

```

/*
 * read_modify_write_sftw_sw0.c
 *
 * Created: 2/3/2022 11:49:03 PM
 * Author : Jason Tan
 */

#include <avr/io.h>

int main(void)
{
    //pointer to PIN2CTRL array of pin configuration registers
    uint8_t* ptr = (uint8_t*)&PORTA.PIN2CTRL;
    PORTC_PIN0CTRL = 0x08; //Enable internal pull up for PC0
    PORTC_PIN1CTRL = 0x08; //Enable internal pull up for PC1

    //DIR is what configure the port pins as inputs or outputs
    VPORTA_DIR = 0x00; //Configure PORTA pins ( PA7, PA6, PA5, PA4, PA3, PA2) as the
        inputs
    VPORTC_DIR = 0x00; //Configure PORTC pins ( PC1, PC0) as the inputs
    VPORTD_DIR = 0xFF; //Configure PORTD pins (PD07 through PD00) as output pins

    VPORTB_DIR = 0x08; //PB3 output for LED0
    PORTB_PIN2CTRL = 0x08; //Pull up enable for SW0

    //Configure PA7 - PA2 as input buffers with internal pull up resistors
    for(uint8_t i = 0; i < 8; i++){
        *(ptr + i) |= PORT_PULLUPEN_bm;
    }

    uint8_t n = 3; //Field value starting from 3
    uint8_t field_mask = 0x0F; //Field mask
    uint8_t field_val; //Read PA3-PA2 and PC1-PC0.
    VPORTD_OUT = ~((VPORTA_IN & 0xFC) | (VPORTC_IN & 0x03));

    while (1)
    {
        //Check for if SW0 is not pressed meaning that sends a logic 1
        //if((VPORTB_IN & PIN2_bm)){
        //    VPORTD_OUT = ~((VPORTA_IN & 0xFC) | (VPORTC_IN & 0x03));
        //}
        //Check for if SW0 is press meaning that sends a logic 0
        field_val = ((VPORTC_IN & 0x03) | (VPORTA_IN & 0x0C));
        if(!(VPORTB_IN & PIN2_bm)){
            //
            //          0b1000 0111          0b1111 PA3 PA2 PC1 PC0
            //          -> 1 PA3 PA2 PC1 PC0 000
            VPORTD_OUT = (VPORTD_OUT & ~(field_mask << n)) | (((field_val &
                field_mask) << n) ^ 0x78);
        }
    }
}

```

```
}

/*
//If PB2 is at logic 0 which means SW0 is pressed
if(~(VPORTB_IN & PIN2_bm)){

    //When PC0 is pressed which sends logic 0
    if(~(VPORTC_IN & PIN0_bm)){
        PORTD_OUT &= ~(1 << 3); //Clears bit 3 to turn on the LED
    }
    //When PC0 is not pressed
    else if((VPORTC_IN & PIN0_bm)){
        PORTD_OUT |= (1 << 3); //Set bit 3 to turn off the LED
    }

    //When PC1 is pressed which sends logic 0
    if (~(VPORTC_IN & PIN1_bm)){
        PORTD_OUT &= ~(1 << 4); //Clears bit 4 to turn on the LED
    }
    //When PC1 is not pressed
    else if((VPORTC_IN & PIN1_bm)){
        PORTD_OUT |= (1 << 4); //Set bit 4 to turn off the LED
    }

    //When PA2 is pressed which sends logic 0
    if (~(VPORTA_IN & PIN2_bm)){
        PORTD_OUT &= ~(1 << 5); //Clears bit 5 to turn on the LED
    }
    //When PA2 is not pressed
    else if((VPORTA_IN & PIN2_bm)){
        PORTD_OUT |= (1 << 5); //Set bit 5 to turn off the LED
    }

    //When PA3 is pressed which sends logic 0
    if (~(VPORTA_IN & PIN3_bm)){
        PORTD_OUT &= ~(1 << 6); //Clears bit 6 to turn on the LED
    }
    //When PA3 is not pressed
    else if((VPORTA_IN & PIN3_bm)){
        PORTD_OUT |= (1 << 6); //Set bit 6 to turn off the LED
    }
}

//If PB2 is at logic 1 which means SW0 not pressed
else if ((VPORTB_IN & PIN2_bm)){

    //From Lab 2 Task 1
```

```
//When PC0 is pressed which sends logic 0
if(~(VPORTC_IN & PIN0_bm)){
    PORTD_OUT &= ~(1 << 0); //Clears bit 0 to turn on the LED
}
//When PC0 is not pressed
else if((VPORTC_IN & PIN0_bm)){
    PORTD_OUT |= (1 << 0); //Set bit 0 to turn off the LED
}

//When PC1 is pressed which sends logic 0
if (~(VPORTC_IN & PIN1_bm)){
    PORTD_OUT &= ~(1 << 1); //Clears bit 1 to turn on the LED
}
//When PC1 is not pressed
else if((VPORTC_IN & PIN1_bm)){
    PORTD_OUT |= (1 << 1); //Set bit 1 to turn off the LED
}

//When PA2 is pressed which sends logic 0
if (~(VPORTA_IN & PIN2_bm)){
    PORTD_OUT &= ~(1 << 2); //Clears bit 2 to turn on the LED
}
//When PA2 is not pressed
else if((VPORTC_IN & PIN2_bm)){
    PORTD_OUT |= (1 << 2); //Set bit 2 to turn off the LED
}

//When PA3 is pressed which sends logic 0
if (~(VPORTA_IN & PIN3_bm)){
    PORTD_OUT &= ~(1 << 3); //Clears bit 3 to turn on the LED
}
//When PA3 is not pressed
else if((VPORTC_IN & PIN3_bm)){
    PORTD_OUT |= (1 << 3); //Set bit 3 to turn off the LED
}

//When PA4 is pressed which sends logic 0
if (~(VPORTA_IN & PIN4_bm)){
    PORTD_OUT &= ~(1 << 4); //Clears bit 4 to turn on the LED
}
//When PA4 is not pressed
else if((VPORTC_IN & PIN4_bm)){
    PORTD_OUT |= (1 << 4); //Set bit 3 to turn off the LED
}

//When PA5 is pressed which sends logic 0
if (~(VPORTA_IN & PIN5_bm)){
```

```
        PORTD_OUT &= ~(1 << 5); //Clears bit 5 to turn on the LED
    }
    //When PA5 is not pressed
    else if((VPORTC_IN & PIN3_bm)){
        PORTD_OUT |= (1 << 5); //Set bit 5 to turn off the LED
    }

    //When PA6 is pressed which sends logic 0
    if (~(VPORTA_IN & PIN6_bm)){
        PORTD_OUT &= ~(1 << 6); //Clears bit 6 to turn on the LED
    }
    //When PA5 is not pressed
    else if((VPORTC_IN & PIN6_bm)){
        PORTD_OUT |= (1 << 6); //Set bit 6 to turn off the LED
    }

    //When PA7 is pressed which sends logic 0
    if (~(VPORTA_IN & PIN6_bm)){
        PORTD_OUT &= ~(1 << 7); //Clears bit 7 to turn on the LED
    }
    //When PA5 is not pressed
    else if((VPORTC_IN & PIN6_bm)){
        PORTD_OUT |= (1 << 7); //Set bit 7 to turn off the LED
    }
}

*/

}
```