**Date Submitted: 11.13.2018**

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

**Task 01: Changing hello world content**

**Youtube Link:**

**Modifications**

*/\* XDC Module Headers \*/*

*/\* TI-RTOS Header files \*/*

#include <xdc/std.h>

#include <ti/sysbios/BIOS.h>

#include <ti/sysbios/knl/Task.h>

#include <ti/drivers/GPIO.h>

#include <ti/sysbios/knl/Clock.h>

*/\* Example/Board Header files \*/*

#include "Board.h"

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)

#pragma DATA\_ALIGN(urgentWorkTaskStack, 8)

#define STACKSIZE 1024

**static** uint8\_t workTaskStack[STACKSIZE];

**static** uint8\_t urgentWorkTaskStack[STACKSIZE];

void doUrgentWork(void)

{

GPIO\_write(Board\_GPIO\_LED1, Board\_GPIO\_LED\_OFF);

FakeBlockingFastWork(); */\* Pretend to do something useful but time-consuming \*/*

GPIO\_write(Board\_GPIO\_LED1, Board\_GPIO\_LED\_ON);

}*//end doUrgentWork*

void doWork(void)

{

GPIO\_write(Board\_GPIO\_LED0, Board\_GPIO\_LED\_OFF);

FakeBlockingSlowWork(); */\* Pretend to do something useful but time-consuming \*/*

GPIO\_write(Board\_GPIO\_LED0, Board\_GPIO\_LED\_ON);

}*//edn doWork*

Void workTaskFunc(UArg arg0, UArg arg1)

{

**while** (1) {

*/\* Do work \*/*

doWork();

*/\* Wait a while, because doWork should be a periodic thing, not continuous.\*/*

*// myDelay(2400000);*

Task\_sleep(Clock\_tickPeriod/2); */\*1000µs/2=500ms\*/*

}

}*//end workTaskFunc*

*/\*\* ======== main ========*

*\**

*\*/*

int main(void)

{

Board\_initGeneral();

GPIO\_init();

*/\* Set up the led task \*/*

Task\_Params workTaskParams;

Task\_Params\_init(&workTaskParams);

workTaskParams.stackSize = STACKSIZE;

workTaskParams.priority = 1;

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**"

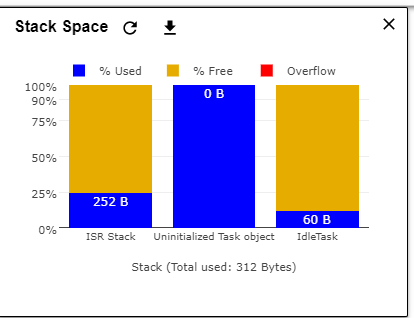
" bne.n myDelay**\n**"

