

**Univerza v Ljubljani**  
Fakulteta za elektrotehniko  
Študijsko leto: 2000/2001  
Smer: elektronika, 4.letnik



---

# **SEMINARSKA NALOGA**

ELEKTRONSKA VEZJA

## **MODUL ZA TVORJENJE SIGNALOV ZA ELEKTROPORACIJO LIPIDNIH DVOSLOJEV**

Mentor: Marko Topič	Pripravil: Peter Kramar
------------------------	----------------------------

Ljubljana, 27.9.2002



# KAZALO

<b>1. UVOD</b>	<b>5</b>
<b>2. MATERIALI IN METODE</b>	<b>6</b>
<b>2.1. Modul za tvorjenje signalov</b>	<b>6</b>
2.1.1. Digitalno – analogni pretvornik	7
2.1.2. Napetostni izhodni del	7
2.1.2.1. <i>Unipolarno – bipolarni tokovno – napetostni pretvornik</i>	7
2.1.2.2. <i>Ojačevalnik – slabilnik</i>	8
2.1.2.3. <i>Napetostni sledilnik – ojačevalnik <math>A=2</math></i>	9
2.1.3. Časovna ločljivost	10
2.1.4. Krmiljenje	10
<b>2.2. Zgradba programa</b>	<b>11</b>
<b>2.3. Pristop k meritvam</b>	<b>11</b>
<b>3. REZULTATI</b>	<b>15</b>
<b>4. ZAKLUČEK</b>	<b>22</b>
<b>5. LITERATURA</b>	<b>23</b>
<b>6. DODATEK</b>	<b>24</b>
6.1. Izvorna koda mikroprocesorja	24
6.2. Električna shema modula za tvorjenje poljubnih signalov	32



## 1. UVOD

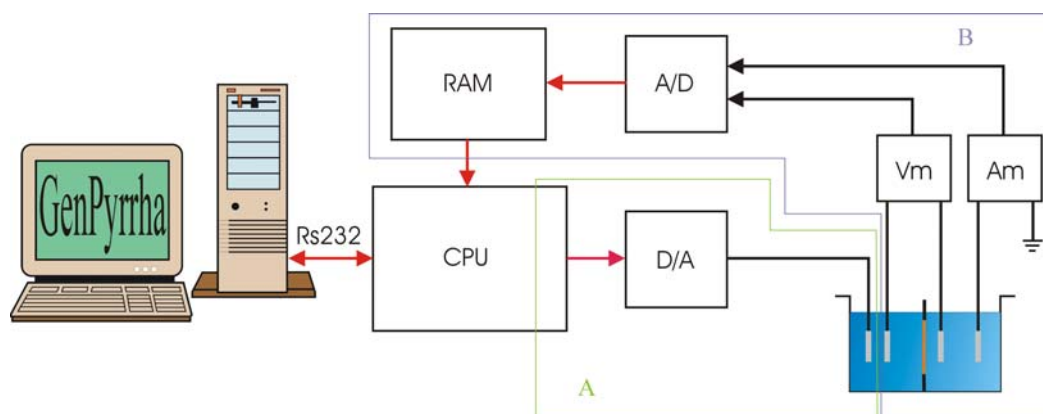
Elektroporacija je pojav, pri katerem nastanejo zaradi že relativno kratkotrajnega visokonapetostnega električnega pulza strukturne spremembe v celični membrani. Največkrat jih imenujemo kar »pore«. Ob zadostnem številu le-teh in njihovi ustrezni velikosti, se poveča prepustnost plazmaleme. To povečanje prepustnosti membrane omogoča ionom in molekulam, za katere je sicer celična membrana nepremagljiva ovira, neposreden vstop v celično notranjost. Sprememba prepustnosti je lahko ireverzibilna (celica odmre) ali reverzibilna [1].

Elektroporacijo uporabljamo za vnos snovi v celice na različnih področjih biotehnologije, biologije in medicine. Nekatera področja uporabe so: molekularna genetika, genska terapija, elektrokemoterapija, vnašanje beljakovin v celično membrano, vnašanje antibiotikov v krvne celice in vnos zdravilnih učinkovin skozi kožo. (Natančni molekularni mehanizmi elektroporacije še vedno niso dobro znani) [2].

Ravninski lipidni dvosloj je umetna membrana, ki služi kot model celične membrane. Umetno membrano si lahko predstavljamo kot diferencialno majhen košček celične membrane. Z njim lahko enostavneje iščemo vplivne parametre poracije, na primer frekvenco, napetost in obliko signala [3].

Za elektroporacijo ravninskih lipidnih dvoslojev potrebujemo sistem ki zmore tvoriti poljubne signale do amplitude enega volta ter merilnik toka in napetosti. V tem delu bomo opisali modul za tvorjenje signalov, ki pa omogoča generiranje signalov napetosti do  $\pm 8V$ . Tako je mogoče generator poljubnih signalov uporabiti tudi v drugih vejah elektronike.

Ravninski lipidni dvosloj tvorimo v posebni komori, ki ima prostor za elektrode in posebno oblikovano okence za opazovanje dvosloja z mikroskopom. Poleg komore potrebujemo še sistem (slika 1), ki omogoča tvorjenje in zajemanje signalov za preučevanje poracije lipidnih dvoslojev.



Slika 1. Blokovna shema sistema za elektroporacijo lipidnega dvosloja.

