locked resource */
        resource += 1;
        /* Unlock resource */

        Semaphore_post(semHandle);

        Task_sleep(sleepTickCount);
```

```
        finishCount++;
        if (finishCount == 5) {
            System_printf("Calling BIOS_exit from task2\n");
            BIOS_exit(0);
        }
    }
}
```

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



3. Screenshots of the IDE, physical setup, debugging process - Provide screenshot of successful compilation, screenshots of registers, variables, graphs, etc.



4. Declaration

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

"This assignment submission is my own, original work".

Name of the Student

Sai Balaji Jai Kumar