

Joseph Sharp Halpin  
CpE 403 Section 1001  
Date Submitted: 11/21/2018

Youtube Link Task 1: <https://youtu.be/DYjWn7Hyed8>

Youtube Link Task 3: <https://youtu.be/n-Me-LxXcR8>

Youtube Link Task 4: <https://youtu.be/MdbNl2ZlgCc>

## Task 2:

### Sensor Controller Studio:

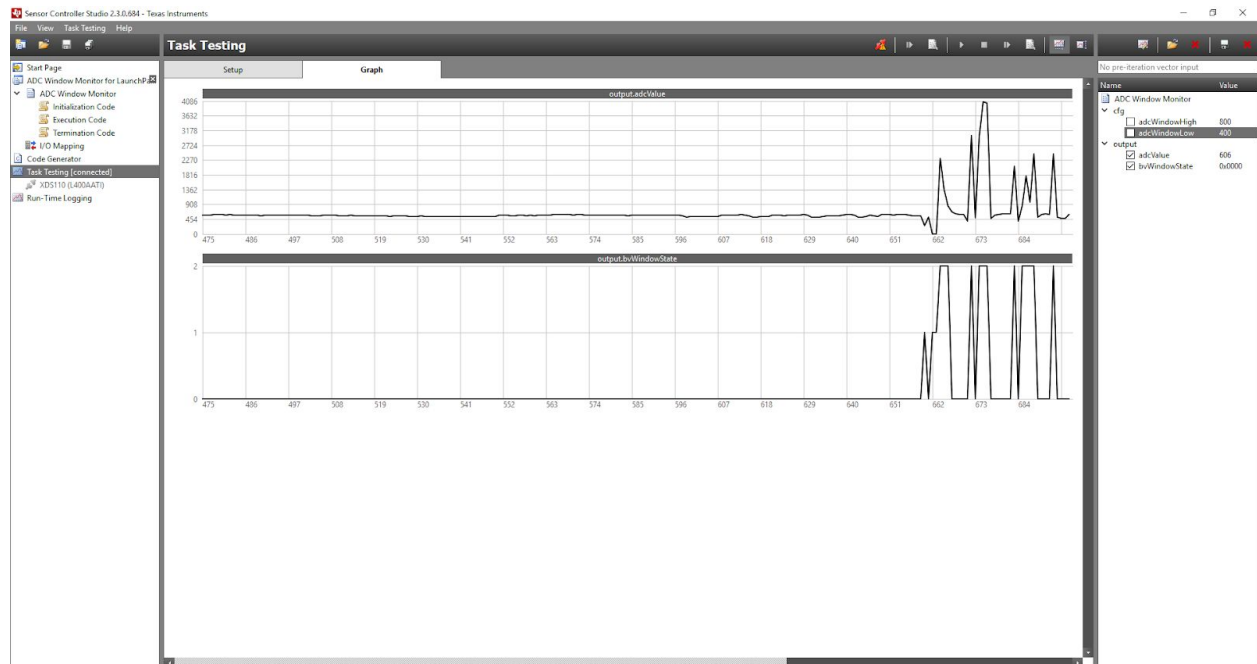
The screenshot shows the Sensor Controller Studio 2.3.0.684 - Texas Instruments interface. The 'I/O Mapping' tab is active, displaying the configuration for the CC1350 LaunchPad. The 'Task / I/O Function' column lists the 'adc level trigger' task with its functions: A: INPUT, O: green led, O: low, and O: high. The 'I/O Name and Board Function' column lists the corresponding hardware functions: DIO29 - A6 (Analog In), DIO7 - GREEN LED (PWM Out), DIO28 - A5 (Analog In), and DIO30 - A7 (Analog In).

Below the I/O Mapping table, the 'CC1350, Revision -, Package QFN48 7x7 RGG' table provides a detailed mapping of I/O pins to their AUX Mapping, Pin Number, Board Mapping, and Currently Mapped I/O Function(s).

I/O Name	AUX Mapping	Pin Number	Board Mapping	Currently Mapped I/O Function(s)
DIO1	14	6	RF SW	-
DIO2	13	7	UART RXD	-
DIO3	12	8	UART TXD	-
DIO4	11	9	I2C SCL	-
DIO5	10	10	I2C SDA	-
DIO6	9	11	RED LED (PWM Out)	-
DIO7	8	12	GREEN LED (PWM Out)	O: green led
DIO23	7	36	A0 (Analog In)	-
DIO24	6	37	A1 (Analog In)	-
DIO25	5	38	A2 (Analog In)	-
DIO26	4	39	A3 (Analog In)	-
DIO27	3	40	A4 (Analog In)	-
DIO28	2	41	A5 (Analog In)	O: low
DIO29	1	42	A6 (Analog In)	A: INPUT
DIO30	0	43	A7 (Analog In)	O: high

