**Date Submitted: November 12, 2019**

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

**Task 01 (ADC Gui):**

Youtube Link: https://youtu.be/0uyVUwZCRqo

**Modified Code:**

/\* For usleep() \*/

**#include** <unistd.h>

**#include** <stdint.h>

**#include** <stddef.h>

/\* Driver Header files \*/

**#include** <ti/drivers/GPIO.h>

**#include** <ti/drivers/ADC.h>

// #include <ti/drivers/I2C.h>

// #include <ti/drivers/SPI.h>

// #include <ti/drivers/UART.h>

// #include <ti/drivers/Watchdog.h>

/\* Driver configuration \*/

**#include** "ti\_drivers\_config.h"

/\* global variableS FOR GUI COMPOSER \*/

uint16\_t adcValue = 0;

uint16\_t threshold = 100;

uint16\_t alert = 0;

/\*

\* ======== mainThread ========

\*/

**void** \***mainThread**(**void** \*arg0)

{

/\* ~10 loops/second \*/

uint32\_t time = 100000; // update ~10/second

/\* Call driver init functions \*/

**GPIO\_init**();

**ADC\_init**();

// I2C\_init();

// SPI\_init();

// UART\_init();

// Watchdog\_init();

/\* Open ADC Driver \*/

ADC\_Handle adc;

ADC\_Params params;

**ADC\_Params\_init**(&params);

adc = **ADC\_open**(CONFIG\_ADC\_0, &params);

**if** (adc == NULL) {

// Error initializing ADC channel 0

**while** (1);

}

**while** (1) {

int\_fast16\_t res;

res = **ADC\_convert**(adc, &adcValue);

**if** (res == ADC\_STATUS\_SUCCESS) {

**if**(adcValue >= threshold) {

**GPIO\_write**(CONFIG\_GPIO\_LED\_0, CONFIG\_GPIO\_LED\_ON);

alert = 1;

} **else**{

**GPIO\_write**(CONFIG\_GPIO\_LED\_0, CONFIG\_GPIO\_LED\_OFF);

alert = 0;

