EECS 2032 Lab 9 Report

Jorra Singh - 215709876
Fabian Cojman - 216299976

Problem Statement :

For this lab, students were asked to implement some interrupts. The program starts with both LED's in the on state. Students are asked to implement an interrupt service routine that turns the red LED off and then sets a global variable. This global variable will then be used to configure the logic of the next portion of code. The next portion of code checks the value of the flag, and if the first part was done correctly, the second part will turn off the green LED exactly three seconds after the red LED was turned off.

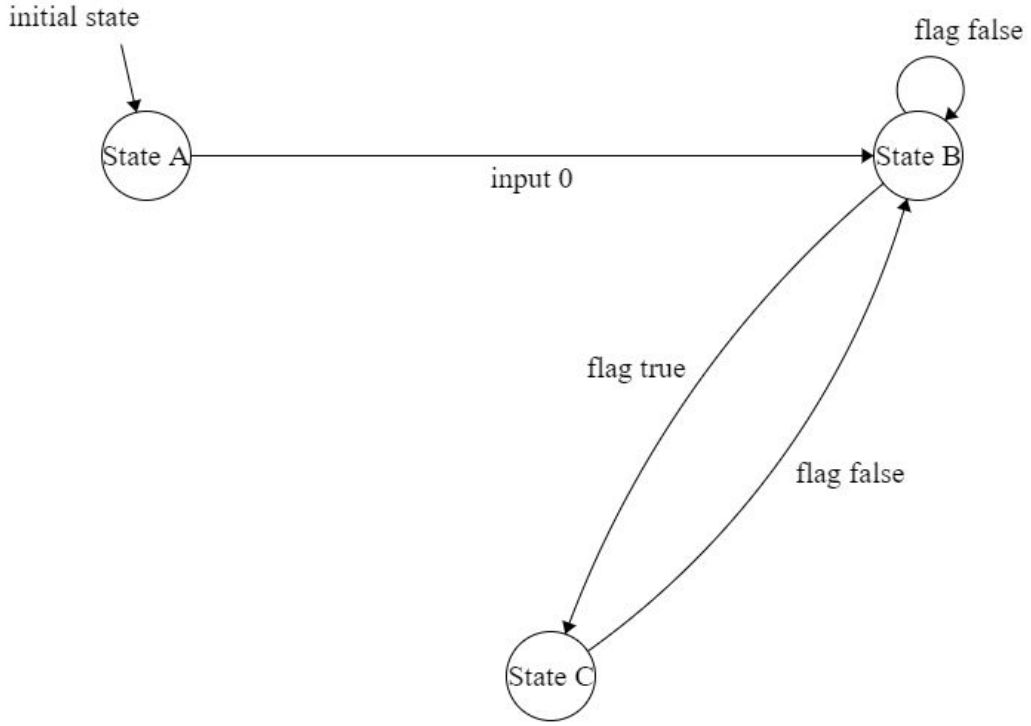
Diagrams:

State Diagram (written in pseudo) :

For this lab we have three possible states. They are listed below…

State A: LEDG is on and LEDR is on
State B: LEDG is off and LEDR is on
State C: LEDG is off and LEDR is off

To represent the three states above, we have the given input…
Input 0: SW3 is pushed
Input 1: Flag is true
Input 2: Flag is false



Explanation of State Diagram :

There are two LED's for this lab. A red LED, and a green LED, henceforth referred to as LEDR and LEDG respectively. There is only one switch used for this lab. Switch 3, henceforth referred to as SW3. The schematic for the FRDM-KL43Z is shown below, and the wiring for both LED's and the switch are shown as well.

As shown above, the initial state of both LED's is the on State. After SW3 is opened, LEDR is in the ON state. This is represented by input 0. After this is pressed we have two different possibilities. The flag is set to true, and the second part of the lab proceeds as expected, or the flag is set to false and the second part of the lab never runs correctly. If the first portion of the lab is performed incorrectly, then the flag will not be set to true. Ie. LEDG will never be turned off because it interprets the flag as false.

The second possibility is the correct one, in the sense that the flag is set correctly, and the next portion of the code can run accordingly. This is represented by flag true and State C. This is the desired state. (Acceptance state)

Code:

```
#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"
/* TODO: insert other include files here. */

/* TODO: insert other definitions and declarations here. */

/*
 * @brief   Application entry point.
 */



int main(void) {

      /* Init board hardware. */
    BOARD_InitBootPins();
    BOARD_InitBootClocks();
    BOARD_InitBootPeripherals();
      /* Init FSL debug console. */
    BOARD_InitDebugConsole();


    //turn on both LEDS
    //SW3 service interrupt routine turns red led off sets global flag
    //main program waits 3 seconds then turns green led off


    //turns on the ports
    SIM->SCGC5 |= ((1<<11) | (1<<12) | (1<<13));

    //configures portD pin 5
    PORTD->PCR[5] = 0x100; //port D pin 5 GPIO (mux = 1) PS=PE=0 no pull up
or down
    //configures portE pin 11
    PORTE->PCR[31] = 0x100; //port E pin 31 GPIO (mux = 1) PS=PE=0 no pull up
or down
    //configure portC
    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 31 of port E to 1 (pin 11 is output)
    PORTC->PCR[3] |= 0x103; //configure the mux and pull up resistor

    //turn on the LEDS
```

```
    PTD->PDOR = 0; // set G_LED on
    PTE->PDOR = 0; // set R_LED on


    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();


    //enable the 3 second clock delay
    SysTick->CTRL |= 1<<0 | 1<<2;
    SysTick->LOAD = 0x8954400;


    while(1) {
      if(PORT_PCR_ISF_MASK & (1 << 3)){ //check the state of the flag
            while((SysTick->CTRL & 0x10000) == 0) {} // wait for clock
            PTD->PDOR = (1<<5); // set G_LED off
      }
    }

    return 0 ;

}

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] |= PORT_PCR_ISF_MASK; //clear the flag
            PTE->PDOR = (1<<31); // set R_LED off
            break;
    }
}
```

