



# Mikroprocesorová technika

A3		Přehrávání vzorů		
Tenk Jakub			1/6	Známka:
5. 2. 2021		Datum odevzdání:	18. 2. 2021	Odevzdáno:



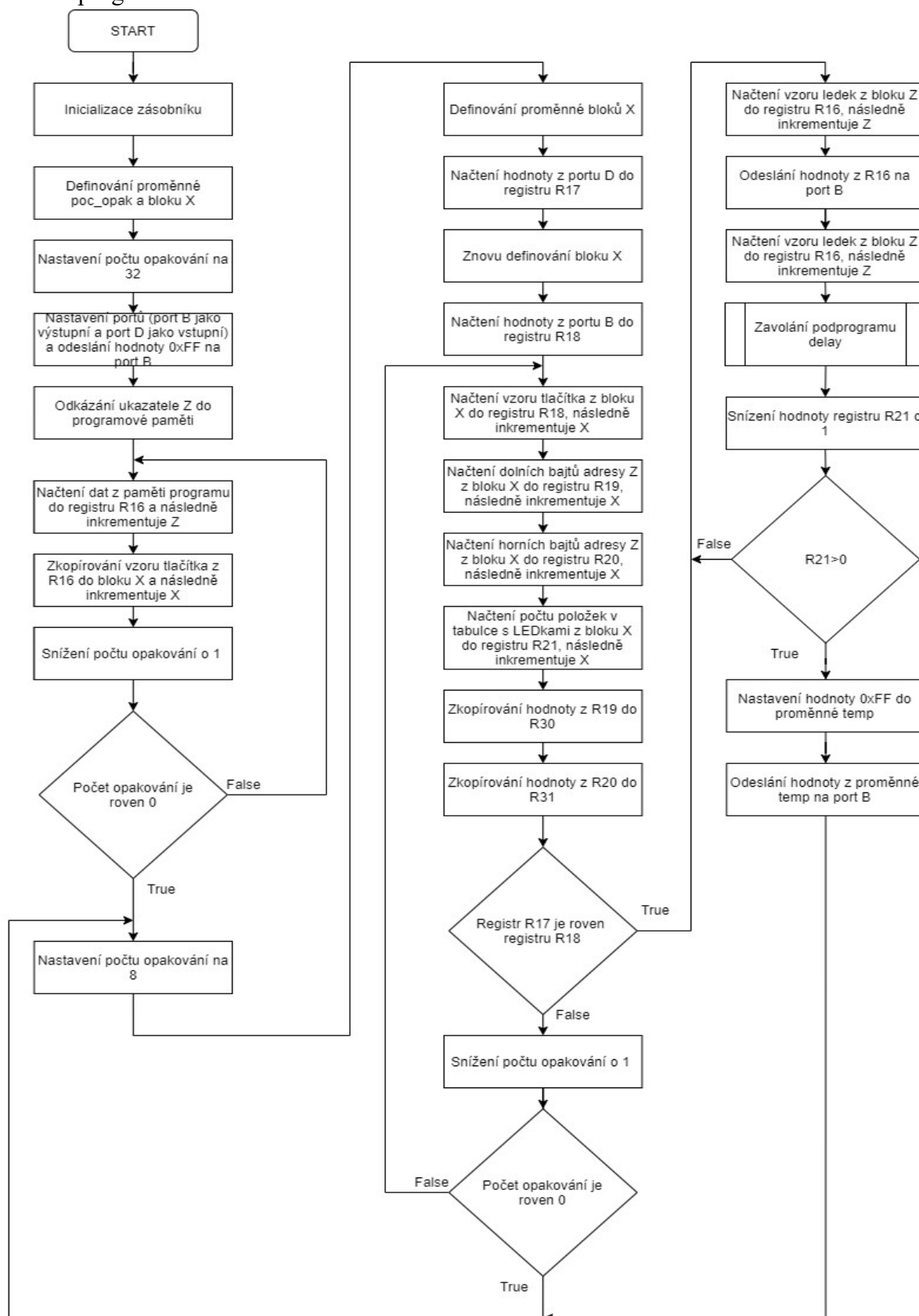
### **Zadání:**

Sestavte v jazyce JSA (assembler) program, který bude "obsluhovat" přípravek:

- 8 tlačítek a 8 LED svítivých diod a obsahovat funkce:
- při stisku zvoleného tlačítka se přehraje na osmici LED diod zvolená sekvence vzorů. Sekvence budou mít od 2 do 6 vzorů,
- jednotlivé vzory budou uloženy v programové paměti ve formě tabulky,
- časový interval mezi změnami jednotlivých vzorů sekvence realizujte s využitím HW komponenty Timer (HW delay),
- časový interval nastavte přibližně v intervalu od 250ms do 350ms.