}

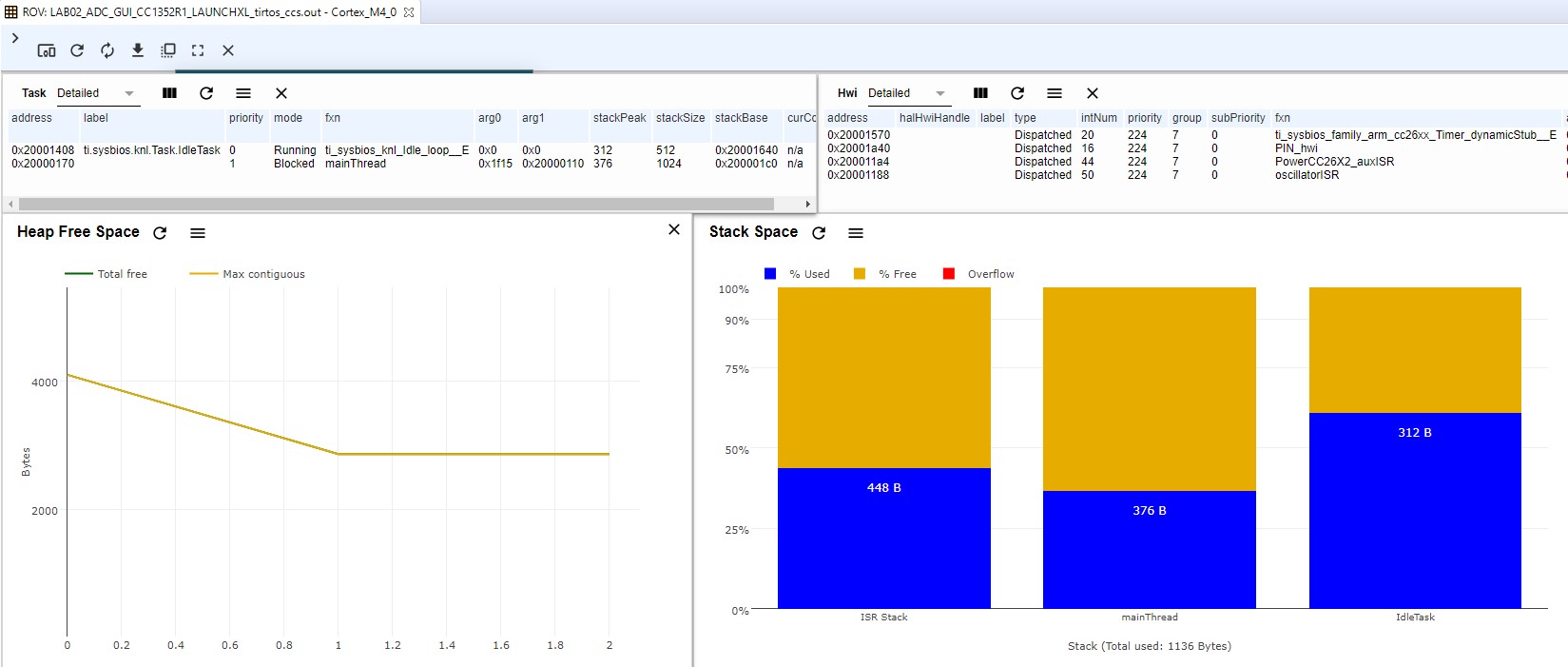
}

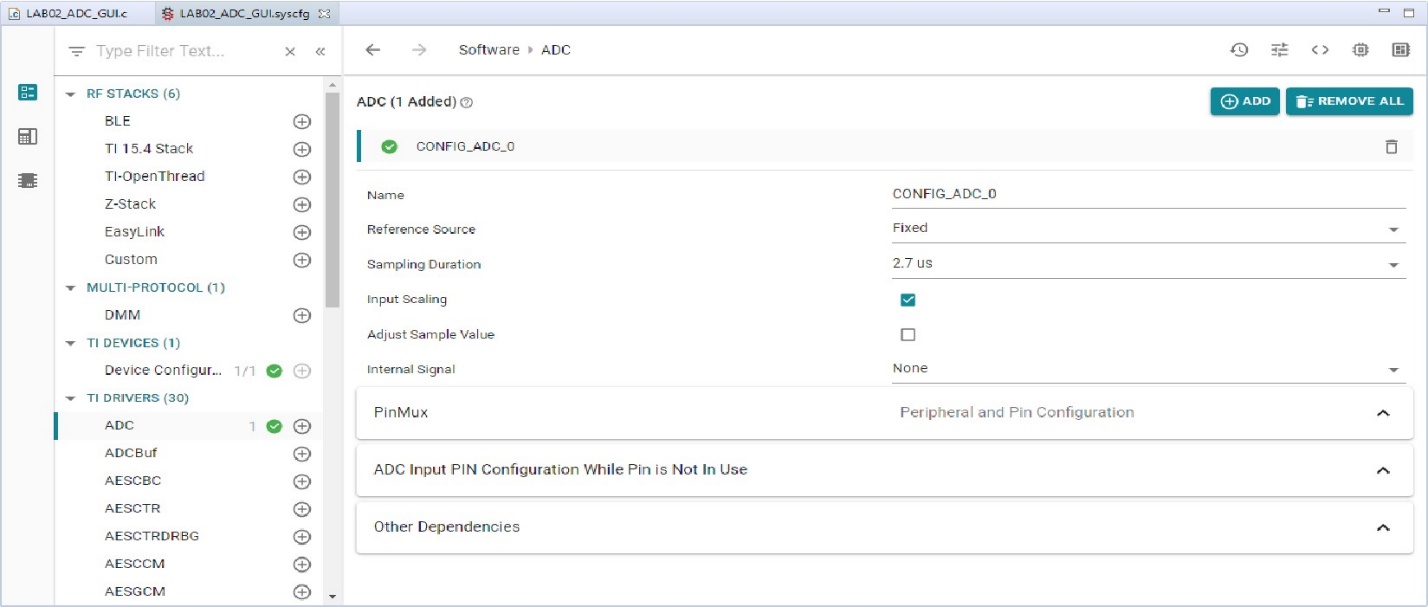
**usleep**(time);

}

}

**Screenshots of ROV view and SysCfg Modifications**



****

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

**Task 02 (RTOS Example):**

Youtube Link: https://youtu.be/dQ99x-8o9Iw

**Modified Code:**

/\*

\* ======== empty.c ========

\*/

/\* For usleep() \*/

**#include** <unistd.h>

**#include** <stdint.h>

**#include** <stddef.h>

/\* Driver Header files \*/

**#include** <ti/drivers/GPIO.h>

**#include** <ti/drivers/ADC.h>

**#include** <ti/display/Display.h>

// #include <ti/drivers/I2C.h>

// #include <ti/drivers/SPI.h>

// #include <ti/drivers/UART.h>

// #include <ti/drivers/Watchdog.h>

/\* Driver configuration \*/

**#include** "ti\_drivers\_config.h"

/\* global variableS FOR GUI COMPOSER \*/

uint16\_t adcValue = 0;

uint16\_t threshold = 100;

uint16\_t trigger = 0;

/\*

\* ======== gpioButtonFxn0 ========

\* Callback function for the GPIO interrupt on Board\_GPIO\_BUTTON0.

