

Vaja 9

Digitalno-analogni (D/A) in analogno-digitalni (A/D) pretvornik

- 1. Napišite program, ki bo na izhodu D/A pretvornika generiral nizko frekvenčni signal trikotne oz. sinusne oblike.**
- 2. Oblika signala naj bo nastavljiva s tipkama T0 (trikotni) in T1 (sinus)**
- 3. Frekvenca signala naj bo nastavljiva s potenciometrom, ki je vezan na vhod A/D pretvornika.**
- 4. Na LCD prikazovalniku naj bo izpisana nastavljena frekvenca in oblika signala**

Frekvenca : 1-10Hz nastavljiva s potenciometrom
Izbira signala : T0 – trikoten, T1 – sinus
LCD : prikazuje izbran signal in frekvenco
npr:

f = 3 Hz
signal = triang

- 1. Izvedba z zanko**
- 2. Izvedba s prekinitvami**
- 3. Dodatek: izvedba z RTOS (uporabite gonilnike za LCD in tipke)**

D/A pretvornik (adc_dac.h, adc_dac.c)

Analogen signal generiramo s pomočjo D/A pretvornika:

- 10 biten => digitalni vzorci 0-1023 -> analogna napetost 0-3.3V
- digitalne vzorce v določenem časovnem zaporedju pretvarjamo v analogno napetost
- dostopen preko vrat P13 (med DAC pinom P0_25 na LPC2138 in vrati P13 je še ojačevalnik)

Inicializacija:

dac_init();

Pretvorba:

DACR register : vrednost ki se zapiše v bite bit15...bit6 se pretvori v analogno napetost

npr.

DACR = 1023 << 6; / na izhodu D/A pretvornika se pojavi napetost 3.3V (malo manj)*/*

A/D pretvornik (adc_dac.h, adc_dac.c)

Frekvenco nastavljamo s potenciometrom, ki je priključen na vhod A/D pretvornika:

- LPC2138 ima dva 8-kanalna pretvornika AD0, AD1
- 10 biten => analogna napetost 0-3.3V -> digitalni vzorci 0-1023
- potenciometer je vezan na pin P0_27 na katerem je na voljo kanal AD0.0

Inicializacija (AD0):

adc0_init(adc0_pins0, adc0_pins1, mode);

adc0_pins0 : maska za izbiro A/D funkcija na pinih 0-15:

0, adc0_6 (P0.4), adc0_7 (P0.5)

adc0_pins1 : maska za izbiro A/D funkcija na pinih 15-31 :

0, adc0_0 (P0.27), adc0_1 (P0.28), adc0_2 (P0.29),
adc0_3 (P0.30) , adc0_4 (P0.25), adc0_5 (P0.26)

mode : način pretvorbe: 0-zahtevana, 4,5,6...11 – neprekinjena (št. ciklov oz. natančnost pretvorbe)

A/D pretvornik (adc_dac.h, adc_dac.c)

A/D pretvorba:

AD0CR register:

- kontrolni register za izbiro načina pretvorbe (neprekinjena ali zahtevana)
- enkratno pretvorbo zahtevamo z vpisom 1 na bit24 (maska *start_now*):

AD0CR = AD0CR | start_now;

AD0DR register:

- podatkovni register
- biti bit15...bit6 (maska *ls*) vsebujejo pretvorjeno vrednost
- pretvorjena vrednost je veljavna, ko je bit31 (maska *done*) postavljen
(AD0DR & done) != 0 => preberi vrednost *val=(AD0DR & ls)>>6;*

Generiranje analognih signalov:

Signale tabeliramo: digitalne vzorce za trikotni in sinusni signal shranimo v zbirke

```
#define NS 10      /* št. vzorcev */
...
int triang[NS];    /* tabela vzorcev za trikotni signal */
int sine[NS];      /* tabela vzorcev za sinusni signal */
int f=1;           /* frekvenca generiranega signala */
int nciklov;       /* št. ciklov VPB do nasl. zamenjave vzorca na D/A */
int idx=0;         /* indeks trenutnega vzorca */
...
int *tabela;       /* kazalec na trenutno izbran signal */

...
tabela = triang;   /* izberemo trikotni signal */
...
tabela = sine;     /* izberemo sinusni signal */
...
```

Generiranje analognih signalov:

Simetrični signali => tabeliramo $\frac{1}{2}$ periode (lahko bi tudi samo $\frac{1}{4}$ => težja implementacija)

=> vzorce beremo naprej in nato nazaj 0,1,2,...NS-2, NS-1, NS-2,...2,1,0,...

=> $2 \cdot (NS-1)$ vzorcev za 1 periodo signala

=> V tabeli moramo zajeti celoten razpon vrednosti signala (0-1023)

Za $i=0,1,2,\dots,NS-1$

*triang[i] = 1023 * i / (NS-1);*

*sine[i] = 1023 * (sin((2 * i + NS - 1) * 3.14 / (2 * (NS - 1))) + 1) / 2;*

Zaporedna vzorca sta razmaknjena za:

$$t = 1 / (2 \cdot (NS-1) \cdot f)$$

Št. ciklov vodila VPB med dvema zaporednima vzorcema :

*nciklov = fVPB / (2 * (NS-1) * f);* */* oz. t*fVPB * /*

Izvedba z zanko:

start_up():

1. Inicializiramo timer1

main():

1. Inicializiramo spremenljivke, tabeliramo signale, določimo št. ciklov med vzorci

2. Poženemo števec timer1

3. gledamo stanje števca in ob primernih trenutkih zapišemo trenutni vzorec signala na D/A pretvornik

4. gledamo tipke in zamenjamo signal, če je potrebno

5. Preberemo vrednost z A/D pretvornika, preračunamo št. ciklov med vzorci, če je potrebno

6. Pripravimo niz za LCD

7. Pokličemo lcd_driver()

Q: kakšno je največje št. vzorcev NS, za katere signal še pravilno generiramo

Q: kakšna je največja frekvenca signala, ki ga še lahko generiramo na ta način

Izvedba s prekinitvami:

Časovno kritičen del programa prepustimo strojni opremi (timer1):

Uporabimo timer1

=> inicializacija timer1

=> inicializacija VIC

Prekinitvena rutina

- 1. skrbi samo za pravočasen zapis trenutnega vzorca na D/A pretvornik (in pobriše prekinitvene zastavice ter resetira VIC)**

main():

- 1. Skrbi za vse kar ni časovno kritično**
- 2. Lahko skrbi za tipke in izbiro signala**
- 3. bere A/D pretvornik (potenciometer) in preračunavanje frekvenc in št. ciklov**
- 4. priprava teksta za LCD, klic gonilnika za LCD**

Izvedba z RTOS:

RTOS preko gonilnikov skrbi za:

tipke

LCD

periodičen zapis vzorca na D/A pretvornik

V funkciji main():

beremo iz medpomnilnika za tipke in menjamo obliko signala, če je potrebno

beremo A/D pretvornik in menjamo frekvenco

pripravimo besedilo za LCD