**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"

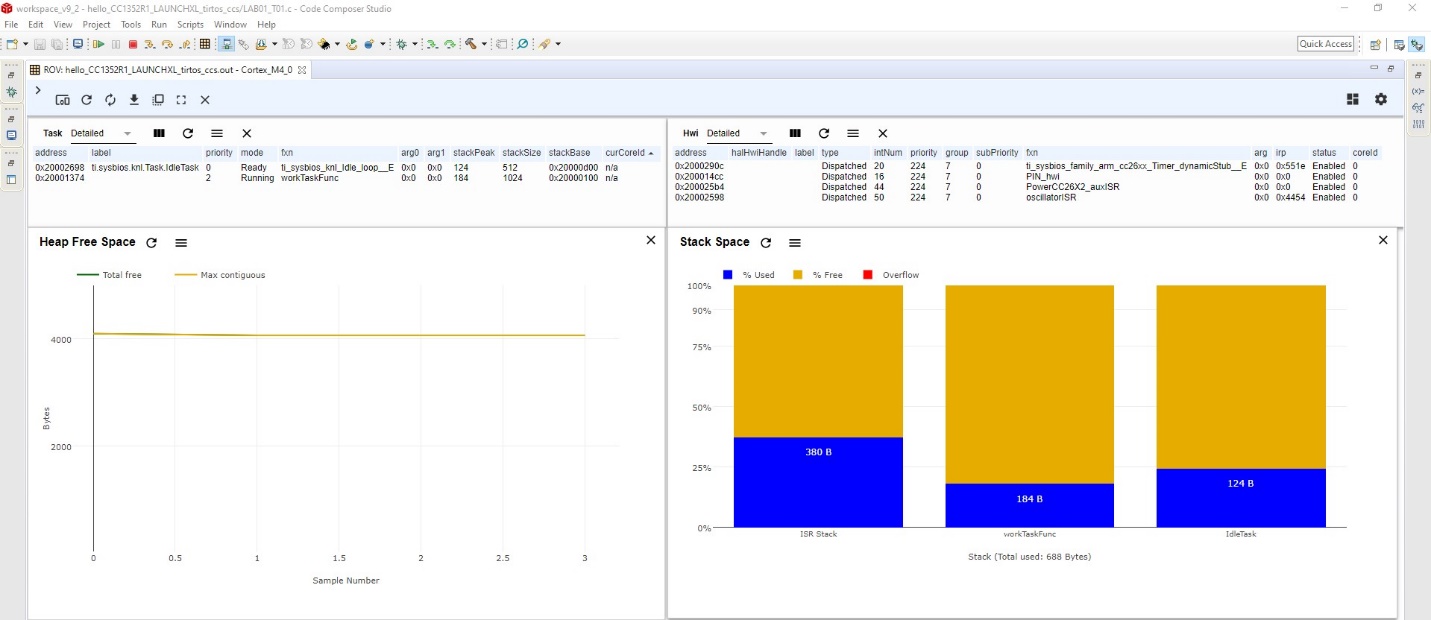
" bx lr\n");

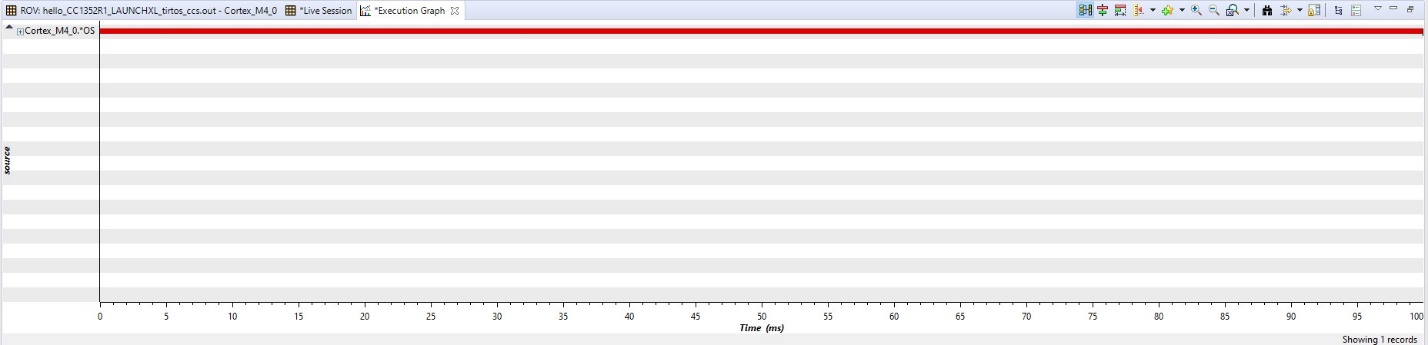
**------------------------------------------------------------------------------------**

