Geovanni Portillo
Github root directory: https://github.com/portig1/submissions_E2

Date Submitted: **November 11, 2019**

Task 01:

Youtube Link: https://youtu.be/9rmPEYZaULg

Modified Code:

```
/* TI-RTOS Header files */
#include <xdc/std.h>
#include <ti/sysbios/BIOS.h>
#include <ti/sysbios/knl/Task.h>

#include <ti/drivers/GPIO.h>

/* Example/Board Header files */
#include "ti_drivers_config.h" //enable ability to use red led. no other leds are
defined.
//#include <ti/boards/CC1352R1_LAUNCHXL/Board.h>
//Board.h was not defined for CC1352 but boards folder has a defined version
//CC1352R1 header does not work with GPIO_write functions. Result is pin states are
not changed
//Used syscfg tool to enable the green LED

void myDelay(int count);

/* Could be anything, like computing primes */
#define FakeBlockingSlowWork() myDelay(12000000)
#define FakeBlockingFastWork() myDelay(2000000)

Task_Struct workTask;
/* Make sure we have nice 8-byte alignment on the stack to avoid wasting memory */
#pragma DATA_ALIGN(workTaskStack, 8)
#define STACKSIZE 1024
static uint8_t workTaskStack[STACKSIZE];

void doUrgentWork(void)
{
    GPIO_write(CONFIG_GPIO_LED_1, CONFIG_GPIO_LED_OFF);
    FakeBlockingFastWork(); /* Pretend to do something useful but time-consuming */
    GPIO_write(CONFIG_GPIO_LED_1, CONFIG_GPIO_LED_ON);
}

void doWork(void)
{
    GPIO_write(CONFIG_GPIO_LED_0, CONFIG_GPIO_LED_OFF);
    FakeBlockingSlowWork(); /* Pretend to do something useful but time-consuming */
    GPIO_write(CONFIG_GPIO_LED_0, CONFIG_GPIO_LED_ON);
}

void workTaskFunc(UArg arg0, UArg arg1)
```

```
{
    while (1) {
        /* Do work */
        doWork();

        /* Wait a while, because doWork should be a periodic thing, not continuous.*/
        myDelay(24000000);
    }
}
/*
* * ======== main ========
*
*/

int main(void)
{
    Board_init();
    GPIO_init();

    /* Set up the led task */
    Task_Params workTaskParams;
    Task_Params_init(&workTaskParams);
    workTaskParams.stackSize = STACKSIZE;
    workTaskParams.priority = 2;
    workTaskParams.stack = &workTaskStack;

    Task_construct(&workTask, workTaskFunc, &workTaskParams, NULL);

    /* Start kernel. */
    BIOS_start();

    return (0);
}
/*
* ======== myDelay ========
* Assembly function to delay. Decrements the count until it is zero
* The exact duration depends on the processor speed.
*/
__asm(  " .sect \".text:myDelay\"\n"
        " .clink\n"
        " .thumbfunc myDelay\n"
        " .thumb\n"
        " .global myDelay\n"
        "myDelay:\n"
        " subs r0, #1\n"
        " bne.n myDelay\n"
        " bx lr\n");


-------------------------------------------------------------------------------
```

