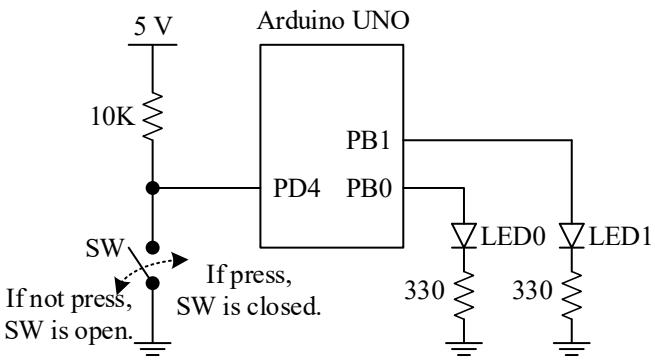
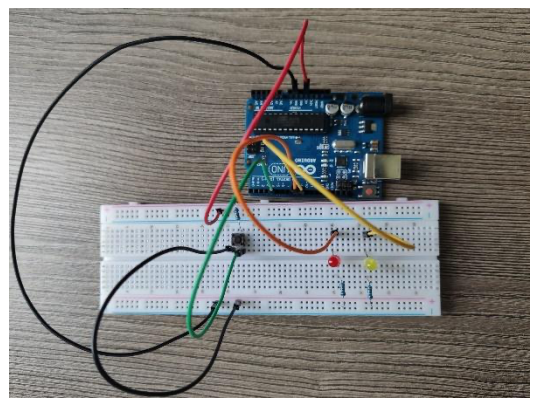
Chaniporn Nerunchorn 6322771450
Ornnicha Phueaksri 6322773837

SIIT SINCE 1992	**School of ICT** **Sirindhorn International Institute of Technology**
CSS332 Microcontrollers and Applications	**Lab 7: AVR Interrupt Programming**

Instructions: Answer the following exercises. During the lab class, please feel free to ask the instructor, the TAs, or other students if there is a question. When finishing all of them, the students can ask a TA to check the answers. The students submit this lab sheet with the answers to the Google Classroom (no submission, no score). (In the Google Classroom, do not forget to press the Confirm button to submit the work.)

Exercise 1: (**Using Timer0 overflow interrupt**). We have connected a circuit as shown below.





We would like to write an Assembly program to do the following tasks.

- Task1 – The microcontroller monitors the status of the switch SW: if we press the switch SW, LED0 is on; if we do not press the switch SW, LED0 is off.
- Task2 – The microcontroller keeps turning on and off LED1, every 1 second.

As a result, the microcontroller will do Task1 in the main program and do Task2 by using the Timer0 overflow interrupt. Timer0 will be set up as follows:

- normal mode, pre-scaling number = 1024, the Timer0 overflow interrupt happens every 5000 μs.

As a result, we need that every 200 Timer0 overflow interrupts will turn the LED1 on or off. Answer the following questions.

a) What are the values of these Timer0 registers?

TCNT0	=	178
TCCR0A	=	0x00
TCCR0B	=	0x05

b) To use the Timer0 overflow interrupt, what should be the value of the TIMSK0 register?

TIMSK0 = 0X01

c) Complete the following Assembly program.

