

---

Nikola Knežević

## Povezivanje A/D konvertora na PC računar u funkciji simuliranja osciloskopa

---

*Cilj ovog projekta je bilo simuliranje osciloskopa pomoću PC računara i A/D konvertora povezanog na paralelni port računara. Napravljen je uređaj koji se može koristiti za dobijanje grafika funkcija koje daje generator funkcija sa maksimalnom amplitudom od 5 V. Takođe, realizovano je i očitavanje za promenljivu vremensku bazu. Greška uređaja 0.14 V, tako da se on može primenjivati u slučajevima kada nam je samo dovoljan oblik, a ne potpuno tačne vrednosti na grafiku.*

---

### Uvod

#### Povezivanje na paralelni port

Svaki paralelni port ima 3 registra: *port a* ima 8 bita za digitalni izlaz, *port b* ima 5 bita za digitalni ulaz i *port c* ima 4 bita koji se mogu definisati ili za ulaz ili za izlaz (Werhimer 1996).

Pinovi paralelnog porta:

pin – funkcija

- 1 port c – prvi bit, vrednost 1 (least significant bit – LSB)
- 2 port a – prvi bit, vrednost 1 (LSB)
- 3 port a – drugi bit
- 4 port a – treći bit
- 5 port a – četvrti bit
- 6 port a – peti bit
- 7 port a – šesti bit
- 8 port a – sedmi bit
- 9 port a – osmi bit (most significant bit)
- 10 port b – sedmi bit
- 11 port b – osmi bit – invertovan

---

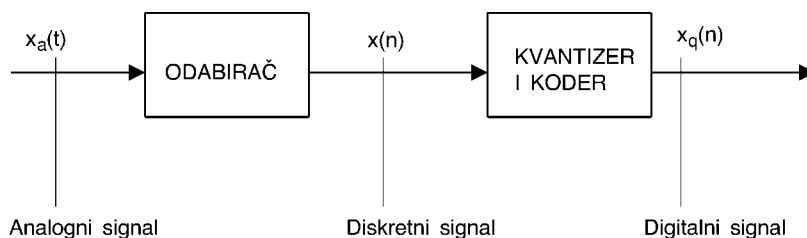
*Nikola Knežević  
(1980), Beograd,  
Vladimira Popovića  
34, učenik 3. razreda  
Matematičke  
gimnazije u Beogradu*

- 12 port b – šesti bit
- 13 port b – peti bit
- 14 port c – drugi bit – invertovan
- 15 port b – četvrti bit
- 16 port c – treći bit
- 17 port c – četvrti bit – invertovan
- 18–25 – masa

Iako neki paralelni portovi imaju 'bi-direkcionu' konfiguraciju, koja dozvoljava da port a može da se koristi i za ulaz i za izlaz, mnogi stariji kompjuteri nemaju ovu mogućnost i port a se može koristiti samo za izlaz.

### Semplovanje i konverzija

U cilju obrade signala koji se javljaju u prirodi korišćenjem računarskih sistema, signali se moraju konvertovati u digitalni oblik, tj. u niz binarnih brojeva. Pri tome se obavlja proces dvostruke diskretizacije: diskretizacija po vremenu (odabiranje, odmeravanje, engl. sampling) i diskretizacija po amplitudi (kvantovanje, engl. quantization). Kompletan proces digitalizacije naziva se analogno-digitalna (A/D) konverzija i obavlja se pomoću A/D konvertora. Proces A/D konverzije može se predstaviti u vidu blok šeme na sledećoj šemi (slika 1):



Slika 1.  
Blok-šema A/D konverzije.

Figure 1.  
The block scheme of an A/D conversion.

Pod analognim signalom podrazumevamo kontinualni signal kod koga amplituda signala može imati bilo koju vrednost iz dozvoljenog opsega. Druga važna klasa signala definisana je samo u diskretnim vremenskim trenucima, pa se takvi signali nazivaju vremenski diskretni signali. Amplituda diskretnih signala je kontinualna veličina iz dozvoljenog opsega. Ako se i amplituda diskretnog signala diskretizuje, dobija se digitalni signal koji je jedini signal koji se može obrađivati računarom (Popović 1997).