\*/

**void** **gpioButtonFxn0**(uint\_least8\_t index)

{

/\* Clear the GPIO interrupt and decrement threshold \*/

**if**(threshold < 250){ // Ensure threshold doesn't go below zero

threshold = 0;

}

**else** {

threshold -= 250; // decrement by 250

}

}

**void** **gpioButtonFxn1**(uint\_least8\_t index)

{

/\* Clear the GPIO interrupt and increment threshold \*/

**if**(threshold > 4095){ // Ensure threshold doesn't go above max ADC range

threshold = 4095;

}

**else** {

threshold += 250; // increment by 250

}

}

/\*

\* ======== gpioButtonFxn1 ========

\* Callback function for the GPIO interrupt on Board\_GPIO\_BUTTON1.

\* This may not be used for all boards.

\*/

/\*

\* ======== mainThread ========

\*/

**void** \***mainThread**(**void** \*arg0)

{

/\* ~10 loops/second \*/

uint32\_t time = 100000; // update ~10/second

/\* Call driver init functions \*/

**GPIO\_init**();

**ADC\_init**();

// I2C\_init();

// SDSPI\_init();

// SPI\_init();

// UART\_init();

// Watchdog\_init();

/\* Open Display Driver \*/

Display\_Handle displayHandle;

Display\_Params displayParams;

Display\_Params\_init(&displayParams);

displayHandle = Display\_open(Display\_Type\_UART, NULL);

/\* Open ADC Driver \*/

ADC\_Handle adc;

ADC\_Params params;

**ADC\_Params\_init**(&params);

adc = **ADC\_open**(CONFIG\_ADC\_0, &params);

**if** (adc == NULL) {

// Error initializing ADC channel 0

**while** (1);

}

/\* install Button callback \*/

**GPIO\_setCallback**(CONFIG\_GPIO\_BTN1, gpioButtonFxn0);

**GPIO\_setCallback**(CONFIG\_GPIO\_BTN2, gpioButtonFxn1);

/\* Enable interrupts \*/

**GPIO\_enableInt**(CONFIG\_GPIO\_BTN1);

**GPIO\_enableInt**(CONFIG\_GPIO\_BTN2);

**while** (1) {

int\_fast16\_t res;

res = **ADC\_convert**(adc, &adcValue);

**if** (res == ADC\_STATUS\_SUCCESS) {

Display\_printf(displayHandle, 1, 0, "ADC Reading %d", adcValue);

**if**(adcValue >= threshold){

**GPIO\_write**(CONFIG\_GPIO\_LED\_0, CONFIG\_GPIO\_LED\_ON);

trigger = 1;

} **else**{

**GPIO\_write**(CONFIG\_GPIO\_LED\_0, CONFIG\_GPIO\_LED\_OFF);

trigger = 0;