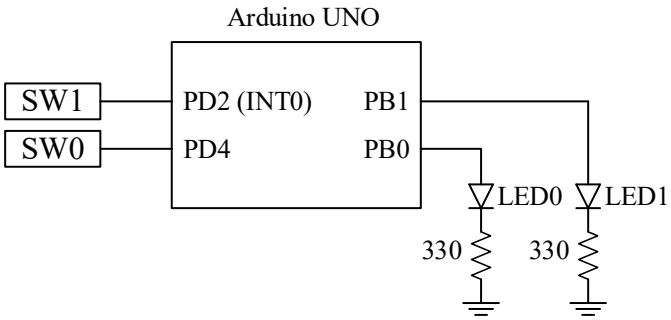
```
.ORG    0x0
        JMP     MAIN
.ORG    0x20
        JMP     T0_OV_ISR

;Start the main program
.ORG    0x100
MAIN:   LDI     R16, HIGH(RAMEND)
        OUT     SPH, R16
        LDI     R16, LOW(RAMEND)
        OUT     SPL, R16

        CALL    PIN_SETUP       ;Set up pin modes
        LDI     R16, (1<<TOIE0)
        STS     TIMSK0, R16     ;Enable the Timer0 overflow interrupt
        SEI                     ;Enable the global interrupt
        CALL    TIMER0_SETUP    ;Set up Timer0
        LDI     R21, 200        ;A dummy variable used in ISR

LOOP:   SBIC    PIND, 4     ;Main task
        RJMP    L_OFF
        RJMP    L_ON
L_OFF:  CBI     PORTB, 0    ;Turn LED0 off
        RJMP    LOOP
L_ON:   SBI     PORTB, 0    ;Turn LED0 on
        RJMP    LOOP
;End of the main program

PIN_SETUP:      ;Subroutine to set up pin modes
        SBI     DDRB, 0     ;Set the pin PB0 as the output -> LED0
        CBI     PORTB, 0    ;PB0=0
        SBI     DDRB, 1     ;Set the pin PB0 as the output -> LED1
        CBI     PORTB, 1    ;PB1=0
        CBI     DDRD, 4     ;Set the pin PD4 as the input
        SBI     PORTD, 4    ;Set pull-up resistor on PD4
        RET                 ;Return to the main program
```

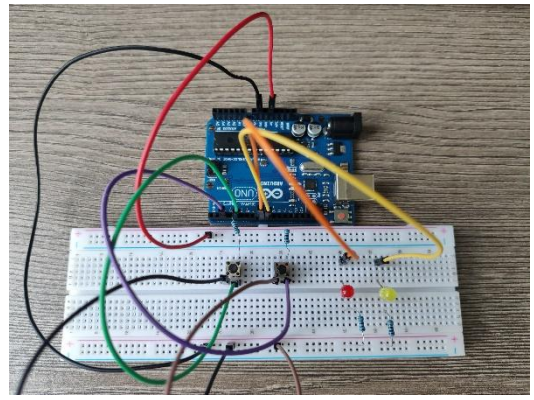
Time delay 5000 $= \left[\left(\frac{(255-A+1)\times P}{3}\right)\times 3\right] + 2 + 14\Big)\times 0.0625$

```
TIMER0_SETUP:   ;Subroutine to set up Timer0
        LDI     R20, 177
        OUT     TCNT0, R20  ;Set the initial value of Timer0
        LDI     R20, 0x00
        OUT     TCCR0A, R20 ;Set the normal mode,
        LDI     R20, 0x05
        OUT     TCCR0B, R20 ;Pre-scaling 1024 and start the clock
        RET                 ;Return to the main program

.ORG    0x200
T0_OV_ISR:      ;ISR of the Timer0 overflow interrupt
        DEC     R21
        BRNE    HERE
        LDI     R21, 200        ;200 loops needed for delay 1 second
        IN      R17, PORTB      ;Read PORTB
        LDI     R18, (1<<1)     ;R18=0b00000010
        EOR     R17, R18
        OUT     PORTB, R17      ;Toggle PB1
HERE:   LDI     R18, 177
        OUT     TCNT0, R18      ;Set up Timer0 initial value
        RETI                    ;Return to the main program
```

d) Upload your assembly program to your Arduino UNO board. Take a video to demonstrate your result. Name it "Ex1" and submit to the Google Classroom.

Exercise 2: (**Using the external interrupt INT0**). We have connected a circuit as shown below.





We would like to write an Assembly program to do the following tasks.

- Task1 – The microcontroller monitors the status of the switch SW0: if we press the switch SW0, LED0 is on; if we do not press the switch SW0, LED0 is off.
- Task2 – The microcontroller monitors the status of the switch SW1: if we press and release the switch SW1, the LED1 will be toggled (on □ off or off □ on).

As a result, the microcontroller will do Task1 in the main program and do Task2 by using the external interrupt INT0 (falling-edge trigger). Answer the following questions.

a) To use the external interrupt INT0, what should be the values of the EIMSK and EICRA registers?