A/D konvertor služi da analogni signal koji daje neki izvor, ili u ovom slučaju generator funkcija, pretvori u digitalni signal. Time se postiže da se očitavanjem sa A/D konvertora dobije trenutni napon koji daje izvor. Tada se preko paralelnog porta očitaju vrednosti pinova A/D konvertora i to se prevede u napon po formuli:

$$\text{Napon} = \text{binar. izlaz A/D konvertora} \times \text{max. napon} / \text{max. bin. broj} \quad (1)$$

gde se pod opsegom merenja podrazumeva maksimalni napon koji dati A/D konvertor može konvertovati u digitalni signal, a maksimalni binarni broj zavisi od rezolucije A/D konvertora. Od opsega i maksimalnog binarnog broja koji A/D konvertor može da obradi zavisi i greška merenja napona. Na primer, kod 8-bitnog konvertora uzima se da greška iznosi opseg/255.

Veličina maksimalnog napona koji se može konvertovati zavisi od A/D konvertora. Neki A/D konvertori mogu konvertovati samo napone u opsegu od 0 do 5 V, pa su pogodni za rad sa PC računarima jer se naponski nivoi A/D konvertora poklapaju sa ulaznim logičkim kolima za PC (tj. minimalni napon koji printerski port može prepoznati je 0 V, a maksimalni 5 V, bez opasnosti po port). Za takvu realizaciju mi smo koristili konvertor ADC0803, a pored njega mogli smo koristiti sledeće A/D konvertore, čija je specifikacija data na slici 3: ADC0801, ADC0802, ADC0803, ADC0804 i ADC0805.

Brzina konverzije ovih A/D konvertora je 100 s, a rade na principu sukcesivne aproksimacije. Ovaj metod se zasniva na uzastopnim promenama referentnog napona A/D konvertora u zavisnosti da li komparator (koji se nalazi u sastavu A/D konvertora) pokazuje da li je ulazni napon veći ili manji od referentnog. Promena stanja prestaje kada se izlaz komparatora (0 ili 1) promeni. Na taj način dobija se približna vrednost ulaznog napona.

## Aparatura

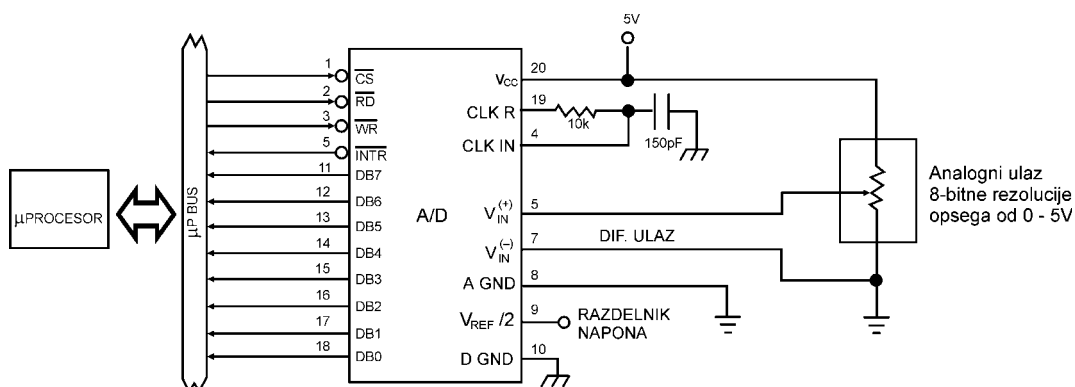
### Komponente

Za realizaciju ovog projekta korišćeni su sledeći delovi:

1. jedan protoboard
2. jedan A/D konvertor (u ovom projektu korišćen je Philips ADC0803LCN)
3. otpornik od 10 k $\Omega$
4. kondenzator od 150 pF
5. dva otpornika jednakih otpornosti reda veličine više k (ovde su korišćeni otpornici od 3.3 k $\Omega$ ), u funkciji razdelnika napona da bi se na pinu 9 dobilo  $V_{\text{ref}}/2$
6. 25 pinski sub-D muški konektor
7. PC računar sa paralelnim portom

## Realizacija

Povezivanjem ovih komponenti kao na slici 2 i startovanjem programa koji sam napisao, imamo mogućnost direktnog očitavanja vrednosti napona, ali sa kašnjenjem koje unosi sporo očitavanje računara sa A/D konvertora. Brzina ovog programa je oko 5700 semplova u sekundi.



