CPE403 – Advanced Embedded Systems

Design Assignment 4

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Meral Abu-Jaser

Email: abujaser@unlv.nevada.edu

Github Repository link (root): https://github.com/MeralAbuJaser/Advanced-Embedded-Systems

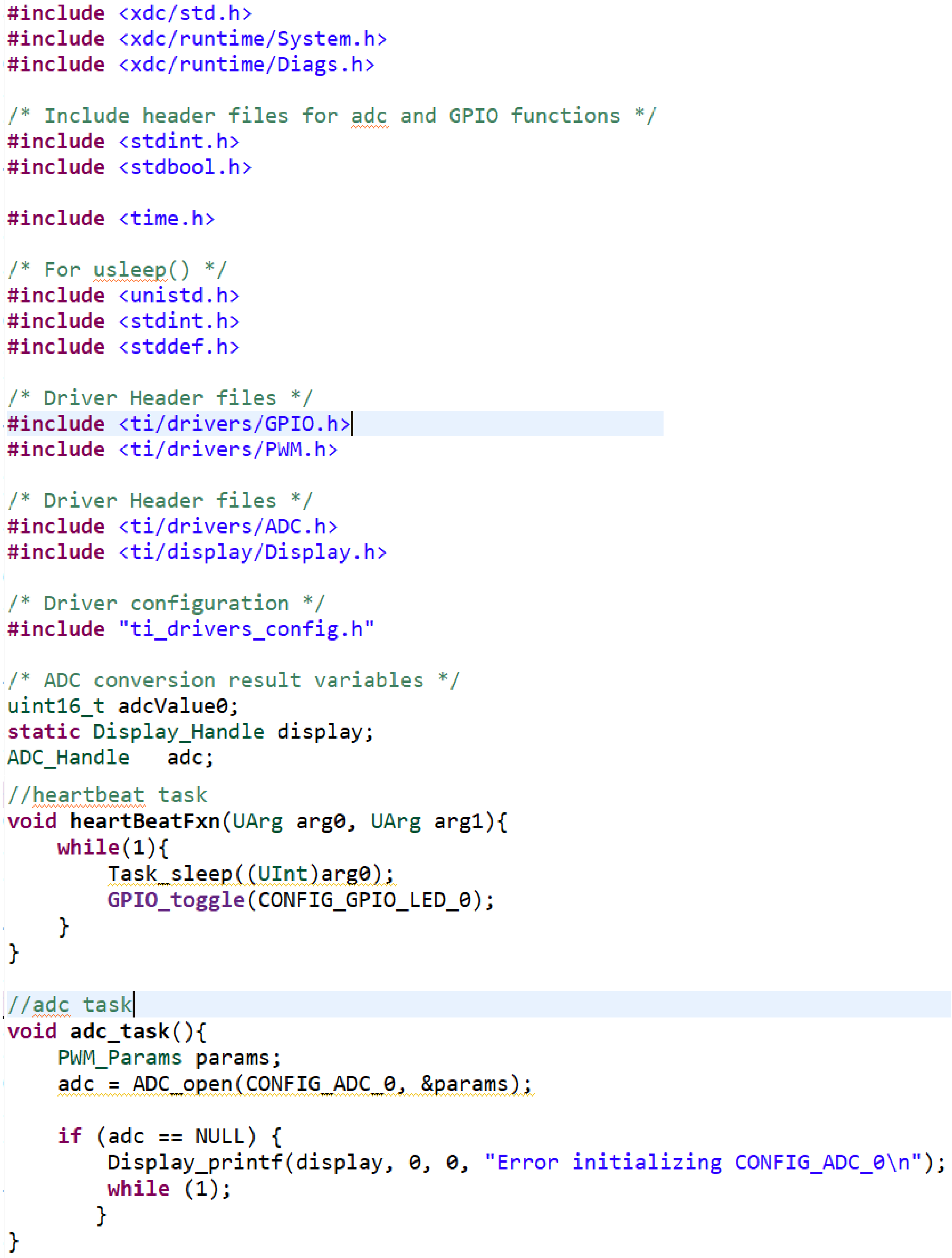
Youtube Playlist link (root): https://www.youtube.com/playlist?list=PLmRQUGgBgm2dlt37RCWlrSrGj7KxvDKmj

Follow the submission guideline to be awarded points for this Assignment.

