Chaniporn Nerunchorn 6322771450
Ornnicha Phueakstri 6322738837

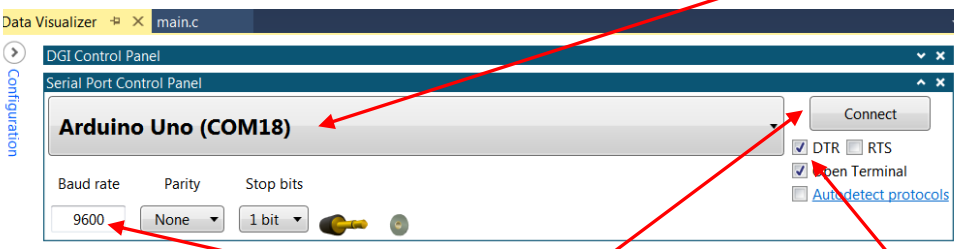
SIIT SINCE 1992	**School of ICT** **Sirindhorn International Institute of Technology**
CSS332 Microcontrollers and Applications	**Lab 9: AVR Serial Port and ADC Programming in C**

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.)

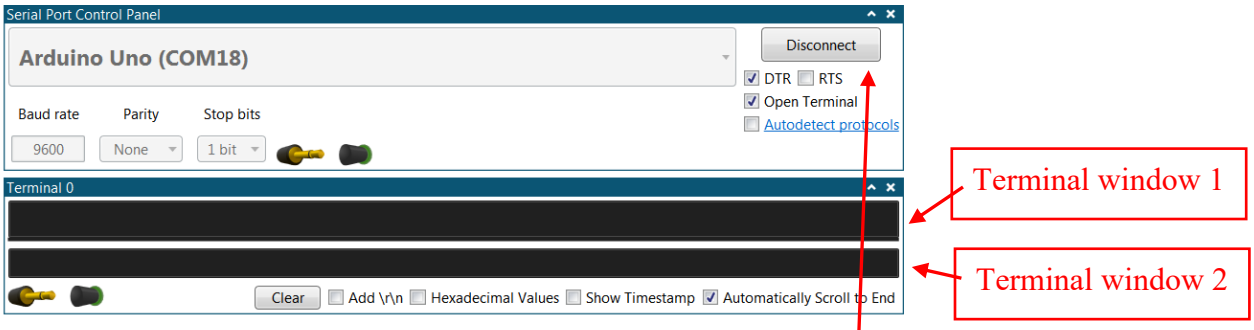
To Open the Terminal in Atmel Studio

In **Exercises 1 – 3**, you will need to open the Terminal in the Atmel Studio 7 to see the result. When you do these exercises, please follow these steps.

i) Connect Arduino UNO to the computer.
ii) Choose Tools > Data Virtualizer.
iii) We will see this window (now Arduino UNO is connecting to COM18)



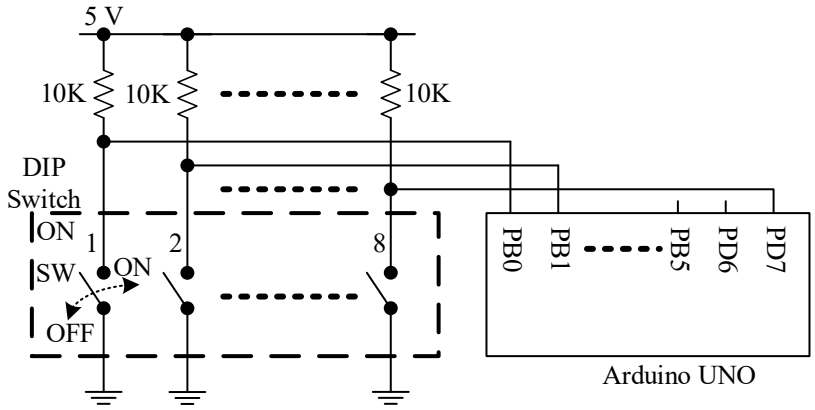
iv) Choose the baud rate to 9600 and click the Connect button. Choose DTR and Open Terminal (the other are left unchecked). Two Terminal windows will appear.



