

Střední Průmyslová Škola Elektrotechnická Havířov	<b>Protokol do MIT</b>	Třída: 4.C
		Skupina: 3
<b>A/D převodník</b>	Zpráva číslo: 5	
	Dne: 11.04.2007	
	Soupis použitých přístrojů: přípravek s $\mu$ C 8051 přípravek ATADC8R přípravek s LCD zdroj	
	Jméno učitele: Ing. Baarová	
	Jméno:	
	Známka:	

## ZADÁNÍ:

1. Vytvořte testovací program, který bude převádět spojitý signál na digitální. Dále vypočítejte konstantu tak, abychom na připojených LED diodách při 5 V zobrazili 100.

2. Vytvořte elektronickou váhu a zobrazujte na LCD převedené napětí, hmotnost předmětu a cenu. Vstupní napětí zvyšujte od 0 po 5 V. Při 1 V bude hmotnost 50 g. Níže jsou uvedeny hmotnosti s cenou:

do 100 g = 20 Kc

do 150 g = 40 Kc

do 200 g = 60 Kc

nad 200 g = 100 Kc

## TEORIE:

A/D převodníky se používají k převodu vstupního analogového (spojitého) signálu na výstupní digitální hodnoty vyjádřené binárním kódem.

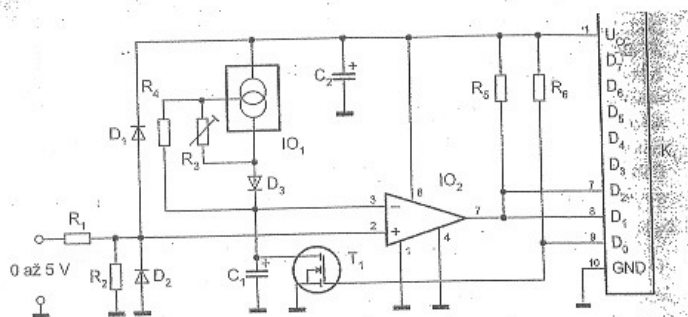
### DRUHY A/D PŘEVODNÍKŮ:

1. S dvojitou integrací
2. S převodem na napětí na frekvenci
3. Převodník Sigma-Delta
4. S postupnou aproximací
5. Paralelní

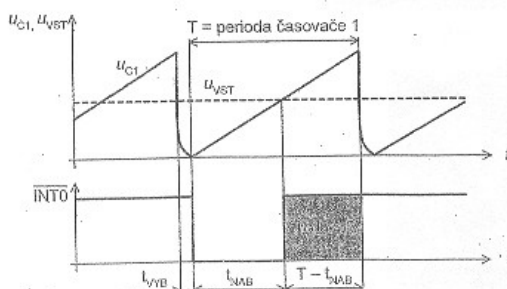
### PŘÍPRAVEK ATADC8R (A/D PŘEVODNÍK):

Jedná se o 8-bitový A/D převodník. Měřicí cyklus začíná vybitím kondenzátoru C1 pomocí tranzistoru T1. Po rozpojení tranzistoru se kondenzátor začne nabíjet z proudového zdroje IO1. Rychlost nabíjení je dána kapacitou kondenzátoru a trimrem R3. Dioda D3 a rezistor R3 slouží jako teplotní kompenzace proudového zdroje IO1. Napětí z kondenzátoru C1 je přivedeno na invertující vstup komparátoru IO2 a tam se porovná se vstupním napětím, které je přivedeno na neinvertující vstup IO2. Protože na proudovém zdroji IO1 musí zůstat určitá minimální napěťová rezerva, nemůže být vstupní napětí 0 až 5 V přivedeno přímo, ale přes dělič R1 a R2 (tedy rozsah 0 až 2,5 V). Diody D1 a D2 chrání přípravek před velkým nebo záporným vstupním napětím. Výstup komparátoru IO2 je přiveden na vstup D1.

