

Střední Průmyslová Škola Elektrotechnická Havířov	<h1>Protokol do MIT</h1>	Třída: 4.C
<h2>D/A převodník</h2>		Skupina: 3
		Zpráva číslo: 4
	Dne: 26.02.2007	
	Soupis použitých přístrojů: přípravek s μC 8051 osciloskop přípravek s D/A převodníkem	
	Jméno učitele: Ing. Baarová	
	Jméno:	
	Znamka:	

ZADÁNÍ:

Vytvořte program pro uC 8051, který bude po zadání vstupní kombinace generovat určitý průběh signálu. Máte za úkol vytvořit 5 různých průběhů: pila nahoru, pila dolů, trojúhelník, obdélník a sinus. Všechny signály mají stejnou frekvenci a to $f = 100 \text{ Hz}$.

TEORIE:

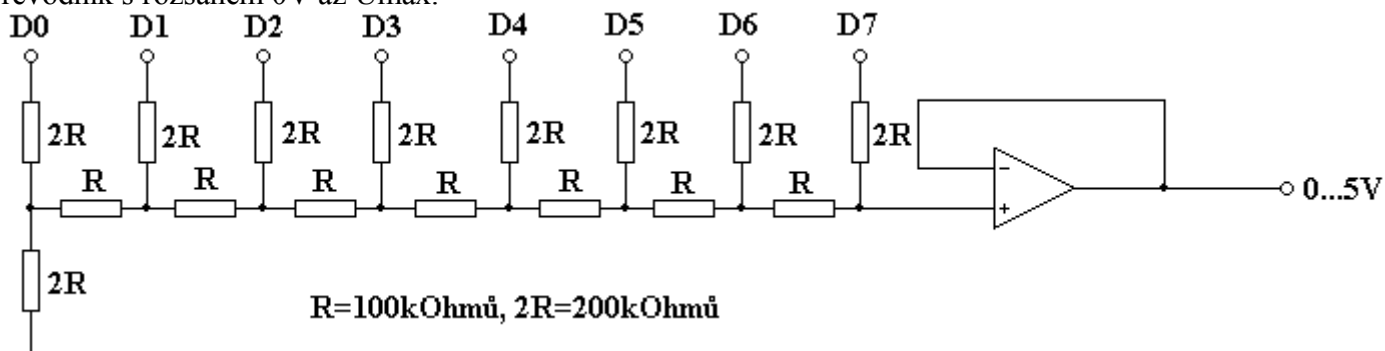
D/A převodníky se používají k převodu vstupní digitální hodnoty vyjádřené binárním kódem na odpovídající výstupní hodnotu spojitého signálu, což je výstupní napětí. D/A převodníky se používají u zvukových a obrazových výstupů počítačů, datových přenosech.

TYPY D/A PŘEVODNÍKŮ:

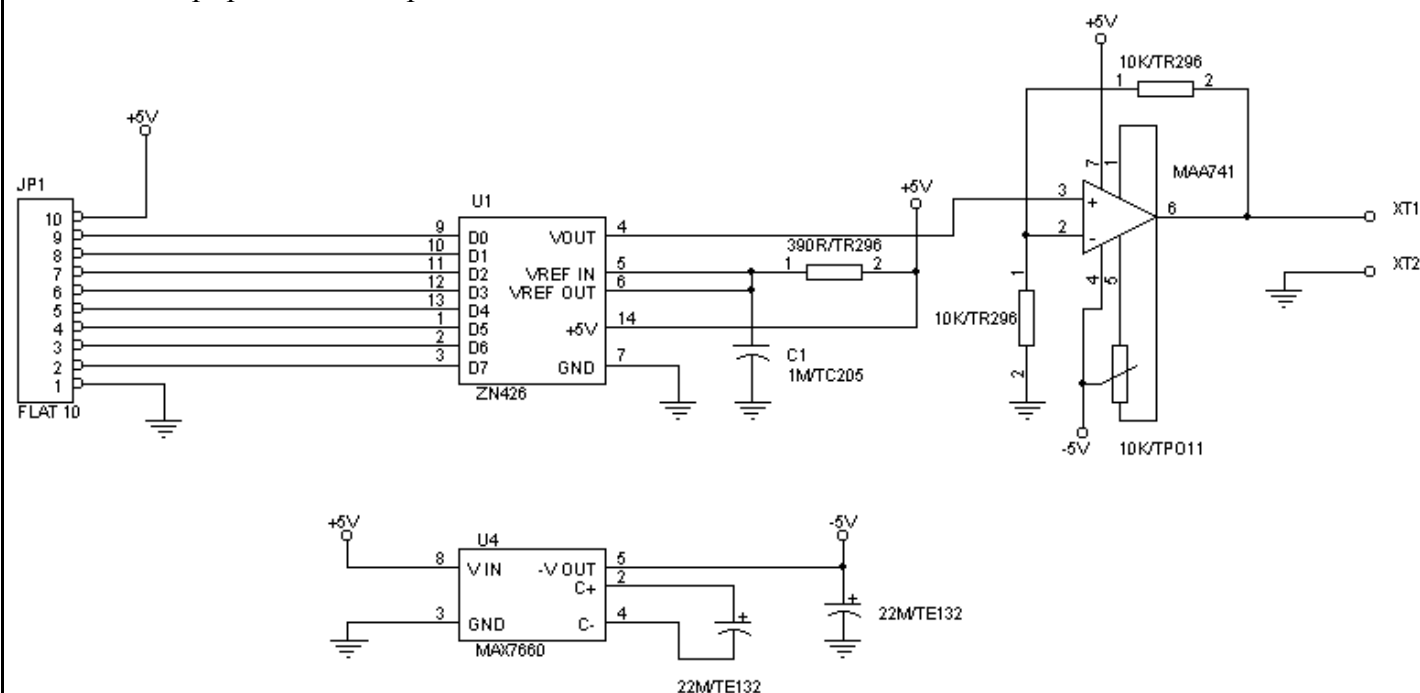
- 1) podle výstupu:
 - a) s proudovým výstupem
 - b) s napěťovým výstupem
- 2) podle vstupního kódu:
 - a) binární kód
 - b) BCD kód

D/A PŘEVODNÍK R-2R (8-bitový):

Jedná se o jednoduchý převodník realizovaný odporovou sítí R-2R. Přivedením U_{vst} na vstup získáme převodník s rozsahem 0V až U_{max} .



Školní přípravek – D/A převodník:



TEST PŘÍPRAVKU:

;20h 0 1 2 3 4 5 6 7

;21h 8 9 10 11 12 13 14 15

org 0

main:

mov r7, #11 ;nastaveni pocitadla => 11 hodnot v tabulce

mov dptr, #tab ;adresu tabulky vlozime do ukazatele dat

repeat:

mov a, #0 ;vymazeme akumulator

movc a, @a+dptr

mov 21h, a ;na tuto adresu dame hodnotu z akumulatoru, kterou cheme reverzovat

call rever ;reverzace => prehozeni bitu

mov p2, 20h ;reverzovanou hodnotu vysleme na port2

inc dptr ;posuneme se v ukazateli dat

djnz r7, repeat ;opakujeme tolikrat, kolik je hodnot v tabulce => dokud nevykreslime cely prubeh

signalu

jmp main ;vse se opakuje od zacatku

rever:

mov c, 15

mov 0, c

mov c, 14

mov 1, c

mov c, 13

mov 2, c

mov c, 12

mov 3, c

mov c, 11

mov 4, c

mov c, 10

mov 5, c

mov c, 9

mov 6, c

mov c, 8

mov 7, c

ret

tab: db 0, 25, 50, 75, 100, 125, 150, 175, 200, 225, 250

nop

end

VÝPOČTY:

Všechny průběhy s $f = 100$ Hz:

$$\text{doba trvání periody } T = \frac{1}{f} = \frac{1}{100} = 0,0010 \text{ s} = 10\text{ms}$$

1) pila nahoru

$$\text{doba trvání 1 vzorku } \frac{T}{\text{počet vzorků}} = \frac{T}{18} = \frac{10000}{18} = 555,56 \text{ us}$$

2) pila dolů

$$\text{doba trvání 1 vzorku} = \frac{T}{\text{počet vzorků}} = \frac{T}{18} = \frac{10000}{18} = 555,56 \text{ us}$$

3) trojúhelníkový průběh

$$\text{doba trvání 1 vzorku} = \frac{T}{\text{počet vzorků}} = \frac{T}{21} = \frac{10000}{21} = 476,19 \text{ us}$$

4) obdelníkový průběh

$$\text{doba trvání 1 vzorku} = \frac{T}{\text{počet vzorků}} = \frac{T}{2} = \frac{10000}{2} = 5 \text{ ms}$$

