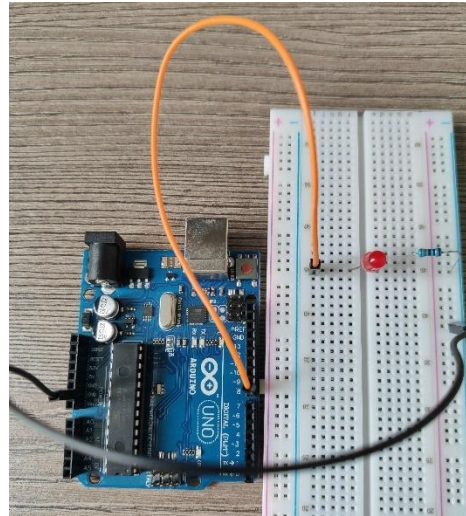
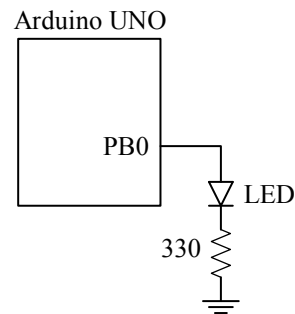


Exercise 1:

In this exercise, we will connect a circuit and write a C program to turn on and off an LED every 1 second. Connect the circuit as shown below.



Write the C program as shown below which will turn on/off the LED every 1 second and upload to the Arduino UNO. Note that the instructor uses this exercise as demonstration, please see the video.

```
1  #include <avr/io.h>
2  #define F_CPU 16000000UL
3  #include <util/delay.h>
4
5  void PIN_SETUP() {           //A function to set up the pin modes
6      DDRB |= (1<<0);          //Set PB0 as an output -> LED
7      PORTB &= ~(1<<0);        //Set PB0 = 0
8  }
9
10 int main() {
11     PIN_SETUP();              //Call the PIN_SETUP function
12     while (1) {               //Repeat the while loop forever
13         PORTB |= (1<<0);      //Turn the LED on
14         _delay_ms(1000);      //for 1 second
15         PORTB &= ~(1<<0);    //Turn the LED off
16         _delay_ms(1000);      //for 1 second
17     }
18     return (0);
19 }
```

Take a video to demonstrate your result. Name it “Ex1” and submit to the Google Classroom.

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

```

1  #include <avr/io.h>
2  #define F_CPU 16000000UL // Define CPU clock speed as 16MHz
3  #include <util/delay.h>
4
5  void PIN_SETUP() {
6      DDRB |= (1<<0); // Set PB0 (digital pin 8 on Arduino UNO) as an output
7      PORTB &= ~(1<<0); // Ensure PB0 is LOW initially (LED off)
8  }
9
10 int main() {
11     PIN_SETUP(); // Call function to set up pin modes
12
13     while (1) { // Infinite loop to keep the program running
14         PORTB |= (1<<0); // Turn ON LED by setting PB0 HIGH
15         _delay_ms(1000); // Wait for 1000 milliseconds (1 second)
16         PORTB &= ~(1<<0); // Turn OFF LED by setting PB0 LOW
17         _delay_ms(1000); // Wait for 1000 milliseconds (1 second)
18     }
19     return 0; // Program should never reach here
20 }

```

Bitwise OR (|=) – Setting a Bit

- Used to **set** (turn ON) a specific bit while keeping other bits unchanged.
- Example:

```

c                                                                    Copy Edit
PORTB |= (1 << 0); // Set bit 0 (turn ON PB0)

```

- `(1 << 0)` shifts `1` to position `0`, resulting in `0b00000001`.
- `|=` ensures only bit 0 is set, leaving other bits unchanged.