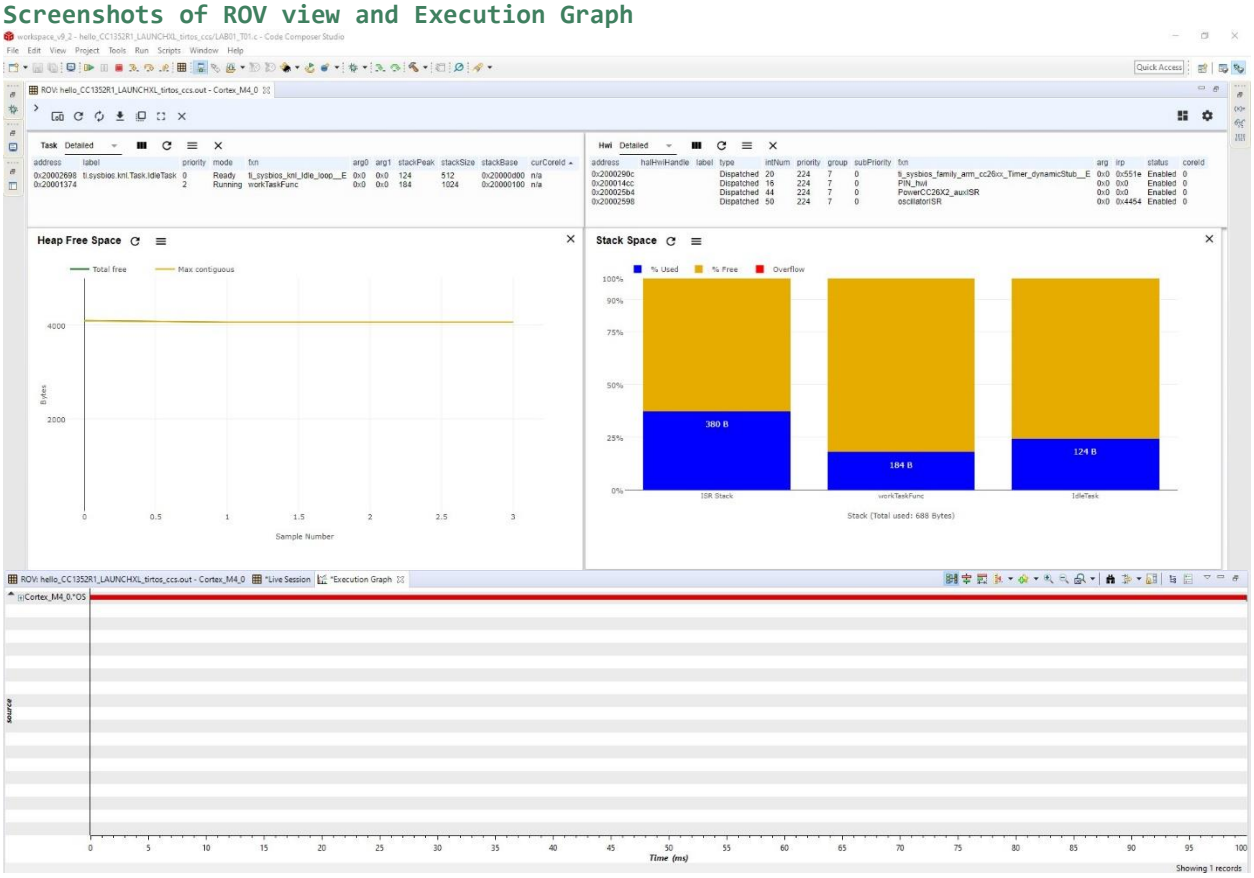
Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Geovanni Portillo
Github root directory: https://github.com/portig1/submissions_E2

Task 02:

No video was done for Task 2 as the task focused on being able to use the ROV and Execution graph

Screenshots of ROV view and Execution Graph



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

Geovanni Portillo
Github root directory: https://github.com/portig1/submissions_E2

Task 03:

Task 3 performs the same function as Task 1
Modified Code:

```
/* TI-RTOS Header files */
#include <xdc/std.h>
#include <ti/sysbios/BIOS.h>
#include <ti/sysbios/knl/Task.h>

#include <ti/drivers/GPIO.h>

/* Example/Board Header files */
#include "ti_drivers_config.h" //enable ability to use red led. no other leds are
defined.
//#include <ti/boards/CC1352R1_LAUNCHXL/Board.h>
//Board.h was not defined for CC1352 but boards folder has a defined version
//CC1352R1 header does not work with GPIO_write functions. Result is pin states are
not changed
//Used syscfg tool to enable the green LED

#include <ti/sysbios/knl/Clock.h> //Used for clock_tickPeriod

void myDelay(int count);

/* Could be anything, like computing primes */
#define FakeBlockingSlowWork() myDelay(12000000)
#define FakeBlockingFastWork() myDelay(2000000)

Task_Struct workTask;
/* Make sure we have nice 8-byte alignment on the stack to avoid wasting memory */
#pragma DATA_ALIGN(workTaskStack, 8)
#define STACKSIZE 1024
static uint8_t workTaskStack[STACKSIZE];

void doUrgentWork(void)
{
    GPIO_write(CONFIG_GPIO_LED_1, CONFIG_GPIO_LED_OFF);
    FakeBlockingFastWork(); /* Pretend to do something useful but time-consuming */
    GPIO_write(CONFIG_GPIO_LED_1, CONFIG_GPIO_LED_ON);
}

void doWork(void)
{
    GPIO_write(CONFIG_GPIO_LED_0, CONFIG_GPIO_LED_OFF);
    FakeBlockingSlowWork(); /* Pretend to do something useful but time-consuming */
    GPIO_write(CONFIG_GPIO_LED_0, CONFIG_GPIO_LED_ON);
}

void workTaskFunc(UArg arg0, UArg arg1)
{
    while (1) {
        /* Do work */
```