- Terminal window 1 will show the data sent from the microcontroller (microcontroller is working as the transmitter). See this window when working on Exercises 1.
- We type a letter/number on Terminal window 2 to send this data to the microcontroller (microcontroller is working as the receiver). Use this window in Exercise 2.

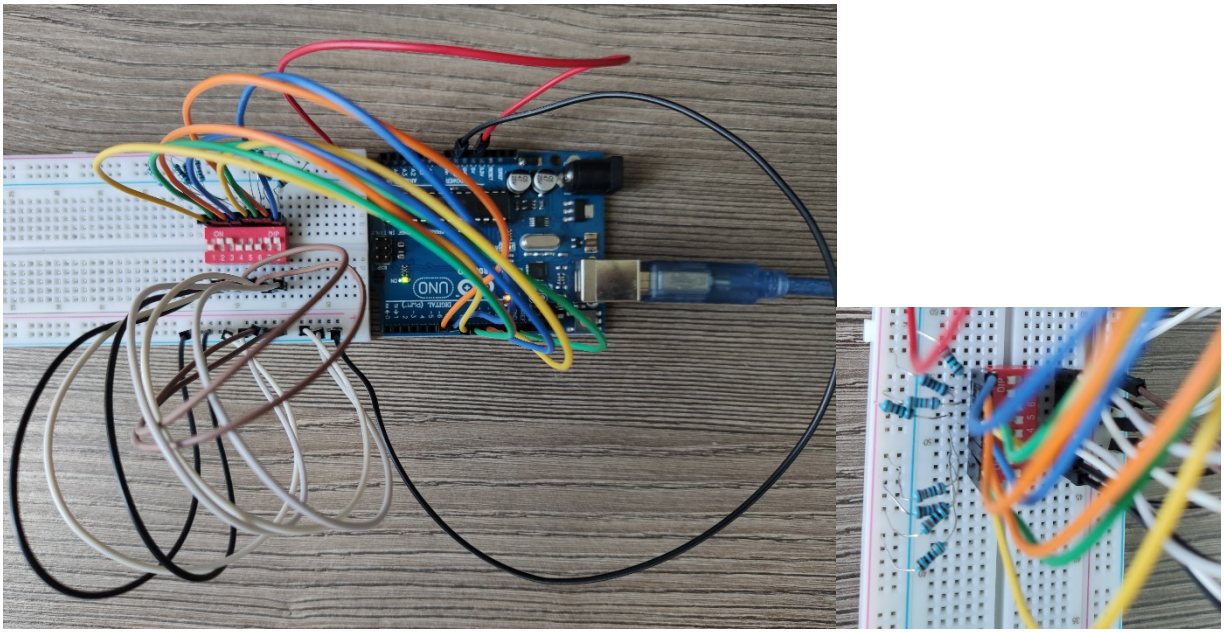
v) When we finish each exercise using Terminal, we must click the Disconnect button to end the communication between Arduino UNO and Terminal; otherwise, we might have the problem on uploading the next program to Arduino UNO.

Exercise 1: (Serial Port Programming) Connect a circuit as shown below. The DIP switches (1 – 8) are connected to the pins PB0, PB1, …, PB5, and *PD6, PD7*, respectively.



Pin Connection

DIP Switch	ATmega 328 Pin	Arduino UNO Pin
1	PB0	8
2	PB1	9
3	PB2	10
4	PB3	11
5	PB4	12
6	PB5	13
7	PD6	6
8	PD7	7



Note that "switch ON" is equivalent to the bit "0" and "switch OFF" is equivalent to the bit "1". The following C program will set the USART in the transmitter mode. The microcontroller sends data out via the Tx and Rx pins. Here, the data will be sent to the computer via the USB port and shown on the Terminal of the Atmel Studio 7.