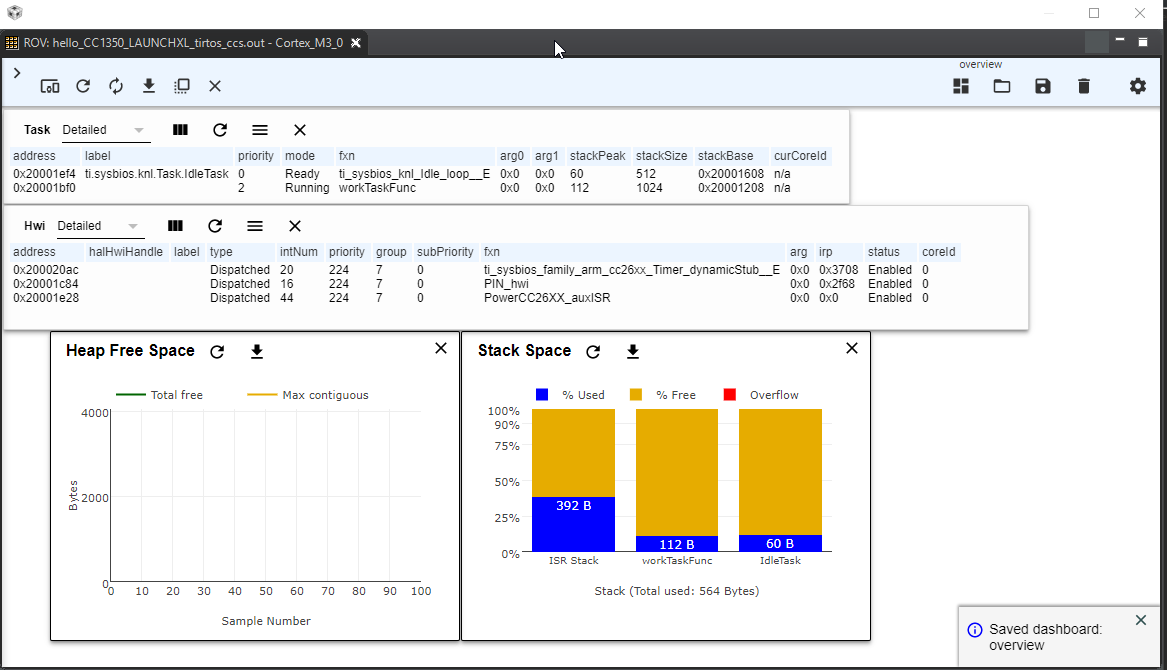
" bx lr**\n**");

**Task 02:** Debugging Tools, Runtime Object View

**Youtube Link: <https://youtu.be/gd6fs_X4gaM>**

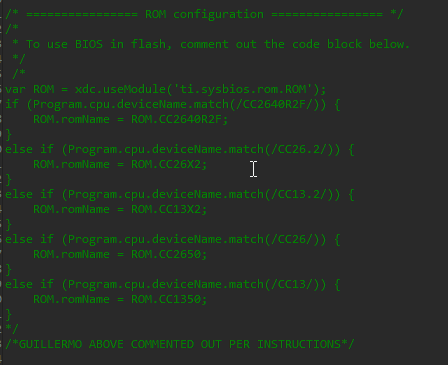
**View**

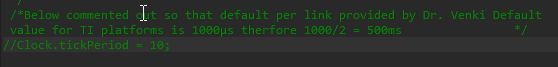
****

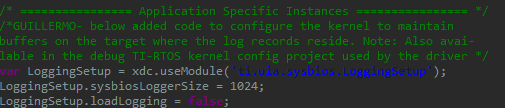


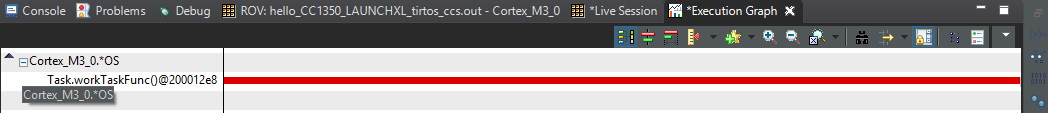
**Added/modification**

****



****

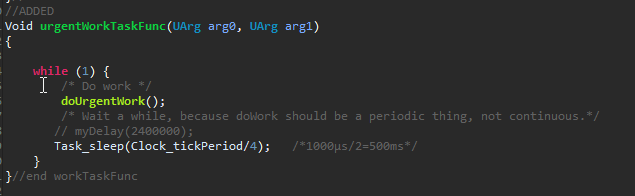




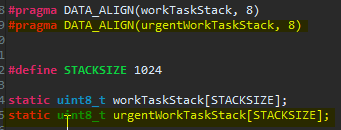
**Task 03: Sleeping Well**

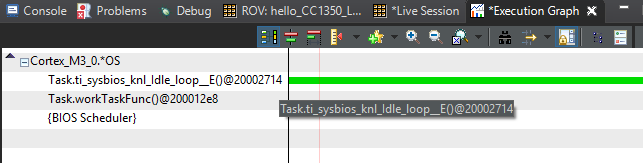
**Youtube Link:** [**https://youtu.be/D6H\_KYKxp9Y**](https://youtu.be/D6H_KYKxp9Y)