Bitwise AND with NOT (&= ~) – Clearing a Bit

- Used to **clear** (turn OFF) a specific bit while keeping other bits unchanged.
- Example:

```

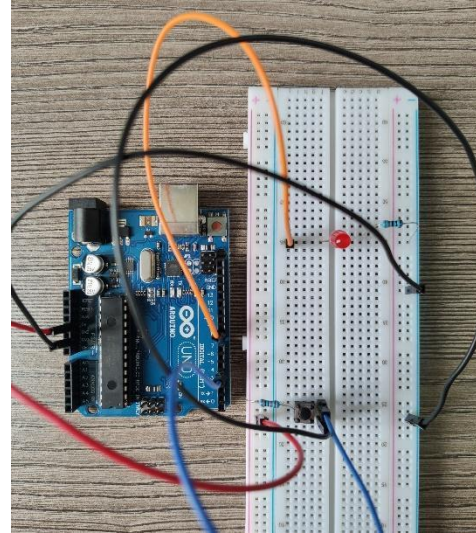
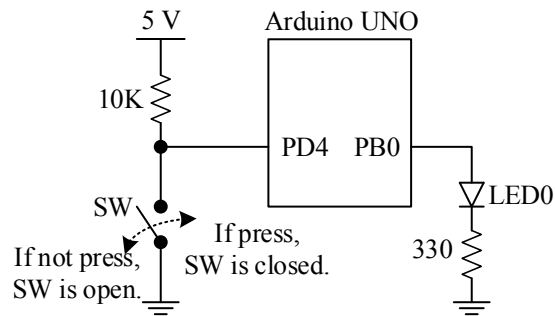
c                                                                    Copy Edit
PORTB &= ~(1 << 0); // Clear bit 0 (turn OFF PB0)

```

- `(1 << 0)` creates `0b00000001`.
- `~(1 << 0)` inverts it to `0b11111110`.
- `&=` ensures only bit 0 is cleared, while other bits remain unchanged.

Exercise 2:

In this exercise, we will connect a circuit and write a C program to turn on and off an LED by pressing a switch. Connect the circuit as shown below.



- a) Please fill in the blanks to complete the following C program in order to fulfill the tasks required above.

```
1  #include <avr/io.h>
2
3  void PIN_SETUP() {           //A function to set up the pin modes
4      DDRB |= (1<<0);          //Set PB0 as an output -> LED0
5      PORTB &= ~(1<<0);        //Set PB0 = 0
6      DDRD &= ~(1<<4);         //Set PD4 as an input -> SW
7      PORTD |= (1<<4);         //Pull-up resistor for PD4
8  }
9
10 int main() {
11     PIN_SETUP();              //Call the PIN_SETUP function
12     while (1) {               //Repeat forever
13         if ( ~PIND & (1<<4) ) { //Check PD4=0 (press the switch)?
14             PORTB |= (1<<0);   //Yes, PB0=1 -> LED0 on
15         } else {
16             PORTB &= ~(1<<0);  //No, PB0=0 -> LED0 off
17         }
18     }
19     return (0);
20 }
```

- b) Upload your C 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/1zQ05sgDaikkTC0O4dgxmKRmrSGcYfmPN?usp=share_link

```

1  #include <avr/io.h>
2
3  void PIN_SETUP() {
4      DDRB |= (1<<0); // Set PB0 (digital pin 8) as an output
5      PORTB &= ~(1<<0); // Ensure PB0 is LOW initially (LED OFF)
6
7      DDRD &= ~(1<<4); // Set PD4 (digital pin 4) as an input
8      PORTD |= (1<<4); // Enable internal pull-up resistor on PD4 (default HIGH)
9  }
10
11 int main() {
12     PIN_SETUP();
13
14     while (1) { // Infinite loop to continuously check button state
15         if(~PIND & (1<<4)) { // Check if button (PD4) is pressed (logic LOW)
16             PORTB |= (1<<0); // Turn ON LED (set PB0 HIGH)
17         } else {
18             PORTB &= ~(1<<0); // Turn OFF LED (set PB0 LOW)
19         }
20     }
21     return (0); // Program should never reach here
22 }

```