EIMSK	=	0x01
EICRA	=	0x02

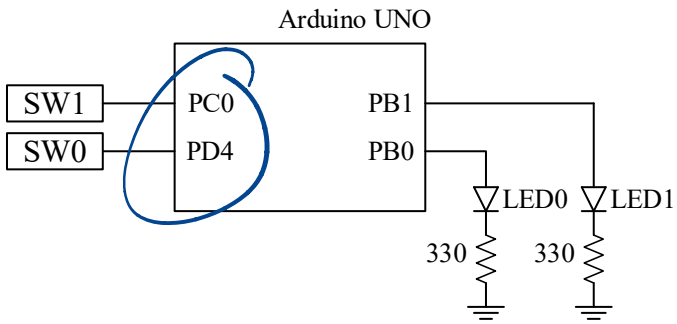
b) Complete the following Assembly program.

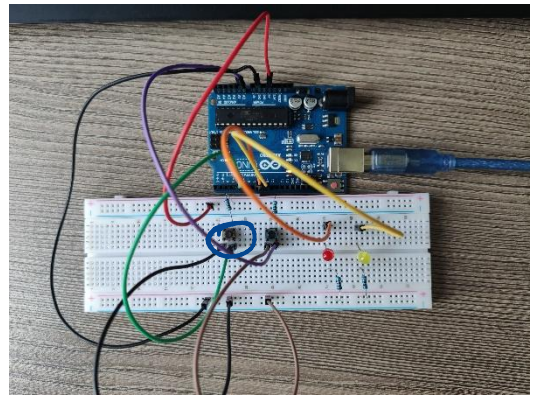
```
.ORG    0x0
        JMP     MAIN
.ORG    0x02
        JMP     INT0_ISR

```

```
;Start the main program
.ORG    0x100
MAIN:   LDI     R16, HIGH(RAMEND)
        OUT     SPH, R16
        LDI     R16, LOW(RAMEND)
        OUT     SPL, R16

        CALL    PIN_SETUP       ;Set up pin modes
        LDI     R16, (1<<INT0)
        OUT     EIMSK, R16      ;Enable the external interrupt INT0
        LDI     R16, (1<<ISC01)
        STS     EICRA, R16      ;Set falling-edge trigger
        SEI                     ;Enable the global interrupt

LOOP:   SBIC    PIND, 4     ;Main task
        RJMP    L_OFF
        RJMP    L_ON
L_OFF:  CBI     PORTB, 0    ;Turn LED0 off
        RJMP    LOOP
L_ON:   SBI     PORTB, 0    ;Turn LED0 on
        RJMP    LOOP
;End of the main program

PIN_SETUP:      ;Subroutine to set up pin modes
        SBI     DDRB, 0     ;Set the pin PB0 as the output -> LED0
        CBI     PORTB, 0    ;PB0=0
        SBI     DDRB, 1     ;Set the pin PB0 as the output -> LED1
        CBI     PORTB, 1    ;PB1=0
        CBI     DDRD, 4     ;Set the pin PD4 as the input
        SBI     PORTD, 4    ;Set pull-up resistor on PD4
        SBI     PORTD, 2    ;Set pull-up resistor on PD2 (INT0)
        RET                 ;Return to the main program

.ORG    0x200
INT0_ISR:       ;ISR of the INT0 external hardware
        IN      R17, PORTB      ;Read PORTB
        LDI     R18, (1<<1)     ;R18=0b00000010
        EOR     R17, R18
        OUT     PORTB, R17      ;Toggle PB1
        RETI                    ;Return to the main program
```

c) Upload your assembly program to your Arduino UNO board. Take a video to demonstrate your result. Name it "Ex2" and submit to the Google Classroom.

Exercise 3: (**Using the pin change interrupt**). We have connected a circuit as shown below.





We would like to write an Assembly program to do the following tasks.

- Task1 – The microcontroller monitors the status of the switch SW0: if we press the switch SW0, LED0 is on; if we do not press the switch SW0, LED0 is off.
- Task2 – The microcontroller monitors the status of the switch SW1: if we press and release the switch SW1, the LED1 will be toggled (on □ off or off □ on).

As a result, the microcontroller will do Task1 in the main program and do Task2 by using the pin change interrupt (via the switch SW1 connected to the pin PC0). Answer the following questions.

a) To use the pin change interrupt as specified above, what should be the values of the following registers?

