Lab assignment – 02-leds

1. LED example. Submit:
    - Tables for DDRB, PORTB, and their combination,

DDRB	Description
0	Input pin
1	Output pin

PORTB	Description
0	Output low value
1	Output high value

DDRB	PORTB	Direction	Internal pull-up resistor	Description
0	0	input	no	Tri-state, high-impedance
0	1	input	depends on PUD	pull-up resistor activation
1	0	output	no	output low
1	1	output	no	output high

PUD ... can be set to disable all pull-ups in all ports (0 -> pull up, 1 -> Hi-z)

- Table with input/output pins available on ATmega328P,

Port	Pin	Input/output usage?
A	x	Microcontroller ATmega328P does not contain port A
B	0	Yes (Arduino pin 8)
	1	Yes (Arduino pin ~9)
	2	Yes (Arduino pin ~10)
	3	Yes (Arduino pin ~11)
	4	Yes (Arduino pin 12)
	5	Yes (Arduino pin 13)
	6	No ... crystal 16 MHz
	7	No ... crystal 16 MHz
C	0	Yes (Arduino pin A0 ... analog)
	1	Yes (Arduino pin A1)
	2	Yes (Arduino pin A2)
	3	Yes (Arduino pin A3)
	4	Yes (Arduino pin A4)
	5	Yes (Arduino pin A5)
	6	No ... reset
	7	Microcontroller ATmega328P does not contain port PC7
D	0	Yes (Arduino pin RX<-0)
	1	Yes (Arduino pin TX->0)
	2	Yes (Arduino pin 2)
	3	Yes (Arduino pin ~3)
	4	Yes (Arduino pin 4)
	5	Yes (Arduino pin ~5)
	6	Yes (Arduino pin ~6)
	7	Yes (Arduino pin 7)

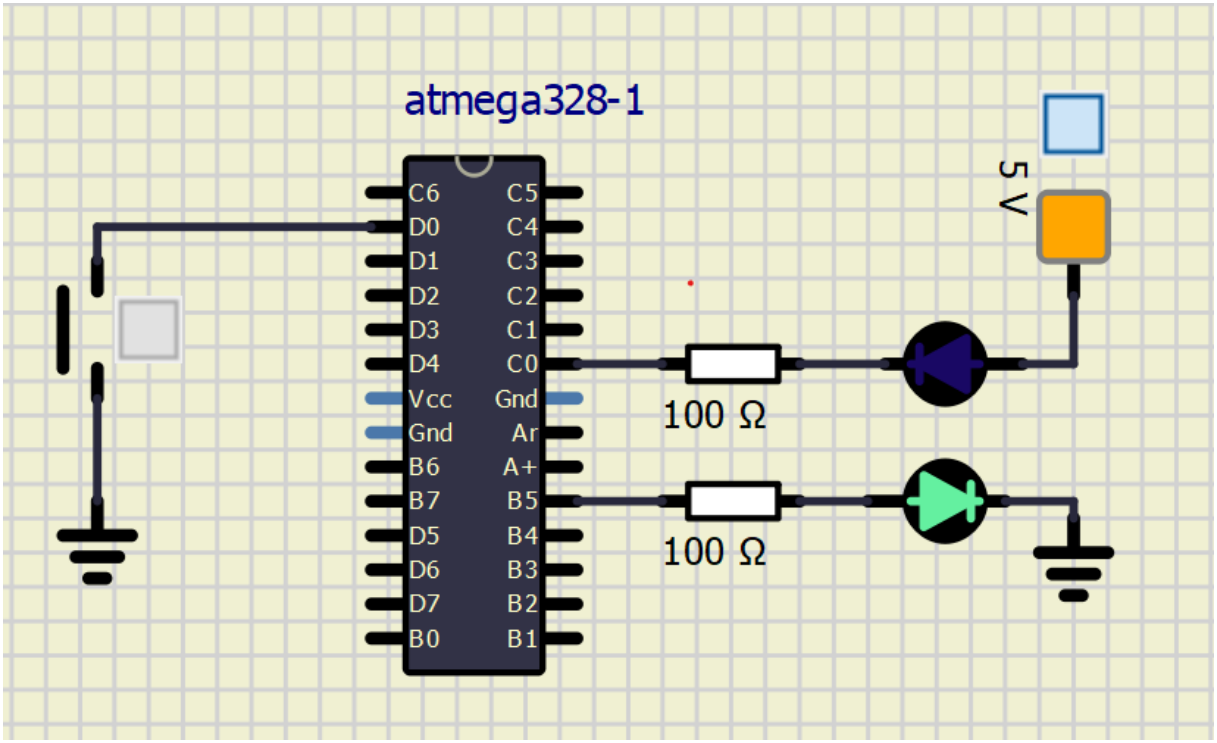
~ ... a PWM (Pulse-width modulation) signal can be generated on these pins

