

Kod zapalający wszystkie diody:

```
org 0
    LJMP start
org 30h
start:
    mov P0, #0x00      ; zgaś diodę
    mov P1, #0x00
    mov P2, #0x00
    mov P3, #0x00
loop:
    LJMP loop          ; w nieskończoność
end
```

Kod migoczący na opóźnieniu pętlami około 3 Hz:

```
org 0
LJMP start

org 30h
start:
    mov A, #00h        ; początek opóźnienia
loop:
    mov R0, #20
loop1:
    mov R1, #50
loop2:
    mov R2, #70
loop3:
    djnz R2, loop3
    djnz R1, loop2
    djnz R0, loop1      ; koniec opóźnienia
    xrl A, #0FFh        ; odwróć wartość
    mov P0, A
    sjmp loop          ; zapętl
```

Kod układu zapalającego diodę z okresem 20ms:

```
.ORG 0
    ljmp main
.ORG 0x0B
    ljmp timer

.ORG 0x030
main:
    mov TMOD, #0x1        ; ustaw tryb 16 bitów
    mov IE, #0x82         ; włącz przerwania
    mov TH0, #0xbe        ; hex(2**16-20000*10/12) - python, pierwszy bajt tu
    mov TL0, #0xe6        ; drugi bajt tu
    setb TR0
    jmp $                 ; w nieskończoność
timer:
    clr TR0               ;wyłącz timer
    mov TH0, #0xbe        ; te same ustawienia
    mov TL0, #0xe6
    setb TR0              ;włącz timer
    xrl A, #0xFF          ; zmień kolor
    mov P0, A
    reti
END
```

Dokładnie ten sam kod co poprzednio tyle, że dioda jest zapalana tylko co 50 przerwanie -  $50 * 20\text{ms} = 1\text{s}$

```
timer:
    clr TR0
    mov TH0, #0xbe
    mov TL0, #0xe6
    setb TR0
    djnz R1, once_a_second ; pomija co pięćdziesiąte
    mov R1, #50
    xrl A, #0xFF
    mov P0, A
    reti
once_a_second:
    reti
END
```

## Nieprzetestowany przykład PWM z możliwością wpływu na wartość.

```
.ORG 0
    LJMP    main

.ORG 0x03
    LJMP    change

.ORG 0x0B
    LJMP    timer

.ORG 0x30
main:
    SETB    EA
    SETB    ET0
    SETB    TR0
    MOV     R1,    #0x7f    ; domyślne wypełnienie
    MOV     TL0,    R1
    SETB    IT0            ; ustaw zbocze
    JMP     $

timer:
    CJNE    P1,    #1,    zero_value
one_value:
    ; jak aktualnie jest jedynka
    CLR     TR0
    MOV     P1,    #0
    MOV     TL0,    R1
    SETB    TR0
    RETI

zero_value:
    ; jak aktualnie jest zero
    CLR     TR0
    MOV     P1,    #1
    MOV     A,    #0xff
    SUBB    A,    R1
    MOV     TL0,    A
    SETB    TR0
    RETI

change:
    CLR     EA    ; wyłącz przerwania
    INC     R1    ; zwiększ wartość wypełnienia o 1
    SETB    EA    ; włącz przerwania
    RETI

END
```