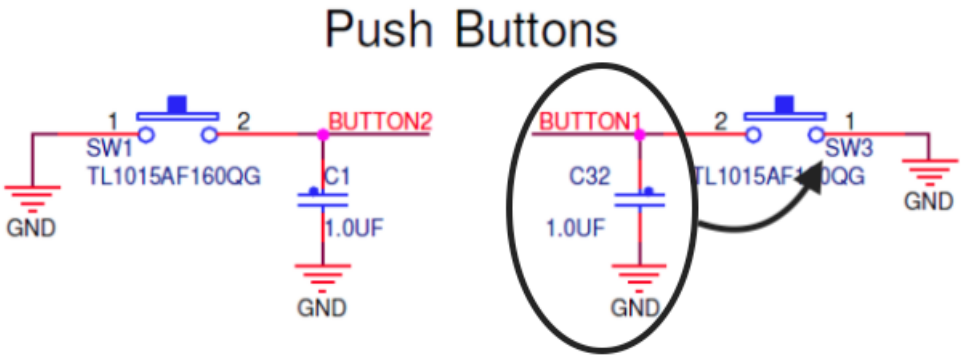
Sidharth Sudarsan
EECS 2032 - Introduction to Embedded Systems W2020
Instructor: Mokhtar Aboelaze

Lab 9 Report
ID: 216697120
Due Date: 6^{th} April, 2020

Problem 1
When your program starts, both LEDs are **ON**.
When we push **SW3**, the interrupt service routine turns the red LED **OFF** and sets a flag (global variable).
The main program (after checking the flag) will turn **OFF** the green LED 3 seconds later.



SW3 is connected to **BUTTON1** and when pushed it goes to ground.

BUTTON1 is connected to **PTC3**, i.e. Pin 3 of Port C. We have to configure that pin for interrupt. We have to configure Pin 3 of Port C first to be input, since the switch connects it to ground, we must have a pull up resistor and configure it for an interrupt.

We will use the SysTick Timer to make the required delays which has its own interrupt service routine, *SysTick_Handler(void)*.

To configure the SysTick counter, we will use the system clock and the interrupt. The routing needs to be triggered every 3 seconds. Since the system clock is 48MHz, we need to initialize the counter to:

48000000 × 3 = 144000000 = 0xF8954400

```c
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "MKL43Z4.h"
#include "fsl_debug_console.h"

void PORTC_PORTD_IRQHandler(void) {
   switch(PORTC->ISFR) { // when there is an interrupt, the corresponding bit=1
        case(0b1000): // there is an interrupt on pin 3
            PORTC->PCR[3] |= PORTC_PCR_ISF_MASK; //clear the flag
            PTE->PDOR = (1 << 31); // turn red LED OFF
            break;
}

void SysTick_Handler(void);

int main() {
  /* Init board hardware. */
  BOARD_InitBootPins();
  BOARD_InitBootClocks();
  BOARD_InitBootPeripherals();
  /* Init FSL debug console. */
  BOARD_InitDebugConsole();
  PRINTF("Hello World \n");

  __disable_irq();
  NVIC_DisableIRQ(PORTC_PORTD_IRQn);

  SIM->SCGC5 |= ((1 << 11) | (1 << 12) | (1 << 13));
  PORTC->PCR[3] |= 0x103;
  PORTD->PCR[5] = 0x100; // PORT D PIN 5 GPIO (mux = 1) PS=PE=0 no pull up or down - LEDG
  PORTE->PCR[31] = 0x100; // PORT E PIN 31 GPIO (mux = 1) PS=PE=0 no pull up or down - LEDR

  PTC->PDDR &=~0x08;
  PTD->PDDR |= (1 << 5); // set bit 5 of PORT D to 1 (PIN 5 is output)
  PTE->PDDR |= (1 << 31); // set bit 5 of PORT E to 1 (PIN 31 is output)

  PTD->PDOR = (0 << 5);
  PTE->PDOR = (0 << 31);

  PORTC->PCR[3] &= ~0xF0000;   //Set field IRQC to 0000 (disable ISF)
  PORTC->PCR[3] |= 0xA0000; //Set field IRQC to 1010 (ISF flag and falling edge)
  NVIC_SetPriority(PORTC_PORTD_IRQn, 192);
  NVIC_ClearPendingIRQ(PORTC_PORTD_IRQn);
  NVIC_EnableIRQ(PORTC_PORTD_IRQn);
  __enable_irq();

  SysTick->CTRL |= 1<<0 | 1<<2; //use system clock, initialize, No interrupt
  SysTick->LOAD = 0xF8954400;

  while(1) {
       while((SysTick->CTRL & 0x10000) == 0) {} // wait for the flag to set

       PTE->PDOR = (1 << 31); //turn green LED OFF
  }
}
```