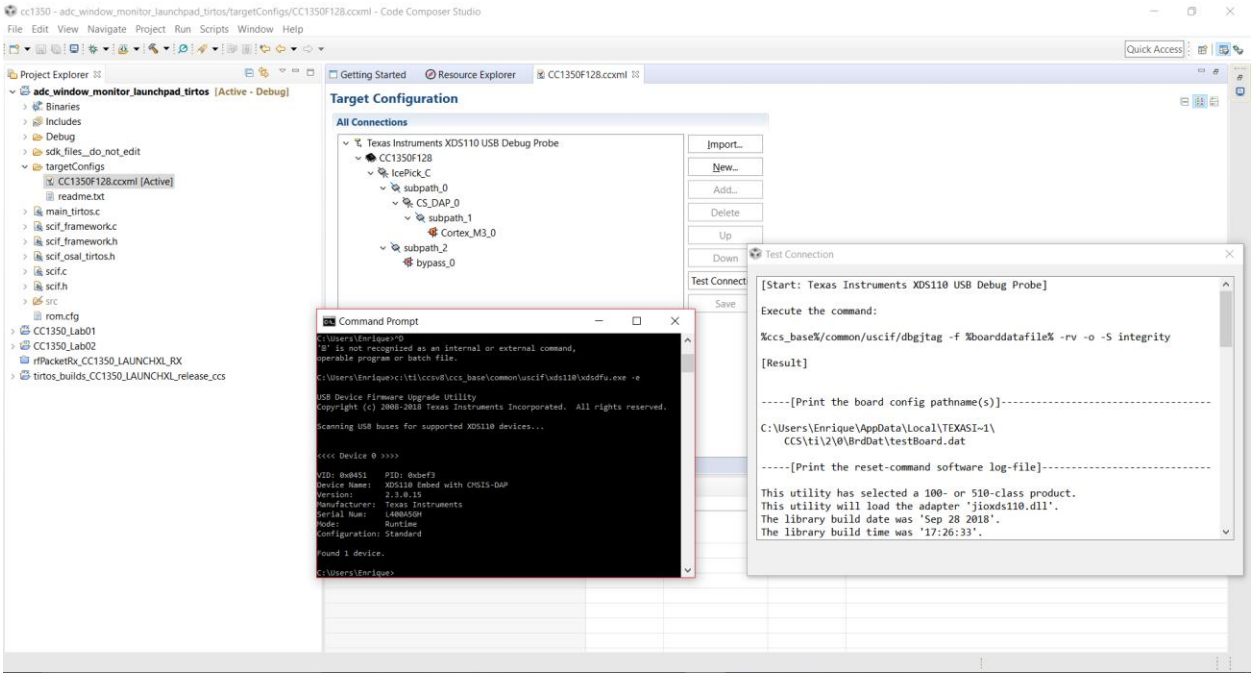
Enrique Saldana

CC1350 Lab04

https://github.com/enri10

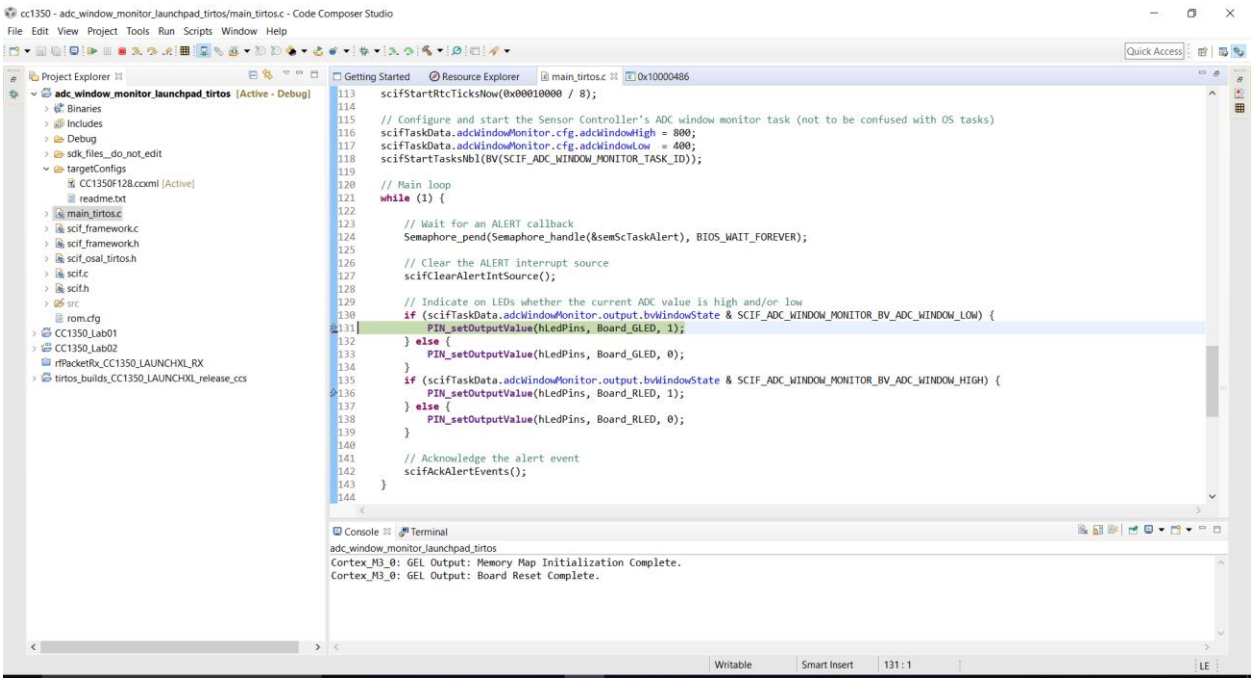
Lab04 Task1 : Set up Project in SCS

Youtube Link: ----------



Lab04 Task2: Download and Debug with CCS

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



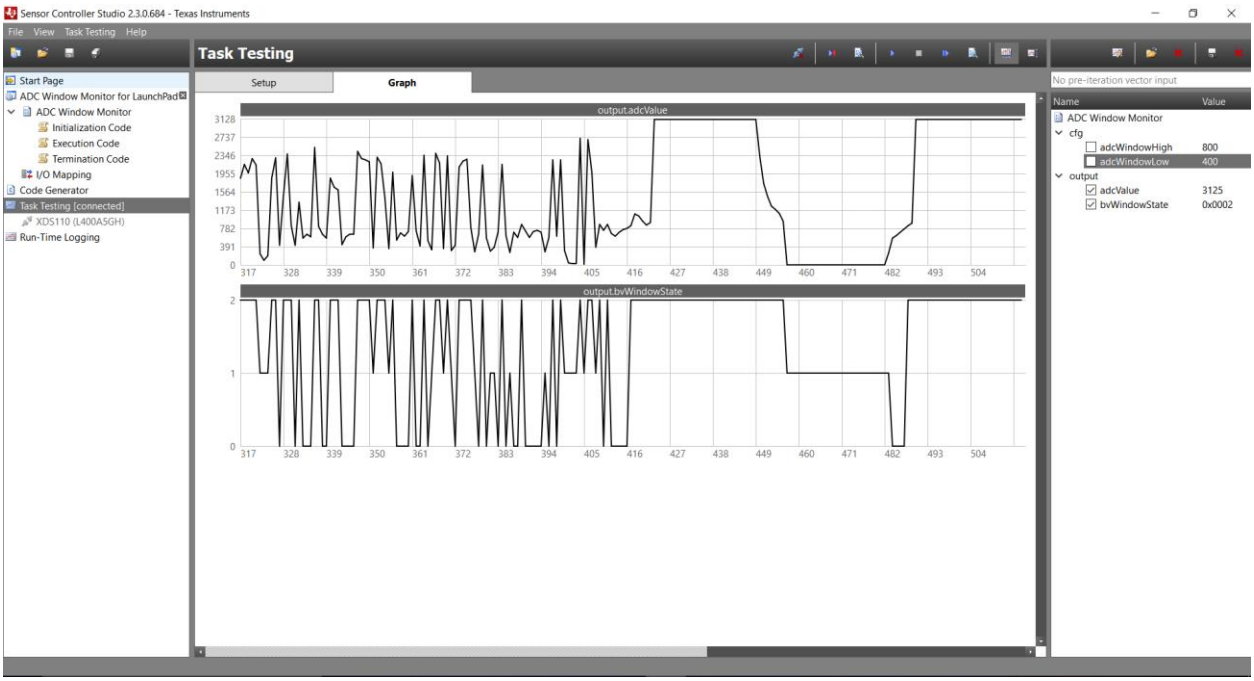

```
#include "ex_include_tirtos.h"
#include "scif.h"


#define BV(n)               (1 << (n))


// Display error message if the SCIF driver has been generated with incorrect
operating system setting
#if !(defined(SCIF_OSAL_TIRTOS_H) || defined(SCIF_OSAL_TIDPL_H))
    #error "SCIF driver has incorrect operating system configuration for this
example. Please change to 'TI-RTOS' or 'TI Driver Porting Layer' in the Sensor
Controller Studio project panel and re-generate the driver."
#endif

// Display error message if the SCIF driver has been generated with incorrect target
chip package
#ifndef SCIF_TARGET_CHIP_PACKAGE_QFN48_7X7_RGZ
    #error "SCIF driver has incorrect target chip package configuration for this
example. Please change to 'QFN48 7x7 RGZ' in the Sensor Controller Studio project
panel and re-generate the driver."
#endif


// Task data
Task_Struct myTask;
Char myTaskStack[1024];


// Semaphore used to wait for Sensor Controller task ALERT event
static Semaphore_Struct semScTaskAlert;
```