## Task 3:

### Sensor Controller Studio Graph:



## Sensor Controller Studio Assembly:

**Task Debugging - Execution Code**

```

; Assert the power-down request. We'll make sure that it has taken effect
; or been completely ignored when waking up again
ioset    #0, [!TOP_MAC_PuRDwnREQ]

/noPowerDown:
; Sleep until the next event
sleep

0069 ---- bdb7 sleepInstr:      sleep

006a ---- 241a AdIdDIacquire:    iobtst    #0, [!TOP_SMPH_SMPH0]
006b ---- a5fe                  biob0     AdIdDIacquire
006c ---- adb7                  rts

006d ---- 641a AdIdDIRelease:    ioset     #0, [!TOP_SMPH_SMPH0]
006e ---- adb7                  rts

; RTC ticks until next execution, one word for each task
pFwTaskExecuteScheduleTable:
dw #0

; Task code function pointers for each task code block, one word for each task
pFwTaskInitializeFuncTable:
dw #adcWindowMonitor/initialize
pFwTaskExecuteFuncTable:
dw #adcWindowMonitor/execute
pFwTaskTerminateFuncTable:
dw #adcWindowMonitor/terminate

; Run-time logging log requests and masks, one bit for each struct, one word for each task
pRtlTaskLogReqTable:
dw #0x0000
pRtlTaskLogTaskTable:
dw #0x0000

; Internal control data
fwCtrlInt:
dw #0 ; ID of the currently running task
dw #0x0000 ; LSB = Normal data exchange, MSB = Overflow or underflow
dw #0xffff ; Can generate an ALERT interrupt now? (0xffff = yes, 0x0000 = no)

CPU    Value    Flag    Value    Control    Value
R0     0x0008   Z FLAG   0        EXC VECTOR 0x00
R1     0x0000   V FLAG   0        WAKEUP     0
R2     0x0000   C FLAG   0        EVENTS     0x00
R3     0x0000   N FLAG   0        PC         0x0008
R4     0x0000   SLEEP   0        OP CODE    0xf502
R5     0x0000   WEV     0        LOOP START 0x0009
R6     0x0001   SELF STOP 1        LOOP END   0x000b
R7     0x0000   BUS ERROR 0        LOOP COUNT 0x01

```

The assembly code shows the execution of the task, including power-down assertions, sleep instructions, and task scheduling. The CPU register dump at the bottom shows the state of the processor, including flags (Z, V, C, N, S, WEV, SELF STOP, BUS ERROR) and control registers (EXC VECTOR, WAKEUP, EVENTS, PC, OP CODE, LOOP START, LOOP END, LOOP COUNT).

## Task 5:

### main\_tirtos.c:

```
1 /*
2  * Copyright (c) 2016-2018, Texas Instruments Incorporated
3  * All rights reserved.
4  *
5  * Redistribution and use in source and binary forms, with or without
6  * modification, are permitted provided that the following conditions
7  * are met:
8  *
9  * * Redistributions of source code must retain the above copyright
10 *   notice, this list of conditions and the following disclaimer.
11 *
12 * * Redistributions in binary form must reproduce the above copyright
13 *   notice, this list of conditions and the following disclaimer in the
14 *   documentation and/or other materials provided with the distribution.
15 *
16 * * Neither the name of Texas Instruments Incorporated nor the names of
17 *   its contributors may be used to endorse or promote products derived
18 *   from this software without specific prior written permission.
19 *
20 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
21 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
22 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
23 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
24 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
25 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
26 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
27 * OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
28 *
29 * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
30 * OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
31 * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
32 */
33
34 /*
35  * ===== main_tirtos.c =====
36  */
37 #include <stdint.h>
38
39 /* POSIX Header files */
40 #include <pthread.h>
41
42 /* RTOS header files */
43 #include <ti/sysbios/BIOS.h>
44
45 /* Example/Board Header files */
46 #include "Board.h"
47
48 extern void *tirtosScThread(void *arg0);
49
50 /* Stack size in bytes */
51 #define THREADSTACKSIZE    1024
```

```

52
53 /*
54 * ===== main =====
55 */
56 int main(void)
57 {
58     pthread_t      thread;
59     pthread_attr_t  attrs;
60     struct sched_param priParam;
61     int            retc;
62
63     /* Call driver init functions */
64     Board_initGeneral();
65
66     /* Initialize the attributes structure with default values */
67     pthread_attr_init(&attrs);
68
69     /* Set priority, detach state, and stack size attributes */
70     priParam.sched_priority = 1;
71     retc = pthread_attr_setschedparam(&attrs, &priParam);
72     retc |= pthread_attr_setdetachstate(&attrs, PTHREAD_CREATE_DETACHED);
73     retc |= pthread_attr_setstacksize(&attrs, THREADSTACKSIZE);
74     if (retc != 0) {
75         /* failed to set attributes */
76         while (1) {}
77     }
78
79     retc = pthread_create(&thread, &attrs, tirtosScThread, NULL);
80     if (retc != 0) {
81         /* pthread_create() failed */
82         while (1) {}
83     }
84
85     BIOS_start();
86
87     return (0);
88 }

```