5) sinusový průběh

$$\text{doba trvání 1 vzorku} = \frac{T}{\text{počet vzorků}} = \frac{T}{24} = \frac{10000}{24} = 416,67 \text{ us}$$

$$1 \text{ vzorek } k + \sin(x) \cdot k = 128 + \sin(0) \cdot 128 = 128$$

SAMOTNÝ PROGRAM:

```
org 0
```

main:

```
jnb P0.0, SawUp ;bude se provadet pila nahoru, jestli bude bit P0.0 = 0
jnb P0.1, SawDown ;bude se provadet pila dolu, jestli bude bit P0.1 = 0
jnb P0.2, Triangle ;bude se provadet trojuhelnik, jestli bude bit P0.2 = 0
jnb P0.3, Rectangle ;bude se provadet obdelnik, jestli bude bit P0.3 = 0
jnb P0.4, Sinus ;bude se provadet sinus, jestli bude bit P0.4 = 0
```

```
jmp main ;bity bude porad hledat, dokud se nektery nezmeni => pak bude provadet
urcity prubeh
```

SawUp:

```
mov r7, #18 ;nastaveni pocitadla, do r7 dame pocet hodnot v tabulce
mov dptr, #SawUpTab ;do ukazatele dat dame adresu, kde zacina tabulka
```

rSawUp:

```
mov a, #0 ;vynuluje akumulator
movc a, @a+dptr ;da do akumulatoru hodnotu z tabulky
mov 21h, a ;hodnotu z akumulatoru da na adresu 21h
call rever ;provede reverzaci
mov P2, 20h ;po reverzaci je hodnota na adrese 20h, kterou vysleme na port2
inc dptr ;posuneme se v ukazateli dat o 1
call fSaw ;zpozdeni
call fSaw
```

```
nop ;zpozdeni => abychom měli frekvenci co nejbliže 100 Hz
```

```
nop
```

```
nop
```

```
nop
```

```
nop
```

```
nop
```

```
djnz r7, rSawUp ;pocitadlo hodnot v tabulce
```

```
jmp main ;skoci na zacatek, aby zkontroloval, jestli ma dal provadel pilu nahoru =>
```

jestli je P0.0 = 0

SawDown:

```
mov r7, #18 ;nastaveni pocitadla, do r7 dame pocet hodnot v tabulce
mov dptr, #SawDownTab ;do ukazatele dat dame adresu, kde zacina tabulka
```

rSawDown:

```
mov a, #0 ;vynuluje akumulator
movc a, @a+dptr ;da do akumulatoru hodnotu z tabulky
mov 21h, a ;hodnotu z akumulatoru da na adresu 21h
call rever ;provede reverzaci
mov P2, 20h ;po reverzaci je hodnota na adrese 20h, kterou vysleme na port2
inc dptr ;posuneme se v ukazateli dat o 1
call fSaw ;zpozdeni
call fSaw
```

```
nop ;zpozdeni => abychom měli frekvenci co nejlize 100 Hz
```

```
nop
```

```
nop
```

```
nop
```

```
nop
```

```
nop
```

```
djnz r7, rSawDown ;pocitadlo hodnot v tabulce
```

```
jmp main ;skoci na zacatek, aby zkontroloval, jestli ma dal provadel pilu dolu =>
```

jestli je P0.1 = 0

Triangle:

```
mov r7, #21 ;nastaveni pocitadla, do r7 dame pocet hodnot v tabulce
mov dptr, #TriangleTab ;do ukazatele dat dame adresu, kde zacina tabulka
```

rTriangle:

```
mov a, #0 ;vynuluje akumulator
movc a, @a+dptr ;da do akumulatoru hodnotu z tabulky
mov 21h, a ;hodnotu z akumulatoru da na adresu 21h
call rever ;provede reverzaci
mov P2, 20h ;po reverzaci je hodnota na adrese 20h, kterou vysleme na port2
inc dptr ;posuneme se v ukazateli dat o 1
call fTriangle ;zpozdeni
call fTriangle
```

```
nop ;zpozdeni => abychom měli frekvenci co nejlize 100 Hz
```

```
nop
```

```
djnz r7, rTriangle ;pocitadlo hodnot v tabulce
```

```
jmp main ;skoci na zacatek, aby zkontroloval, jestli ma dal provadel trojuhelnik =>
```