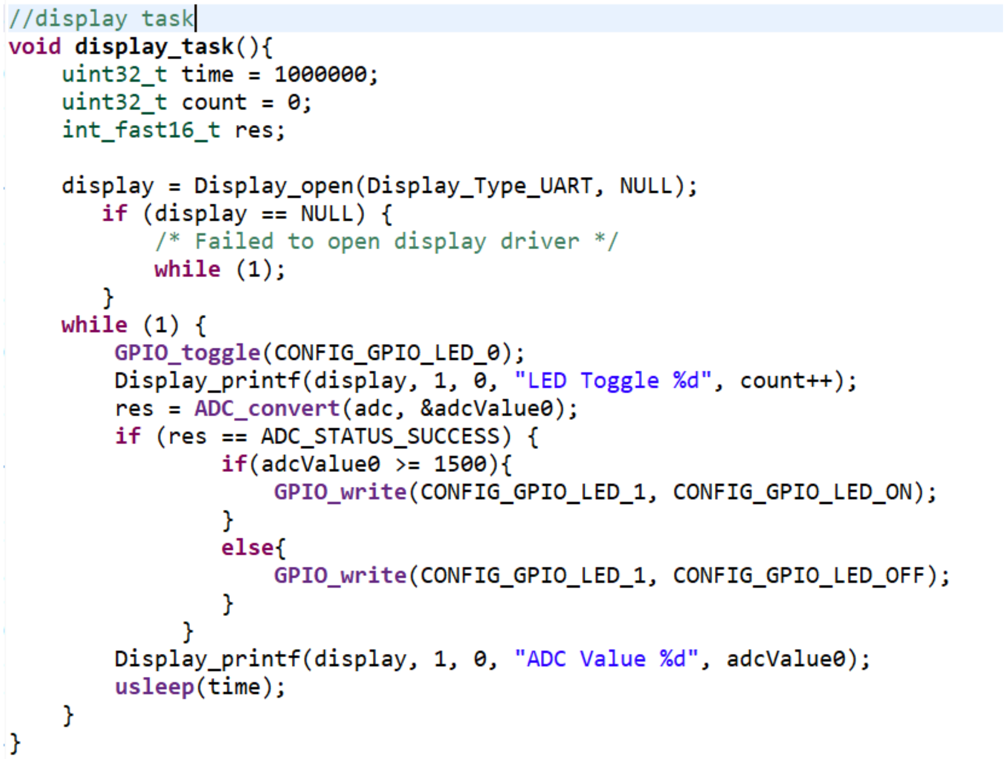
Submit the following for all Assignments:

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.

