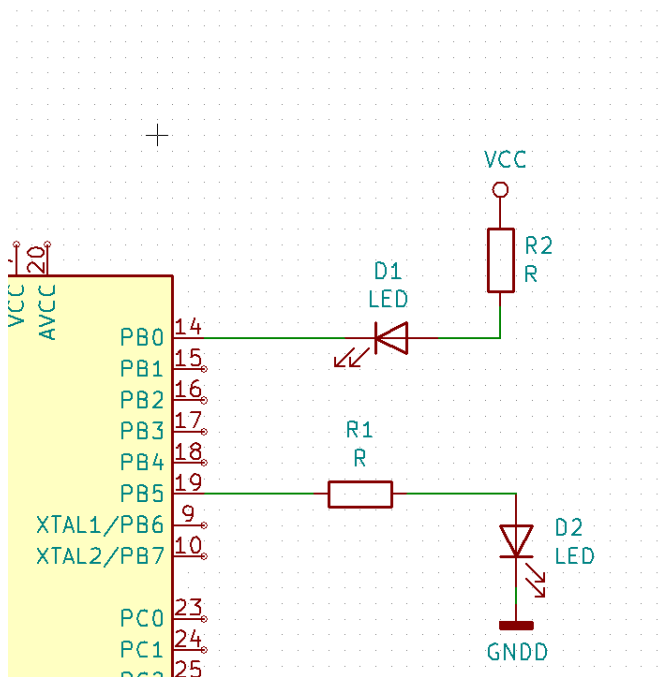


1. Schéma zpojení diod

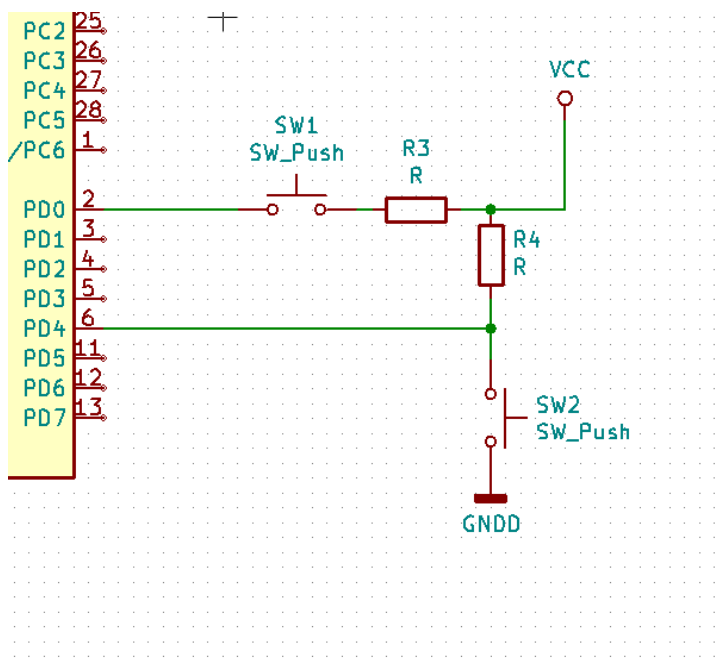


V případě active-low je připojeným pinem katoda diody v druhém případě to je anoda.

Výpočet předřadného rezistoru LED

LED	Vcc	Id	Ud	Rd
Červená	5	20 mA	2	150 Ohm
Modrá	5	20 mA	4	50 Ohm

Zapojení tlačítek k mikrokontroléru



DDRB	Popis
0	Vstupní pin
1	Výstupní pin

PORTB	Popis
0	Výstup low
1	Výstup high

DDRB	PORTB	Direction	Internal pull-up resistor	Description
0	0	input	no	Tri-state, high-impedance
0	1	Input	Yes	Může poskytnout proud skrz aktivní pullup
1	0	Out	No	Výstup nízká úroveň
1	1	out	no	Výstup vysoká úroveň

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 pin10)
	3	Yes (Arduino pin 11)
	4	Yes (Arduino pin 12)
	5	Yes (Arduino pin 13)

	6	No (Crystal oscillator)
	7	No (Crystal oscillator)
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 (Reset pin)
	7	No (Does not exist)
D	0	Yes (Arduino pin RX<-0)
	1	Yes (Arduino pin RX->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)

