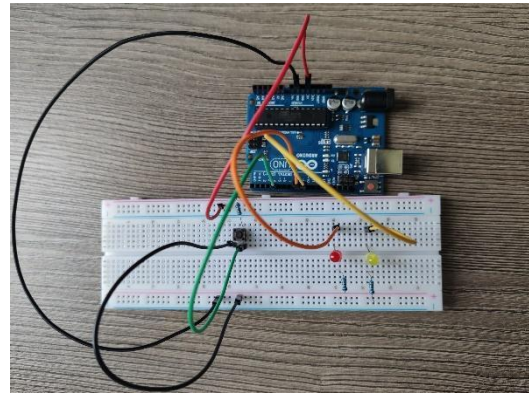
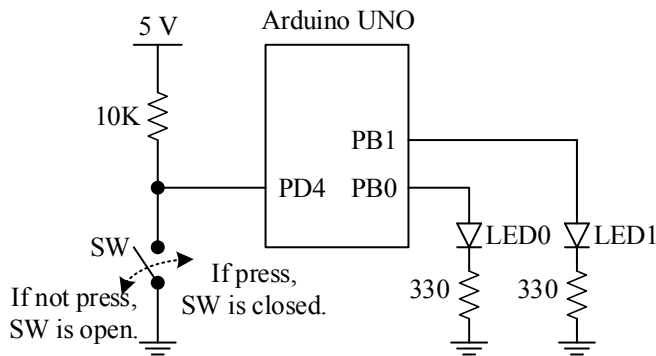


Exercise 3:

(Using Timer0 overflow interrupt). Similar to Exercise 1 in Lab 7, we will use the Timer0 overflow interrupt to help create a time delay but in a C program. We have connected a circuit as shown below.



We would like to write an C 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 (every 1 second). Recall the following related registers are set up as follows (from Exercise 1 in Lab 7):

- TCNT0 = 0xB2
- TCCR0A = 0x00
- TCCR0B = 0x05
- TIMSK0 = 0x01

Answer the following questions.

- a) Similar to the structure of the Assembly program shown in Exercise 1 in Lab 7, we have the following C program to fulfill the tasks required above. Please fill in the blanks.

```
1  #include <avr/io.h>
2  #include <avr/interrupt.h>
3                                     #define F_CPU 16000000UL
4                                     #include <util/delay.h>
5
6  unsigned char z = 200;
7
8  void PIN_SETUP() {                //A function to set up the pin modes
9      DDRB |= (1<<0);               //Set PB0 as an output -> LED0
10     PORTB &= ~(1<<0);              //Set PB0 = 0
11     DDRB |= (1<<1);               //Set PB1 as an output -> LED1
12     PORTB &= ~(1<<1);              //Set PB1 = 0
13     DDRD &= ~(1<<4);               //Set PD4 as an input -> SW
14     PORTD |= (1<<4);               //Pull-up resistor for PD4
15 }
```

```

15 void TIMER0_SETUP() { //A function to set TIMER0
16     TCNT0 = 0xB2; //Set initial value of Timer0
17     TCCR0A = 0x00; //Set the normal mode
18     TCCR0B = 0x05; //Pre-scaling = 1024, start the timer
19 }
20

21 int main() {
22     PIN_SETUP(); //Call the PIN_SETUP function
23     TIMER0_SETUP(); //Call the TIMER0_SETUP function
24     TIMSK0 = (1<<TOIE0); //Enable the Timer0 Overflow Interrupt
25     sei(); //Enable the global interrupt
26     while (1) {
27         if ( ~PIND & (1<<4) ) { //Check PD4=0?
28             PORTB |= (1<<0); //Yes, PB0=1 -> LED0 on
29         } else {
30             PORTB &= ~(1<<0); //No, PB0=0 -> LED0 off
31         }
32     }
33     return (0);
34 }
35

36 ISR (TIMER0_OVF_vect) { //ISR of Timer0 interrupt
37     z--; //Decrease z by 1
38     if (z==0) { //Check z=0
39         z = 200; //Yes, reset z = 200
40         PORTB ^= (1<<1); //and toggle PB1
41     }
42     TCNT0 = 0xB2; //Set TCNT0 = 0xB2
43 }

```

Note: A red box highlights the code between lines 31 and 32, which is `_delay_ms(100); // between line 31-32`. A red arrow points from this box to the closing brace of the while loop in line 31.

- b) Upload your C program to your Arduino UNO board. Take a video to demonstrate your result. Name it “Ex3” and submit to the Google Classroom.

https://drive.google.com/drive/folders/1zQ05sgDaikkTC0O4dgxmKRmrSGcYfmPN?usp=share_link

```

1  #include <avr/io.h>
2  #include <avr/interrupt.h>
3  #define F_CPU 16000000UL
4  #include <util/delay.h>
5
6  unsigned char z = 200; // Counter variable
7
8  void PIN_SETUP() {
9      DDRB |= (1<<0); // Set PB0 as an output -> LED0
10     PORTB &= ~(1<<0); // Ensure PB0 starts LOW (LED0 OFF)
11     DDRB |= (1<<1); // Set PB1 as an output -> LED1
12     PORTB &= ~(1<<1); // Ensure PB1 starts LOW (LED1 OFF)
13     DDRD &= ~(1<<4); // Set PD4 as an input -> SW
14     PORTD |= (1<<4); // Enable internal pull-up resistor for PD4 (default HIGH)
15 }
16
17 void TIMER0_SETUP() {
18     TCNT0 = 0xB2; // Load initial timer value to adjust timing
19     TCCR0A = 0x00; // Normal mode operation
20     TCCR0B = 0x05; // Prescaler = 1024, starts Timer0
21 }
22
23 int main() {
24     PIN_SETUP(); // Configure input/output pins
25     TIMER0_SETUP(); // Configure Timer0
26     TIMSK0 = (1<<TOIE0); // Enable Timer0 overflow interrupt
27     sei(); // Enable global interrupts
28
29     while (1) {
30         if (~PIND & (1<<4)) { // Check if switch (PD4) is pressed (active low)
31             PORTB |= (1<<0); // Turn LED0 ON (PB0=1)
32         } else {
33             PORTB &= ~(1<<0); // Turn LED0 OFF (PB0=0)
34         }
35         _delay_ms(100); // Debounce delay for the switch
36     }
37     return 0;
38 }
39
40 ISR(TIMER0_OVF_vect) { // Timer0 overflow interrupt service routine
41     z--; // Decrement counter variable
42     if (z == 0) { // If counter reaches zero
43         z = 200; // Reset counter for approximately 1-second delay
44         PORTB ^= (1<<1); // Toggle LED1 (PB1) state
45     }
46     TCNT0 = 0xB2; // Reload timer count for next cycle
47 }

```