**Date Submitted: 11/04/2019**

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

**Task 01:**

Youtube Link: <https://www.youtube.com/watch?v=djN0R5nj5Vs>

**Modified Schematic (if applicable): N/A**

**Modified Code:**

**/\* For usleep() \*/**

**#include <unistd.h>**

**#include <stdint.h>**

**#include <stddef.h>**

**/\* Driver Header files \*/**

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

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

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

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

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

**/\* Board Header file \*/**

**#include "Board.h"**

**/\***

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

**\*/**

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

**{**

**/\* 1 second delay \*/**

**uint32\_t time = 1;**

**/\* Call driver init functions \*/**

**GPIO\_init();**

**// I2C\_init();**

**// SPI\_init();**

**// UART\_init();**

**// Watchdog\_init();**

**/\* Configure the LED pin \*/**

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

**/\* Turn on user LED \*/**

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

**while (1) {**

**sleep(time);**

**GPIO\_toggle(Board\_GPIO\_LED0);**

**}**

**}**

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

**Task 02:**

Youtube Link: <https://www.youtube.com/watch?v=qyd_K8_GodA>

**Modified Schematic (if applicable): N/A**

**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>**

**/\* Board Header file \*/**

**#include "Board.h"**

**/\***

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

**\*/**

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

**{**

**/\* 1 second delay \*/**

**uint32\_t time = 1;**

**uint16\_t adcValue0 = 0;**

**//uint32\_t adcValue0MicroVolt;**

**uint16\_t threshold = 675;**

**uint16\_t trigger = 0;**

**/\* Call driver init functions \*/**

**GPIO\_init();**

**ADC\_init();**

**// I2C\_init();**

**// SPI\_init();**

**// UART\_init();**

**// Watchdog\_init();**

**ADC\_Handle adc;**

**ADC\_Params params;**

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

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

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

**//Display\_printf(display, 0, 0, "Error initializing ADC channel 0\n");**

**while (1);**

**}**

**/\* Configure the LED pin \*/**

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

**while (1)**

**{**

**int\_fast16\_t res;**

**res = ADC\_convert(adc, &adcValue0);**

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

**{**

**//Display\_printf(displayHandle, 1, 0, "ADC Reading %d", adcValue0);**

**if(adcValue0 >= threshold)**

**{**

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

**trigger = 1;**

**}**

**else**

**{**

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

**trigger = 0;**

**}**

**}**

**sleep(time);**

**}**

**}**

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

**Task 03:**

Youtube Link: <https://www.youtube.com/watch?v=SP10OqZu7YI>

**Modified Schematic (if applicable): N/A**

**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>**

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

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

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

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

**/\* Board Header file \*/**

**#include "Board.h"**

**/\***

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

**\*/**

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

**{**

**/\* 1 second delay \*/**

**uint32\_t time = 1;**

**uint16\_t adcValue0 = 0;**

**//uint32\_t adcValue0MicroVolt;**

**uint16\_t threshold = 660;**

**uint16\_t trigger = 0;**

**/\* Call driver init functions \*/**

**GPIO\_init();**

**ADC\_init();**

**// I2C\_init();**

**// SPI\_init();**

**// UART\_init();**

**// Watchdog\_init();**

**Display\_Handle displayHandle;**

**Display\_Params displayParams;**

**Display\_Params\_init(&displayParams);**

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

**ADC\_Handle adc;**

**ADC\_Params params;**

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

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

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

**Display\_printf(displayHandle, 0, 0, "Error initializing ADC channel 0\n");**

**while (1);**

**}**

**/\* Configure the LED pin \*/**

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

**while (1)**

**{**

**int\_fast16\_t res;**

**res = ADC\_convert(adc, &adcValue0);**

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

**{**

**Display\_printf(displayHandle, 1, 0, "ADC Reading %d", adcValue0);**

**if(adcValue0 >= threshold)**

**{**

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

**trigger = 1;**

**}**

**else**

**{**

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

**trigger = 0;**

**}**

**}**

**sleep(time);**

**}**

**}**

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

**Task 04:**

Youtube Link: <https://www.youtube.com/watch?v=OK4Dm4CJ6cY>

**Modified Schematic (if applicable): N/A**

**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>**

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

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

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

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

**/\* Board Header file \*/**

**#include "Board.h"**

**uint16\_t threshold = 0;**

**uint16\_t trigger = 0;**

**uint16\_t adcValue0 = 0;**

**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 > 3845)**

**{**

**// Ensure threshold doesn't go above max ADC range**

**threshold = 4095;**

**}**

**else**

**{**

**threshold += 250; // increment by 250**

**}**

**}**

**/\***

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

**\*/**

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

**{**

**/\* 1 second delay \*/**

**uint32\_t time = 1;**

**/\* Call driver init functions \*/**

**GPIO\_init();**

**ADC\_init();**

**// I2C\_init();**

**// SPI\_init();**

**// UART\_init();**

**// Watchdog\_init();**

**Display\_Handle displayHandle;**

**Display\_Params displayParams;**

**Display\_Params\_init(&displayParams);**

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

**ADC\_Handle adc;**

**ADC\_Params params;**

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

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

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

**Display\_printf(displayHandle, 0, 0, "Error initializing ADC channel 0\n");**

**while (1);**

**}**

**/\* Configure the LED pin \*/**

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

**/\* 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, &adcValue0);**

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

**{**

**Display\_printf(displayHandle, 1, 0, "ADC Reading %d", adcValue0);**

**if(adcValue0 >= threshold)**

**{**

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

**trigger = 1;**

**}**

**else**

**{**

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

**trigger = 0;**

**}**

**}**

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

**sleep(time);**

**}**

**}**

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