**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**

****

****

**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 \*/

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"

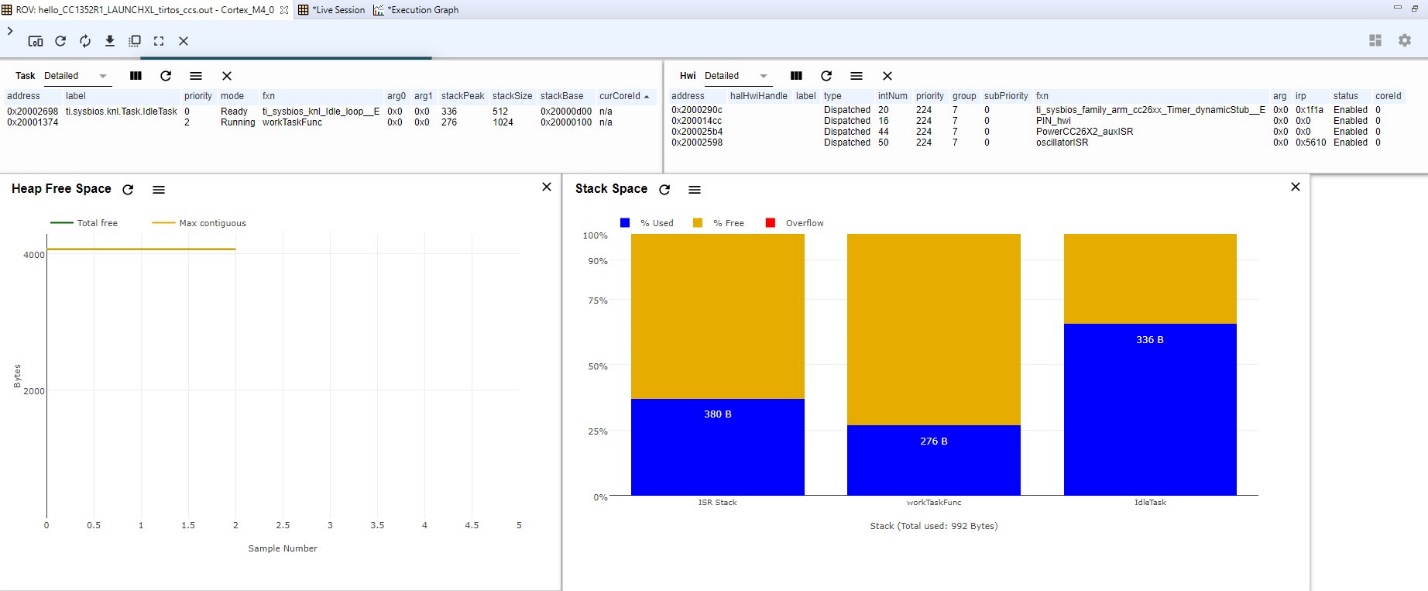
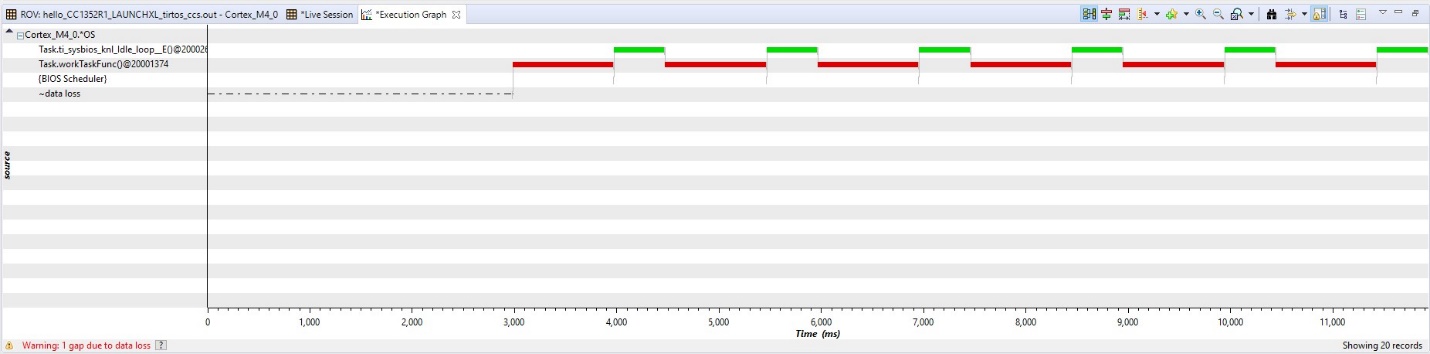
" .global myDelay\n"

"myDelay:\n"

" subs r0, #1\n"

" bne.n myDelay\n"

" bx lr\n");