jestli je P0.2 = 0

Rectangle:

```
mov P2, #255 ;na port2 vysleme hodnotu #255, aby byla amplituda obdelniku max.
call fRectangle ;zpozdeni
call fRectangle
```

```
mov P2, #0 ;na port2 vysleme hodnotu #0 => druha polovina prubehu obdelniku
```

```
call fRectangle
```

```
call fRectangle
```

```
jmp main ;skoci na zacatek, aby zkontroloval, jestli ma dal provadel obdelnik =>
```

jestli je P0.3 = 0

Sinus:

```
mov r7, #24 ;nastaveni pocitadla, do r7 dame pocet hodnot v tabulce
mov dptr, #SinusTab ;do ukazatele dat dame adresu, kde zacina tabulka
```

rSinus:

```

mov    a, #0                ;vynuluje akumulator
movc   a, @a+dptr           ;da do akumulatoru hodnotu z tabulky
mov    21h, a               ;hodnotu z akumulatoru da na adresu 21h
call   rever                ;provede reverzaci
mov    P2, 20h              ;po reverzaci je hodnota na adrese 20h, kterou vysleme na port2
inc    dptr                 ;posuneme se v ukazateli dat o 1
call   fSinus               ;zpozdeni
call   fSinus
djnz   r7, rSinus           ;pocitadlo hodnot v tabulce
jmp    main                 ;skoci na zacatek, aby zkontroloval, jestli ma dal provadel obdelnik =>
jestli je P0.4 = 0

rever:
mov    c, 15
mov    0, c
mov    c, 14
mov    1, c
mov    c, 13
mov    2, c
mov    c, 12
mov    3, c
mov    c, 11
mov    4, c
mov    c, 10
mov    5, c
mov    c, 9
mov    6, c
mov    c, 8
mov    7, c
ret

fSaw:                          ;zpozdeni pro prubehy => pila nahoru a dolu
mov    r6, #255

fSaw2:
djnz   r6, fSaw2
ret

fTriangle:                     ;zpozdeni pro trojuhelnikovy prubeh
mov    r6, #216

fTriangle2:
djnz   r6, fTriangle2
ret

fRectangle:                    ;zpozdeni pro obdelnikovy prubeh
mov    r6, #200
mov    r5, #10

fRectangle2:
djnz   r6, fRectangle2
djnz   r5, fRectangle2
ret

fSinus:                        ;zpozdeni pro sinusovy prubeh
mov    r6, #175

fSinus2:
djnz   r6, fSinus2
ret

```

SawUpTab:

db 0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 255

SawDownTab:

db 255, 240, 225, 210, 195, 180, 165, 150, 135, 120, 105, 90, 75, 60, 45, 30, 15, 0

TriangleTab:

db 0, 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 225, 200, 175, 150, 125, 100, 75, 50, 25, 0

SinusTab:

db 128, 161, 192, 219, 239, 252, 255, 252, 239, 219, 192, 161, 128, 95, 64, 37, 17, 4, 0, 4, 17, 37, 64, 95

nop

end

ZHODNOCENÍ:

Úloha D/A převodník je jedna z jednodušších úloh. Jelikož jsme už realizovali minulý rok, nebylo těžké napsat program pro generování všech pěti signálů. Kámen úrazu byl ovšem opět jako loni ve stanovení správného zpoždění, aby měl průběh frekvenci přesně 100 Hz. Protože jsem program v Keilu krokoval, povedlo se mi jak jsem si myslel správně odečíst dobu vykreslení jednoho vzorku průběhu. Program jsem odzkoušel a zjistil, že jsem někde udělal chybu a všechny průběhy měly dvojnásobnou frekvenci, tedy 200 Hz místo 100 Hz. Bohužel jsem měl již málo času k předvedení úlohy a provedl jsem neprogramátorský krok a zpoždění všude ještě jednou nakopíroval, aby frekvence průběhu byla už opravdu 100 Hz. Sinusový průběh jsem taky neměl přesně 100 Hz, tomu by pomohlo, kdybych dal do zpožďovací smyčky o něco větší hodnotu.

Program by šel určitě dále zjednodušit, ale mě napadlo zrovna tohle řešení. Například by se mohly sjednotit všechny tabulky na stejnou délku, tedy, aby všechny průběhy měly stejný počet vzorků => aby v programu byla jen jedna zpožďovací smyčka pro všechny průběhy. Taky by se dala ještě více sjednotit reverzace společně s posunem v tabulce a výslaním vzorku na port 2, všechno by se to mohlo dát do jednoho podprogramu.