Prema Nikvistovoj (Nyquist) teoremi o odabiranju, koja kaže da se potpuna rekonstrukcija signala iz njegovih odabiraka može izvršiti ako je učestanost odabiranja bar dva puta veća od najviše učestanosti u spektru signala, imamo da se, s obzirom na maksimalnu brzinu semplovanja (5700 u sekundi), signali mogu verno očitavati i predstaviti na ekranu do maksimalne frekvencije spektra signala od 2800 Hz. Ovo znatno ograničava primenu uređaja, ali ukoliko nam nije potrebna precizna slika, već samo približni oblik signala (npr. sinusoida), onda ovaj uređaj može naći svoju širu primenu.

U nastavku je dat program koji sam napisao u programskom jeziku BORLAND TURBO PASCAL 7.0 i on simulira rad osciloskopa, ali je zbog nedostatka bi-direkcionog porta korišćen port b umesto porta a i tada je preciznost merenja bila manja zbog toga što se na portu b mogu koristiti samo 5 pinova za čitanje, pa se opseg smanjio sa 256 na 32 binarna broja što dovodi do toga da formula (1) sada izgleda ovako:

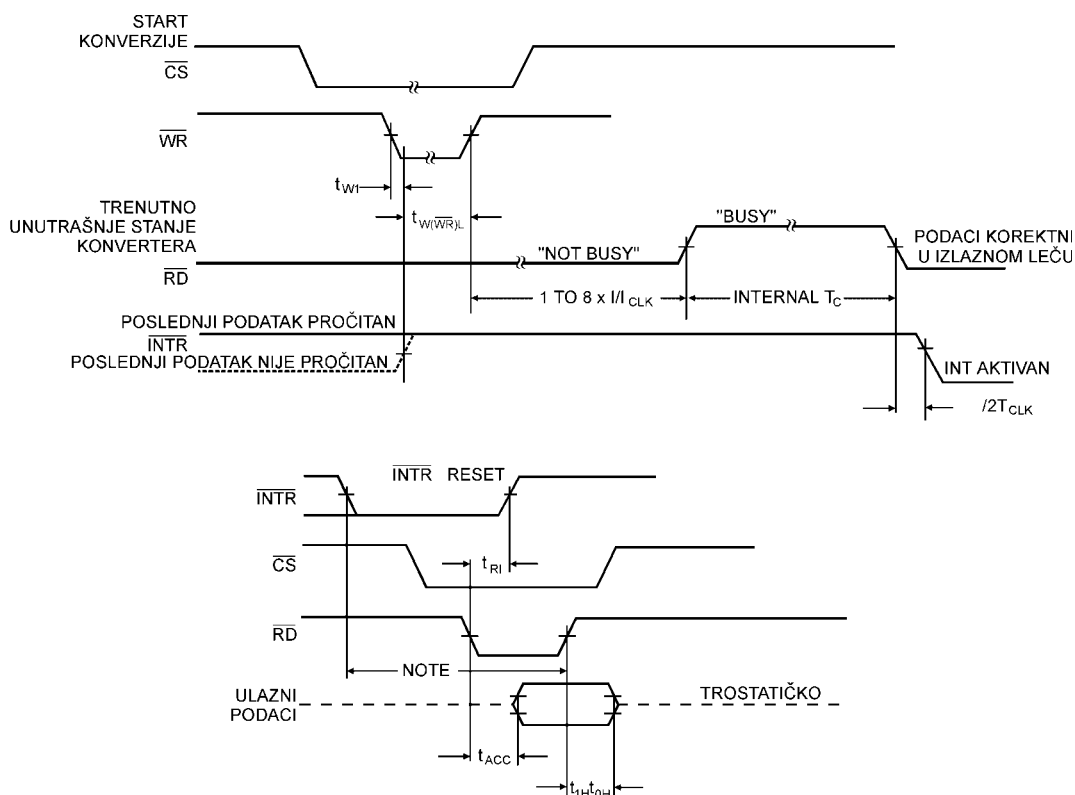
$$\text{Napon [V]} = \text{dekadna predstava dobijenog binarnog broja} \times 5/31,$$

s obzirom da je najveći broj sada 31.