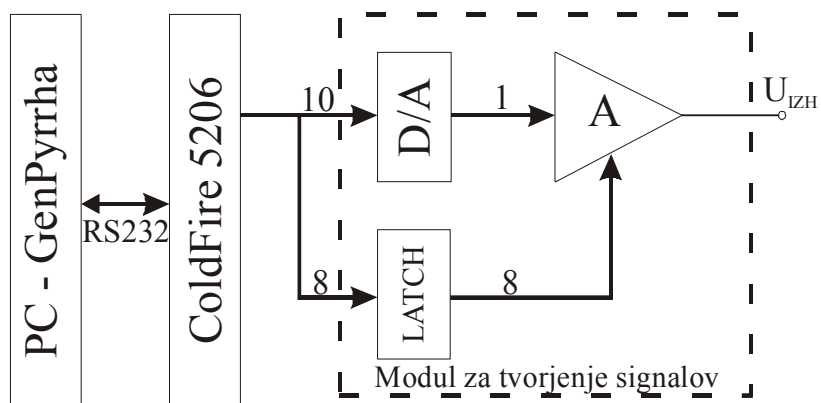
Sistem je sestavljen iz dveh delov: sklopa za tvorjenje signalov (a) in sklopa za merjenje toka in napetosti (b). Natančneje sistem razdelimo na program za krmiljenje sistema (GenPyrrha) [4], mikroprocesor, modul za tvorjenje signalov (D/A), merilnik toka in napetosti z analogno-digitalnim pretvornikom (A/D) in dodatnim spominom za hitro zajemanje podatkov in kasnejšo obdelavo.

Namen elektroporacije lipidnih dvoslojev je preučiti razmere za učinkovitejšo elektroporacijo celic. Cilj seminarske naloge pa je izmiriti frekvenčne lastnosti modula za tvorjenje poljubnih signalov.

## 2. MATERIALI IN METODE

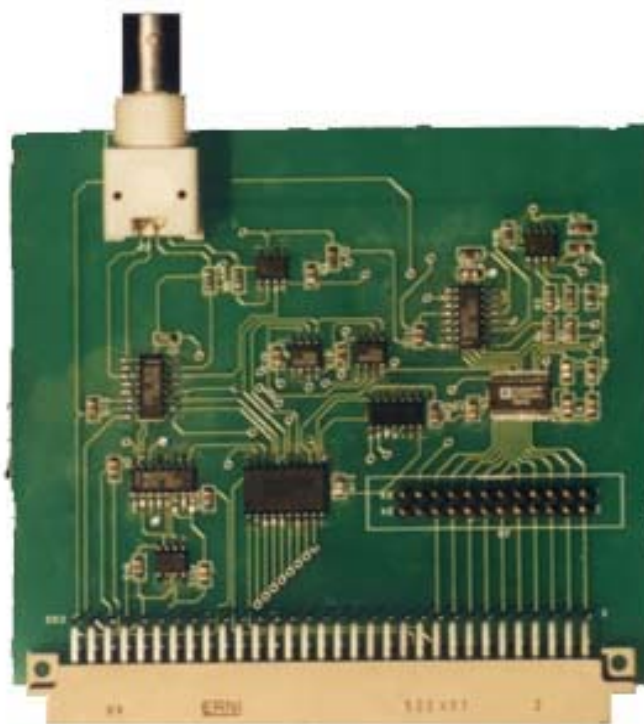
### 2.1. Modul za tvorjenje signalov

Srcce modula za tvorjenje signalov je digitalno-analogni pretvornik. Z njim pretvorimo digitalni signal, iz računalnika, v analogno napetost. Modul je sestavljen iz digitalno-analognega pretvornika, napetostno izhodnega dela, časovnega dela ter krmiljenja modula (slika 2).



Slika 2. Blokovna shema modula za tvorjenje signalov.

Tiskano vezje takega modula, ki je del vezja z mikroprocesorjem vidimo na sliki 3.



Slika 3. Tiskano vezje modula za tvorjenje signalov.

### 2.1.1. Digitalno-analogni pretvornik

Digitalno-analogni pretvornik, ki smo ga uporabili je AD9750 (Analog devices). Njegove dobre lastnosti so visoka hitrost 125MSPS (tisoč točk na sekundo), 10 bitna ločljivost in paralelni vpis. Deset vhodov, od D0 (LSB) do D9 (MSB) je pripeljanih na predpomnilnik. Le-ta služi za prepis informacije v notranjost pretvornika ob urinem impulzu. Deset bitna informacija nastavi stikalno matriko tako, da imamo na izhodu željen analogni tok. Izhod digitalno-analognega pretvornika je tokovni in je razdeljen na  $I_{OUTA}$  in  $I_{OUTB}$ . Integrirano vezje vsebuje dve različni in ločeni ozemljitvi za analogni in digitalni del vezja. Poleg ozemljitev imamo posledično tudi dve ločeni napajalni napetosti vrednosti 5V. Najpomembnejše pri pretvorbi je, da imamo pred prožitvijo urinega impulza najmanj  $t_s=2ns$  stabilno informacijo na vhodu digitalno-analognega pretvornika. Digitalno-analogni pretvornik proži na prvo fronto urinega impulza.