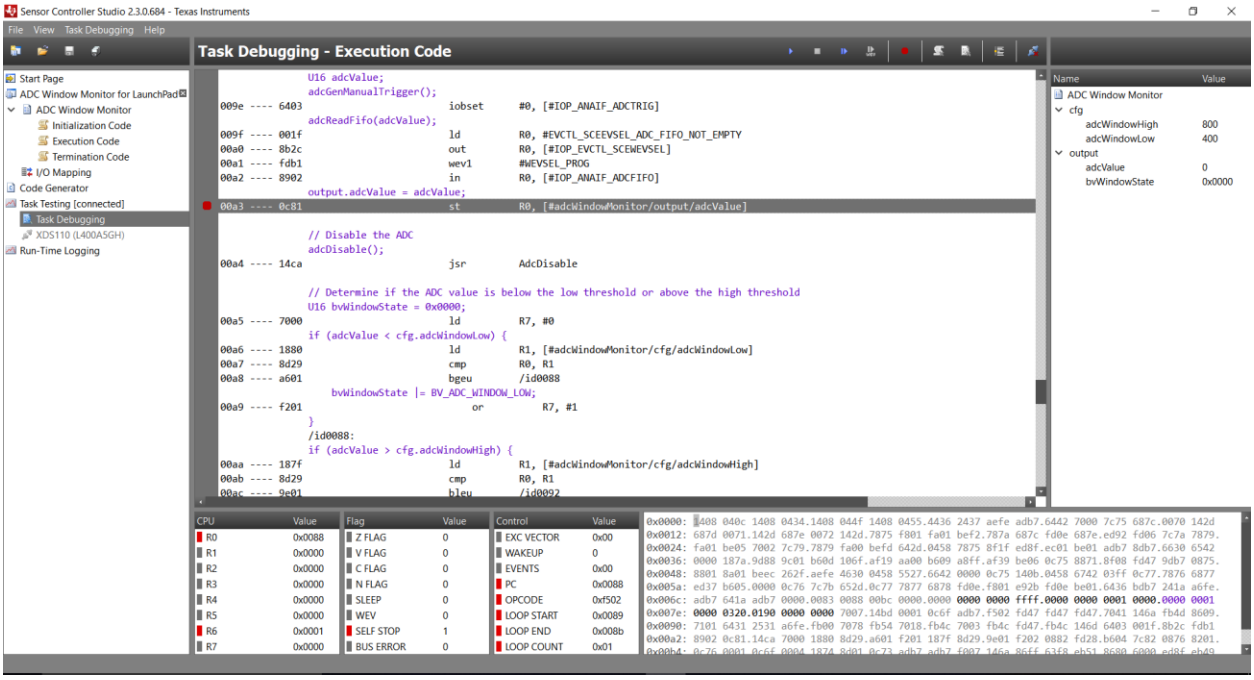
```
void scCtrlReadyCallback(void) {

} // scCtrlReadyCallback



void scTaskAlertCallback(void) {

    // Wake up the OS task
    Semaphore_post(Semaphore_handle(&semScTaskAlert));

} // scTaskAlertCallback



PIN_Config pLedPinTable[] = {
    Board_GLED | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL | PIN_DRVSTR_MAX,
    Board_RLED | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL | PIN_DRVSTR_MAX,
    PIN_TERMINATE
};
PIN_State ledPinState;


void taskFxn(UArg a0, UArg a1) {
    PIN_Handle hLedPins;

    // Enable LED pins
    hLedPins = PIN_open(&ledPinState, pLedPinTable);

    // Initialize the Sensor Controller
    scifOsalInit();
    scifOsalRegisterCtrlReadyCallback(scCtrlReadyCallback);
    scifOsalRegisterTaskAlertCallback(scTaskAlertCallback);
    scifInit(&scifDriverSetup);
    scifStartRtcTicksNow(0x00010000 / 8);

    // Configure and start the Sensor Controller's ADC window monitor task (not to be
confused with OS tasks)
    scifTaskData.adcWindowMonitor.cfg.adcWindowHigh = 800;
    scifTaskData.adcWindowMonitor.cfg.adcWindowLow  = 400;
    scifStartTasksNbl(BV(SCIF_ADC_WINDOW_MONITOR_TASK_ID));

    // Main loop
    while (1) {

        // Wait for an ALERT callback
        Semaphore_pend(Semaphore_handle(&semScTaskAlert), BIOS_WAIT_FOREVER);

        // Clear the ALERT interrupt source
        scifClearAlertIntSource();