The program reads the inputs from the pins PB0, PB1, …, PB5, PD6, and PD7 (as shown on the circuit above) and displays on the Terminal, every 5 seconds. These inputs represent an 8-bit binary number (where PD7 is the MSB and PB0 is LSB). The program will display this binary number on the Terminal (note that the binary number here will be interpreted as an ASCII code. For example, the binary number 0b00110101 will be shown as a number 5 on the Terminal).

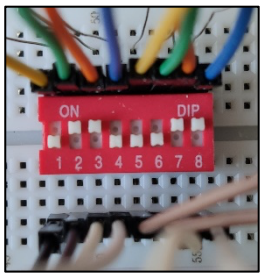
a) Complete this C program.

```
1   #include <avr/io.h>
2   #define F_CPU 16000000UL
3   #include <util/delay.h>
4
5   void usart_init (void) {
6       UCSR0B=(1<<TXEN0);                  //Enable USART transmitter
7       UCSR0C=(1<<UCSZ01)|(1<<UCSZ00); //Async, 8 bits,
8       UBRR0L= 103                         //Baud rate = 9600
9   }
10
```

```
11  void pin_setup (void) { //Set input pins
12      DDRB = 0x00;
13      PORTB = 0xFF;
14      DDRD = 0x00;
15      PORTD = 0xFF;
16  }
17
18  int main(void) {
19      usart_init();
20      pin_setup();
21      while (1) {
22          while (!(UCSR0A&(1<<UDRE0)));          //Wait until UDRE0 flag = 1
23          UDR0= (PIND & 0xC0) | (PINB & 0b00111111);//Store inputs in UDR0
24          _delay_ms(5000);                       //Wait for 5 seconds
25      }
26      return 0;
27  }
```

opposite

b) Upload the code to the Arduino UNO board. Open the Terminal in the Atmel Studio 7 (steps explained on the first page). Set the DIP switch to represent the binary number 0b00111001 (the ASCII code of the decimal number ***9***).



Do you see the number ***9*** on the Terminal? Note that "switch ON" is equivalent to the bit "0" and "switch OFF" is equivalent to the bit "1".

Yes [10011100 → 0b111001 = 9]

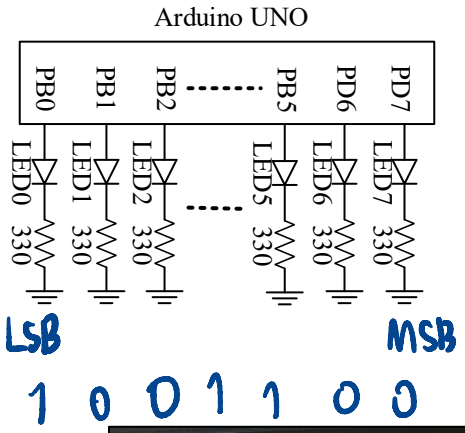
c) Repeat the Step b) above and setting the DIP switch such that the capital letter ***A*** is shown on the Terminal. Record the video to demonstrate your result, name it "Ex1", and submit to the Google Classroom.

What binary number do we have to set (using the DIP switches) such that the capital letter ***A*** shown on the Terminal?

0b01000001

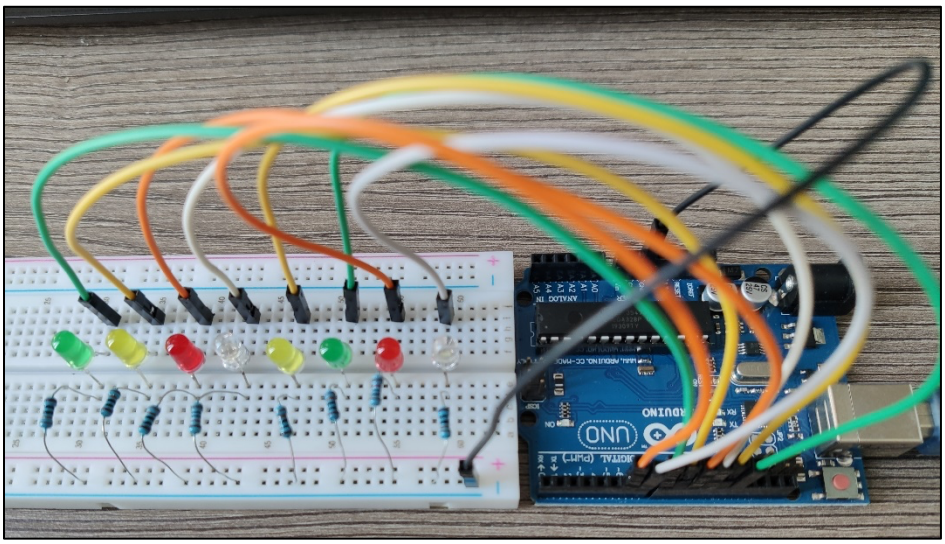
Press the Disconnect button on the terminal to end the communication between Arduino UNO.