**Added/Modified**

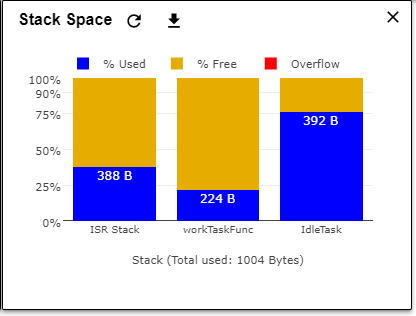
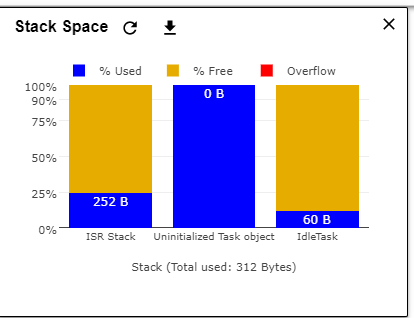
****

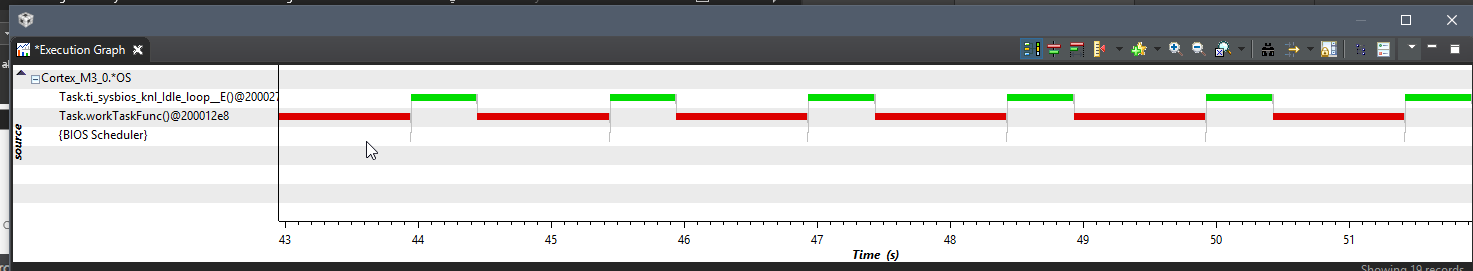
**Not sure if this is what the instructions meant**