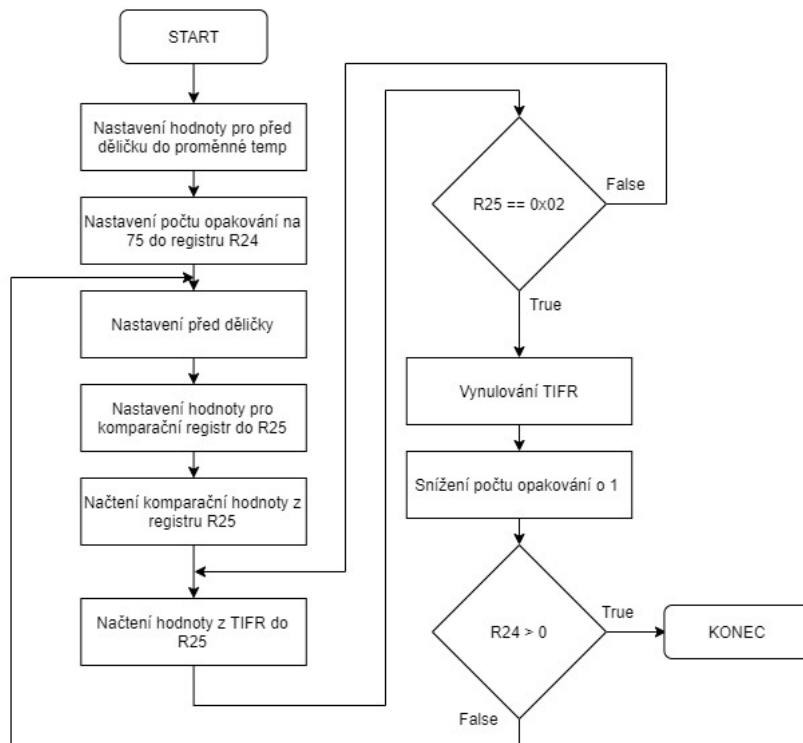


## Vývojový diagram: Hlavní program





## Podprogram delay



## Výpis programu:

```
>> <<
>> Úloha č. 6 <<
>> Rozsvícení sekvence vzorů LED podle stisknutého tlačítka <<
>> Vytvořil Jakub Tenk <<
>> <<
.nolist >> Vypne debug
.include "m128def.inc" >> Přiřazení jmen registru
.list >> Zapne debug

.dseg
.org 0x200 >> Interní RAM - počáteční adresa

.cseg
.org 0x0000 >> Začátek paměti flash

.def poc_opak = R22 >> Definování proměnných (poc_opak, temp)
.def temp = R23

ldi temp, LOW(RAMEND) >> Inicializace zásobníku
out SPL, temp
ldi temp, HIGH(RAMEND)
out SPH, temp

ldi poc_opak, 32 >> Nastavení hodnoty počtu opakování na 32

ldi XL, 0x00 >> Nastavení dolního bajtu adresy X
ldi XH, 0x04 >> Nastavení horního bajtu adresy X
```



```
ldi R16, 0xFF      ; >> Nastavení portu B jako výstupní a portu D jako vstupní
ldi R17, 0x00
out DDRB, R16
out DDRD, R17
ldi R16, 0b11111111 ; >> Nastavení hodnoty, která zhasne všechny ledky do R16
out PORTB, R16      ; >> Odeslání hodnoty pro zhasnutí ledek na port B

ldi ZL, LOW(tb_tlac<<1) ; >> Odkázání ukazatele Z do programové paměti na adresu, kde se nachází tb_tlac
ldi ZH, HIGH(tb_tlac<<1)

load_tb:
lpm R16, Z+        ; >> Přesun hodnoty z paměti programu z adresy uložené v registru Z do registru R16 s následnou inkrementací
st X+, R16          ; >> Zkopírování hodnoty z R16 do datové paměti na adresu uloženou v registru X s následnou inkrementací
dec poc_opak        ; >> Snížení počtu opakování o 1
brne load_tb        ; >> Skok na návěští load_tb pokud poc_opak > 0

main:
ldi poc_opak, 8      ; >> Nastavení hodnoty počtu opakování na 8
ldi XL, 0x00         ; >> Nastavení dolního bajtu adresy X
ldi XH, 0x04         ; >> Nastavení horního bajtu adresy X
in R17, pinD         ; >> Přečtení hodnoty z portu D a uložení do R17

loop:
ld R18, X+           ; >> Zkopírování hodnoty z datové paměti do registru R18 s následnou inkrementací
ld R19, X+           ; >> Zkopírování hodnoty z datové paměti do registru R19 s následnou inkrementací
ld R20, X+           ; >> Zkopírování hodnoty z datové paměti do registru R20 s následnou inkrementací
ld R21, X+           ; >> Zkopírování hodnoty z datové paměti do registru R21 s následnou inkrementací
mov R30, R19         ; >> Odkázání ukazatele Z do programové paměti na adresu, kde se nachází daná tabulka led vzorů
mov R31, R20

cp R17, R18          ; >> Porovnání vstupní hodnoty z portu D se vzorem tlačítka
brq output           ; >> Pokud se hodnoty rovnají skočí na návěští output

dec poc_opak         ; >> Snížení počtu opakování o 1
brne loop            ; >> Skok na návěští loop pokud poc_opak > 0
rjmp main            ; >> Skok do hlavního programu

output:
lpm R16, Z+          ; >> Načtení vzoru ledek do registru R16 s následnou inkrementací
out PORTB, R16       ; >> Odešle hodnotu z R16 na port B
lpm R16, Z+          ; >> Načtení vzoru ledek do registru R16 s následnou inkrementací (tento vzor se nepoužívá, slouží jen k tomu, aby
ukazatel poskočil znovu o 1)
call delay           ; >> Zavolání podprogramu jménem delay
dec R21              ; >> Snížení hodnoty registru R21 o 1
brne output          ; >> Skok na návěští output pokud R21 > 0

ldi temp, 0b11111111 ; >> Nastavení hodnoty do proměnné temp
out PORTB, temp      ; >> Odeslání hodnoty uložené v proměnné temp (slouží k zhasnutí ledek, po dokončení sekvence)
rjmp main            ; >> Skok na začátek hlavního programu (návěští main)

delay:
; >> HW Delay (+- 300ms)
ldi temp, 0b00001110 ; >> Hodnota pro nastavení před děličky
ldi R24, 75          ; >> Zpoždění se bude opakovat 75x -> 75x4ms=300ms

set_timer:
out TCCR0, temp      ; >> Nastavení před děličky
ldi R25, 250         ; >> Hodnota pro komparační registr
out OCR0, R25        ; >> Načtení komparační hodnoty
```



