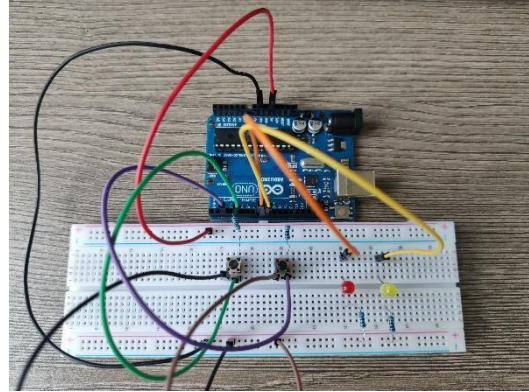
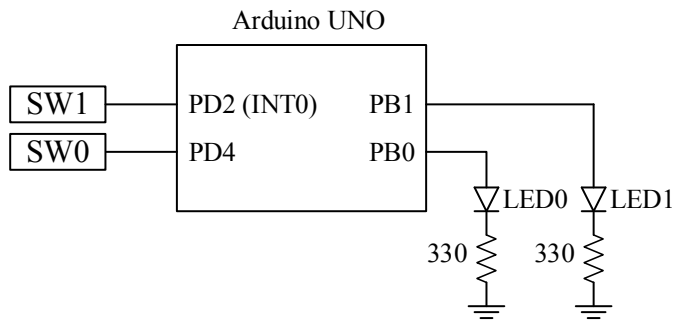


Exercise 2: (Using the external interrupt INT0). We have connected a circuit as shown below.



ନିମ୍ନ ଲିଖିତ
କାର୍ଯ୍ୟକାରୀ

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

ନା toggle
ନା ଲିଖିତ ନା ଲିଖିତ

- 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 (falling edge)

- b) Complete the following Assembly program.

```

1  .ORG    0x0
2      JMP      MAIN
3  .ORG    0x02
4      JMP      INT0_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<<INT0)
15         OUT      EIMSK, R16    ;Enable the external interrupt INT0
16         LDI      R16, (1<<ISC01)
17         STS      EICRA, R16   ;Set falling-edge trigger
18         SEI                      ;Enable the global interrupt
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          SBI      PORTD, 2     ;Set pull-up resistor on PD2 (INT0)
37          RET
38
39 .ORG    0x200
40 INT0_ISR: ;ISR of the INT0 external hardware
41          IN       R17, PORTB   ;Read PORTB
42          LDI      R18, (1<<1)  ;R18=0b00000010
43          EOR      R17, R18
44          OUT      PORTB, R17   ;Toggle PB1
45          RETI                  ;Return to the main program

```

Handwritten notes in the first code block:

- Line 14: $1 \ll 0$ (next to $(1 \ll \text{INT0})$)
- Line 15: 0×01 (next to EIMSK)
- Line 16: 0×01 (next to $(1 \ll \text{ISC01})$)
- Line 16: $1 \ll 1$ (next to $(1 \ll \text{ISC01})$)
- Line 16: 10 (circled, next to $(1 \ll \text{ISC01})$)

- 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.

https://drive.google.com/drive/folders/1u1haDM-cXIsXZlc9xzIBD9-gBf_8yR9h?usp=share_link

```

1      .ORG 0x0           ; Reset vector
2      JMP MAIN           ; Jump to the main program
3
4      .ORG 0x02          ; External Interrupt 0 vector
5      JMP INT0_ISR       ; Jump to interrupt service routine (ISR) for INT0
6
7      .ORG 0x100         ; Start of main program code
8  MAIN:  LDI R16, HIGH(RAMEND) ; Load high byte of RAMEND to R16
9         OUT SPH, R16        ; Set stack pointer high byte
10        LDI R16, LOW(RAMEND) ; Load low byte of RAMEND to R16
11        OUT SPL, R16        ; Set stack pointer low byte
12
13        CALL PIN_SETUP      ; Call subroutine to initialize I/O pins
14
15        LDI R16, (1<<INT0)  ; Enable external interrupt INT0 } Enables INTO in EIMSK
16        OUT EIMSK, R16      ; Set INT0 enable bit in EIMSK
17        LDI R16, (1<<ISC01) ; Set INT0 to trigger on falling edge } Configures it for falling edge trigger in EICRA
18        STS EICRA, R16     ; Store configuration in EICRA register
19        SEI                ; Enable global interrupts
20
21  LOOP:  SBIC PIND, 4        ; Skip next instruction if switch SW0 (PIND4) is clear
22        RJMP L_OFF         ; If not pressed, turn off LED0
23        RJMP L_ON          ; If pressed, turn on LED0
24
25  L_OFF: CBI PORTB, 0        ; Clear bit 0 of PORTB (turn off LED0)
26        RJMP LOOP          ; Repeat loop
27
28  L_ON:  SBI PORTB, 0        ; Set bit 0 of PORTB (turn on LED0)
29        RJMP LOOP          ; Repeat loop
30
31        ; Subroutine to configure I/O pins
32  PIN_SETUP:
33        SBI DDRB, 0         ; Set PORTB0 (LED0) as output
34        CBI PORTB, 0        ; Ensure LED0 is initially off
35        SBI DDRB, 1         ; Set PORTB1 (LED1) as output
36        CBI PORTB, 1        ; Ensure LED1 is initially off
37        CBI DDRD, 4         ; Set PIND4 (SW0) as input
38        SBI PORTD, 4        ; Enable pull-up resistor for SW0
39        SBI PORTD, 2        ; Enable pull-up resistor for SW1 (INT0)
40        RET                ; Return from subroutine
41
42        .ORG 0x200         ; Start of INT0 ISR (Interrupt Service Routine)
43  INT0_ISR:
44        IN R17, PORTB       ; Read current state of PORTB
45        LDI R18, (1<<1)    ; Load mask for LED1 (PORTB1)
46        EOR R17, R18        ; Toggle LED1 state using XOR
47        OUT PORTB, R17     ; Output new state to PORTB
48        RETI               ; Return from interrupt

```

Monitor the switch (SW0)

```

pressed / \ not pressed
  LED ON  \  LED OFF

```

When SW1 is pressed and released, an interrupt is triggered due to the falling edge

The microcontroller jumps to the interrupt service routine (ISR) at address 0x02