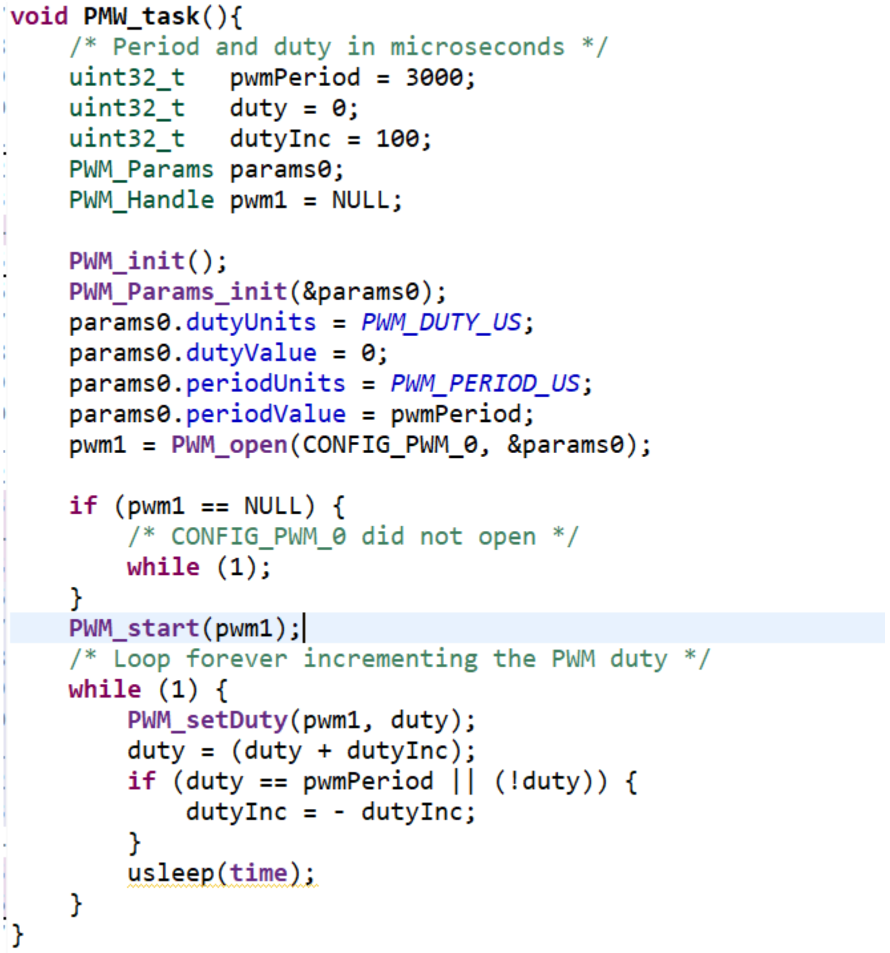
1. Task 01



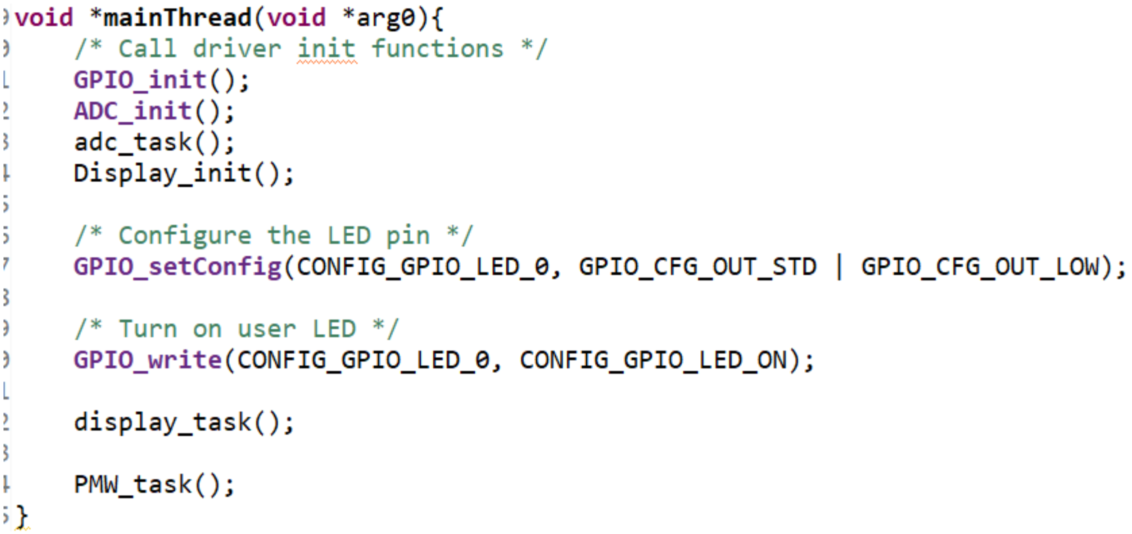
```
#include <xdc/std.h>
#include <xdc/runtime/System.h>
#include <xdc/runtime/Diags.h>

/* Include header files for adc and GPIO functions */
#include <stdint.h>
#include <stdbool.h>

#include <time.h>

/* For usleep() */
#include <unistd.h>
#include <stdint.h>
#include <stddef.h>

/* Driver Header files */
#include <ti/drivers/GPIO.h>
#include <ti/drivers/PWM.h>

/* Driver Header files */
#include <ti/drivers/ADC.h>
#include <ti/display/Display.h>

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

/* ADC conversion result variables */
uint16_t adcValue0;
static Display_Handle display;
ADC_Handle   adc;
//heartbeat task
void heartBeatFxn(UArg arg0, UArg arg1){
    while(1){
        Task_sleep((UInt)arg0);
        GPIO_toggle(CONFIG_GPIO_LED_0);
    }
}

//adc task
void adc_task(){
    PWM_Params params;
    adc = ADC_open(CONFIG_ADC_0, &params);

    if (adc == NULL) {
        Display_printf(display, 0, 0, "Error initializing CONFIG_ADC_0\n");
        while (1);
       }
}
```



```
//display task
void display_task(){
    uint32_t time = 1000000;
    uint32_t count = 0;
    int_fast16_t res;

    display = Display_open(Display_Type_UART, NULL);
       if (display == NULL) {
           /* Failed to open display driver */
           while (1);
       }
    while (1) {
        GPIO_toggle(CONFIG_GPIO_LED_0);
        Display_printf(display, 1, 0, "LED Toggle %d", count++);
        res = ADC_convert(adc, &adcValue0);
        if (res == ADC_STATUS_SUCCESS) {
                if(adcValue0 >= 1500){
                    GPIO_write(CONFIG_GPIO_LED_1, CONFIG_GPIO_LED_ON);
                }
                else{
                    GPIO_write(CONFIG_GPIO_LED_1, CONFIG_GPIO_LED_OFF);
                }
             }
        Display_printf(display, 1, 0, "ADC Value %d", adcValue0);
        usleep(time);
    }
}
```