compare:

```
in R25,TIFR          ; >> Načtení hodnoty z TIFR do registru R25
cpi R25,0x02          ; >> Porovnání hodnot
brne compare         ; >> Pokud hodnoty TIFR a 0x07 nejsou stejné, tak skočí na návěští compare
out TIFR, R25         ; >> Vynulování TIFR
dec R24              ; >> Odečtení cyklu
brne set_timer       ; >> Pokud registr s odečítáním cyklu není roven 0 skočí na návěští set_timer
ret                  ; >> Vrací se do hlavního programu
```

; >> Tabulky led vzorů a tlačítek

beg\_led1:

```
.db 0b01111111
.db 0b10111111
.db 0b11011111
.db 0b11101111
.db 0b11110111
.db 0b11111011
```

end\_led1:

beg\_led2:

```
.db 0b11111110
.db 0b11111101
.db 0b11111011
.db 0b11110111
```

end\_led2:

beg\_led3:

```
.db 0b10011001
.db 0b10111101
```

end\_led3:

beg\_led4:

```
.db 0b10101010
.db 0b01010101
.db 0b00100100
```

end\_led4:

beg\_led5:

```
.db 0b01111110
.db 0b10111101
.db 0b11011011
.db 0b11100111
.db 0b11011011
```

end\_led5:

beg\_led6:

```
.db 0b11110000
.db 0b00001111
```

end\_led6:

beg\_led7:

```
.db 0b11001100
.db 0b00110011
.db 0b10111101
```

end\_led7:

beg\_led8:

```
.db 0b11100001
```



```
.db 0b11000011  
.db 0b10000111  
.db 0b11000011  
.db 0b11100001
```

end\_led8:

tb\_tlac:

```
.db 0b01111111, low(beg_led1<<1), high(beg_led1<<1), (end_led1-beg_led1)  
.db 0b10111111, low(beg_led2<<1), high(beg_led2<<1), (end_led2-beg_led2)  
.db 0b11011111, low(beg_led3<<1), high(beg_led3<<1), (end_led3-beg_led3)  
.db 0b11101111, low(beg_led4<<1), high(beg_led4<<1), (end_led4-beg_led4)  
.db 0b11110111, low(beg_led5<<1), high(beg_led5<<1), (end_led5-beg_led5)  
.db 0b11111011, low(beg_led6<<1), high(beg_led6<<1), (end_led6-beg_led6)  
.db 0b11111101, low(beg_led7<<1), high(beg_led7<<1), (end_led7-beg_led7)  
.db 0b11111110, low(beg_led8<<1), high(beg_led8<<1), (end_led8-beg_led8)
```

### **Závěr:**

Tato úloha pro mě nebyl moc těžká. Nejvíce mi usnadnilo práci to, že už při hodině jsme si navrhli nejlepší možnost tabulek, se kterými jsem následně pracoval. Nejdříve jsem sestavil celý program, tak aby byl funkční na ATmega2560 a otestoval ho na živo. Když vše fungovalo, tak jsem přepsal pár věcí, aby to fungovalo s ATmega128.