TX->1

- C code with two LEDs and a push button,

```
/* Defines -----------------------------------------
#define LED_GREEN   PB5     // AVR pin where green LED
#define LED_RED     PC0
#define BTN     PD0         // button
#define BLINK_DELAY 250
#ifndef F_CPU
#define F_CPU 16000000      // CPU frequency in Hz requ
#endif

/* Includes ----------------------------------------
#include <util/delay.h>     // Functions for busy-wait
#include <avr/io.h>         // AVR device-specific IO c

/* Functions ---------------------------------------
/**
 * Main function where the program execution begins. To
 * when a push button is pressed.
 */
int main(void)
{
    /* GREEN LED */
    // Set pin as output in Data Direction Register...
    DDRB = DDRB | (1<<LED_GREEN);
    // ...and turn LED off in Data Register
    PORTB = PORTB & ~(1<<LED_GREEN);

    /* second (RED) LED */
    DDRC = DDRC | (1<<LED_RED);
    PORTC = PORTC & ~(1<<LED_RED);

    /* BUTTON */
    DDRD = DDRD & ~(1<<BTN);
    PORTD = PORTD | (1<<BTN);

    // Infinite loop
    while (1)

    {
        // Pause several milliseconds
        _delay_ms(BLINK_DELAY);

        if(bit_is_clear(PIND, BTN))
        {
            PORTB = PORTB ^ (1<<LED_GREEN);
            PORTC = PORTC ^ (1<<LED_RED);
        }
    }

    // Will never reach this
    return 0;

}
```

- Screenshot of SimulIDE circuit.



2. Knight Rider application. Submit:

   - C code

```c
void fce (int i)
{
    switch(i)
    {
        case 0:
            PORTC = PORTC ^ (1<<LED_RED0);
            break;
        case 1:
            PORTC = PORTC ^ (1<<LED_RED0);
            PORTC = PORTC ^ (1<<LED_RED1);
            break;
        case 2:
            PORTC = PORTC ^ (1<<LED_RED1);
            PORTC = PORTC ^ (1<<LED_RED2);
            break;
        case 3:
            PORTC = PORTC ^ (1<<LED_RED2);
            PORTC = PORTC ^ (1<<LED_RED3);
            break;
        case 4:
            PORTC = PORTC ^ (1<<LED_RED3);
            PORTC = PORTC ^ (1<<LED_RED4);
            break;
        default:
            PORTC = PORTC ^ (1<<LED_RED0);
    }
}
```

```c
int main(void)
{
    /* RED LEDs */
    DDRC = DDRC | (1<<LED_RED0);
    PORTC = PORTC & ~(1<<LED_RED0);
    DDRC = DDRC | (1<<LED_RED1);
    PORTC = PORTC & ~(1<<LED_RED1);
    DDRC = DDRC | (1<<LED_RED2);
    PORTC = PORTC & ~(1<<LED_RED2);
    DDRC = DDRC | (1<<LED_RED3);
    PORTC = PORTC & ~(1<<LED_RED3);
    DDRC = DDRC | (1<<LED_RED4);
    PORTC = PORTC & ~(1<<LED_RED4);

    /* BUTTON */
    DDRD = DDRD & ~(1<<BTN);
    PORTD = PORTD | (1<<BTN);
    // Infinite loop
    while (1)
    {
        if(bit_is_clear(PIND, BTN))
        {
            for(int i = 0; i <= 4; i++)
            {
                _delay_ms(BLINK_DELAY);
                fce(i);
            }
            for(int i = 4; i >= 0; i--)
            {
                _delay_ms(BLINK_DELAY);
                fce(i);
            }
        }
    }
    return 0;
}
```

Fce -> knightRider

Int -> uint8