LED L- připojena k pinu PB5

Kód k blikání LED

```
/* Defines -----*/
#define LED_GREEN    PB5    // AVR pin where green LED is connected
#define LED_RED      PC0
#define BLINK_DELAY  500
#define Button       PD0

#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
#include <avr/sfr_defs.h>

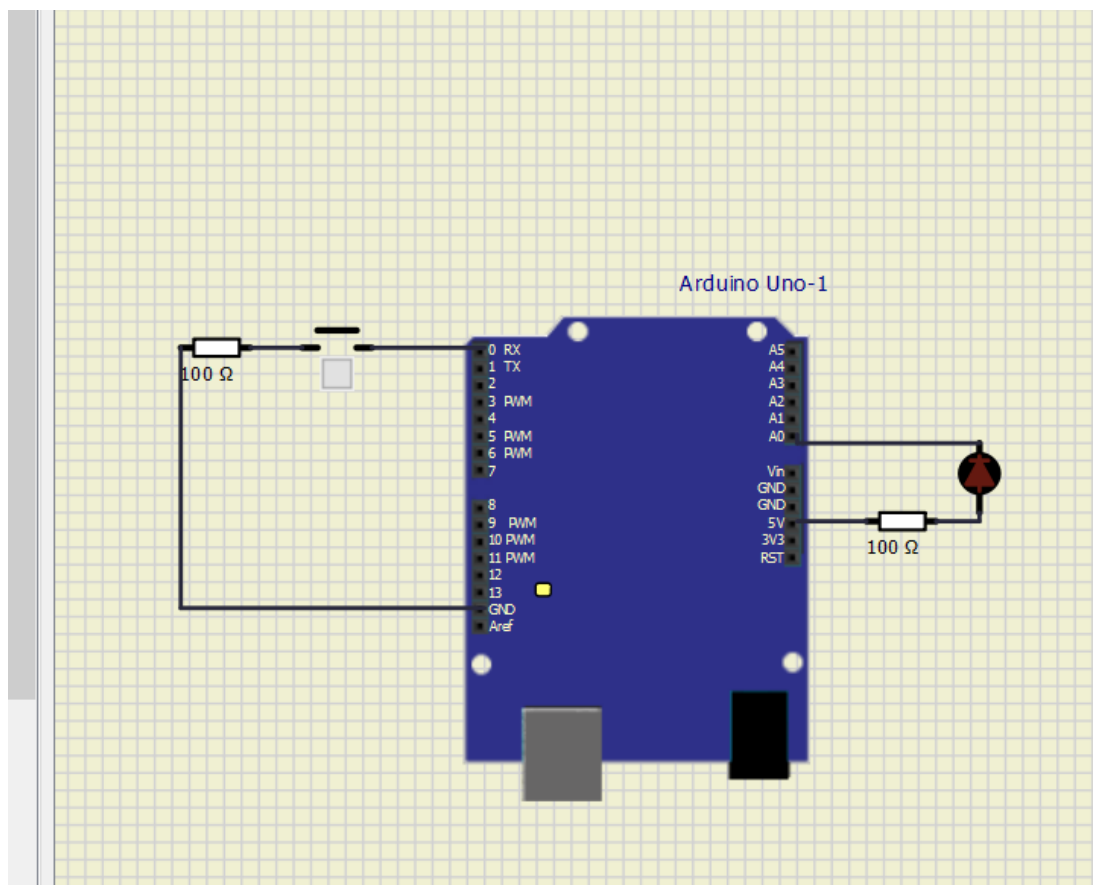
/* Functions -----*/
/**
 * Main function where the program execution begins. Toggle two LEDs
 * when a push button is pressed.
 */
int main(void)
{
    /* GREEN LED */
    // Set pin as output in Data Direction Register...
    DDRB = DDRB | (1<<LED_GREEN);
    // ...and turn LED off in Data Register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

    /* second LED */
    // WRITE YOUR CODE HERE
    DDRC = DDRC | (1<<LED_RED);
    // ...and turn LED off in Data Register
    PORTC = PORTC & ~(1<<LED_RED);

    // Button
    DDRB = DDRB | (0<<Button);
    PORTD=PORTD| (1<<Button);
    // Infinite loop
    while (1)
    {
        // Pause several milliseconds

        // WRITE YOUR CODE HERE
        if (bit_is_clear(PIND,0))
        {
            PORTB = PORTB ^ (1<<LED_GREEN);
            PORTC = PORTC ^ (1<<LED_RED);
            _delay_ms(BLINK_DELAY);
        }
    }

    // Will never reach this
    return 0;
}
```



Zapojení pro blikání s tlačítkem v SimulIDE.

Kód Knight Rider

```

/* Defines -----*/
#define LED_GREEN    PB5    // AVR pin where green LED is connected
#define LED_RED      PC0
#define BLINK_DELAY  100
#define Button       PD0

#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
#include <avr/sfr_defs.h>
/* 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...
    /* second LED */
    // WRITE YOUR CODE HERE
    //init data register
    DDRC = 0xFF;
    PORTC=0xFF;

    // Button
    DDRB = DDRB | (1<<Button);
    PORTD=PORTD | (1<<Button);

    //
    PORTC=PORTC & ~(1<<0);
    // Infinite loop
    int i=0;
    int up=1;
    while (1)
    {
        // Pause several milliseconds

        // WRITE YOUR CODE HERE
        if (bit_is_clear(PIND,0) && up!=0) // if button pressed and direction is up
        {
            PORTC = PORTC ^ (1<<i);
            PORTC = PORTC ^ (1<<(i+1));
            i=i+1;
        }

        if (bit_is_clear(PIND,0) && up==0)
        {
            PORTC = PORTC ^ (1<<i);
            PORTC = PORTC ^ (1<<(i-1));
            i=i-1;
        }

        if(i==4 || i==0) //switch up/down direction
            up^=1;

        _delay_ms(BLINK_DELAY);
    }

    // Will never reach this
    return 0;
}

```