PCICR	=	0x02
PCMSK0	=	–
PCMSK1	=	0b00000001
PCMSK2	=	–

b) Complete the following Assembly program.

```
.ORG    0x0
        JMP     MAIN
.ORG    0x08
        JMP     PCINT8_ISR

```

```
;Start the main program
.ORG    0x100
MAIN:   LDI     R16, HIGH(RAMEND)
        OUT     SPH, R16
        LDI     R16, LOW(RAMEND)
        OUT     SPL, R16

        CALL    PIN_SETUP   ;Set up pin modes
        LDI     R16, (1<<PCIE1)
        STS     PCICR, R16  ;Enable pin change interrupt PORTC
        LDI     R16, 0x01
        STS     PCMSK1, R16 ;Enable interrupt from pin PC0
        SEI                 ;Enable the global interrupt
        LDI     R20, 2      ;A dummy variable used in ISR

LOOP:   SBIC    PIND, 4     ;Main task
        RJMP    L_OFF
        RJMP    L_ON
L_OFF:  CBI     PORTB, 0    ;Turn LED0 off
        RJMP    LOOP
L_ON:   SBI     PORTB, 0    ;Turn LED0 on
        RJMP    LOOP
;End of the main program

```

```
PIN_SETUP:      ;Subroutine to set up pin modes
        SBI     DDRB, 0     ;Set the pin PB0 as the output -> LED0
        CBI     PORTB, 0    ;PB0=0
        SBI     DDRB, 1     ;Set the pin PB1 as the output -> LED1
        CBI     PORTB, 1    ;PB1=0
        CBI     DDRD, 4     ;Set the pin PD4 as the input
        SBI     PORTD, 4    ;Set pull-up resistor on PD4
        SBI     PORTC, 0    ;Set pull-up resistor on PC0
        RET                 ;Return to the main program

```

```
.ORG    0x200
PCINT8_ISR:     ;ISR of the PORTC Pin Change Interrupt
        DEC     R20
        BRNE    HERE
        IN      R17, PORTB      ;Read PORTB
        LDI     R18, (1<<1)     ;R18=0b00000010
        EOR     R17, R18
        OUT     PORTB, R17      ;Toggle PB1
        LDI     R20, 2          init. R20 again
HERE:   RETI                    ;Return to the main program
```

c) Upload your assembly program to your Arduino UNO board. Take a video to demonstrate your result. Name it "Ex3" and submit to the Google Classroom.

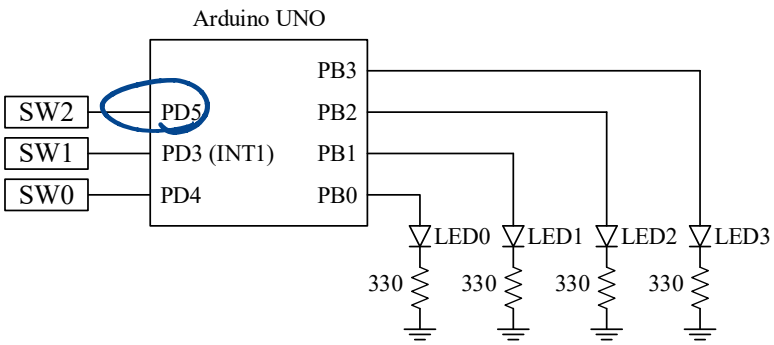
Exercise 4: Connect the circuit as shown below. Write an Assembly program to do the following tasks.

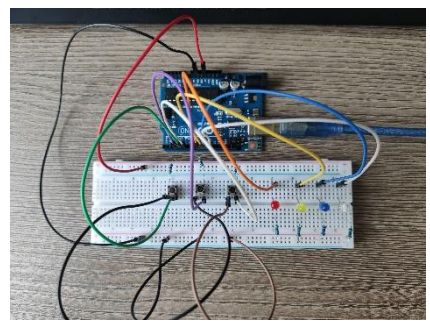
- Task1 – The microcontroller monitors the status of the switch SW0: if we press the switch SW0, LED0 is on; if we do not press the switch SW0, LED0 is off.
- Task2 – The microcontroller monitors the status of the switch SW1: if we press and release the switch SW1, the LED1 will be toggled (on off or off on).
- Task3 – The microcontroller monitors the status of the switch SW2: if we press and release the switch SW2, the LED2 will be toggled (on off or off on).
- Task4 – The microcontroller keeps turning on and off LED3, every 1 second.