```

```
        // Indicate on LEDs whether the current ADC value is high and/or low
        if (scifTaskData.adcWindowMonitor.output.bvWindowState &
SCIF_ADC_WINDOW_MONITOR_BV_ADC_WINDOW_LOW) {
            PIN_setOutputValue(hLedPins, Board_GLED, 1);
        } else {
            PIN_setOutputValue(hLedPins, Board_GLED, 0);
        }
        if (scifTaskData.adcWindowMonitor.output.bvWindowState &
SCIF_ADC_WINDOW_MONITOR_BV_ADC_WINDOW_HIGH) {
            PIN_setOutputValue(hLedPins, Board_RLED, 1);
        } else {
            PIN_setOutputValue(hLedPins, Board_RLED, 0);
        }

        // Acknowledge the alert event
        scifAckAlertEvents();
    }

} // taskFxn

int main(void) {
    Task_Params taskParams;

    // Initialize the board
    Board_initGeneral();
#ifdef Board_shutDownExtFlash
    Board_shutDownExtFlash();
#endif

    // Configure the OS task
    Task_Params_init(&taskParams);
    taskParams.stack = myTaskStack;
    taskParams.stackSize = sizeof(myTaskStack);
    taskParams.priority = 3;
    Task_construct(&myTask, taskFxn, &taskParams, NULL);

    // Create the semaphore used to wait for Sensor Controller ALERT events
    Semaphore_Params semParams;
    Semaphore_Params_init(&semParams);
    semParams.mode = Semaphore_Mode_BINARY;
    Semaphore_construct(&semScTaskAlert, 0, &semParams);

    // Start TI-RTOS
    BIOS_start();
    return 0;

} // main
```