Exercise 2: (Serial Port Programming) Connect a circuit as shown below. Eight LEDs are connected to the pins PB0, PB1, …, PB5, and *PD6, PD7* and are called as LED0, LED1, …, LED7, respectively. Let LED7 represent the MSB and LED0 represent the LSB of an 8-bit binary number.



Pin Connection

LED	ATmega 328 Pin	Arduino UNO Pin
0	PB0	8
1	PB1	9
2	PB2	10
3	PB3	11
4	PB4	12
5	PB5	13
6	PD6	6
7	PD7	7



The following C program will set the USART in the receiver mode. The microcontroller receives data via the Tx and Rx pins. Here, the data will be sent from the computer via the USB port to the microcontroller. Any data typed in the Terminal will be sent to the microcontroller.

The program receives an input from the Terminal of the Atmel Studio 7 by typing a number or character and pressing the Enter key. This input will be interpreted as an ASCII code and shown on the eight LEDs above.

a) Complete this C program.

```
1   #include <avr/io.h>
2
3   void usart_init(void) {
4       UCSR0B = (1<<RXEN0);                //Enable USART receiver
5       UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);   //Async, 8 bits,
6       UBRR0L = 103;                       //Baud rate = 9600
7   }
8
9   void pin_setup(void) {  //Set output pins
10      DDRB = 0xFF;
11      PORTB = 0x00;
12      DDRD = 0xFF;
13      PORTD =0x00;
14  }
15
16  int main(void) {
17      pin_setup();
18      usart_init();
19      while (1) {          //Receive the input and show it on output pins
20          while (!(UCSR0A&(1<<RXC0))); //Wait until RXC0 flag = 1
21          PORTB = (UDR0 & 0x3F);       //Show the input on pins PB0 - PB5
22          PORTD = (UDR0 & 0xC0);       //Show the input on pins PD6 - PD7
23      }
24      return 0;
25  }
```

b) Upload the code to the Arduino UNO board. Open the Terminal in the Atmel Studio 7 (steps explained on the first page). On the Terminal, type the number 9 and press the Enter button.

- Which LEDs are on and off?
- If "LED on" represents "1" and "LED off" represents "0", what is the binary number represented by these 8 LEDs?

0b 00 111001 → 9

c) We would like to turn on (to represent "1") the following LEDs: LED6, LED5, LED2, and LED0, while the other LEDs are off. What letter/number should we type on the Terminal? Record the video to demonstrate the work, name it "Ex2", and submit to the Google Classroom.

0b01100101 → e

Press the Disconnect button on the terminal to end the communication between Arduino UNO.

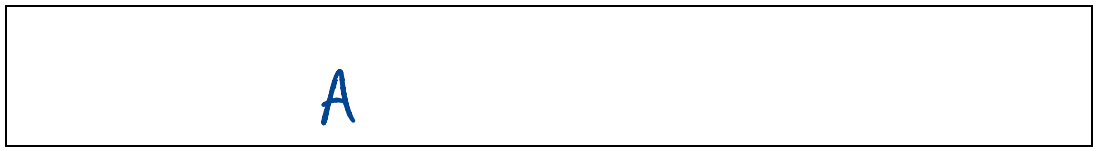
LED 0 1 2 3 4 5 6 7
1 0 1 0 0 1 1 0
⇌ 1 1 0 0 1 0 1 → ASCII

Exercise 3: (Serial Port Programming) The C program below will set the USART in both transmitter and receiver modes. The program receives an input letter from Terminal, convert it into the corresponding capital letter, and, then, show the result to Terminal. (Remark: no circuit connection.)

a) Complete this C program.

