

MIKROPROCESOROVÁ TECHNIKA

| Třída: A3 | Úloha č. 10 | _{Název:} Dynamický displej a matice tlačítek | |
|--------------|---------------|--|---------|
| Jméno: | Datum zadání: | Datum odevzdání: | Známka: |
| Jakub Tenk | 8. 4. 2021 | 18. 5. 2021 | |

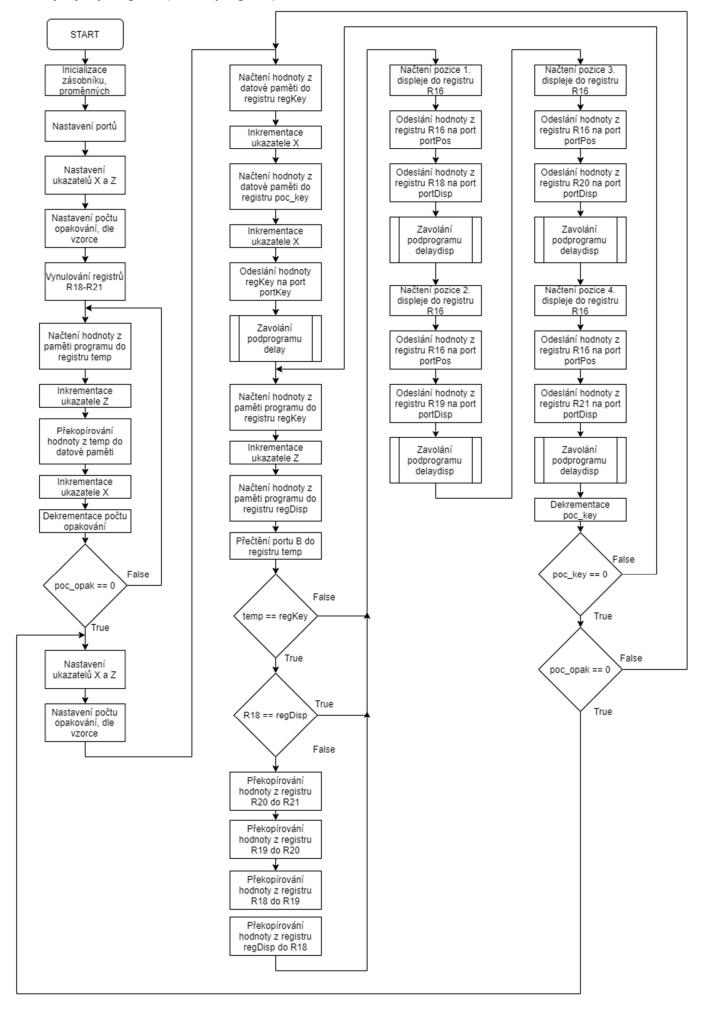
Zadání:

Vytvořte program v JSA, který bude zobrazovat na dynamickém displeji hodnoty stisknuté na připojené klávesnici (matice tlačítek - viz. minulé cvičení). Zvolené hodnoty se budou postupně posouvat po displeji zleva doprava.

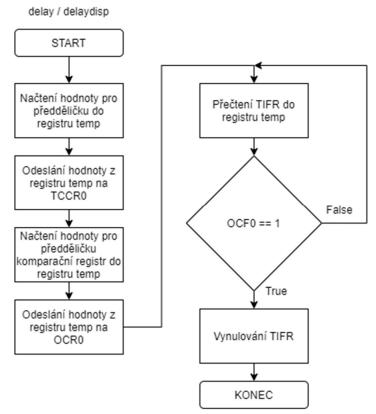
Postup:

- 1. Při spuštění program inicializuje zásobník, port a vynuluje registry R18-R21.
- 2. Načtení tabulky tb sloupce a přelití do datové paměti.
- 3. Pokračování do hlavního programu na návěští main.
- **4.** Načtení vzoru sloupce a odeslání jej na port B (aktivace sloupce klávesnice).
- **5.** Načtení vzoru tlačítka s následnou inkrementací, načtení vzoru displeje s následnou inkrementací.
- **6.** Přečtení portu B a odeslání hodnoty do registru temp.
- **7.** Pokud vzor tlačítka v sloupci není shodný s hodnotou z registru temp pokračuje bodem **11**.
- **8.** Skok na návěští output.
- 9. Pokud registr R18 je shodný s registrem regDisp, tak program pokračuje bodem 9.
- 10. Přelití hodnot z registrů do registrů:
 - a. Hodnota z registru R20 se překopíruje do registru R21.
 - **b.** Hodnota z registru R19 se překopíruje do registru R20.
 - **c.** Hodnota z registru R18 se překopíruje do registru R19.
 - d. Hodnota z registru regDisp se překopíruje do registru R18.
- 11. Skok na návěští continue.
- 12. Rozsvícení hodnoty z R18 na 1. pozici displeje.
- 13. Rozsvícení hodnoty z R19 na 2. pozici displeje.
- 14. Rozsvícení hodnoty z R20 na 3. pozici displeje.
- 15. Rozsvícení hodnoty z R21 na 4. pozici displeje.
- **16.** Dekrementace počtu tlačítek ve sloupci (poc key).
- **17.** Pokud počet tlačítek ve sloupci (poc_key) je větší jak nula, program pokračuje bodem **5**.
- **18.** Dekrementace počtu opakování (poc_opak).
- 19. Pokud počet opakování (poc opak) je větší jak nula, program pokračuje 4.
- **20.** Skok na začátek hlavního programu, na návěští main.

Vývojový diagram (hlavní program):



Vývojový diagram (podprogram delay a delaydisp):