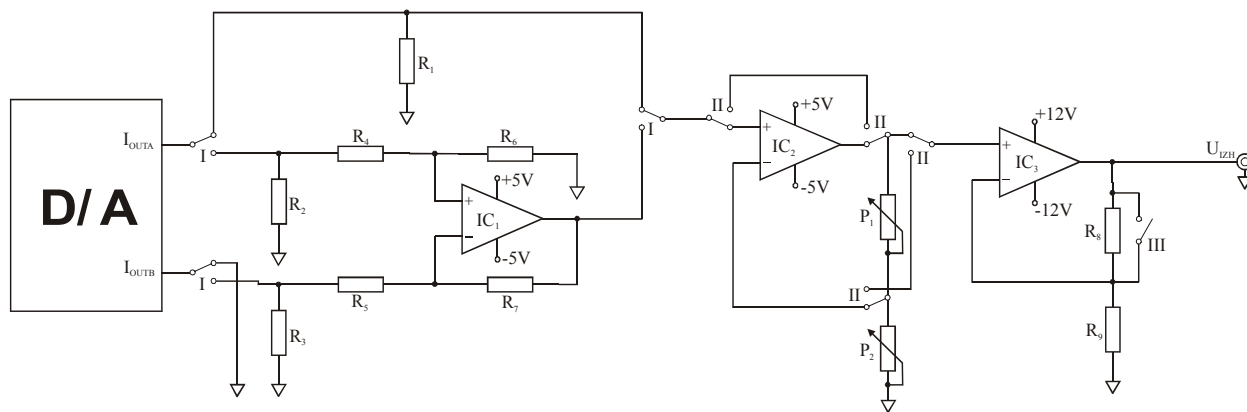
### 2.1.2. Napetostni izhodni del

Tokovni izhod digitalno-analognega pretvornika je potrebno pretvoriti v napetost, nato pa ustrezno ojačati in prirediti izhodnemu bremenu. Slika 4 prikazuje celotni napetostni izhodni del. V prvem delu pretvorimo tok v napetost preko upora  $R_1$  ali  $R_2$  in  $R_3$ , pri čemer imamo možnost oblikovati unipolarne ali pa bipolarne signale, kar je odvisno od položaja preklopnika I. Naslednja stopnja je napetostni ojačevalnik ali slabilnik. Funkcijo nastavimo s stikalom II. Zadnja stopnja je napetostni ojačevalnik, ki nam signal ojači za faktor 2. Kadar želimo imeti slabilnik preklopnik III preklopimo tako, da iz napetostnega ojačevalnika postane emitorski sledilnik. Emitorski sledilnik nam loči odvisnost izhoda od preostalega vezja glede bremena na izhodu.

#### 2.1.2.1 Unipolarno – bipolarni tokovno-napetostni pretvornik

Ločitev na unipolarni in bipolarni del navidez ni potrebna. Vendar ob premisleku ugotovimo, da ima unipolarni signal boljšo napetostno ločljivost tako pri risanju signala kot pri tvorjenju signala z analogno-digitalnim pretvornikom. Osnovni položaj stikala I nam zagotavlja, da imamo izhod analogno-digitalnega pretvornika  $I_{OUTA}$  pripeljan preko upora, kjer se izhodni tok spremeni v napetost. Izbira upora  $R_1$  je odvisna od zahtev analogno digitalnega pretvornika ter od zahtev za vhod v naslednjo stopnjo *ojačevalnik-slabilnik*. Izračun za unipolarno stopnjo [5] nam pokaže, da moramo izbrati vrednost upora  $R_1=2.2k\Omega$ , če želimo, da bo maksimalna napetost na izhodu tokovno-napetostnega pretvornika 2V. Izhod  $I_{OUTB}$  pa je priključen na skupno točko.

Ravno tako želimo pri bipolarnem tokovno – napetostnem pretvorniku izračunati [5] vrednosti uporov odštevalnika, da na izhodu dobimo napetost  $\pm 2V$ . Ob izbiri bipolarnega pretvornika sta oba tokovna izhoda digitalno-analognega pretvornika spremenjena v napetost preko  $50\Omega$  uporov  $R_2$  in  $R_3$  in priključena na vhod odštevalnika napetosti.  $I_{OUTA}$  in  $I_{OUTB}$  imata komplementarno vrednost toka na izhodu in tako tvorimo bipolaren električni signal. Upori na odštevalniku morajo biti  $R_4=R_5=220\Omega$  in  $R_6=R_7=500\Omega$ , da na izhodu ponovno zagotovimo  $\pm 2V$ .



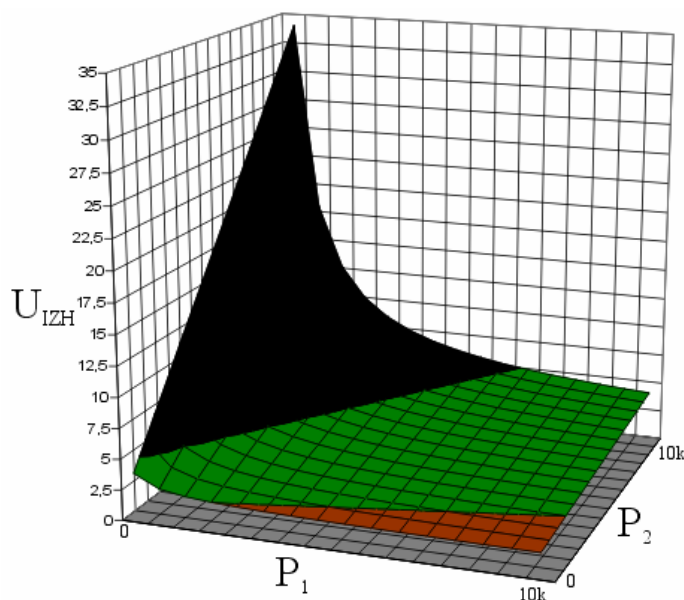
Slika 4. Napetostni izhodni del. Preklopniki na sliki so v položaju 0 (Tabela 1).

