**Date Submitted: 11/17/2019**

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

**Task 01: Program the CC1350 to use ADC, LED, and Push Button**

Youtube Link:

<https://youtu.be/IptNcHX4CRQ>

**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/display/Display.h>

/\* Board Header file \*/

#include "Board.h"

// Global Variables

uint16\_t adcValue = 0;

uint16\_t threshold = 100;

uint16\_t trigger = 0;

// Function Prototypes

void gpioButtonFxn0(uint\_least8\_t index);

void gpioButtonFxn1(uint\_least8\_t index);

/\*

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

\*/

void \*mainThread(void \*arg0)

{

/\* 1 second delay \*/

uint32\_t time = 100000;

/\* Call driver init functions \*/

GPIO\_init();

ADC\_init();

/\* Configure the LED pin \*/

GPIO\_setConfig(Board\_GPIO\_LED0, GPIO\_CFG\_OUT\_STD | GPIO\_CFG\_OUT\_LOW);

GPIO\_setConfig(Board\_GPIO\_BUTTON0, GPIO\_CFG\_IN\_PU | GPIO\_CFG\_IN\_INT\_FALLING);

GPIO\_setConfig(Board\_GPIO\_BUTTON1, GPIO\_CFG\_IN\_PU | GPIO\_CFG\_IN\_INT\_FALLING);

/\* Turn on user LED \*/

GPIO\_write(Board\_GPIO\_LED0, Board\_GPIO\_LED\_ON);

// ADC

ADC\_Handle adc;

ADC\_Params params;

ADC\_Params\_init(&params);

adc = ADC\_open(Board\_ADC0, &params);

if (adc == NULL) {

// ADC\_open() failed

while (1);

}

// UART

Display\_Handle displayHandle;

Display\_Params displayParams;

Display\_Params\_init(&displayParams);

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

/\* install Button callback \*/

GPIO\_setCallback(Board\_GPIO\_BUTTON0, gpioButtonFxn0);

GPIO\_setCallback(Board\_GPIO\_BUTTON1, gpioButtonFxn1);

/\* Enable interrupts \*/

GPIO\_enableInt(Board\_GPIO\_BUTTON0);

GPIO\_enableInt(Board\_GPIO\_BUTTON1);

while (1) {

int\_fast16\_t res;

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

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

Display\_printf(displayHandle, 1, 0, "Threshold %d", threshold);

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

if (adcValue >= threshold) { // arbitrary threshold

GPIO\_write(Board\_GPIO\_LED0, Board\_GPIO\_LED\_ON);

trigger = 1;

}

else {

GPIO\_write(Board\_GPIO\_LED0, Board\_GPIO\_LED\_OFF);

trigger = 0;

}

}

usleep(time);

}

}

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 > 4096) { // Ensure threshold doesn't go above max ADC range

threshold = 4096;

}

else {

threshold += 250; // increment by 250

}

}

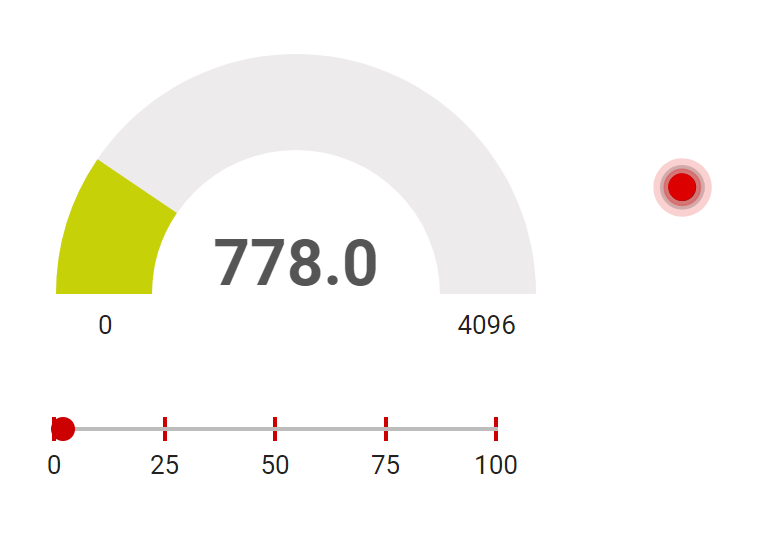
**------------------------------------------------------------------------------------**

**Task 02: GUI Composer**

Youtube Link:

<https://youtu.be/-7rh06Nbavc>

**Modified Code:**

****