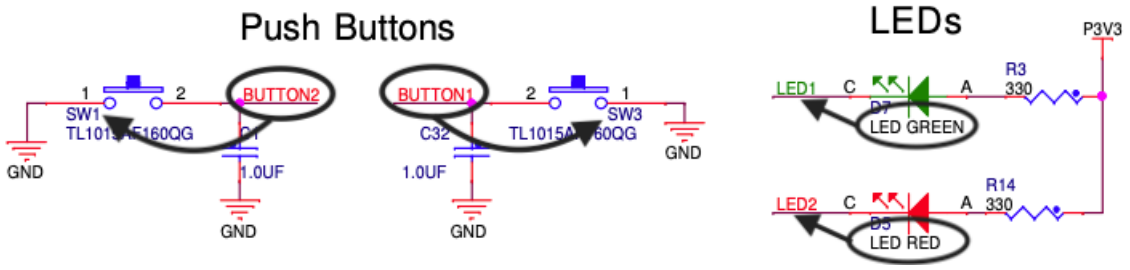
Sidharth Sudarsan
Gurban Jumagulyyev
EECS 2032 - Introduction to Embedded Systems W2020

Lab 8 Report: Group Work
Due Date: 6^{th} April, 2020
Instructor: Mokhtar Aboelaze

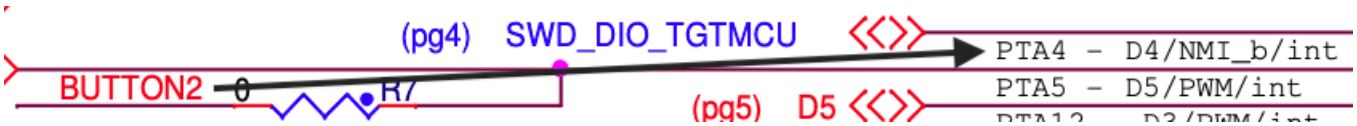
Part 1

Write a program to do the following

1. Initial state before you press any switch, the two LED's are OFF
2. If you press SW1 and SW3 is not pressed, both LED's are ON
3. If you press SW3 and SW1 is pressed, LEDG is ON, LEDR is OF
4. If both are not pressed, LED's are OFF
5. If both are pressed, releasing any switch will make LEDG OFF, LEDR ON

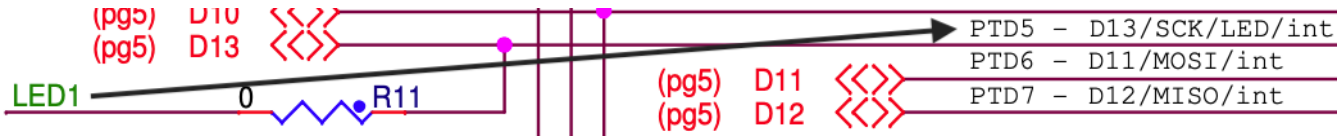


SW1 is connected to **BUTTON2** and **SW3** is connected to **BUTTON1**
LED GREEN is connected to **LED1** and **LED RED** is connected to **LED2**

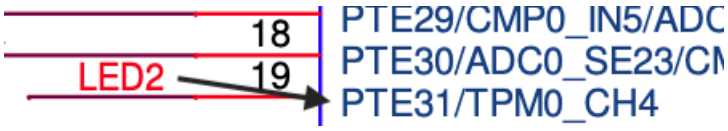


BUTTON2 is connected to **PTA4** (*Port A Pin 4*)

BUTTON1 is connected to **PTC3** (*Port C Pin 3*)