### 2.1.2.2. Ojačevalnik – stabilnik

Na vhodu v ojačevalnik ali stabilnik imamo najvišjo napetost  $\pm 2V$ . Napetost lahko ojačujemo, ko je preklopnik II v osnovnem položaju. Najvišja vrednost izhodne napetosti je  $\pm 5V$ , vzrok temu pa sta potenciometra  $P_1$  in  $P_2$ . Za operacijski ojačevalnik smo uporabili OPA627 (Burr-Brown), ki ga odlikujeta hitrost in dobre šumne lastnosti. Upora ob operacijskem ojačevalniku določata napetostno ojačanje po enačbi:

$$U_{izh} = \left(1 + \frac{P_1}{P_2}\right) \cdot U_{vh} \quad (1)$$

Graf napetosti, ki jih lahko dosežemo z ojačevalnikom je prikazan na sliki 5, s črno obarvano področje predstavlja področje nasičenja.



Slika 5. Graf napetostnega ojačanja  $U_{IZH}[V]$  v odvisnosti od vrednosti potenciometrov  $P_1[\Omega]$  in  $P_2[\Omega]$ .

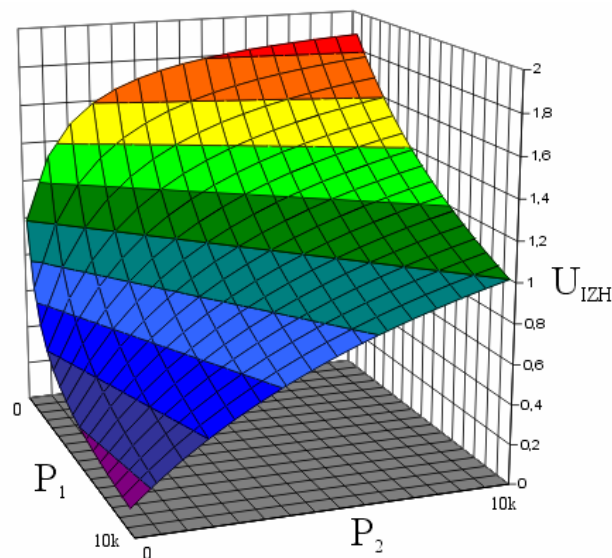
Stabilnik je vključen v vezje, ko je preklopnik II preklopljen. Enačba, ki opisuje slabljenje je preprost delilnik napetosti:



$$U_{izh} = \frac{P_2}{P_1 + P_2} \cdot U_{vh} \quad (2)$$

Na sliki 6 vidimo porazdelitev izhodnih napetosti glede na vrednost upornosti obeh potenciometrov.

Upora sta nastavljiva, zato lahko spreminjamo ojačanje in s tem razpon izhodne napetosti. Izhodno napetost si izbere uporabnik že v programu GenPyrrha [4]. Program po posebnem algoritmu preveri, kateri upori ustrezajo željeni napetosti ter pošlje zahtevane parametre mikroprocesorju. Mikroprocesor nato ustrezno nastavi vrednost obeh potenciometrov.



Slika 6. Graf slabljenja  $U_{IZH}$ [V] vezja v odvisnosti od vrednosti potenciometrov  $P_1$ [ $\Omega$ ] in  $P_2$ [ $\Omega$ ].

Potenciometra  $P_1$  in  $P_2$  sta integrirana X9C103 (XICOR) [6]. Dobra lastnost integriranega potenciometra je, da ima veliko notranjo matriko uporov na istem substratu, rezultat pa je manjši šum ter malo porabljenega prostora na tiskanem vezju. Potenciometer vsebuje matriko stotih uporov po  $100\Omega$ . Njegova slaba stran je, da deluje le do  $\pm 5V$ , zato smo dodali tretjo stopnjo, kjer signal ojačimo.

### 2.1.2.3 Napetostni sledilnik – ojačevalnik $A=2$

Na izhodu potrebujemo poseben člen, ki nam zagotavlja, da se nam napetost znotraj posameznih ojačevalnih stopenj ne seseda. Zato uporabimo napetostni sledilnik, *buffer*. Vendar pa želimo imeti na izhodu napetost do  $\pm 10V$ , zato je potrebno signal ojačati za faktor  $A=2$ . Za preklap med funkcijama uporabljamo stikalo III. Napetosti  $\pm 10V$  ne moremo doseči, ker je najvišji razpon napetosti, ki ga lahko dosežemo na stikalu  $\pm 8V$ . Zato se pač zadovoljimo le s to napetostjo.