U programu se nalazi deo koda za pretvaranje iz 8-bitnog u 5-bitni broj. To se radi odsecanjem zadnja tri bita pročitana sa porta i tada, za dati maksimalni napon koji se može konvertovati, greška iznosi 0.14 V.

Slika 2.  
Način povezivanja  
A/D konvertora i  
računara.  
(NS 1994).

Figure 2.  
The way to connect  
the A/D converter  
with the PC.



Karakteristike korišćenog A/D konvertora date su na slici 3, gde se sa vremenskih dijagrama vidi kako se startuje konverzija i kako se učitavaju podaci sa A/D konvertora.

Na slici 3 su sa INT, CS, RD i WR označene komande A/D konvertoru koje znače:

- INT – interrupt i služi za proveru stanja A/D konvertora
- CS – chip select
- RD – read
- WR – write.

Podaci dati na slikama 2 i 3 uzeti su za National Semiconductor čip, iako se radilo sa Philips čipom. Specifikacija je ista jer su u pitanju isti modeli čipova.

Pri izradi uređaja i izračunavanju brzine semplovanja korišćen je računar sledeće konfiguracije: PC 486 DX4-S/100 MHz, 8 Mb RAM, 16 bitni I/O kontroler, 425 Mb Western Digital Hard Drive. Program je podjednako dobro radio i pod DOS i pod Windows 95 operativnim sistemom. Za ubrzanje programa (da se ne bi ponovo pisali kritični delovi programa u Asembleru), može se koristiti računar sa boljim I/O kontrolerom.

Slika 3.  
Vremenski dijagrami za start konverzije i čitanje podataka (NS 1994).

Figure 3.  
Time diagrams for starting the conversion and reading the data.

Program je računarska simulacija osciloskopa koji radi na principu očitavanja sa A/D konvertora povezanog na paralelni port računara. Program očitava binarnu vrednost napona i pretvara je u voltažu na sledećem principu: pošto je opseg 5 V, a binarni brojevi se kreću od 0 do 11111 (31 u dekadnom) prava vrednost napona dobija se množenjem binarnog broja sa 5/31.

Konstante i promenljive koje se koriste u programu definisane su na sledeći način:

```

const
  AdrPortA = 888; {adresa porta a}
  AdrPortB = 889; {adresa porta b}
  AdrPortC = 890; {adresa porta c}
  maxarray = 500; {maksimalni broj merenja}
  CS=1;          {aktivan 0001}
  RD=8;          {aktivan 1000}
  WR=4;          {neaktivan 0100}
  inicijalC=4;   {inicijalizacija porta c za upis}
  gornjilevix=00;
  gornjileviy=00;
  donjidesnix=639;
  donjidesniy=479;
  opsegx=donjidesnix-gornjilevix;
  kx=639/opsegx;
  opsegy=donjidesniy-gornjileviy;
  ky=479/opsegy;
  brx=trunc(opsegx/18);
  bry=trunc(opsegy/15);

type
  mat = array[1..maxarray] of byte; {matrica za
  skladistenje uzoraka}
  dat = file of byte; {fajl za skladistenje podataka}

var
  x,y,i:integer;
  grDriver,Grmode:integer;
  a:mat;
  tacka:real;
  c:char;
  vreme:byte;

```

Procedura koja služi da se zada start konverzije A/D konvertoru da bi posle bilo moguće očitati vrednost i nacrtati datu tačku na grafiku izgleda ovako:

```

procedure startuj_konverziju;
{procedura za zadavanje starta konverzije A/D konver-
toru
start konverzije izvrsava niz naredbi koje A/D kon-
vertoru nalazu da pocne sa konverzijom pomocu kon-