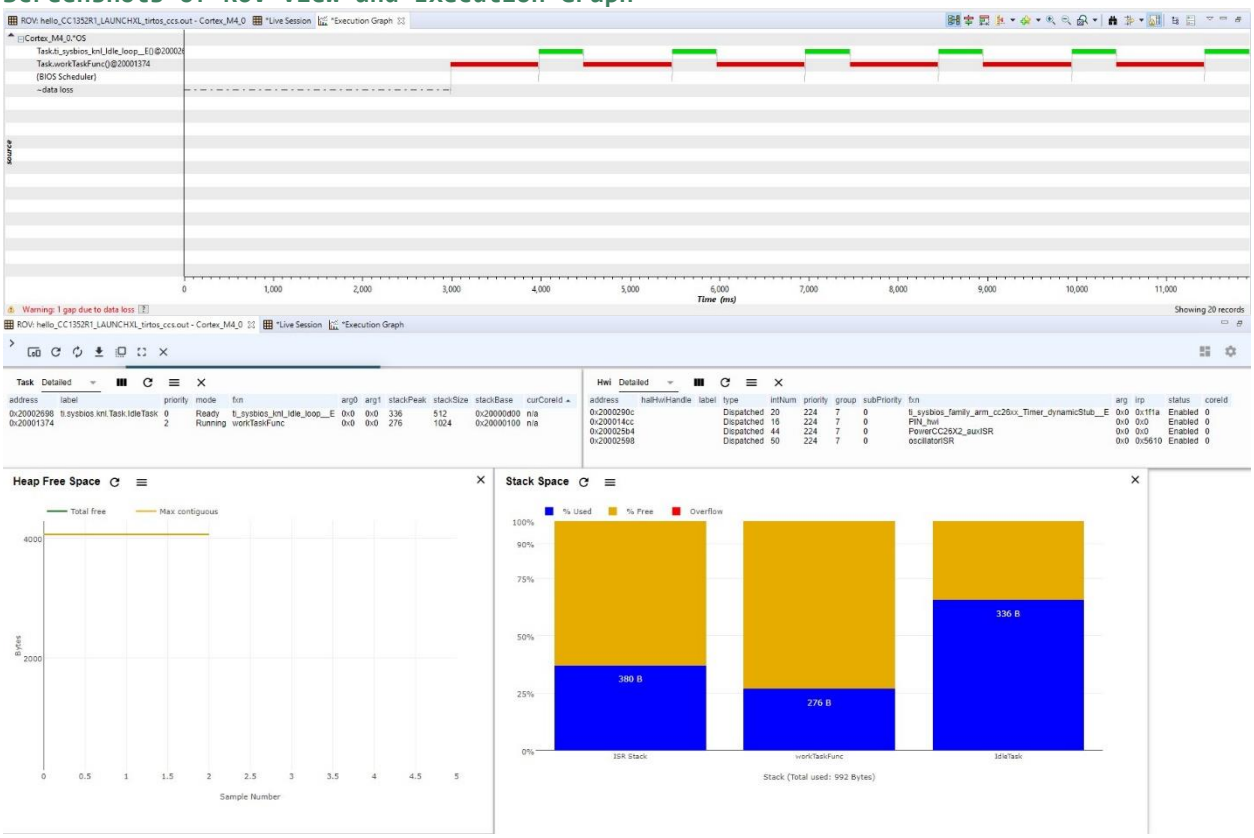
Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

```
        doWork();

        /* Wait a while, because doWork should be a periodic thing, not continuous.*/
        //myDelay(24000000);
        Task_sleep(500 * (1000 / Clock_tickPeriod));
    }
}
/*
* * ======== main ========
*
*/

int main(void)
{
    Board_init();
    GPIO_init();

    /* Set up the led task */
    Task_Params workTaskParams;
    Task_Params_init(&workTaskParams);
    workTaskParams.stackSize = STACKSIZE;
    workTaskParams.priority = 2;
    workTaskParams.stack = &workTaskStack;

    Task_construct(&workTask, workTaskFunc, &workTaskParams, NULL);

    /* Start kernel. */
    BIOS_start();

    return (0);
}
/*
* ======== myDelay ========
* Assembly function to delay. Decrements the count until it is zero
* The exact duration depends on the processor speed.
*/
__asm(  " .sect \".text:myDelay\"\n"
        " .clink\n"
        " .thumbfunc myDelay\n"
        " .thumb\n"
        " .global myDelay\n"
        "myDelay:\n"
        " subs r0, #1\n"
        " bne.n myDelay\n"
        " bx lr\n");
```