Ko ne potrebujemo višjih napetosti od  $\pm 5V$ , oziroma ko uporabljamo napetostni stabilnik imamo stikalo III sklenjeno. Tako imamo upor premoščen in izhod operacijskega ojačevalnika neposredno vezan na negativni vhod operacijskega ojačevalnika, v tem primeru dobimo kar napetostni sledilnik. Napetostni sledilnik nam zagotavlja konstantno napetost na izhodu operacijskega ojačevalnika, ne glede na breme na izhodu. Moč, ki jo breme potrebuje izhaja iz napajalne napetosti operacijskega ojačevalnika. Breme se namreč ne preslika na vhod operacijskega ojačevalnika, kjer imamo upore s katerimi določamo ojačanje oziroma

slabljenje. Tok, ki odteka v operacijski ojačevalnik OPA627 je reda pA, zato ga lahko zanemarimo.

V položaju, ko je stikalo III razklenjeno pa imamo ojačevalnik s faktorjem ojačanja  $A=2$ . To dosežemo tako, da imamo oba upora  $R_8$  in  $R_9$  enaka in po enačbi 1 dobimo željeno ojačanje. V tem primeru gre izhodna napetost do  $\pm 8V$ , kar pomeni, da imamo sedaj na izhodu možnost tvoriti napetost 16Vpp. Zadnji operacijski ojačevalnik ima napajalno napetost  $\pm 8.5V$ , kar pomeni, da je pri  $\pm 8V$  izhodne napetosti še vedno v linearnem področju.

### 2.1.3. Časovna ločljivost

Časovna ločljivost nam predstavlja vzorčevalno frekvenco tvorjenja signala. Odločili smo se za programski način nastavljanja časovnih razdelkov.

Pri tem načinu lahko uporabljamo obstoječ mikroprocesor in RAM, kjer je shranjen signal. Program prebere vrednost signala v pomnilniku in ga pošlje preko D/A pretvornika, kjer nastane analogen signal. Preden prebere iz pomnilnika naslednjo vrednost, lahko določen čas počakamo, tako da dobimo večji zamik med tvorjenimi točkami in s tem daljši čas signala. Vendar pa s podaljševanjem časovne zanke zmanjšujemo ločljivost signala.

Vrednost zakasnilne zanke pošljemo preko krmilnega programa GenPyrrha. Program GenPyrrha ima tudi možnost, da ohrani največjo ločljivost in podaljša vektor signala, kar nas lahko pripelje do »razkošno« dolгих signalov, za katere kmalu zmanjka pomnilnika. Poseben algoritem preračunava, katero metodo je potrebno izbrati, in jo tudi nastavi.

Program, ki je napisan za mikroprocesor se izvaja v flash-u. Ta pa je lahko izredno počasen pri tvorjenju hitrih signalov, tako pri tvorjenju iz flash-a dobimo ločljivost  $10\mu s$  med dvema točkama. Če del programa prenesemo v zunanji RAM in ga od tam tudi izvajamo lahko izboljšamo ločljivost. Hitrost se popravi na  $2\mu s$  med dvema točkama. Za višje hitrosti, pa prenesemo program v statični RAM, ki je v strukturi mikroprocesorja MCF5204. Hitrost tvorjenja signala je v tem primeru  $860ns$  med dvema točkama oziroma  $1.17MHz$ .

Odgovor na vprašanje, zakaj je čas tvorjenja signala tako »dolg«, je jasen. Imamo »optimalno« napisan program,  $20MHz$  frekvenco delovanja procesorja ter RAM, ki za dostop do podatkov potrebuje en čakalni časovni cikel. Ob pregledu izvorne kode, seštevek ciklov ukazov ter upoštevanju, da je en cikel dolg ravno  $50ns$  znese ravno 17 ciklov. Ne glede na to, da uporabljamo mikroprocesor RISC tehnologije, ki je znan po tem, da za svoje osnovne operacije potrebuje izredno malo ciklov.

### 2.1.4. Krmiljenje

Modul krmilimo s pomočjo programa GenPyrrha, ki ima vgrajene že določene izračune za ojačanja, ter optimalno izbiro uporov za napetostni ojačevalnik. Prav tako s pomočjo programa določimo kakšen tip signala (unipolaren ali bipolaren) ter kakšno napetost želimo imeti.

Iz programa podatke preko serijske komunikacije pošljemo mikroprocesorju, ta pa jih posreduje modulu za tvorjenje signalov. Ob tem priključi ustrezen izbiralnik (CS – chip select) ter vpiše vrednost v osem bitno D pomnilno celico 74ACT574.

Krmiljenje uporov nam povzroči, da moramo vpis večkrat ponoviti, saj je vpis v digitalne potenciometre X9C103 serijski. Pri tem potrebujemo za vsakega tri bite, vendar lahko združimo bita INC in U/D iz obeh potenciometrov. Za to smo porabili 4 bite, trije so namenjeni za serijo stikal I, II, in III, zadnjega pa lahko uporabimo za izklop/vklop izhoda signalnega generatorja.

Stikala so realizirana z integriranim vezjem MAX4053, ki predstavlja analogno stikalo. Integrirano vezje vsebuje tri dvopoložajna stikala. Stikala krmilimo posamezno z biti ADDA, ADDB in ADDC [7].

Tabela 1. Položaj stikal in različne funkcije vezja.