```

trolnih bitova u A/D konvertoru; ne postoje side efekti jer program ne menja globalne promenljive vec samo upisuje u kontrolne bitove porta}

```

var
  x:integer;

begin
  port[AdrPortA]:=0; {dodeljivanje logickih jedinica
  svim bitovima porta a}
  port[AdrPortC]:=WR;
  delay(1);
  port[AdrPortC]:=WR+CS;
  delay(1);
  port[AdrPortC]:=CS;
  delay(1);
  port[AdrPortC]:=CS+WR;
  delay(1);
  port[AdrPortC]:=WR;
  delay(1);
  port[AdrPortC]:=CS+WR;
  delay(1);
  port[AdrPortC]:=CS+WR+RD;
end;

```

Još jedna veoma važna procedura je procedura za štampanje očitanih podataka na ekranu. Ona koristi standardne Pascal procedure iz modula graph.tpu kao što su *line*, *outtexty*, *putpixel* i druge.

```

procedure stampa;
{procedura služi za crtanje grafika na osnovu po-
dataka upisanih u datoteku koja je dobijena pomoću
procedure nova merenja i nema nikakvih side efekata
jer samo cita iz datoteke i nista u nju ne upisuje}

var
  i:integer;
  f:dat;
  s:string;
  x:byte;
begin
  grdriver:=detect;
  Initgraph(grdriver,grmode,'c:\tp\bgi');{inicijali-
  zacija grafika}
  setcolor(1);
  for i:=2 to 16 do

    line(gornjilevix+i*brx,gornjileviy,gornjilevix
    +i*brx,donjidesniy-30);
    for i:=0 to 13 do

      line(gornjilevix+30,gornjileviy+i*bry,donjidesnix
      -60,gornjileviy+i*bry);

```

```

setcolor(7);
line(gornjilevix+30,gornjileviy,gornjilevix+30,
donjidesniy-30);
line(gornjilevix+30,donjidesniy-30,donjidesnix-60,
donjidesniy-30);{crtanje koordinatnih osa}
line(gornjilevix+30,gornjileviy,donjidesnix-60,
gornjileviy);
line(donjidesnix-60,donjidesniy-30,donjidesnix-60,
gornjileviy);
for i:=1 to maxarray do {petlja za crtanje svih
tacaka}
putpixel(gornjilevix+i+30,round((donjidesniy-30)/
ky-opsegy*a[i]/(31*ky)),green);
readln;
closegraph;
end;

```

Procedura *novomerenje* služi za očitavanje vrednosti koju je A/D konvertor poslao PC računaru. Ti se podaci skladište u globalni niz, za kasnije korišćenje ili štampanje.

```

procedure novomerenje;
{procedura očitava trenutno stanje porta tj. uzima
konvertovane podatke sa A/D konvertora.
Postoje side efekti,a to su da se podaci prethodnog
merenja gube nepovratno i preko njih se upisuju
rezultati novog merenja}

var x:byte;
    zakasni:integer;

begin
  clrscr;
  write('Unesite vreme očitavanja u sec. ');
  readln(vreme);
  for i:=1 to maxarray do
  begin
    startuj_konverziju; {startovanje konverzije}
    delay(1);
    port[AdrPortC]:=WR+CS; {aktivan CS, neaktivan RD
i WR}
    delay(1);
    port[AdrPortC]:=WR+CS+RD; {inicijalizacija
citanja sa A/D}
    delay(1);
    port[AdrPortA]:=0;
    delay(1);
    a[i]:=(port[AdrPortB]); {citanje vrednosti koje
A/D konvertor daje na svom izlazu}
    a[i]:=a[i]+128;
    a[i]:=a[i]div 8;
    port[AdrPortC]:=WR+CS; {vracanje na staro tj. nova
inicijalizacija porta c za sledeci zorak}
  end;
end;

```



```
    delay(1);
    port[AdrPortC] := WR;
    end;
end;
```

Prethodne procedure mogu se importovati u program koji bi ih iskoristio za povezivanje tačaka i, recimo, iscrtavanje kontinualnog grafika. Procedure *stampaj* štampa samo pojedinačne tačke, bez njihovog povezivanja.