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

Geovanni Portillo
Github root directory: https://github.com/portig1/submissions_E2

Screenshots of ROV view and Execution Graph



Task 04:

Youtube Link: https://youtu.be/X1nR6OiCxjk

Modified Code:

```
/* TI-RTOS Header files */
#include <xdc/std.h>
#include <ti/sysbios/BIOS.h>
#include <ti/sysbios/knl/Task.h>

#include <ti/drivers/GPIO.h>

/* Example/Board Header files */
#include "ti_drivers_config.h" //enable ability to use red led. no other leds are
defined.
//#include <ti/boards/CC1352R1_LAUNCHXL/Board.h>
//Board.h was not defined for CC1352 but boards folder has a defined version
//CC1352R1 header does not work with GPIO_write functions. Result is pin states are
not changed
//Used syscfg tool to enable the green LED

#include <ti/sysbios/knl/Clock.h> //Used for clock_tickPeriod
```

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

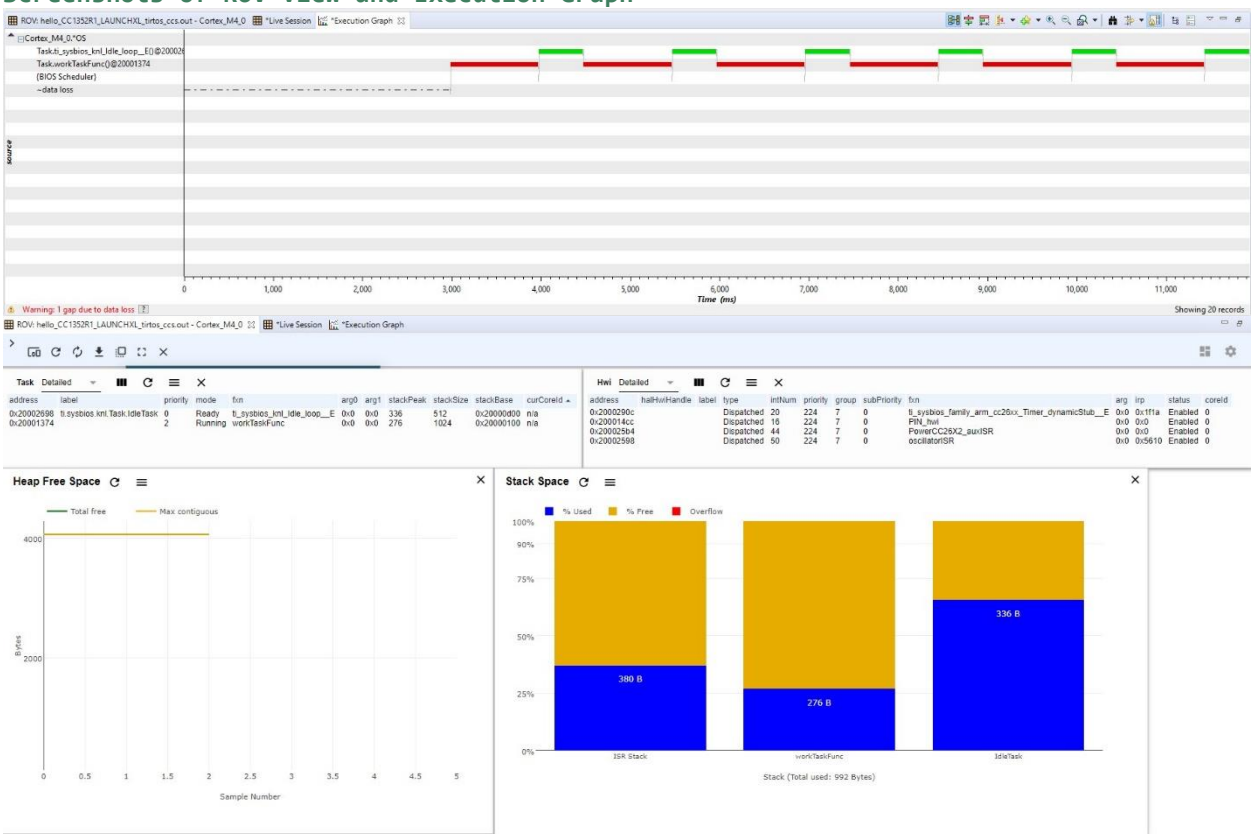
```
void myDelay(int count);

/* Could be anything, like computing primes */
#define FakeBlockingSlowWork() myDelay(12000000)
#define FakeBlockingFastWork() myDelay(2000000)

Task_Struct workTask;
Task_Struct urgentWorkTask;
/* Make sure we have nice 8-byte alignment on the stack to avoid wasting memory */
#pragma DATA_ALIGN(workTaskStack, 8)
#define STACKSIZE 1024
static uint8_t workTaskStack[STACKSIZE];
static uint8_t urgentWorkTaskStack[STACKSIZE];

void doUrgentWork(void)
{
    GPIO_write(CONFIG_GPIO_LED_1, CONFIG_GPIO_LED_OFF);
    FakeBlockingFastWork(); /* Pretend to do something useful but time-consuming */
    GPIO_write(CONFIG_GPIO_LED_1, CONFIG_GPIO_LED_ON);
}

void doWork(void)
{
    GPIO_write(CONFIG_GPIO_LED_0, CONFIG_GPIO_LED_OFF);
    FakeBlockingSlowWork(); /* Pretend to do something useful but time-consuming */
    GPIO_write(CONFIG_GPIO_LED_0, CONFIG_GPIO_LED_ON);