Lab04 Task3: Download and Debug with SCS

Youtube Link: ----------


Sensor Controller Studio 2.3.0.684 - Texas Instruments
File View Task Testing Help
Task Testing
Start Page
ADC Window Monitor for LaunchPad
ADC Window Monitor
Initialization Code
Execution Code
Termination Code
I/O Mapping
Code Generator
Task Testing [connected]
XDS110 (L400A5GH)
Run-Time Logging
Setup
Graph
output.adcValue
output.bvWindowState
No pre-iteration vector input
Name
Value
cfg
adcWindowHigh 800
adcWindowLow 400
output
adcValue 3125
bvWindowState 0x0002


Sensor Controller Studio 2.3.0.684 - Texas Instruments
File View Task Debugging Help
Task Debugging - Execution Code
Start Page
ADC Window Monitor for LaunchPad
ADC Window Monitor
Initialization Code
Execution Code
Termination Code
I/O Mapping
Code Generator
Task Testing [connected]
Task Debugging
XDS110 (L400A5GH)
Run-Time Logging
U16 adcValue;
adcGenManualTrigger();
009e ---- 6403 iobset #0, [#IOP_ANAIF_ADCTRIG]
adcReadFifo(adcValue);
009f ---- 001f ld R0, #EVCTL_SCEEVSEL_ADC_FIFO_NOT_EMPTY
00a0 ---- 8b2c out R0, [#IOP_EVCTL_SCEWEVSEL]
00a1 ---- fdb1 wev1 #WEVSEL_PROG
00a2 ---- 8902 in R0, [#IOP_ANAIF_ADCFIFO]
output.adcValue = adcValue;
00a3 ---- 0c81 st R0, [#adcWindowMonitor/output/adcValue]
// Disable the ADC
adcDisable();
00a4 ---- 14ca jsr AdcDisable
// Determine if the ADC value is below the low threshold or above the high threshold
U16 bvWindowState = 0x0000;
00a5 ---- 7000 ld R7, #0
if (adcValue < cfg.adcWindowLow) {
00a6 ---- 1880 ld R1, [#adcWindowMonitor/cfg/adcWindowLow]
00a7 ---- 8d29 cmp R0, R1
00a8 ---- a601 bgeu /id0088
bvWindowState |= BV_ADC_WINDOW_LOW;
00a9 ---- f201 or R7, #1
}
/id0088:
if (adcValue > cfg.adcWindowHigh) {
00aa ---- 187f ld R1, [#adcWindowMonitor/cfg/adcWindowHigh]
00ab ---- 8d29 cmp R0, R1
00ac ---- 9e01 bleu /id0092
Name
Value
ADC Window Monitor
cfg
adcWindowHigh 800
adcWindowLow 400
output
adcValue 0
bvWindowState 0x0000
CPU
Value
R0 0x0088
R1 0x0000
R2 0x0000
R3 0x0000
R4 0x0000
R5 0x0000
R6 0x0001
R7 0x0000
Flag
Value
Z FLAG 0
V FLAG 0
C FLAG 0
N FLAG 0
SLEEP 0
WEV 0
SELF STOP 1
BUS ERROR 0
Control
Value
EXC VECTOR 0x00
WAKEUP 0
EVENTS 0x00
PC 0x0088
OPCODE 0xf502
LOOP START 0x0089
LOOP END 0x008b
LOOP COUNT 0x01
0x0000: 1408 040c 1408 0434.1408 044f 1408 0455.4436 2437 aefe adb7.6442 7000 7c75 687c.0070 142d
0x0012: 687d 0071.142d 687e 0072 142d.7875 f801 fa01 bef2.787a 687c fd0e 687e.ed92 fd06 7c7a 7879.
0x0024: fa01 be05 7002 7c79.7879 fa00 befd 642d.0458 7875 8f1f ed8f.ec01 be01 adb7 8db7.6630 6542
0x0036: 0000 187a.9d88 9c01 b60d 106f.af19 aa00 b609 a8ff.af39 be06 0c75 8871.8f08 fd47 9db7 0875.
0x0048: 8801 8a01 beec 262f.aefe 4630 0458 5527.6642 0000 0c75 140b.0458 6742 03ff 0c77.7876 6877
0x005a: ed37 b605.0000 0c76 7c7b 652d.0c77 7877 6878 fd0e.f801 e92b fd0e be01.6436 bdb7 241a a6fe.
0x006c: adb7 641a adb7 0000.0083 0088 00bc 0000.0000 0000 0000 ffff.0000 0000 0001 0000.0000 0001
0x007e: 0000 0320.0190 0000 0000 7007.14bd 0001 0c6f adb7.f502 fd47 fd47 fd47.7041 146a fb4d 8609.
0x0090: 7101 6431 2531 a6fe.fb00 7078 fb54 7018.fb4c 7003 fb4c fd47.fb4c 146d 6403 001f.8b2c fdb1
0x00a2: 8902 0c81.14ca 7000 1880 8d29.a601 f201 187f 8d29.9e01 f202 0882 fd28.b604 7c82 0876 8201.