Ukázka z kódu:

```
>>
                    Obsluha matice tlačítek 4x4 a Displeje
Vytvořil Jakub Tenk
  >>
                                                                                                  <<
  >>
                                                                                                  <<
                                     github.com/DrGumik
                                                    ; >> Vypne debug
.include "m128def.inc"
                                                    ; >> Přiřazení jmen registru
; >> Zapne debug
.list
. dseg
.org 0x200
                                                    ; >> Interní RAM - počáteční adresa
.cseg
.org 0x0000
                                                   ; >> Začátek pamětí flash
.def poc_opak = R22
.def poc_key = R17
.def temp = R23
.def regKey = R15
.def regDisp = R16
.equ portKey = POPTB
                                                   ; >> Definování proměnných
                                                    ; >> Registr pro počet tlačítek ve sloupci
                                                    ; >> Temporary registr
                                                   ; >> Registr pro vzorce tlačítek
                                                   ; >> Registr pro vzor čísla pro dislpej na rozsvícení
                                                   ; >> Port klávesnice
; >> Port k rozsvicení displeje
; >> Port k multiplexu displeje
 equ portDisp = PORTA
.equ portDisp = FORIC
.equ portPos = PORTC
.equ ddrKey = DDRB
.equ ddrDisp = DDRA
.equ ddrPos = DDRC
      ldi temp,LOW(RAMEND)
                                                   ; >> Inicializace zásobníku, portů...
      out SPL, temp
ldi temp, HIGH(RAMEND)
      out SPH, temp
      ldi temp, 0x0F
      out ddrKey, temp
ldi temp, 0xFF
out ddrDisp, temp
out ddrPos, temp
      ldi XL, 0x00
ldi XH, 0x01
      ldi poc_opak, 2*(tb_sloupce_end-tb_sloupce_beg)
ldi ZL, IOW(tb_sloupce_beg<<1)
ldi ZH, HIGH(tb_sloupce_beg<<1)</pre>
      clr R18
clr R19
```

```
clr R20
     clr R21
load_tb:
                                        ; >> Načtení tabulky tb_sloupce
      lpm temp, Z+
                                         ; >> z paměti programu do datové paměti
      st X+, temp
      dec poc_opak
     brne load_tb
                                        ; >> Začátek hlavního programu
main:
     ldi XL, 0x00
ldi XH, 0x01
      ldi ZL, LOW(tb_key_beg<<1)
ldi ZH, HIGH(tb_key_beg<<1)</pre>
      ldi poc_opak, (tb_sloupce_end-tb_sloupce_beg)
                                        ; >> Kontrola celého jednoho sloupce
loop:
     ld regKey, X+
ld poc_key, X+
out portKey, regKey
     call delay
key_id:
      lpm regKey, Z+
                                        ; >> Načtení vzorce tlačítka a displeje
      lpm regDisp, Z+
     in temp, pinB
cp temp, regKey
                                        ; >> Pokud regKey == hodnota z portuB, skočí na output
     brne continue
      jmp output
continue:
     ldi R16, 0b00000_000
out portPos, R16
                                         ; >> 1. pos displeje
     out portDisp, R18
call delaydisp
     ldi R16, 0b00000_001
out portPos, R16
                                        ; >> 2. pos displeje
     out portDisp, R19
     call delaydisp
     ldi R16, 0b00000_010
     out portPos, R16
                                        ; >> 3. pos displeje
     out portDisp, R20
     call delaydisp
      ldi R16, 0b00000_011
     out portPos, R16
out portDisp, R21
                                        ; >> 4. pos displeje
     call delaydisp
     dec poc key
     brne key_id
     dec poc_opak
     brne loop
     jmp main
                                    ; >> Rozsvícení displejů
     cp R18, regDisp
                                   ; >> a posunutí do prava
; >> Podmínka cp = kontroluje, že tlačítko může být stisknuto
; >> jen jednou po sobě (tudíž se na displejinevypíše například 1111)
     breq continue
     mov R21, R20
mov R20, R19
mov R19, R18
mov R18, regDisp
jmp continue
                                    ; >> HW Delay (+- 0.78ms)
; >> Hodnota pro nastavení předděličky
delay:
ldi temp, 0b00001101
set_timer:
out TCCR0, temp
ldi temp, 96
out OCR0, temp
                                    ; >> Nastavení předděličky
                                    ; >> Hodnota pro komparační registr
; >> Načtení komparační hodnoty
compare:
                                    ; >> Načtení hodnoty z TIFR do registru temp
; >> Pokud hodnota OCFO == 1, tak program preskočí instrukci jmp compare
     in temp, TIFR
     sbrs temp, 1
     jmp compare
                                    ; >> Vynulování TIFR
; >> Vrací se do hlavního programu
     out TIFR, temp
     ret
delaydisp:
                                    ; >> HW Delay (+- 2ms)
      ldi temp, 0b00001101
                                  ; >> Hodnota pro nastavení předděličky
set_timer_disp:
out_TCCR0, temp
ldi_temp, 255
out_OCR0, temp
                                    ; >> Nastavení předděličky
                                    ; >> Hodnota pro komparační registr
; >> Načtení komparační hodnoty
comparedisp:
in temp, TIFR
                                   ; >> Načtení hodnoty z TIFR do registru temp
; >> Pokud hodnota OCFO == 1, tak program preskočí instrukci jmp compare
     sbrs temp, 1
     jmp comparedisp
out TIFR, temp
                                    ; >> Vynulování TIFR
     ret
```

```
tb_sloupce_beg:
    .db 0b1111_1110, 3 ; >> 1. sloupec, pocet kláves
    .db 0b1111_1101, 4 ; >> 2. sloupec, pocet kláves
    .db 0b1111_1011, 3 ; >> 3. sloupec, pocet kláves
    .db 0b1111_1011, 3 ; >> 3. sloupec, pocet kláves
tb_sloupce_end:

tb_key_beg: ; >> Key_ID , DISP_VZOR / Key_ID2, DISP_VZOR...
    .db 0b1110_1110, 0b0110_0000, 0b1101_1110, 0b0110_0110, 0b1011_1110, 0b1110_0000
    .db 0b1110_1101, 0b1101_1010, 0b1101_1010, 0b1011_0110, 0b1011_1101, 0b1111_1110, 0b01111_1110, 0b1111_1100
    .db 0b1110_1011, 0b1111_0010, 0b1101_1011, 0b1011_1110, 0b1011_1011, 0b1111_0110

tb_key_end:
```

Závěr:

Tato úloha z mého hlediska byla nebyla zas tak těžká, jen to chtělo ve škole odladit, protože můj sestavený modul pro Arduino je jinak zapojen než ten, který máme ve škole. Po odladění programu vše funguje.