

Link to my repository in GitHub

<https://github.com/Konecny343/Digital-electronics-2>

Calculate LED resistor value for typical red and blue LEDs

$$R_{red_led} = \frac{V_{SUPPLY} - V_{LED}}{I} = \frac{5 - 1,8}{20 \cdot 10^{-3}} = \mathbf{160 \, \Omega}$$

$$R_{blue_led} = \frac{V_{SUPPLY} - V_{LED}}{I} = \frac{5 - 3,6}{20 \cdot 10^{-3}} = \mathbf{70 \, \Omega}$$

LED color	Supply voltage	LED current	LED voltage	Resistor value
red	5 V	20 mA	1,8 V	160 Ω
blue	5 V	20 mA	3,6 V	70 Ω

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	Yes	Tri-state, high-impedance
1	0	Output	No	Output low
1	1	Output	No	Output high

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
	7	No
C	0	Yes (Arduino pin A0)
	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
	7	No
D	0	Yes (Arduino pin RX <- 0)
	1	Yes (Arduino pin TX <- 1)
	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)

C code with two LEDs and a push button

```

/*****
 *
 * Alternately toggle two LEDs when a push button is pressed.
 * ATmega328P (Arduino Uno), 16 MHz, AVR 8-bit Toolchain 3.6.2
 *
 * Copyright (c) 2018-2020 Tomas Fryza
 * Dept. of Radio Electronics, Brno University of Technology, Czechia
 * This work is licensed under the terms of the MIT license.
 *
 *****/
/* Defines -----*/
#define LED_GREEN PB5 // AVR pin where green LED is connected
#define LED_RED PC0
#define BTN PD0
#define BLINK_DELAY 250
#ifndef F_CPU
#define F_CPU 16000000 // CPU frequency in Hz required for delay
#endif

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

/* Functions -----*/
/**
 * Main function where the program execution begins. Toggle two LEDs
 * 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);

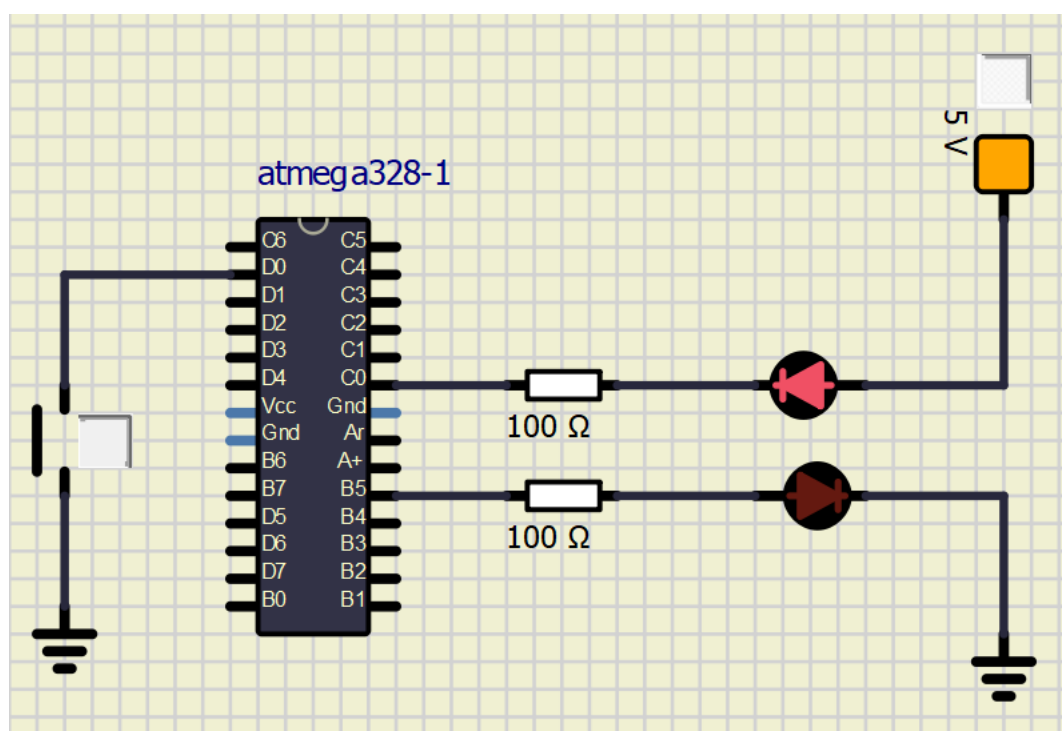
    /* second LED */
    DDRC = DDRC | (1<<LED_RED);
    PORTC = PORTC & ~(1<<LED_RED);

    /* button with pullup resistor*/
    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))
        {
            // Invert LED in Data Register
            // PORTB = PORTB xor 0010 0000
            PORTB = PORTB ^ (1<<LED_GREEN);
            PORTC = PORTC ^ (1<<LED_RED);
        }
    }
    // Will never reach this
    return 0;
}