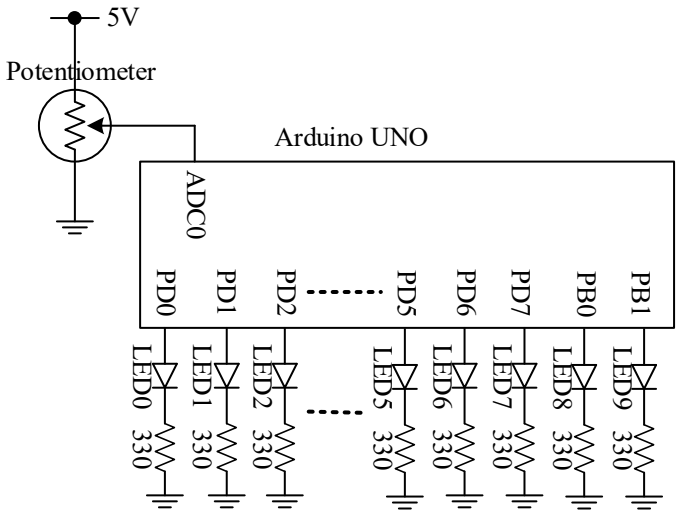
```
#include <avr/io.h>

int main(void) {
    unsigned char ch;

    UCSR0B=(1<<TXEN0)|(1<<RXEN0); //USART setup
    UCSR0C=(1<<UCSZ01)|(1<<UCSZ00);
    UBRR0L=103;

    while (1) {
        while (!(UCSR0A&(1<<RXC0))); //Wait for the input from Terminal
        ch= UDR0
        if (ch>='a'&& ch<='z') {
            ch+=('A'-'a'); //update what stored in ch to be its capital letter
            while (!(UCSR0A&(1<<UDRE0))); //Wait to send ch to Terminal
            UDR0= ch
        }
    }
    return 0;
}
```

b) Connect the Arduino UNO board to the computer. Upload the code to the Arduino UNO board. Open the Terminal in the Atmel Studio 7 (steps explained on the first page). On the Terminal, if we type the small letter ***a*** and press the Enter key, as a return, what will be shown on the Terminal? Record the video to demonstrate your work, name it "Ex3", and submit to the Google Classroom.



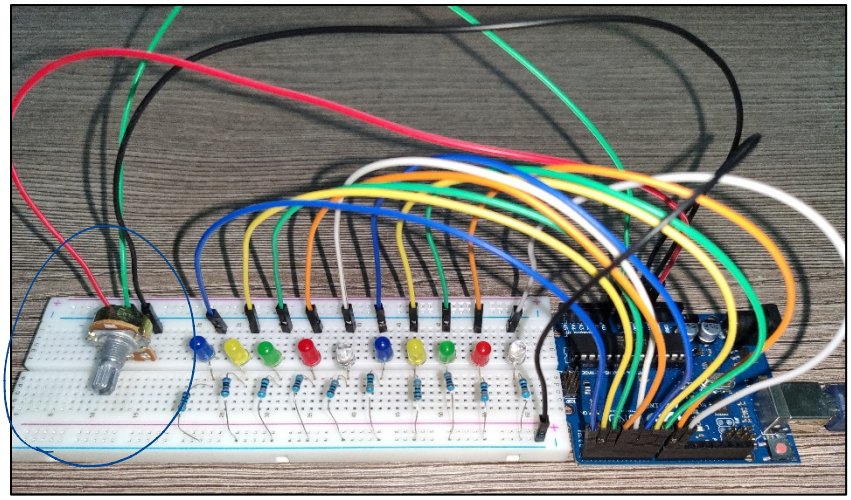
Exercise 4: (ADC Programming) Connect a circuit as shown below. Ten LEDs are connected to the pins PD0, PD1, …, PD7 and *PB0, PB1* are called as LED0, LED1, …, LED9, respectively. Let LED9 (pin PB1) represent the MSB and LED0 (pin PD0) represent the LSB of an 10-bit binary number. A potentiometer (5KΩ) is connected to the pin ADC0 (i.e., the pin A0 on the Arduino UNO board). With the C program below, when rotating the knob of the potentiometer, we will see the changes of the LEDs (on/off) which represents the 10-bit binary number (equivalently, voltage across the pin ADC0).