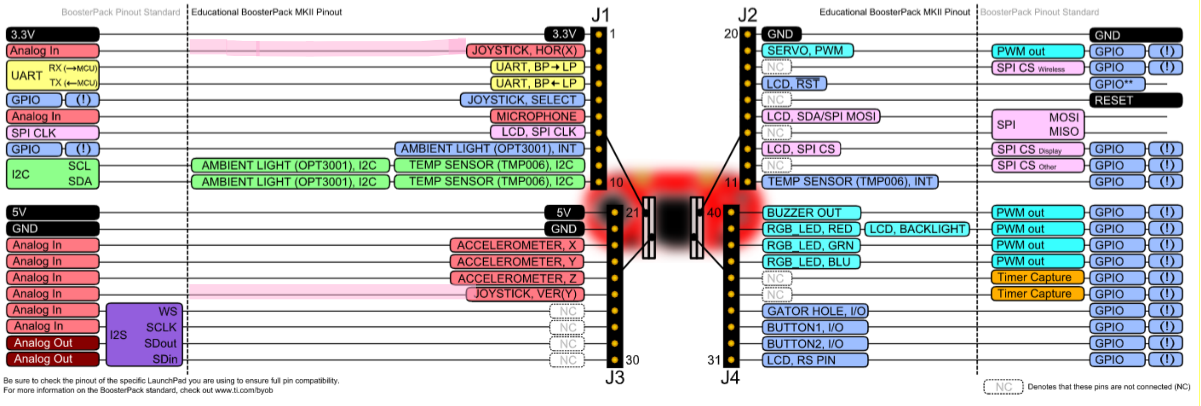


```
void PMW_task(){
    /* Period and duty in microseconds */
    uint32_t   pwmPeriod = 3000;
    uint32_t   duty = 0;
    uint32_t   dutyInc = 100;
    PWM_Params params0;
    PWM_Handle pwm1 = NULL;

    PWM_init();
    PWM_Params_init(&params0);
    params0.dutyUnits = PWM_DUTY_US;
    params0.dutyValue = 0;
    params0.periodUnits = PWM_PERIOD_US;
    params0.periodValue = pwmPeriod;
    pwm1 = PWM_open(CONFIG_PWM_0, &params0);

    if (pwm1 == NULL) {
        /* CONFIG_PWM_0 did not open */
        while (1);
    }
    PWM_start(pwm1);
    /* Loop forever incrementing the PWM duty */
    while (1) {
        PWM_setDuty(pwm1, duty);
        duty = (duty + dutyInc);
        if (duty == pwmPeriod || (!duty)) {
            dutyInc = - dutyInc;
        }
        usleep(time);
    }
}
```