As a result, we design such that:

- the microcontroller will do Task1 in the main program,
- the microcontroller will do Task2 by using the external interrupt INT1 (fall-edge trigger),
- the microcontroller will do Taks3 by using the pin change interrupt (via the switch SW2 connected to the pin PD5),
- the microcontroller will do Task4 by using the Timer0 overflow interrupt (similar to Exercise 1 – used the same setup).

Answer the following questions.





a) To use the external interrupt INT1 for Task2, what should be the values of the EIMSK and EICRA registers?

EIMSK	=	0x02
EICRA	=	0x02

b) To use the pin change interrupt for Task3, what should be the values of the following registers?

PCICR	=	0x04
PCMSK0	=	–
PCMSK1	=	–
PCMSK2	=	0x 20

c) What are the values of these Timer0 registers and TIMSK0 register?

TCNT0	=	178
TCCR0A	=	0X00
TCCR0B	=	0X05
TIMSK0	=	0X02

d) Copy the screenshot of your assembly program and put it here.

1

```
        .org 0x0
         JMP Main

        .org 0x04
         JMP INT1_ISR

        .org 0x0A
         JMP PCINT8_ISR

        .org 0x20
         JMP T0_OV_ISR

    ;Start the main program
    .org 0x100
    Main:
         LDI R16, HIGH(RAMEND)
         OUT SPH, R16
         LDI R16, LOW(RAMEND)
         OUT SPL, R16

         CALL PIN_SETUP
         LDI R16, (1<<TOIE0)
         STS TIMSK0, R16
         SEI
         CALL TIMER0_SETUP
         LDI R21, 178

         LDI R16, (1<<INT1)
         out EIMSK, R16
         LDI R16, (1<<ISC11)
         STS EICRA, R16

         LDI R16, (1<<PCIE2)
         STS PCICR, R16
         LDI R16, 0x20
         STS PCMSK2, R16
         SEI
         LDI R20, 2

         SEI

    LOOP:
         SBIC PIND, 4
         RJMP L_OFF
```

2

```
    LOOP:
         SBIC PIND, 4
         RJMP L_OFF
         RJMP L_ON
    L_OFF:
         CBI PORTB, 0
         RJMP LOOP
    L_ON:
         SBI PORTB, 0
         RJMP LOOP

    PIN_SETUP:
         SBI DDRB, 0
         CBI PORTB, 0
         SBI DDRB, 1
         CBI PORTB, 1
         CBI DDRD, 4
         SBI PORTD, 4
         RET

    TIMER0_SETUP:
         LDI R20, 177
         OUT TCNT0, R20
         LDI R20, 0x00
         OUT TCCR0A, R20
         LDI R20, 0x05
         OUT TCCR0B, R20
         RET

    .org 0x200
    T0_OV_ISR:
         DEC R21
         BRNE HERE
         LDI R21, 200
         IN R17, PORTB
         LDI R18, (1<<3)
         EOR R17, R18
         OUT PORTB, R17
    HERE:
         LDI R18, 178
         OUT TCNT0, R18
         RETI
    .org 0x300
    INT1_ISR:
         IN R17, PORTB
         LDI R18, (1<<1)
```

3

```
    .org 0x200
    T0_OV_ISR:
         DEC R21
         BRNE HERE
         LDI R21, 200
         IN R17, PORTB
         LDI R18, (1<<3)
         EOR R17, R18
         OUT PORTB, R17
    HERE:
         LDI R18, 178
         OUT TCNT0, R18
         RETI
    .org 0x300
    INT1_ISR:
         IN R17, PORTB
         LDI R18, (1<<1)
         EOR R17, R18
         OUT PORTB, R17
         RETI
         .org 0x400
    PCINT8_ISR:
         DEC R20
         BRNE HERE2
         IN R17, PORTB
         LDI R18, (1<<2)
         EOR R17, R18
         OUT PORTB, R17
         LDI R20, 2
    HERE2: RETI
```