I	II	III	Nastavitev
0	0	0	Unipolarni +10V
0	0	1	Unipolarni +5V
0	1	0	Prepovedan položaj
0	1	1	Unipolarni s slabljenjem < +5V
1	0	0	Bipolarni ±10V
1	0	1	Bipolarni ±5V
1	1	0	Prepovedan položaj
1	1	1	Bipolarni s slabljenjem < ±5V

## 2.2. Zgradba programa

V besedilu sem pogosto omenjal program, s katerim lahko kontroliramo obnašanje modula za tvorjenje poljubnih signalov, GenPyrrha. Program je napisan za osebni računalnik, s katerega izvajamo nadzor tudi nad mikroprocesorjem MCF5204. Komunikacija med programoma teče preko serijske komunikacije (RS232). Dela na principu »Handshake« protokola, le da imata svoje lastne ukaze in podatke. Ukazi in podatki so šestnajst bitne besede, ki se v dveh korakih prenesejo od pošiljatelja k prejemniku. Več o tem sem napisal v Prešernovi nalogi [4], zato bom podrobnosti izpustil.

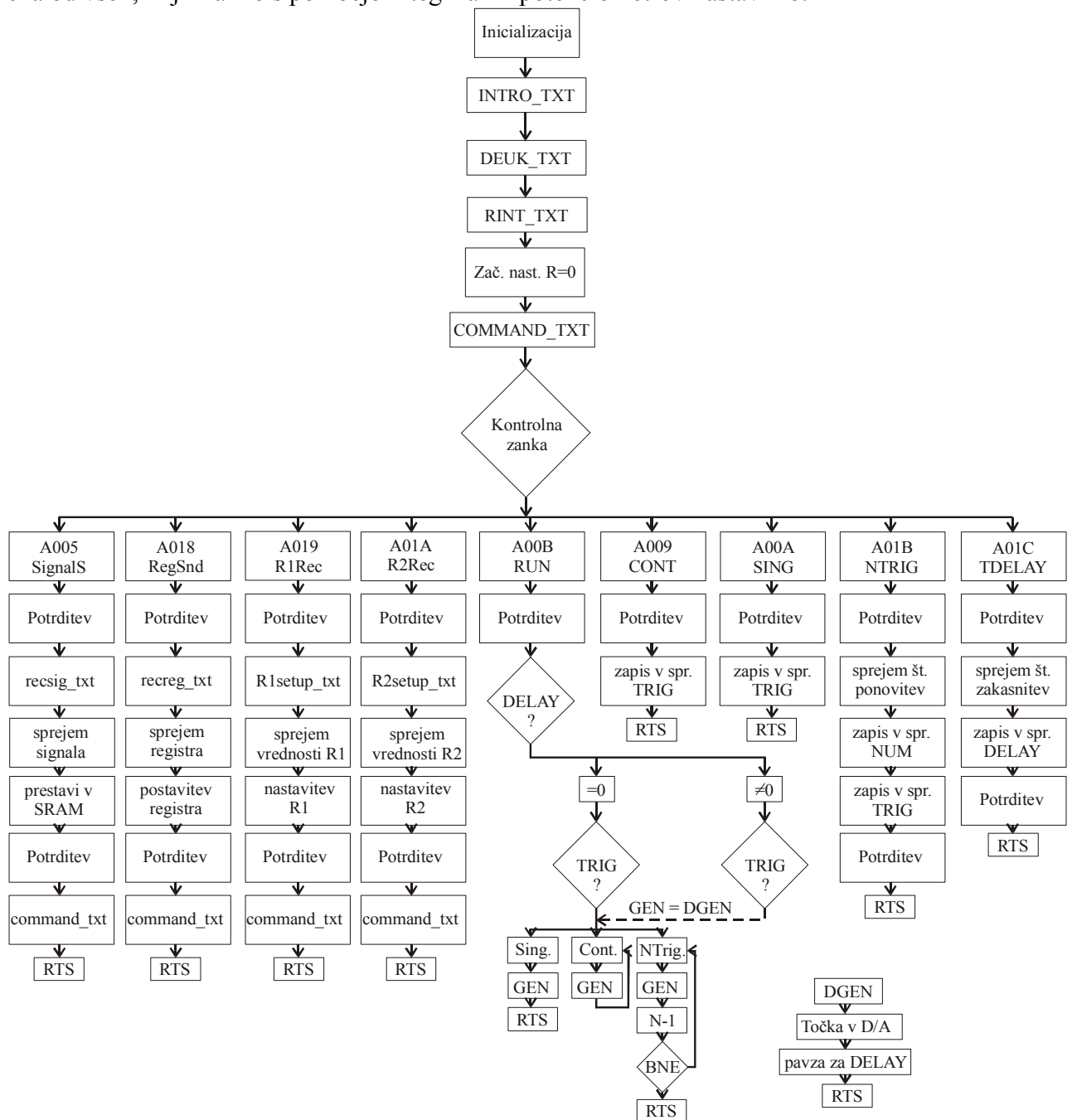
Pravtako je bilo potrebno napisati program, s katerim je »opremljen« mikroprocesor. Program je široko razvejan, saj je potrebno kar nekaj podatkov prenesti na mikroprocesor. Potrebno je vedeti postavitev stikal na modulu za tvorjenje signalov, vrednost uporov, način proženja in ostalo. Za ilustracijo sem dodal blokovno shemo programa (slika 7), ki je sestavljen v assemblerju in je tudi dodan v prilogi.