}

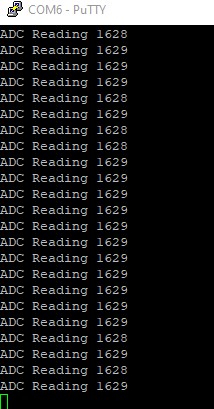
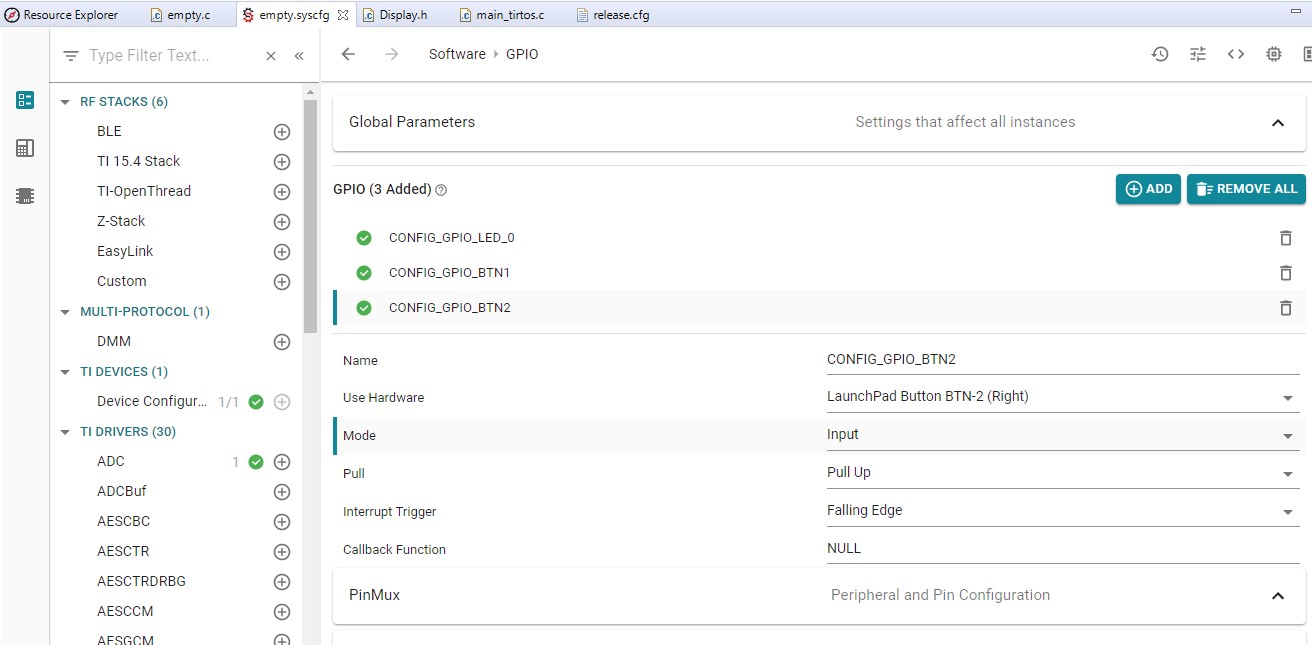
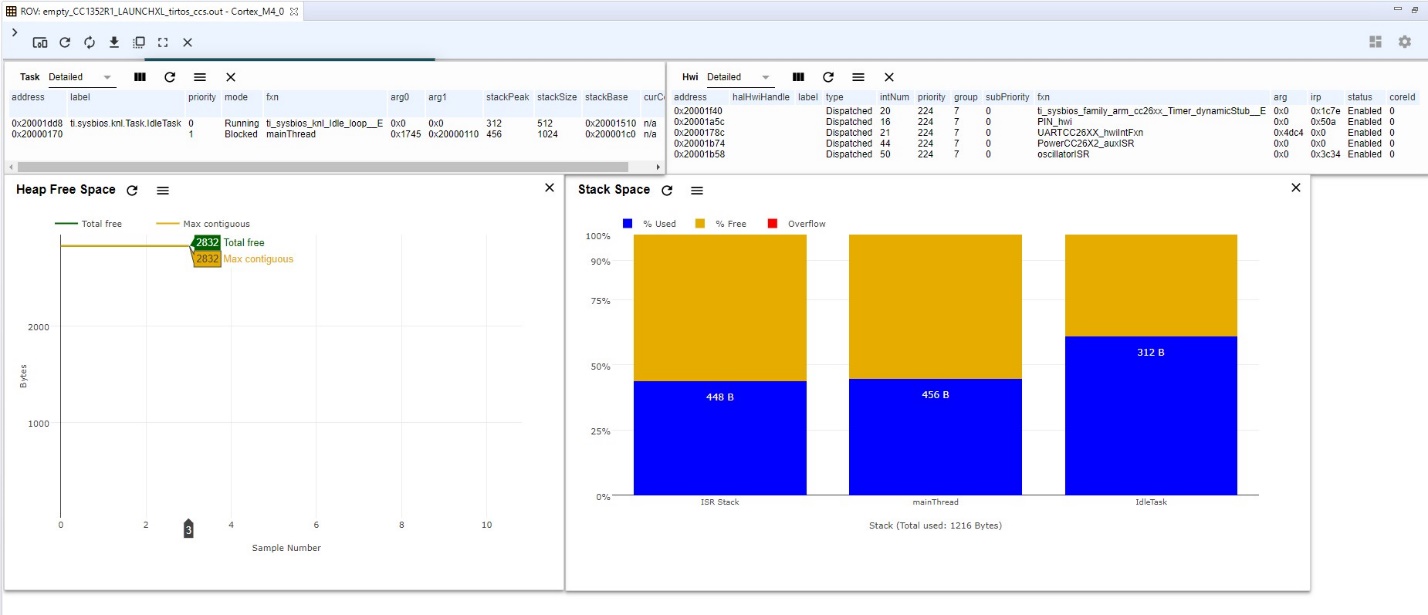
}

**usleep**(time);

}

}

**Screenshots of ROV view and SysCfg Modifications**

****

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