Alert! Disconnect the pins PD0 and PD1 before uploading the code and connect them back when you finished uploading the code. Since connecting PD0 and PD1 during uploading the code will cause an uploading error.



Pin Connection

LED	ATmega 328 Pin	Arduino UNO Pin
0	PD0	0
1	PD1	1
2	PD2	2
3	PD3	3
4	PD4	4
5	PD5	5
6	PD6	6
7	PD7	7
8	PB0	8
9	PB1	9



The following C program will read the input voltage (which is the voltage across the potentiometer) via the pin ADC0 and convert it to a 10-bit binary number. This 10-bit binary number will be shown on the LEDs (LED0 – LED9).

a) Complete this C program.

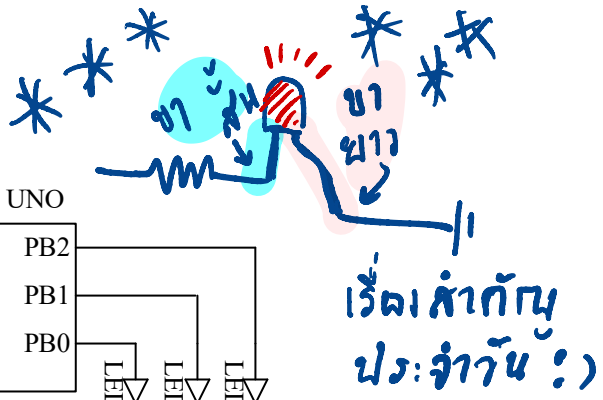
```c
#include <avr/io.h>
#define F_CPU 16000000UL
#include <util/delay.h>

void pin_setup(void) { //Set the pin modes
    DDRC = 0x00;
    DDRB = 0xFF;
    DDRD = 0xFF;
    PORTB = 0x00;
    PORTD = 0x00;
}

void ADC_setup(void) {
    ADCSRA = 0x87; //Enable the ADC and use CK/128
    ADMUX = 0x40 //Vref = AVCC and use ADC0 pin, right-justified
}

int main(void) {
    pin_setup();
    ADC_setup();
    while (1) {
        ADCSRA |= (1<<ADSC);    //Start the ADC conversion
        while ((ADCSRA&(1<<ADIF)) == 0);  //Wait for the conversion
        PORTD = ADCL //Last 8 bits shown on pins D
        PORTB = ADCH //First 8 bits shown on pins B
        _delay_ms(100);
    }
    return 0;
}
```