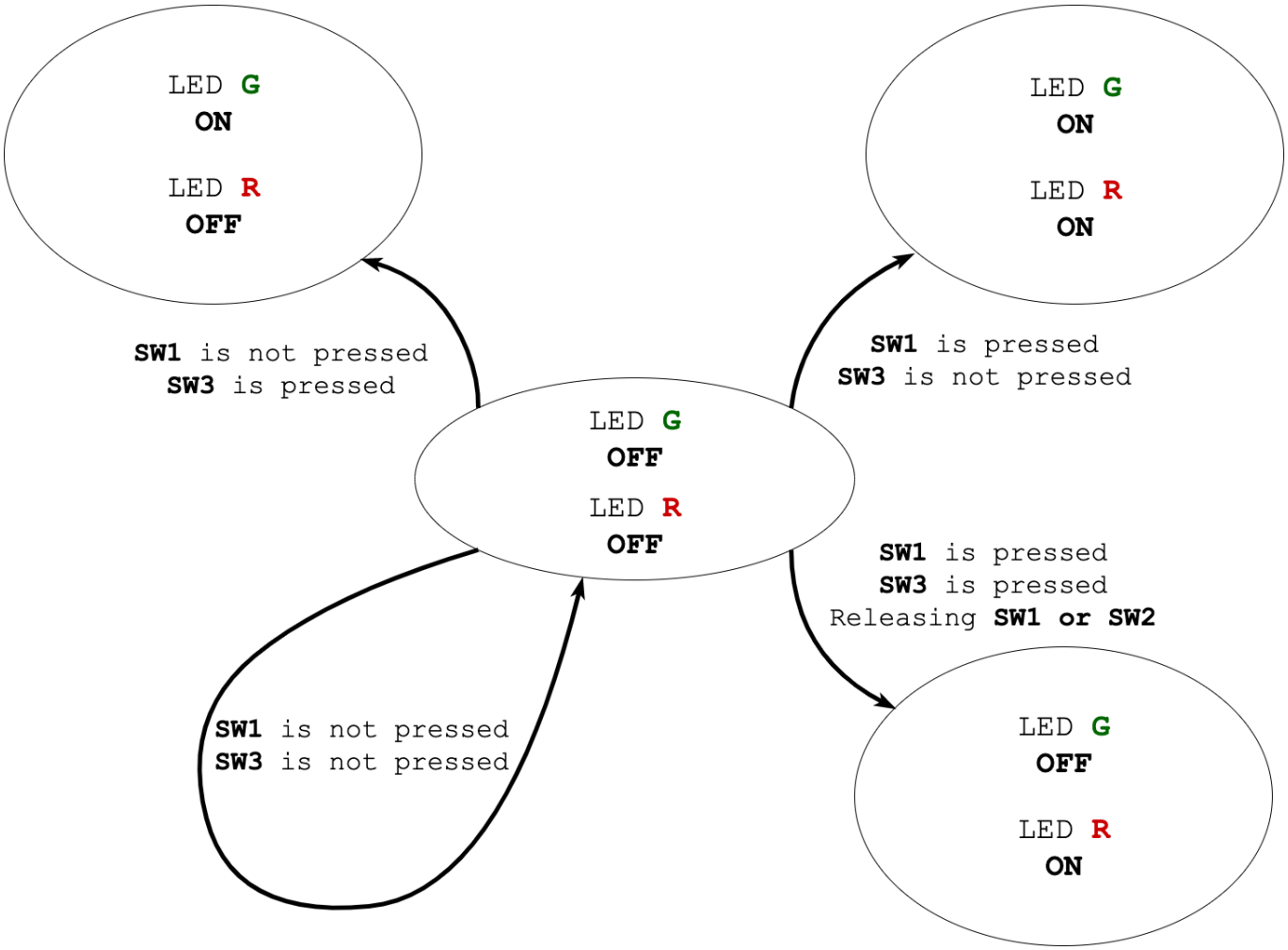


LED1 is connected to **PTD5** (*Port D Pin 5*)



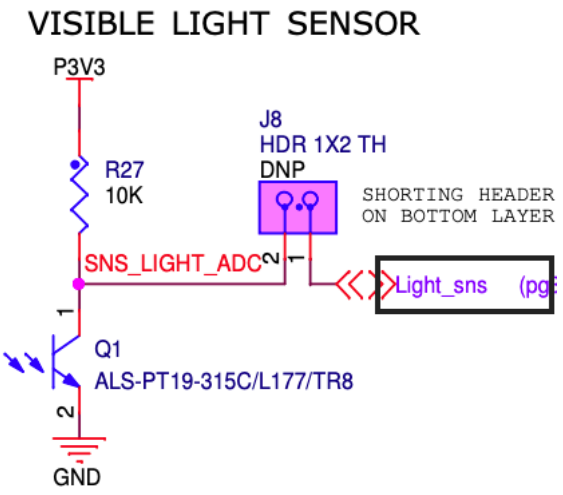
LED2 is connected to **PTE31** (*Port E Pin 31*)