### SCHÉMA A ČASOVÝ DIAGRAM PŘÍPRAVKU ATADC8R:



Obr. 10.24 Schéma zapojení přípravku ATADC8R



## VÝPOČET KONSTANTY:

20 ms = 20000 us (doba nabití C1 na 5 V)

Vypočítali jsme konstantu 200 (délka trvání 1 měření) => na LEDkách se pak zobrazila hodnota 100 při 5 V.

$$\frac{\text{doba nabití kondenzátoru}}{\text{hodnota 0 až 255 (velikost registru)} - \text{vybereme hodnotu, kterou se dá dobře dělit}} = \frac{20000}{100} = 200us$$

Pak vztahem  $100/2=50 \Rightarrow 50/10 \Rightarrow 5$  uložena v akumulátoru; 0 (zbytek po dělení) uložena v registru b

## TESTOVACÍ PROGRAM:

```
org 0

Start:
mov r0, #0 ;vynulovani registru r0

Vybij:
setb P1.0 ;nastavime P1.0 do 1 => vybijeni C
mov r7, #0 ;do r7 nastavime 0 => nastaveni r7 pred odecitanim
djnz r7, $ ;odecitame z r7 -1 dokud nebude 0 => cekame na vybiti C
clr P1.0 ;P1.0 vratime po vybiti C zpět do puvodniho stavu => 0

Mer:
jb P1.1, Zobraz ;kontrolujeme P1.1 => P1.1 v 1 pak ukoncime mereni a
skaceme na zobrazeni
mov r7, #200 ;do r7 dame konstantu 200 => delka trvani 1 mereni (vypocitano)
djnz r7, $ ;cekame 200 us
inc r0 ;pricteme do r0 +1
jmp Mer ;opakujeme tak dlouho dokud vystup z komparatoru není
roven 1

Zobraz:
mov a, r0 ;do akumulatoru dame hodnotu r0
cpl a ;negujeme
mov P2, a ;vysleme hodnotu z akumulatoru na P2

jmp Start

nop
end
```