Schematics of FRDM-KL43Z

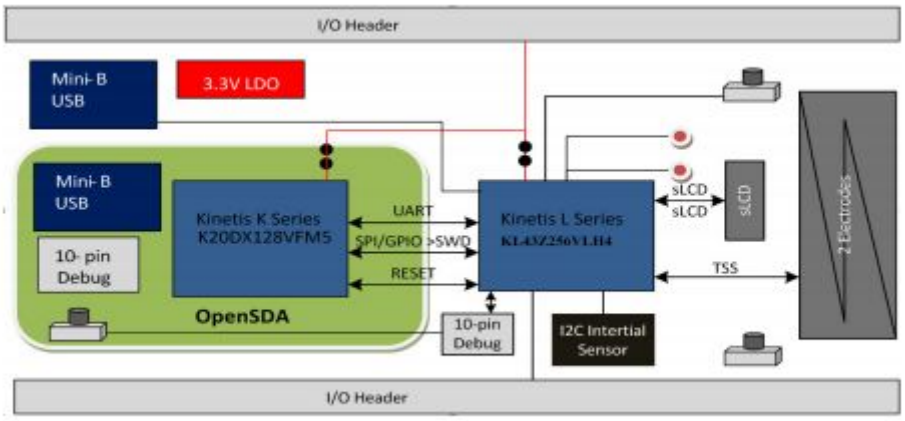


Figure 1: layout of FRDM-KL43X

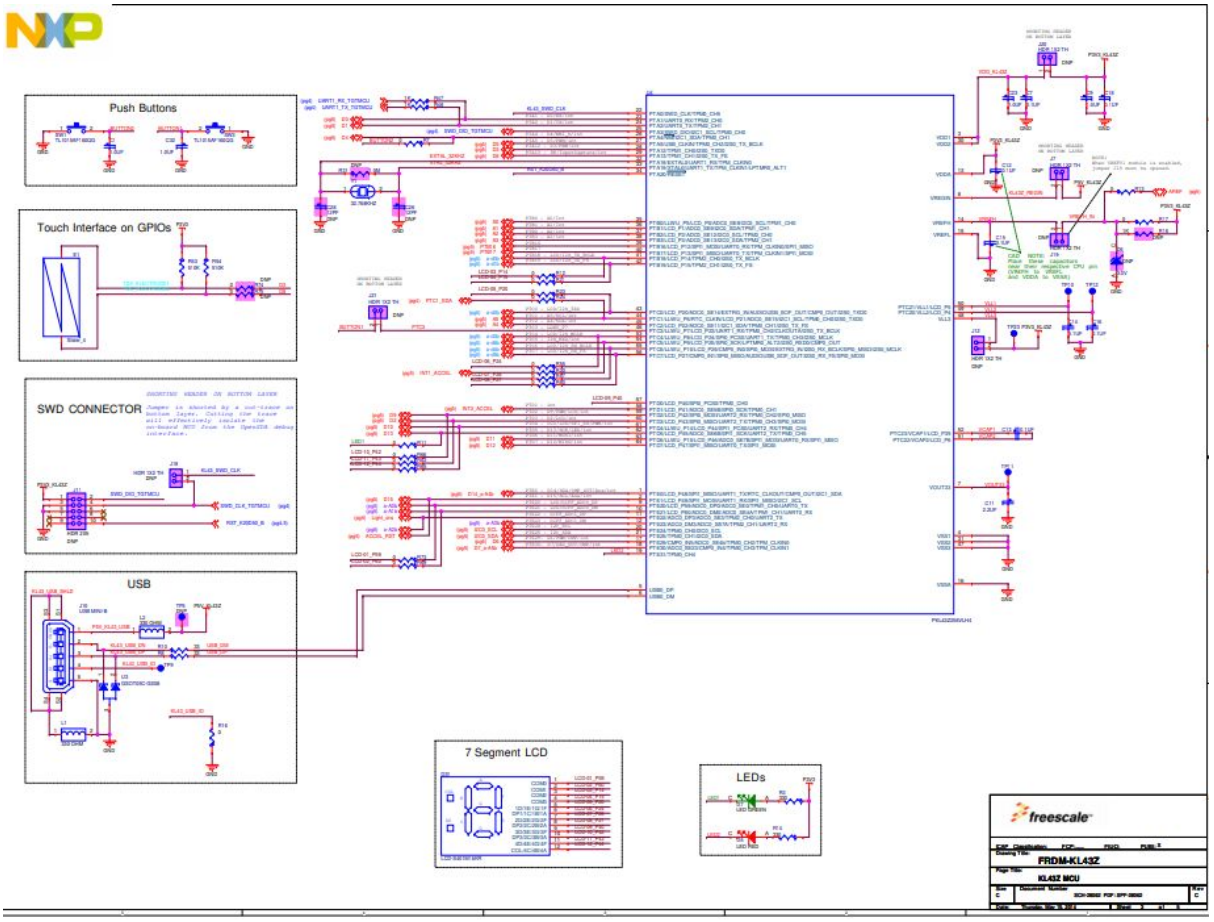


Figure 2: schematic of FRDM-KL43X