## 2.3. Pristop k meritvam

Kot sem omenili v uvodu, je cilj seminarske naloge izmeriti lastnosti signalnega generatorja. Signalni generator ima možnost tvorjenja poljubnih signalov z relativno velikim razponom napetosti. Imamo torej velik spekter različnih kombinacij, pri katerih se moramo zelo omejiti. Izbrali bomo sinusni signal ze enako napetostjo amplitude. Spreminjali pa bomo frekvenco signala pri enaki časovni ločljivosti, kar pomeni da bo razdalja med dvema tvorjenima točkama vedno enaka.

Vse te nastavitve določimo v programu GenPyrrha. Najprej nastavimo maksimalno napetost ter čas ene periode (slika 8). Napetost nastavimo na 1.2V, čas pa na 430μs. Nastavitev časa je

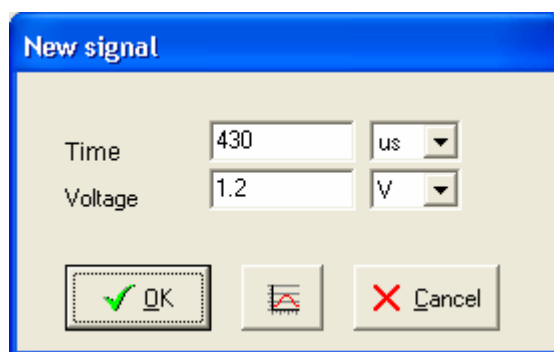
pogojena s tem, da takrat procesor brez zakasnitev pošilja signale na izhod, to pa pomeni, da je takrat časovna ločljivost največja. Frekvenca periodičnega signala je s tem že določena. Če je na ekranu narisana ena perioda, potem imamo frekvenco 2,3kHz. Izbrana napetost pa je le ena od vseh, ki jih lahko s pomočjo integriranih potenciometrov nastavimo.



Slika 7. Blokovna shema programa na mikroprocesorju

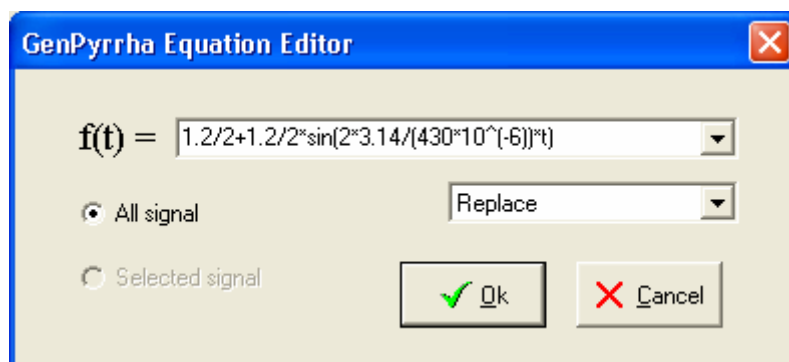
Ko smo določili napetostne in časovne parametre, moramo narisati sinusni signal na površino, ki je namenjena za »opis« signala. Ker je sinus matematična funkcija, si lahko pomagamo z urejevalnikom enačb (slika 9), ki je vgrajen v programu GenPyrrha. Ker smo vzeli, da je signal monopolaren se je potrebno zavedati, da ga moramo dvigniti za določen offset, ki je enak polovica najvišje amplitude. Nato amplitudo zožamo za ravno tak faktor. Frekvenca mora biti ravno prava, tako da pride v celotno okno ena perioda oziroma število period za celoštevilčni faktor.

$$f(t) = \frac{U}{2} + \frac{U}{2} \cdot \sin\left(\frac{2 \cdot \pi \cdot n}{T} \cdot t\right) \quad (3)$$



Slika 8. Okno programa GenPyrrha pri določanju časovnih in napetostnih parametrov signalnega generatorja.

V enačbi (3) vidimo da je  $U$  napetost, ki smo jo vnesli za najvišjo amplitudno vrednost signalnega generatorja,  $T$  perioda celotne slike ter  $n$  celoštevilski faktor, ki nam pove koliko sinusnih period naj bo na zaslону.



