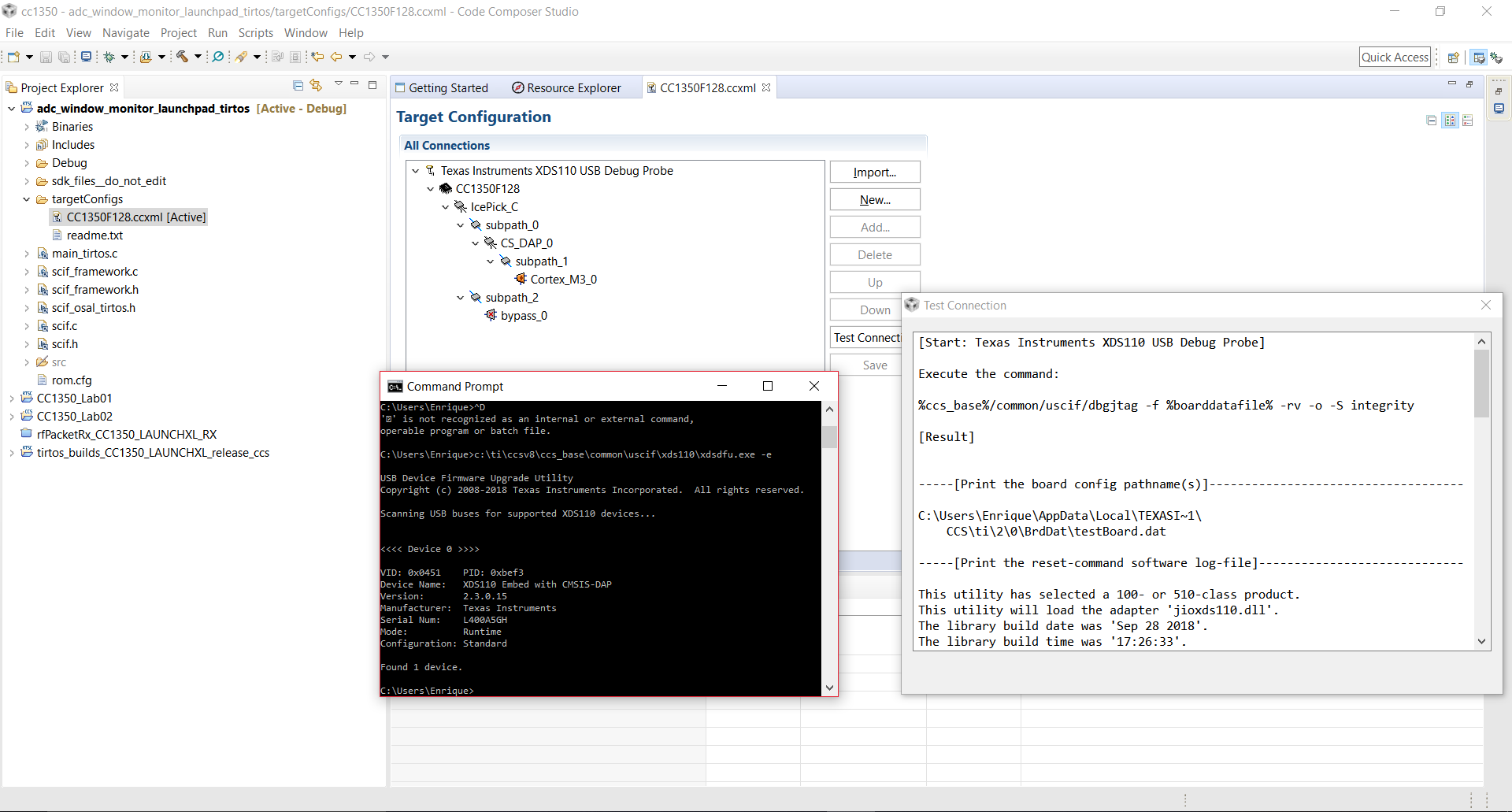
Enrique Saldana

CC1350 Lab04

<https://github.com/enri10>

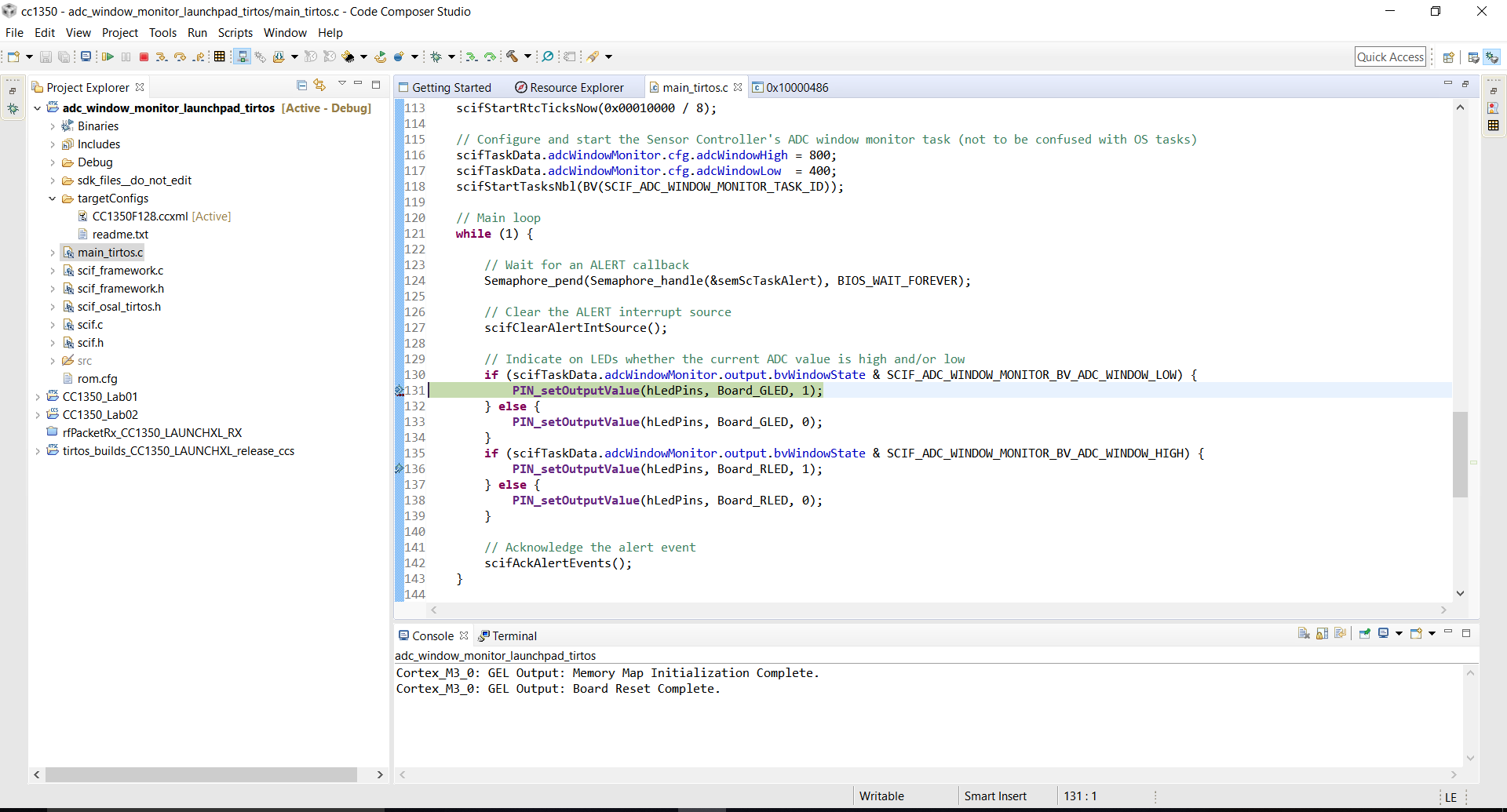
**Lab04 Task1 : Set up Project in SCS**

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



**Lab04 Task2: Download and Debug with CCS**

**Youtube Link:** [**https://youtu.be/SNRNJAyuyEQ**](%20https:/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: --------------------------------------**