}

void workTaskFunc(UArg arg0, UArg arg1)
{
    while (1) {
        /* Do work */
        doWork();

        /* Wait a while, because doWork should be a periodic thing, not continuous.*/
        //myDelay(24000000);
        Task_sleep(500 * (1000 / Clock_tickPeriod));
    }
}

void urgentWorkTaskFunc(UArg arg0, UArg arg1)
{
    while (1) {
        /* Do work */
        doUrgentWork();

        /* Wait a while, because doWork should be a periodic thing, not continuous.*/
        //myDelay(24000000);
        Task_sleep(50 * (1000 / Clock_tickPeriod));
    }
}

/*
* * ======== main ========
*
```

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

```
*/

int main(void)
{
    Board_init();
    GPIO_init();

    /* Set up the led task */
    Task_Params workTaskParams;
    Task_Params_init(&workTaskParams);
    workTaskParams.stackSize = STACKSIZE;
    workTaskParams.priority = 2;
    workTaskParams.stack = &workTaskStack;

    Task_construct(&workTask, workTaskFunc, &workTaskParams, NULL);

    workTaskParams.priority = 3;
    workTaskParams.stack = &urgentWorkTaskStack;

    Task_construct(&urgentWorkTask, urgentWorkTaskFunc, &workTaskParams, NULL);

    /* Start kernel. */
    BIOS_start();

    return (0);
}
/*
* ======== myDelay ========
* Assembly function to delay. Decrements the count until it is zero
* The exact duration depends on the processor speed.
*/
__asm(  " .sect \".text:myDelay\"\n"
        " .clink\n"
        " .thumbfunc myDelay\n"
        " .thumb\n"
        " .global myDelay\n"
        "myDelay:\n"
        " subs r0, #1\n"
        " bne.n myDelay\n"
        " bx lr\n");
```

Geovanni Portillo
Github root directory: https://github.com/portig1/submissions_E2

Screenshots of ROV view and Execution Graph



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