```

Screenshot of SimulIDE circuit for code with two LEDs and a push button



C code Knight Rider

```

/*****
 *
 * Alternately toggle two LEDs when a push button is pressed.
 * ATmega328P (Arduino Uno), 16 MHz, AVR 8-bit Toolchain 3.6.2
 *
 * Copyright (c) 2018-2020 Tomas Fryza
 * Dept. of Radio Electronics, Brno University of Technology, Czechia
 * This work is licensed under the terms of the MIT license.
 *
 *****/

/* Defines -----*/
#define LED_RED5    PB5      // AVR pin where red LED (number five) is connected
#define LED_RED4    PB4      // AVR pin where red LED (number four) is connected
#define LED_RED3    PB3      // AVR pin where red LED (number three) is connected
#define LED_RED2    PB2      // AVR pin where red LED (number two) is connected
#define LED_RED1    PB1      // AVR pin where red LED (number one) is connected
#define DELAY       250
#ifndef F_CPU
#define F_CPU 16000000      // CPU frequency in Hz required for delay
#endif

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

/* Functions -----*/
/**
 * Main function where the program execution begins. Toggle two LEDs
 * when a push button is pressed.
 */
int main(void)
{
    /* LED RED ONE */
    // Set pin as output in Data Direction Register...
    DDRB = DDRB | (1<<LED_RED1);
    // ...and turn LED off in Data Register
    PORTB = PORTB & ~(1<<LED_RED1);

    /* LED RED TWO */
    DDRB = DDRB | (1<<LED_RED2);
    PORTB = PORTB & ~(1<<LED_RED2);

    /* LED RED THREE */
    DDRB = DDRB | (1<<LED_RED3);
    PORTB = PORTB & ~(1<<LED_RED3);

    /* LED RED FOUR */
    DDRB = DDRB | (1<<LED_RED4);
    PORTB = PORTB & ~(1<<LED_RED4);

    /* LED RED FIVE */
    DDRB = DDRB | (1<<LED_RED5);
    PORTB = PORTB & ~(1<<LED_RED5);
}

```

```

// Infinite loop
while (1)
{
    // Move left to right
    PORTB = PORTB ^ (1<<LED_RED1);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED1);

    PORTB = PORTB ^ (1<<LED_RED2);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED2);

    PORTB = PORTB ^ (1<<LED_RED3);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED3);

    PORTB = PORTB ^ (1<<LED_RED4);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED4);

    PORTB = PORTB ^ (1<<LED_RED5);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED5);

    //Move right to left
    PORTB = PORTB ^ (1<<LED_RED5);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED5);

    PORTB = PORTB ^ (1<<LED_RED4);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED4);

    PORTB = PORTB ^ (1<<LED_RED3);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED3);

    PORTB = PORTB ^ (1<<LED_RED2);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED2);

    PORTB = PORTB ^ (1<<LED_RED1);
    _delay_ms(DELAY);
    PORTB = PORTB ^ (1<<LED_RED1);

}

// Will never reach this
return 0;
}

```

Screenshot of SimulIDE circuit for code Knight Rider