## HLAVNÍ PROGRAM A/D PŘEVODNÍKU:

```
;1 V ~ 50 g
;2 V ~ 100 g
;3 V ~ 150 g
;4 V ~ 200 g
;do 100 g = 20 Kc
;do 150 g = 40 Kc
;do 200 g = 60 Kc
;nad 200 g = 100 Kc

;prirazeni pinu k LCD
LCD_EN    bit        P2.0
LCD_RS    bit        P2.2

                org    0

Main:
    call     Init           ;inicializacni cast
    call     Vybij         ;vybijeni kondenzátoru
    call     Mer           ;mereni a nasledne zobrazeni

    jmp     Main

Init:
    mov     r0, #0         ;vynuluje registr r0
    call    ResetLCD      ;resetuje LCD
    ret

Vybij:
    setb    P1.0          ;nastavime P1.0 do 1 => vybijeni C
    mov     r7, #0        ;do r7 nastavime 0 => nastaveni r7 pred odecitanim
    djnz    r7, $         ;odecitame z r7 -1 dokud nebude 0 => cekame na vybiti C
    clr     P1.0          ;P1.0 vratime po vybiti C zpet do puvodniho stavu => 0
    ret

Mer:
    jb     P1.1, Vaha1    ;kontrolujeme P1.1 => P1.1 v 1 pak ukoncime mereni a skaceme na
zobrazeni
    mov     r7, #200      ;do r7 dame konstantu 200 => delka trvani 1 mereni (vypocitano)
    djnz    r7, $         ;cekame 200 us
    inc     r0            ;pricteme do r0 +1
    jmp     Mer           ;opakujeme tak dlouho dokud vystup z komparatoru neni
roven 1

Vaha1:
    cjne    r0, #20, Hop1 ;porovnavame hodnotu r0 s 20

Hop1:
    jnc     Vaha2        ;souvisi s predeslou pominkou, jeli r0 < 20 pak priznak
C=1 a program pokracuje
Vaha1:
    mov     dptr, #Vaha1Tab ;do dptr dame adresu tabulky, ktera se ma zobrazit na LCD
    call    TextLCD      ;podiva se na adresu v dptr a tabulku na teto adrese zobrazi
na LCD
    jmp     Napeti
```

```

Vaha2:
    cjne        r0, #60, Hop2
Hop2:
    jnc        Vaha3
Vaha21:
    mov        dptr, #Vaha2Tab
    call       TextLCD
    jmp        Napeti

Vaha3:
    cjne        r0, #80, Hop3
Hop3:
    jnc        Vaha4
Vaha31:
    mov        dptr, #Vaha3Tab
    call       TextLCD
    jmp        Napeti

Vaha4:
    mov        dptr, #Vaha4Tab
    call       TextLCD

Napeti:
    mov        a, r0                ;prevod namerene hodnoty na napeti U
    pak ji prevedeme na U           ;akumulator naplnime hodnotou z r0 (namerena hodnota),
    mov        b, #2                ;do registru b dame hodnotu 2
    div        ab                    ;delime akumulator registrem b => namerenou hodnotu/2
    mov        b, #10               ;do registru b dame hodnotu 10, v akumulatoru zustala
    celociselna cast po deleni
    div        ab                    ;pokracujeme v deleni
    mov        r1, a                 ;celociselnou cast zalohujeme do registru r1
    mov        r2, b                 ;zbytek po deleni zalohujeme do registru r2

LCD
    mov        a, #083h             ;do akumulatoru dame pozice (pozici na LCD) kde budeme chtit psat po
    call       RidLCD               ;nastavime se na tuto pozici
    mov        a, r1                 ;do akumulatoru dame hodnotu, kterou chceme zobrazit na

LCD
    add        a, #30h              ;pricteme hodnotu 30h ('0'-nula) => prevod na ASCII
    call       ZnakLCD              ;vypiseme znak na LCD

    mov        a, #085h
    call       RidLCD
    mov        a, r2
    add        a, #30h
    call       ZnakLCD

    call       Delay01              ;zpozdeni, aby zobrazovane texty a hodnoty na LCD neblikaly

    ret

Delay01:
    mov        r6, #255
    mov        r5, #255
    mov        r4, #2

```

Delay02:

```
djnz    r6, Delay02
djnz    r5, Delay02
djnz    r4, Delay02
ret
```

Vaha1Tab:

;tabulky ktere budeme zobrazovat na LCD

```
db 'U= , V      '
db ' m<100g c= 20Kc '
```

Vaha2Tab:

```
db 'U= , V      '
db ' m<150g c= 40Kc '
```

Vaha3Tab:

```
db 'U= , V      '
db ' m<200g c= 60Kc '
```

Vaha4Tab:

```
db 'U= , V      '
db ' m>200g c=100Kc '
```

```
#include<LCDkit.inc>
```

;soubor pro ovladani LCD

```
nop
end
```

### ZHODNOCENÍ:

U této úlohy je důležité si uvědomit, že napětí na vstupu je úměrné času nabíjení kondenzátoru. Nesmíme opomenout vybit kondenzátor před měřením. Při měření jsme používali konstantu, kterou jsme museli vypočítat. Jelikož máme k dispozici 8 bitové registry => můžeme do něj uložit hodnotu 0 až 255. My jsme vybrali hodnotu 100, která bude představovat 5 V, a bude se dát dobře dělit. Vypočítaná hodnota představovala délku 1 měření a v našem případě byla 200 us. Důležité taky bylo přidat include soubor pro ovládání LCD, dále taky bylo nutné převést hodnotu na ASCII a až pak ji zobrazit na LCD. Měl jsem menší problém se zobrazením na LCD, konkrétně že zobrazené hodnoty blikaly => proto jsem přidal větší zpoždění => hodnota na LCD bude déle zobrazena.