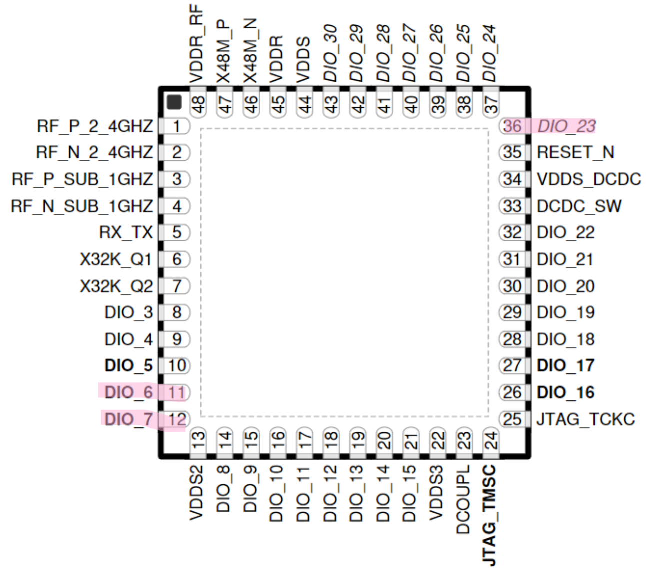


```
void *mainThread(void *arg0){
    /* Call driver init functions */
    GPIO_init();
    ADC_init();
    adc_task();
    Display_init();

    /* Configure the LED pin */
    GPIO_setConfig(CONFIG_GPIO_LED_0, GPIO_CFG_OUT_STD | GPIO_CFG_OUT_LOW);

    /* Turn on user LED */
    GPIO_write(CONFIG_GPIO_LED_0, CONFIG_GPIO_LED_ON);

    display_task();

    PMW_task();
}
```

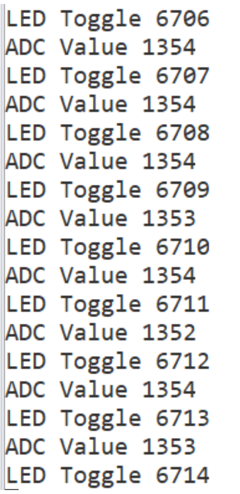
2. Block diagram and/or Schematics showing the components, pins used, and interface.



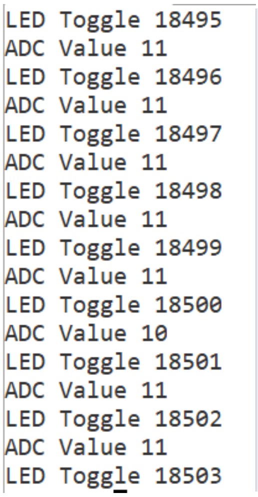
Be sure to check the pinout of the specific LaunchPad you are using to ensure full pin compatibility.
For more information on the BoosterPack standard, check out www.ti.com/byob



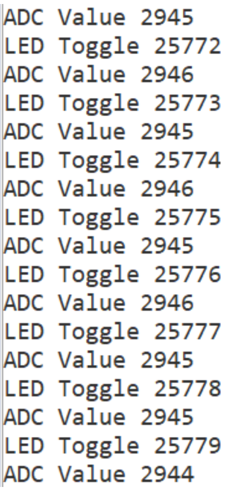
3. Screenshots of the IDE, physical setup, debugging process - Provide screenshot of successful compilation, screenshots of registers, variables, graphs, etc.



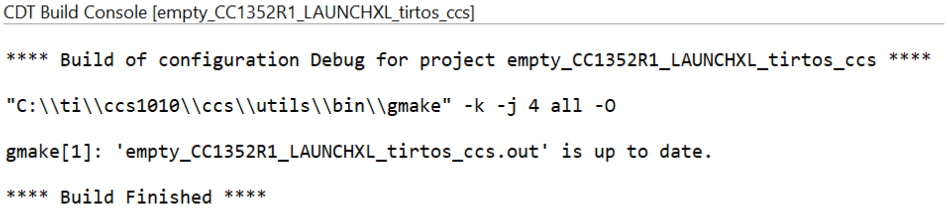
```
LED Toggle 6706
ADC Value 1354
LED Toggle 6707
ADC Value 1354
LED Toggle 6708
ADC Value 1354
LED Toggle 6709
ADC Value 1353
LED Toggle 6710
ADC Value 1354
LED Toggle 6711
ADC Value 1352
LED Toggle 6712
ADC Value 1354
LED Toggle 6713
ADC Value 1353
LED Toggle 6714
```



```
LED Toggle 18495
ADC Value 11
LED Toggle 18496
ADC Value 11
LED Toggle 18497
ADC Value 11
LED Toggle 18498
ADC Value 11
LED Toggle 18499
ADC Value 11
LED Toggle 18500
ADC Value 10
LED Toggle 18501
ADC Value 11
LED Toggle 18502
ADC Value 11
LED Toggle 18503
```