P. 39 0x40 = AVCC = 5V.

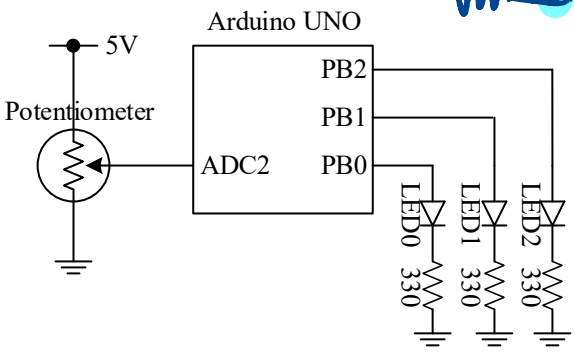
Analog to Digital Converter

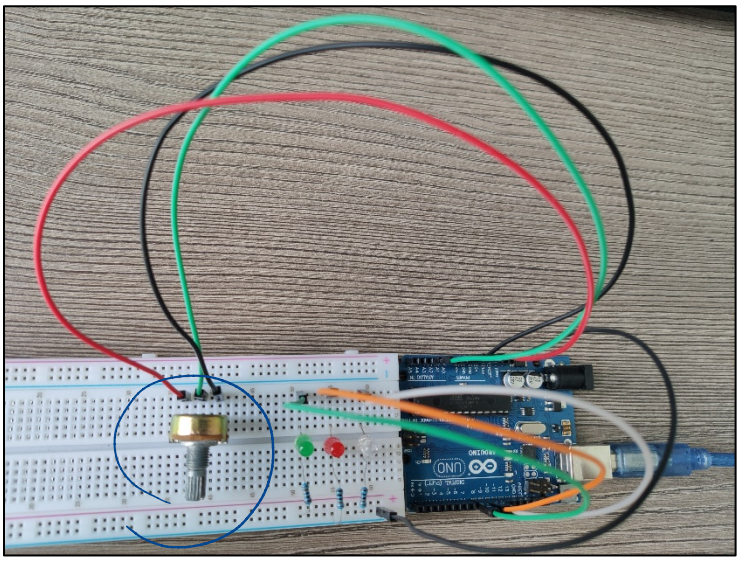
b) Upload the code to the Arduino UNO board (disconnect the pins PD0 and PD1 to the circuit before uploading the code and connect them back after finishing). Rotate the knob of the potentiometer such that the LEDs are on/off to represent this 10-bit binary number: 0b1110100111, where "LED on" represents "1" and "LED off" represents "0". Record the video to demonstrate your work, name it "Ex4", and submit to the Google Classroom. Here, what should be the voltage across the pin ADC0?

$\frac{5V}{1024 \text{ pattern}} = 0.00488 \times 935$
$= 4.56V$



Exercise 5: Connect the circuit as shown below.





Note that the pin ADC2 (of the ATmega328) is the pin A2 on the Arduino UNO board. Write the C program such that:

- if the voltage across the pin ADC2 is lower than or equal to 1.5V, the LED connected to the pin PB0 is on (the other LEDs are off),
- if the voltage across the pin ADC2 is between 1.5V and 3V, the LED connected to the pin PB1 is on (the other LEDs are off),
- if the voltage across the pin ADC2 is higher than or equal to 3V, the LED connected to the pin PB2 is on (the other LEDs are off).

$$1.5\,V = \frac{5V}{1024} \times x$$

$$x = 307.2 \approx 308$$

$$3V = \frac{5V}{1024} \times x$$

$$x = 614.2 \approx 615$$

a) Complete this C program.

```
#include <avr/io.h>

void pin_setup(void) { //Set the pin modes
    DDRC = 0x00;
    DDRB = 0xFF;
    PORTB = 0x00;
}

void ADC_setup(void) {
    ADCSRA = 0x87;  //Enable the ADC and use CK/128
    ADMUX = 0x42;   //Vref = AVCC = 5V and use ADC2 pin, right-justified
}

int main(void) {
    pin_setup();
    ADC_setup();
    while (1) {
        ADCSRA |= (1<<ADSC);    //Start the ADC conversion
        while ((ADCSRA&(1<<ADIF)) == 0);  //Wait for the conversion
        if (ADC <= 308) { //If ADC < or = 1.5V
            PORTB = 0x01; // Set PB0=1
        } else if ( (ADC > 308 ) && (ADC < 615 ) ) {//If 1.5V < ADC < 3V
            PORTB = 0x02; // Set PB1=1
        } else { //If ADC > or = 3V
            PORTB = 0x04; // Set PB2=1
        }
    }
    return 0;
}
```

0x40 ADC0 → 0x42 ADC2

b) Upload the code to the Arduino UNO board. Rotating the knob of the potentiometer will turn on and off the LEDs. Record the video to demonstrate your work, name it "Ex5", and submit to the Google Classroom.