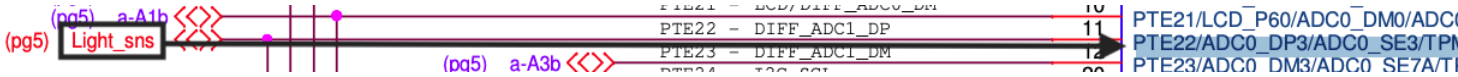
```c
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "MKL43Z4.h"
#include "fsl_debug_console.h"

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

  SIM->SCGC5 |= ((1 << 9) | (1 << 11) | (1 << 12) | (1 << 13));

  PORTA->PCR[4] = 0x103; // PORT A PIN 4 GPIO (mux = 001) PS=PE=1 pull up resistor - SW1
  PORTC->PCR[3] = 0x103; // PORT C PIN 3 GPIO (mux = 001) PS=PE=1 pull up resistor - SW3

  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

  PTA->PDDR &= ~(0x10); // set bit 4 of PORT A to 0 (input)
  PTC->PDDR &= ~(0x08); // set bit 3 of PORT C to 0 (input)

  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)

  //initial state before any switches are pressed, both LEDs are OFF
  PTD->PDOR = (1 << 5);
  PTE->PDOR = (1 << 31);

  volatile static int i = 0;
  volatile static int j = 0;

  while (1) {
    i = PTA->PDIR & (1 << 4);
    j = PTC->PDIR & (1 << 3);

    // if i is false, i.e. pressed and j is true, i.e. not pressed
    if (!i && j) {
      // both LEDs are ON
      PTD->PDOR = (0 << 5);
      PTE->PDOR = (0 << 31);
    }

    // if j is false, i.e. pressed and i is true, i.e. not pressed
    else if (!j && i) {
      // LEDG is ON, LEDR is OFF
      PTD->PDOR = (0 << 5);
      PTE->PDOR = (1 << 31);
    }

    // if i and j are true, i.e. not pressed
    else if (i && j) {
      // both LEDs are OFF
      PTD->PDOR = (1 << 5);
      PTE->PDOR = (1 << 31);
    }

    // if i and j are false, i.e. pressed
    else if (!i && !j) {
      // releasing any switch will make LEDG OFF and LEDR ON
      if (i || j) {
        PTD->PDOR = (1 << 5);
        PTE->PDOR = (0 << 31);
      }
    }
  }
}
```



Finite State Machine *for* **Part 1**

Part 2
FRDM-KL43Z has an on-board light sensor. In this lab, you will get the output of the light sensor to the input of the **ADC**. Convert it to digital value, and display that value on the monitor.





PTE22/ADC0_DP3/**ADC0_SE3**/TPM2_CH0/UART2_TX

Light sensor is connected to *Port E Pin 22*

We will use ***Port E Pin 22*** with **ADC0_SE3**, a single ended ADC channel.

Table 23-2. ADC0 channel assignment

ADC channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 and ADC0_DM0	ADC0_DP0/ADC0_SE0
00001	DAD1	ADC0_DP1 and ADC0_DM1	ADC0_DP1/ADC0_SE1
00010	DAD2	ADC0_DP2 and ADC0_DM2	ADC0_DP2/ADC0_SE2
00011	DAD3	ADC0_DP3 and ADC0_DM3	ADC0_DP3/ADC0_SE3
00100	AD4a	Reserved	ADC0_DM0/ADC0_SE4a

ADC0_DP3/ADC0_SE3 is labelled as channel **00011**.

We will use the single ended mode (not differential), so we have to configure for **SE3**.

```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"

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

    SIM_SCGC5  |= (1<<13);

    //set the MUX bits to 0
    PORTE->PCR[22] &= ~(0x700);

    SIM->SCGC6 |= 0x8000000; Set bit 27 of SCGC6 to enable ADC0

    /* ADICLK=0 (use bus clock), MODE= 10 (10-bit conversion),
    ADLSMP=0 (short sample time), ADIV=00 (divide ratio is 1). */
    ADC0->CFG1 = 0x00000008;

    /* software triggered, no compare function enable, no range enable,
    no DMA, and default voltage reference */
    ADC0->SC2 = 0x00000001;

    for (;;) {

      // channel 3 selected
      ADC0->SC1[0] = 0x03;

      // bit 7 of ADC0_SC1A is 1 when conversion is complete
      while(!(ADC0->SC1[0] & 0x80)) { }

      // Data is read from ADC0_R0
      Data = ADC0->R[0];

      // display digital value on the monitor
      PRINTF(Data);

    }
}
```