Lab 3: Interrupts and Debouncers

Instructor's Guide

Lab Introduction

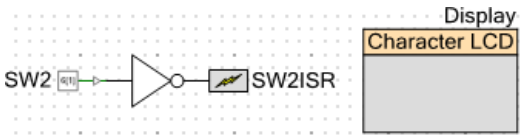
This lab introduces concepts of interrupts and debouncers. Interrupts are an important concept when processing multiple inputs, especially if they need to be handled in a timely manner, so this lab may need to be referred back to when working on the final project.

Note that some of the content of this lab was cut due to the initial lab being too long. This content has been placed into the incomplete Lab 6.

Instructor Review

Part A

This is the first lab where creating the schematic is entirely up to the student. Here is what the schematic should look like:



Here is a sample firmware solution.

SW2ISR.c

```
/*******************************************************************************
*  Place your includes, defines and code here
********************************************************************************/
/* `#START SW2ISR_intc` */

#define TRUE  1
#define FALSE 0

//The global variables described in main.c
uint8 sw2_pressed;
uint16 sw2_count;

/* `#END` */


/****** SNIP ******/
```

```
CY_ISR(SW2ISR_Interrupt)
{
    /*  Place your Interrupt code here. */
    /* `#START SW2ISR_Interrupt` */

      //Increment the count and set the flag to true
      ++sw2_count;
      sw2_pressed = TRUE;

    /* `#END` */
}
```

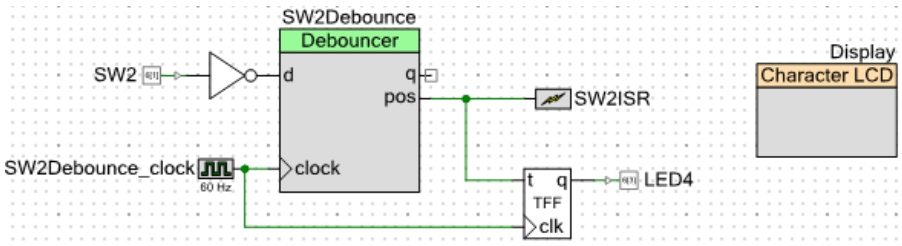
main.c

```
#include <project.h>

#define TRUE  1
#define FALSE 0

//Global boolean for indicating when the pushbutton was pressed
//Declared in SW2ISR.c
extern uint8 sw2_pressed;

//Global counter for how many times the pushbutton was pressed
//Declared in SW2ISR.c
extern uint16 sw2_count;

int main()
{
      //Initialize global variables
      sw2_count = 0;
      sw2_pressed = TRUE;  //Prints initial count

      //Initialize the display
      Display_Start();
      Display_Position(0, 0);
      Display_PrintString("SW2 Count");

      //Initialize the ISR and clears any pending interrupt
      //There may be a pending interrupt due to the power-on
      //  state of the input pin
      SW2ISR_Start();
      SW2ISR_ClearPending();

      //Enable interrupts
      CyGlobalIntEnable;

      for(;;)
      {
            //Do not update the display unless the pushbutton was pressed
            if(sw2_pressed)
            {
                  //Disable interrupts to prevent a race condition
                  //  when using global variables
                  CyGlobalIntDisable;
```

```
                    //Reset the pressed flag and print out the count
                    //Since the count never decrements (unless it overflows),
                    //  no need to clear anything
                    sw2_pressed = FALSE;
                    Display_Position(1, 0);
                    Display_PrintDecUint16(sw2_count);

                    //Re-enable interrupts
                    CyGlobalIntEnable;
                }
        }
}
```

Part B

This part of the lab only requires an update to the schematic. Here is what the updated schematic should look like:



Nothing needs to be changed in the firmware.