

# Osnove mikroprocesorske elektronike

## Vaja 2: Zbirnik

---

### Naloge:

- Ugotovite kaj delajo programi Vaja2a do Vaja2c in jim dodajte komentarje. Ugotovitve preverite s simulatorjem in nato še z MIŠKOM.
- Napišite program, ki:
  - preverja stanje ene tipke in prižge vse diode, če je tipka pritisnjena.
  - preverja stanje tipk in ugasne vse diode, če so pritisnjene vse štiri tipke.
  - preverja stanje tipk T1 in T2 in premika diode v levo, dokler je pritisnjena tipka T1, premika diode v desno dokler je pritisnjena tipka T2 in pusti diode pri miru, če ni pritisnjena nobena tipka. Če sta pritisnjeni obe tipki hkrati, naj se diode vrnejo v začetno stanje (prižgana je skrajno desna dioda).

## Programs:

### Vaja2a

```
    rcall    Initialization
main:
    rcall    Wait
    rcall    ShiftLED
    rjmp     main
```

```
Initialization:
    ldi     r16,0xFF
    out     DDRB,r16
    ldi     r16,0x01
    out     PORTB,r16
    ret
```

```
ShiftLED:
    clc
    in      r16,PORTB
    rol    r16
    brcc   _do_not_carry_one
_carry_one:
    sbr    r16,0x01
_do_not_carry_one:
    out    PORTB,r16
    ret
```

```
Wait:
    ldi     r16,50
_loop1:
    ldi     r17,238
    _loop2:
        ldi     r18,255
        _loop3:
            dec     r18
            brne   _loop3

            dec     r17
            brne   _loop2

    dec     r16
    brne   _loop1
    ret
```

## Vaja2b

```
    rcall    Initialization
main:
    rcall    Wait
    sbic    PIND,5
    rcall    ShiftLED
    rjmp    main
```

```
Initialization:
    ldi    r16,0xFF
    out    DDRB,r16
    ldi    r16,0x01
    out    PORTB,r16
    ldi    r16,0x00
    out    DDRD,r16
    ret
```

```
ShiftLED:
    clc
    in     r16,PORTB
    rol   r16
    brcc  _do_not_carry_one
_carry_one:
    sbr   r16,0x01
_do_not_carry_one:
    out   PORTB,r16
    ret
```

```
Wait:
    ldi    r16,50
_loop1:
    ldi    r17,238
    _loop2:
        ldi    r18,255
        _loop3:
            dec    r18
            brne  _loop3

            dec    r17
            brne  _loop2

    dec    r16
    brne  _loop1
    ret
```

## Vaja2c

```
.dseg
    LEDleft:  .byte 1
    LEDright: .byte 1
    SkipCnt:  .byte 1

.cseg
    rcall      Initialization
main:
    rcall      Wait
    rcall      ShiftLeft
    rcall      ShiftRight
    rcall      ShowLED
    rjmp       main

Initialization:
    ldi  r16,0xFF
    out  DDRB,r16
    ldi  r16,0x00
    out  PORTB,r16
    ldi  r16,0x01
    sts  LEDleft,r16
    ldi  r16,0xc0
    sts  LEDright,r16
    ldi  r16,3
    sts  SkipCnt,r16
    ret

ShiftLeft:
    lds  r16,LEDleft
    lsl  r16
    sbrc r16,4
    sbr  r16,0x01
    sts  LEDleft,r16
    ret

ShiftRight:
    lds  r16,SkipCnt
    dec  r16
    brne _sr_notTimeYet
_sr_itsTime:
    lds  r16,LEDright
    lsr  r16
    sbrc r16,3
    sbr  r16,0x80
    sts  LEDright,r16
    ldi  r16,3
_sr_notTimeYet:
    sts  SkipCnt,r16
    ret

ShowLED:
    lds  r16,LEDleft
    andi r16,0x0F
```

```
lds      r17,LEDright
andi    r17,0xF0
or      r16,r17
out     PORTB,r16
ret
```

Wait:

```
ldi     r16,50
_loop1: ldi     r17,238
        _loop2: ldi     r18,255
                _loop3: dec     r18
                brne   _loop3

                dec     r17
                brne   _loop2

dec     r16
brne   _loop1
ret
```