CPE403 – Advanced Embedded Systems

Design Assignment #4

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Joe Uesato

Email: uesatj1@unlv.nevada.edu

Github Repository link (root): https://github.com/joeuesato/lab_submissions

Youtube Playlist: https://www.youtube.com/playlist?list=PLSBOvuRedzOf8JAhpVx0VsSteisJQUKv3

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.

```
/*
 * Copyright (c) 2015-2019, Texas Instruments Incorporated
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * *  Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * *  Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 *
 * *  Neither the name of Texas Instruments Incorporated nor the names of
 *    its contributors may be used to endorse or promote products derived
 *    from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
 * OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
 * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
 * OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
 * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

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

 #include <ti/drivers/GPIO.h>

 /* Driver configuration */
 #include "ti_drivers_config.h"

 void myDelay(int count);

 /* Could be anything, like computing primes */
 #define FakeBlockingSlowWork()   myDelay(1200000)
 #define FakeBlockingFastWork()   myDelay(200000)

 Task_Struct workTask;
```

```
52.  Task_Struct urgentWorkTask;
53.  /* Make sure we have nice 8-byte alignment on the stack to avoid wasting
   memory */
54.  #pragma DATA_ALIGN(workTaskStack, 8)
55.  #pragma DATA_ALIGN(urgentWorkTaskStack, 8)
56.
57.  #define STACKSIZE 1024
58.  static uint8_t workTaskStack[STACKSIZE];
59.  static uint8_t urgentWorkTaskStack[STACKSIZE];
60.
61.  void doUrgentWork(void)
62.  {
63.      GPIO_write(CONFIG_GPIO_LED_1, CONFIG_LED_OFF);
64.      FakeBlockingFastWork(); /* Pretend to do something useful but time-
   consuming */
65.      GPIO_write(CONFIG_GPIO_LED_1, CONFIG_LED_ON);
66.  }
67.
68.  void doWork(void)
69.  {
70.      GPIO_write(CONFIG_GPIO_LED_0, CONFIG_LED_OFF);
71.      FakeBlockingSlowWork(); /* Pretend to do something useful but time-
   consuming */
72.      GPIO_write(CONFIG_GPIO_LED_0, CONFIG_LED_ON);
73.  }
74.
75.  Void workTaskFunc(UArg arg0, UArg arg1)
76.  {
77.      while (1) {
78.
79.          /* Do work */
80.          doWork();
81.
82.          /* Wait a while, because doWork should be a periodic thing, not
   continuous.*/
83.          //myDelay(24000000);
84.          Task_sleep(500 * (1000 / Clock_tickPeriod));
85.      }
86.  }
87.
88.  Void urgentWorkTaskFunc(UArg arg0, UArg arg1)
89.  {
90.      while (1) {
91.
92.          /* Do work */
93.          doUrgentWork();
94.
95.          /* Wait a while, because doWork should be a periodic thing, not
   continuous.*/
96.          //myDelay(24000000);
97.          Task_sleep(50 * (1000 / Clock_tickPeriod));
98.      }
99.  }
100.
101.       /*
```

```
 *  ======== 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 = 1;
    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");
```

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

CC1352 with MKII plugged in

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

146. Declaration

Please note this is not done, see github for finished code and updated document

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

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