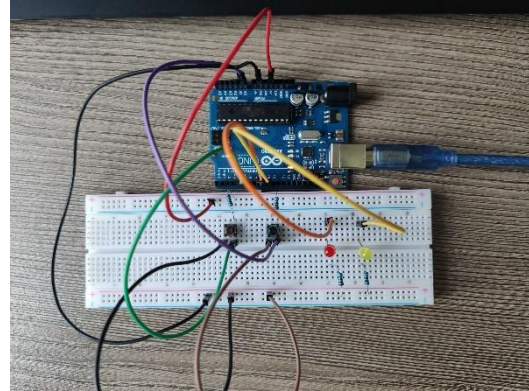
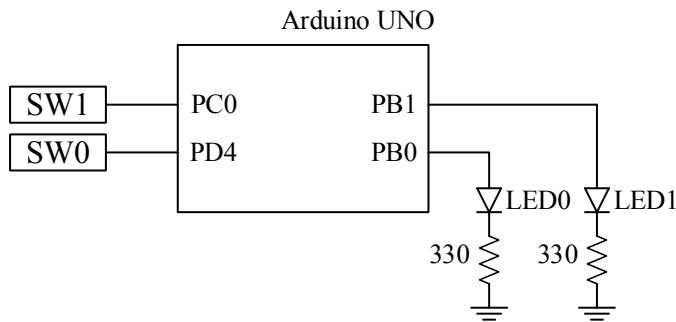


Exercise 5:

(Using the pin change interrupt). Similar to Exercise 3 in Lab 7, we will use the pin change interrupt but in a C program. 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 $\leftrightarrow$ off or off $\leftrightarrow$ 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). Recall the following related registers are set up as follows (from Exercise 3 in Lab 7):

- PCICR = 0x02
- PCMSK0 = -
- PCMSK1 = 0x01
- PCMSK2 = -

Answer the following questions.

- a) Similar to the structure of the Assembly program shown in Exercise 3 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
4  unsigned char z = 2;
5
6  void PIN_SETUP() {           //A function to set up the pin modes
7      DDRB |= (1<<0);          //Set PB0 as an output -> LED0
8      PORTB &= ~(1<<0);        //Set PB0 = 0
9      DDRB |= (1<<1);          //Set PB1 as an output -> LED1
10     PORTB &= ~(1<<1);         //Set PB1 = 0
11     DDRD &= ~(1<<4);          //Set PD4 as an input -> SW
12     PORTD |= (1<<4);          //Pull-up resistor for PD4
13     PORTC |= (1<<0);          //Pull-up resistor for PC0 (Pin Change Interrupt)
14 }
15

```

```

16 int main() {
17     PIN_SETUP();           //Call the PIN_SETUP function
18     PCICR = 0x02;          //Enable the PORTC Pin Change Interrupt
19     PCMSK1 = 0x01;         //Enable the interrupt from PC0
20     sei();                 //Enable the global interrupt
21     while (1) {
22         if ( ~PIND & (1<<4) ) { //Check PD4=0?
23             PORTB |= (1<<0);    //Yes, PB0=1 -> LED0 on
24         } else {
25             PORTB &= ~(1<<0);   //No, PB0=0 -> LED0 off
26         }
27     }
28     return (0);
29 }
30
31 ISR (PCINT1_vect) {        //ISR of INT0 interrupt
32     z--;
33     if (z==0) {
34         PORTB ^= (1<<1);      //and toggle PB1
35         z=2;
36     }
37 }

```

- b) Upload your C program to your Arduino UNO board. Take a video to demonstrate your result. Name it “Ex5” 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
4  unsigned char z = 2;
5
6  void PIN_SETUP() { // A function to set up the pin modes
7      DDRB |= (1<<0); // Set PB0 as an output -> LED0
8      PORTB &= ~(1<<0); // Set PB0 = 0
9      DDRB |= (1<<1); // Set PB1 as an output -> LED1
10     PORTB &= ~(1<<1); // Set PB1 = 0
11     DDRD &= ~(1<<4); // Set PD4 as an input -> SW
12     PORTD |= (1<<4); // Pull-up resistor for PD4
13     PORTC |= (1<<0); // Pull-up resistor for PC0 (Pin Change Interrupt)
14 }
15
16 int main() {
17     PIN_SETUP(); // Call the PIN_SETUP function
18     PCICR = 0x02; // Enable the PORTC Pin Change Interrupt
19     PCMSK1 = 0x01; // Enable the interrupt from PC0
20     sei(); // Enable the global interrupt
21
22     while (1) {
23         if (~PIND & (1<<4)) { // Check PD4=0?
24             PORTB |= (1<<0); // Yes, PB0=1 -> LED0 on
25         } else {
26             PORTB &= ~(1<<0); // No, PB0=0 -> LED0 off
27         }
28     }
29
30     return 0;
31 }
32
33 ISR (PCINT1_vect) { // ISR of INT0 interrupt
34     z--;
35     if (z == 0) {
36         PORTB ^= (1<<1); // and toggle PB1
37         z = 2;
38     }
39 }

```