## Zaključak

Ovim postupkom moguće je do grafika željene funkcije doći samo uz pomoć računara, bez korišćenja osciloskopa. U praksi računar daje nepreciznije podatke od osciloskopa. To se naročito pokazalo pri izradi ovog uređaja, usled toga što se 8-bitni konvertor morao koristiti kao 5-bitni.

Preciznost se može povećati korišćenjem 8-bitnog porta, ili zamenjivanjem 8-bitnog A/D konvertora nekim konvertorom sa većom rezolucijom. Tako bi se dobilo na preciznosti, ali bi se izgubilo na brzini, jer se tada očitavanje sa uređaja mora vršiti u više ciklusa. Brzina je ograničena i brzinom A/D konvertora, odnosno njegovom nemogućnošću da izvrši onoliko konverzija koliko to od njega PC računar može da zahteva.

## Komentar

Pri realizaciji ovog uređaja naišli smo na neke probleme na koje treba ukazati onima koji bi pokušali realizaciju nečeg sličnog. Prvo, A/D konvertor nije dovoljno brz da obradi onoliko podataka u sekundi, koliko računar to od njega može zahtevati. Zato je neophodno da se između svake dve naredbe A/D konvertoru ubaci pauza (delay). Tako se program znatno uspori, ali ne dolazi do zagušenja koje može prouzrokovati dobijanje pogrešnih podataka. Drugo, nije potrebno čitati INT, jer će i tada doći do pucanja programa pošto A/D konvertor veoma sporo šalje obaveštenje. Program radi mnogo bolje kada se INT zanemari i time se i štedi jedan bit na portu. Sve ostalo treba uraditi po navedenoj šemi i uređaj će dobro raditi.

---

## Literatura

NS (National Semiconductor) 1994. NationalSemiconductor ADC0801 / ADC0802 / ADC0803 / ADC0804 / ADC0805 8-bit P Compatible A/D Converters Catalogue.

Popović, M. 1997. *Digitalna obrada signala*. Beograd: Naučna knjiga

Werhimer, D. 1996. Inexpensive interfacing experiments on the PC. Griep International Conference, Ljubljana, Slovenia.

## Connecting an A/D Converter to a PC Computer in Order to Simulate Oscilloscope

The goal of this project has been to simulate an oscilloscope by using a PC computer and an A/D converter, connected to the computer's parallel port. The constructed device can be used to plot the output of a function generator, with a maximum amplitude of 5 V. The measurement error is 0.14 V. Taking this into account the device could effectively be used to determine the general shape of the output function.

All connections should be made as shown on the Figure 2. Due to the compatibility between the National Semiconductor ADC0801, ADC0802, ADC0803, ADC0804, ADC0805 and the Philips ADC0803 A/D converters, the data given for the National Semiconductor chips applies to Philips as well.

All the data needed for starting the conversion and reading the output value are shown on time diagrams (Figure 3). All crucial A/D commands are given on those diagrams, where INT means interrupt (last data was or was not read), CS-chip select, RD-read and WR-write.

Maximum sampling speed of this device is 5700 Hz, which means that signals with maximum frequency of 2800 Hz can correctly be shown on the screen (according to the Nyquist sampling theorem).

During the realization of this device we have experienced several problems that, I feel, should be mentioned for those who would want to try to make the same or similar device.

The first thing is that A/D converter is not fast enough to manage as many data per second as the computer can ask him to. That is why it is necessary to put a delay between every two A/D converter commands. This way, the program is slowed down, but there is no jam, which can cause wrong information reading.

The second thing concerns the INT. There is no need to read the INT, because the reading would provoke the program to crash. This happens because the converter sends the INT to slow. Program works much better if the INT is not taken in consideration thus, we also save one bit on the port. Everything else should be done as shown on the Figure 2 and the device will work properly.