****



**3.1**

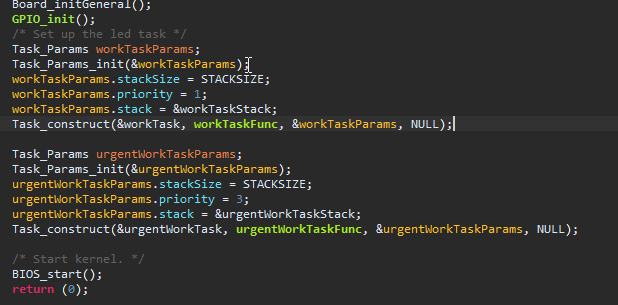
****

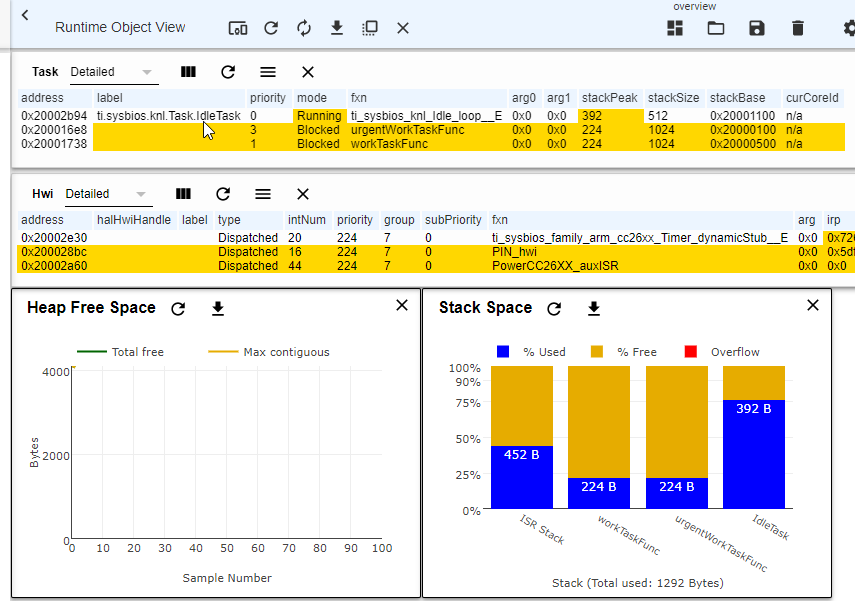
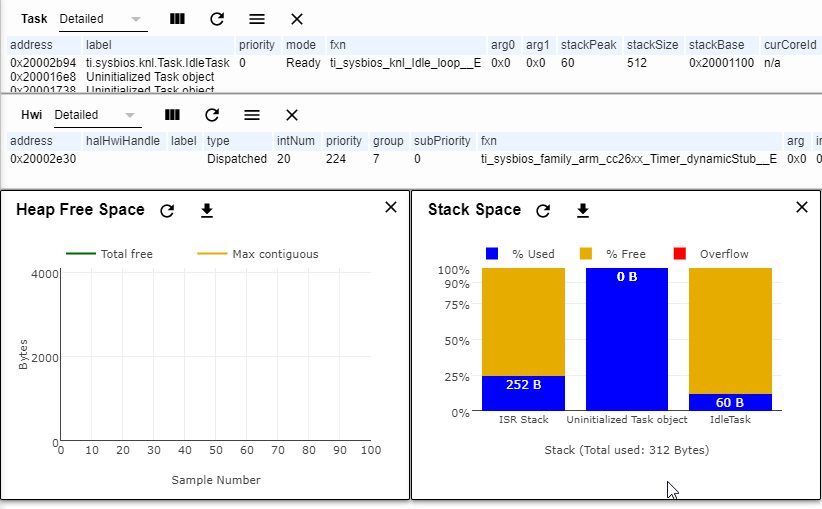
****

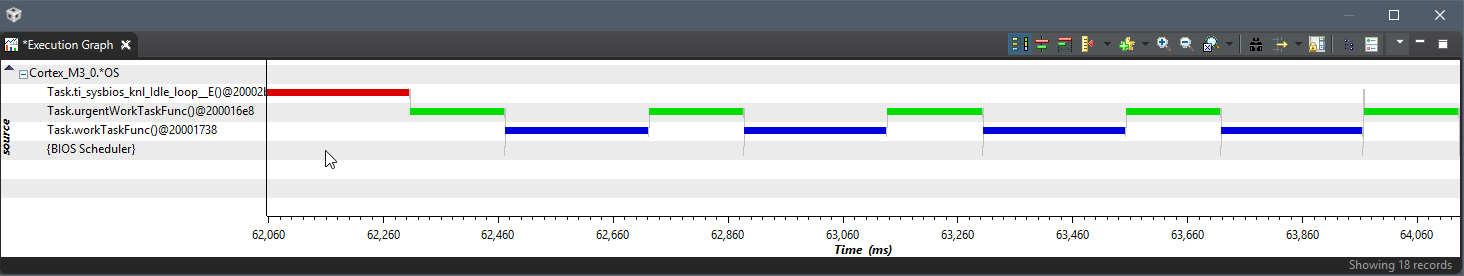
**Task 04: Executing urgent work**

**Youtube Link: N/A**

**Added/Modified**

****

****

****