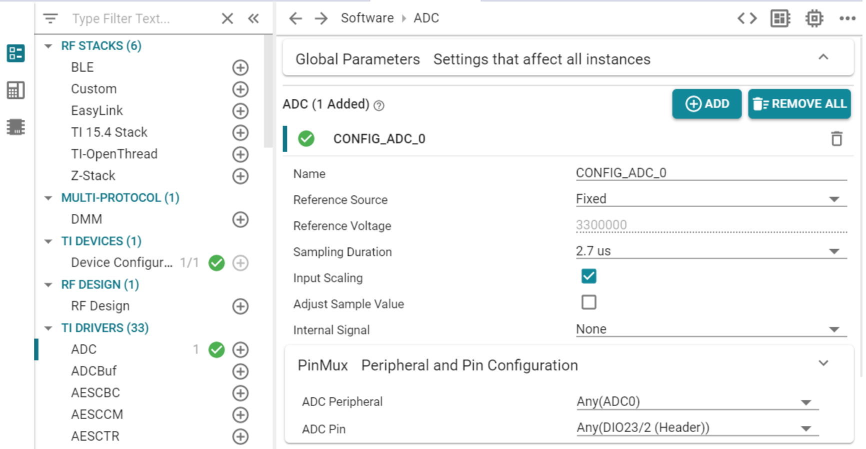


```
ADC Value 2945
LED Toggle 25772
ADC Value 2946
LED Toggle 25773
ADC Value 2945
LED Toggle 25774
ADC Value 2946
LED Toggle 25775
ADC Value 2945
LED Toggle 25776
ADC Value 2946
LED Toggle 25777
ADC Value 2945
LED Toggle 25778
ADC Value 2945
LED Toggle 25779
ADC Value 2944
```



CDT Build Console [empty_CC1352R1_LAUNCHXL_tirtos_ccs]

```
**** Build of configuration Debug for project empty_CC1352R1_LAUNCHXL_tirtos_ccs ****

"C:\\ti\\ccs1010\\ccs\\utils\\bin\\gmake" -k -j 4 all -O

gmake[1]: 'empty_CC1352R1_LAUNCHXL_tirtos_ccs.out' is up to date.

**** Build Finished ****
```



Global Parameters Settings that affect all instances

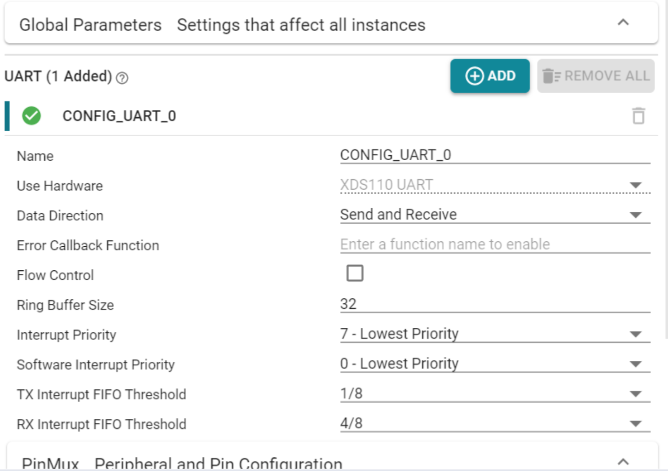
ADC (1 Added)

CONFIG_ADC_0

Name	CONFIG_ADC_0
Reference Source	Fixed
Reference Voltage	3300000
Sampling Duration	2.7 us
Input Scaling	☑
Adjust Sample Value	☐
Internal Signal	None

PinMux Peripheral and Pin Configuration

ADC Peripheral	Any(ADC0)
ADC Pin	Any(DIO23/2 (Header))



Global Parameters Settings that affect all instances

UART (1 Added)

CONFIG_UART_0

Name	CONFIG_UART_0
Use Hardware	XDS110 UART
Data Direction	Send and Receive
Error Callback Function	Enter a function name to enable
Flow Control	☐
Ring Buffer Size	32
Interrupt Priority	7 - Lowest Priority
Software Interrupt Priority	0 - Lowest Priority
TX Interrupt FIFO Threshold	1/8
RX Interrupt FIFO Threshold	4/8

PinMux Peripheral and Pin Configuration

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

“This assignment submission is my own, original work”.
Meral Abu-Jaser