

/\*

Popis:

Vybrané ledky budou blikat cca 2x za vteřinu

Ledky které mají takto blikat jsou vybrány konstantou "LED" a čas svícení je dán děličkou časovače a šířkou jeho registru.

V tomto příkladu bude použit čítač/časovač "0" který má šířku registru 8 bitů to znamená že počítá do 255.

Čítač bude mít zapnutou děličku hodinového signálu s dělitelem 1024.

Výsledná frekvence tedy bude  $1\ 000\ 000/1024/255 = 3,8\ \text{Hz}$  - při každém přetečení budeme negovat stav ledek, takže se rozsvítí a zhasnou přibližně 2x za sekundu.

Přetečení čítače způsobí přerušeni, ve kterém bude negován stav ledek.

\*/

```
#include <avr/io.h> //Vlož knihovnu vstupů a výstupů (PORT, DDR, PIN)
#include <avr/interrupt.h> //Vlož knihovnu vektorů přerušeni (ISR() )

#define LED 0b01110101; //Ledky, které se mají rozsvítit při stisku tlačítka

/*****Obsluha přerušeni*****/
ISR(TIMER0_OVF_vect) //Vektor přerušeni při přetečení čítače/časovače "0"
{
    PORTB = ~PORTB; //vyměň na portu "B" všechny nuly za jedničky a naopak
}

/*****HLAVNÍ FUNKCE*****/

int main (void) //hlavní funkce
{
    TIMSK = 0b00000001; //vybereme přerušeni od TCO
    sei(); //povol přerušeni
    TCCR0 = 5; //spust' časovač "0" s předděličkou 1024

    DDRB = 0xff; //(0xff = 0b11111111) -> Piny 0 - 7 portu "B" budou výstupní
    PORTB = LED; //rozsvit' vybrané ledky

    while (1) // Nekonečná smyčka (dokud 1 = pořád)
    {
        ; //nic nedělej (prázdný příkaz)
        //- zde by mohl být program, který nebude zdržován blikáním
    } //Konec cyklu "while (1)" - program skočí zpět na jeho začátek

} //konec funkce main() - sem se program nikdy nedostane
//((závorka tu však musí být jinak by překladač nahlásil chybu)
```