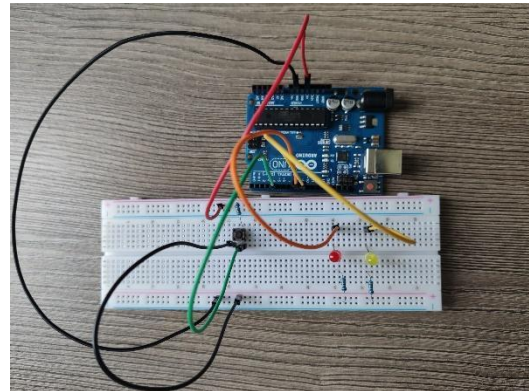
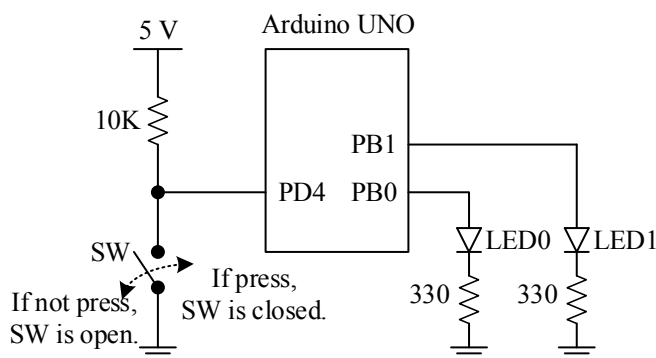


	School of ICT Sirindhorn International Institute of Technology
CSS332 Microcontrollers and Applications	Lab 7: AVR Interrupt Programming

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 μ s.

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 = 0x82
TCCR0A = 00 = 0x00

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

$$5000 = \left[\left(\left(\frac{(255 - A + 1) \times 1024}{3} \right) \times 3 \right) + 2 + 14 \right] \times 0.0625$$

$$\begin{aligned} A &= 177.89 \\ &\approx 178 \\ &= 0x82 \end{aligned}$$



TCCR0B = 101 = 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.

```
1  .ORG    0x00
2      JMP    MAIN
3  .ORG    0x20
4      JMP    T0_OV_ISR
5
6  ;Start the main program
7  .ORG    0x100
8  MAIN:  LDI    R16, HIGH(RAMEND)
9          OUT    SPH, R16
10         LDI    R16, LOW(RAMEND)
11         OUT    SPL, R16
12
13         CALL   PIN_SETUP    ;Set up pin modes
14         LDI    R16, (1<<TOIE0)
15         STS    TIMSK0, R16    ;Enable the Timer0 overflow interrupt
16         SEI    ;Enable the global interrupt
17         CALL   TIMER0_SETUP ;Set up Timer0
18         LDI    R21, 200      ;A dummy variable used in ISR
19
20 LOOP:  SBIC    PIND, 4      ;Main task
21         RJMP   L_OFF
22         RJMP   L_ON
23 L_OFF: CBI     PORTB, 0     ;Turn LED0 off
24         RJMP   LOOP
25 L_ON:  SBI     PORTB, 0     ;Turn LED0 on
26         RJMP   LOOP
27 ;End of the main program
28
29 PIN_SETUP: ;Subroutine to set up pin modes
30         SBI     DDRB, 0     ;Set the pin PB0 as the output -> LED0
31         CBI     PORTB, 0    ;PB0=0
32         SBI     DDRB, 1     ;Set the pin PB0 as the output -> LED1
33         CBI     PORTB, 1    ;PB1=0
34         CBI     DDRD, 4     ;Set the pin PD4 as the input
35         SBI     PORTD, 4    ;Set pull-up resistor on PD4
36         RET
37
```

```

38  TIMER0_SETUP:  ;Subroutine to set up Timer0
39      LDI        R20, 0xB2
40      OUT        TCNT0, R20 ;Set the initial value of Timer0
41      LDI        R20, 0x00
42      OUT        TCCR0A, R20 ;Set the normal mode,
43      LDI        R20, 0x05
44      OUT        TCCR0B, R20 ;Pre-scaling 1024 and start the clock
45      RET
46
47  .ORG    0x200
48  T0_OV_ISR:      ;ISR of the Timer0 overflow interrupt
49      DEC        R21
50      BRNE       HERE
51      LDI        R21, 200 ;200 loops needed for delay 1 second
52      IN         R17, PORTB ;Read PORTB
53      LDI        R18, (1<<1) ;R18=0b00000010
54      EOR        R17, R18
55      OUT        PORTB, R17 ;Toggle PB1
56  HERE:  LDI        R18, 0xB2
57      OUT        TCNT0, R18 ;Set up Timer0 initial value
58      RETI

```

- 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.