**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

**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 ========

\*

\*/

**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"

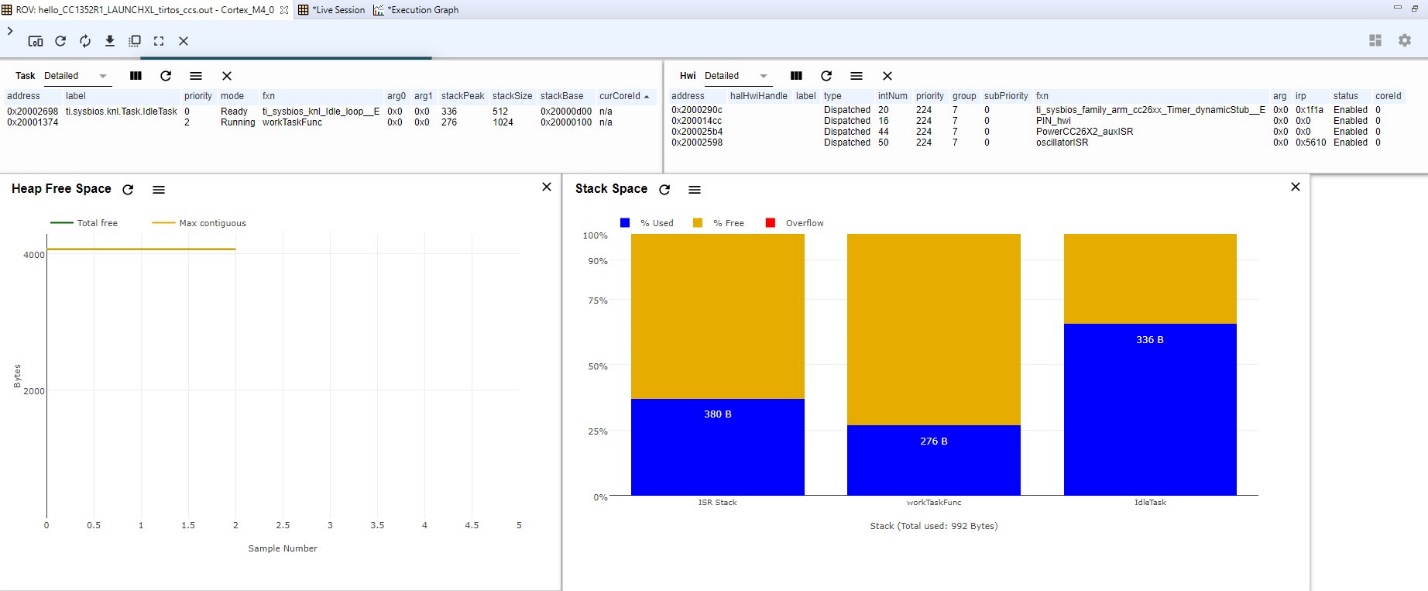
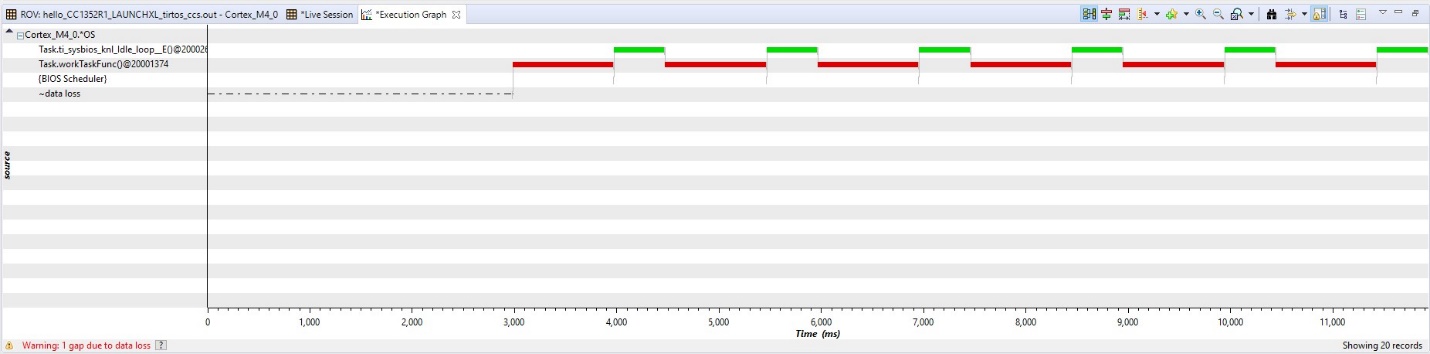
"myDelay:\n"

" subs r0, #1\n"

" bne.n myDelay\n"

" bx lr\n");

**Screenshots of ROV view and Execution Graph**



**------------------------------------------------------------------------------------**