https://drive.google.com/drive/folders/1u1haDM-cXIsXZIC9xzIBD9-gBf_8yR9h?usp=share_link

16	0x001E	TIMSK0 COMPS	Timer/Counter0 compare match 0
17	0x0020	TIMSK0 OVF	Timer/Counter0 overflow
18	0x0022	SDI STC	SDI serial transfer complete

The program automatically jumps to T0_OV_ISR to execute Task 2

```

1  .ORG 0x00    ; Set the origin for the reset vector
2  JMP MAIN    ; Jump to the main program
3
4  .ORG 0x200    ; Interrupt Vector Table for Timer0 Overflow Interrupt
5  JMP T0_OV_ISR ; Jump to the Timer0 Overflow Interrupt Service Routine
6
7  .ORG 0x100    ; Start of the main program
8  MAIN:
9      LDI R16, HIGH(RAMEND) ; Load high byte of RAMEND into R16
10     OUT SPH, R16          ; Set the high byte of the stack pointer
11     LDI R16, LOW(RAMEND)  ; Load low byte of RAMEND into R16
12     OUT SPL, R16          ; Set the low byte of the stack pointer
13
14     CALL PIN_SETUP        ; Call subroutine to set up pins
15
16     LDI R16, (1<<TOIE0)   ; Enable Timer0 overflow interrupt
17     STS TIMSK0, R16       ; Store in Timer Interrupt Mask Register
18     SEI                  ; Enable global interrupts
19
20     CALL TIMER0_SETUP      ; Call subroutine to set up Timer0
21     LDI R21, 200          ; Load 200 into R21 (Counter for 1-second toggling)
22
23     LOOP:
24         SBIC PIND, 4       ; Skip next instruction if bit 4 of PIND is clear (switch pressed)
25         RJMP L_OFF        ; If switch is not pressed, jump to turn LED0 off
26         RJMP L_ON         ; Otherwise, turn LED0 on
27
28     L_OFF:
29         CBI PORTB, 0       ; Clear bit 0 of PORTB (Turn LED0 off)
30         RJMP LOOP
31
32     L_ON:
33         SBI PORTB, 0       ; Set bit 0 of PORTB (Turn LED0 on)
34         RJMP LOOP
35
36     ; Subroutine to configure I/O pins
37     PIN_SETUP:
38         SBI DDRB, 0        ; Set PB0 as output (LED0)
39         CBI PORTB, 0       ; Ensure LED0 is off initially
40         SBI DDRB, 1        ; Set PB1 as output (LED1)
41         CBI PORTB, 1       ; Ensure LED1 is off initially
42         CBI DDRD, 4        ; Set PD4 as input (Switch)
43         SBI PORTD, 4       ; Enable pull-up resistor on PD4
44         RET                ; Return from subroutine
45
46     ; Subroutine to configure Timer0
47     TIMER0_SETUP:
48         LDI R20, 0xB2      ; Load initial value for Timer0
49         OUT TCNT0, R20     ; Set Timer0 initial value
50         LDI R20, 0x00      ; Set normal mode (TCCR0A = 0)
51         OUT TCCR0A, R20    ; Store in Timer Control Register A
52         LDI R20, 0x05      ; Set pre-scaler to 1024 and start Timer0 (TCCR0B = 0x05)
53         OUT TCCR0B, R20    ; Store in Timer Control Register B
54         RET                ; Return from subroutine
55
56     .ORG 0x200 ; Start of Timer0 Overflow Interrupt Service Routine
57     T0_OV_ISR:
58         DEC R21            ; Decrement overflow counter
59         BRNE HERE         ; If not zero, skip the toggling part
60         LDI R21, 200       ; Reset counter to 200 (1-second cycle)
61         IN R17, PORTB      ; Read current PORTB state
62         LDI R18, (1<<1)    ; Load mask for PB1 (LED1)
63         EOR R17, R18       ; Toggle PB1 state using XOR
64         OUT PORTB, R17     ; Output new state to PORTB
65
66     HERE:
67         LDI R18, 0xB2      ; Reload initial Timer0 value
68         OUT TCNT0, R18     ; Store in Timer0 counter register
69         RETI              ; Return from interrupt

```

Set up stack pointer

TIMSK0 is configured to enable the Timer0 overflow interrupt
SEI (Set global interrupt enable) allows interrupts to trigger

Monitor the switch
pressed / not pressed
LED ON / LED OFF

At the same time, Timer0 is counting up in the background independently.

When Timer0 overflows (reaches its max value 0xFF → 0x00), the Timer0 overflow interrupt (ISR) is triggered!

PB0 (LED0) and PB1 (LED1) are set as outputs

PD4 (switch input) with pull-up resistor

8-bit XOR doesn't toggle when 0010-0010! happens every 1 second.

ensures consistent 5 ms timing

Toggling part