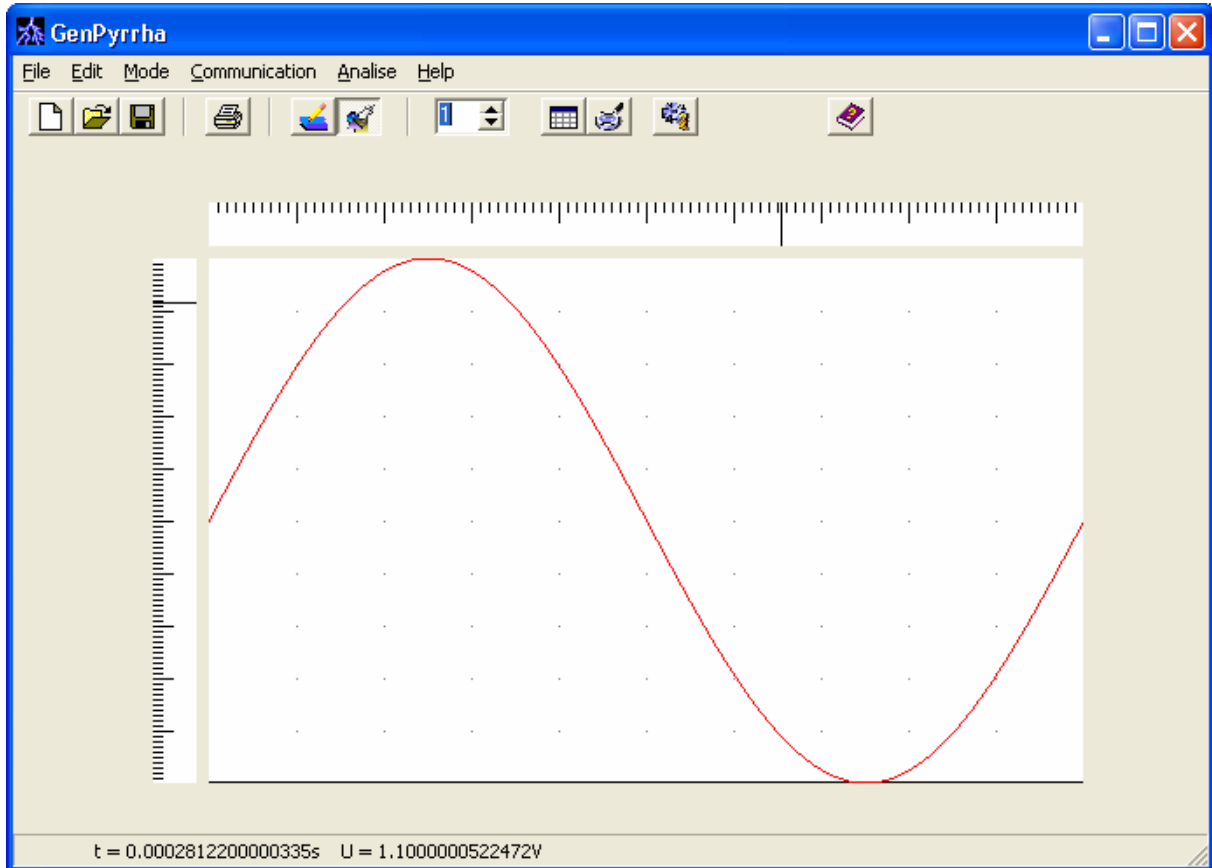
Slika 9. Okno programa GenPyrrha za urejanje enačb.

Izris ene periode vidimo lahko na sliki 10. Ko signal izrišemo, ga je potrebno poslati mikroprocesorju. Pod funkcijo »Communication->Send Signal« pošljemo procesorju vektor točk. Ob tem pa je potrebno poslati procesorju še parametre, ki določajo čas in napetost, da se postavijo posamezni registri v vezju. To naredimo s pomočjo funkcije »Communication->Send Register«. Procesor ima sedaj na svojem RAM-u vektor s 500 točkami ter postavljene registre na modulu za tvorjenje signalov. Funkcija »Communication->Start« nas pripelje do izbire, kako želimo tvoriti signale. Izbiramo lahko med ponavljajočim *continuous trigger*, enkratnim proženjem, *single trigger* ter številčnim proženjem, *numbered triggering*, kjer povemo kolikokrat želimo narisan signal prožiti (slika 11).

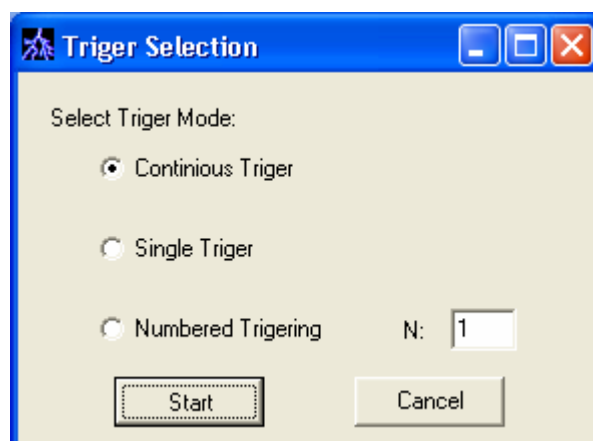
Pri poskusih za to seminarsko nalogo sem se odločil za kontinuitetno proženje, tako da lahko signal spremljamo z osciloskopom v avtomatskem načinu ter s spektralnim analizatorjem, kjer pa je potrebno imeti periodični signal.

S pritiskom na tipko »Start« procesor začne tvoriti signal, ki ga lahko opazujemo na izhodu modula za tvorjenje signalov. Zaradi začetnih pomankljivosti pri programiranju v assemblerju pa je potrebno pri kontinuitetnem proženju po vsaki zamenjavi parametrov ali vektorja signala resetirati procesor. Razlog za to je, da se mikroprocesor »vrti« po zanki, ko mu vnesemo

vektor in želimo kontinuitetno proženje. Pri tem prihranimo nekaj časa, ker je maksimalni del osredotočen na tvorjenje signala in ne na prgledovanje kdaj je prišlo do prekinitve. V kasnejši izvedbi bo potrebno strukturo mikroprocesorskega programa spremeniti in ga prenesti v višji nivo z možnostjo večopravnosti. Vendar pa bo zato potrebno plačati z nižjo frekvenco tvorjenja signala.



Slika 10. Osnovno okno programa GenPyrrha, kjer imamo izrisano eno periodo.



Slika 11. Okno programa GenPyrrha, z možnostjo izbire proženja.

### 3. REZULTATI

Naloga je bila preveriti kvaliteto signalnega generatorja. Za to sem potreboval pogledati v spekter sinusnega signala. V pomoč pri merjenju sta mi bila osciloskop Lecroy 9310C in spektralni analizator HP3598.

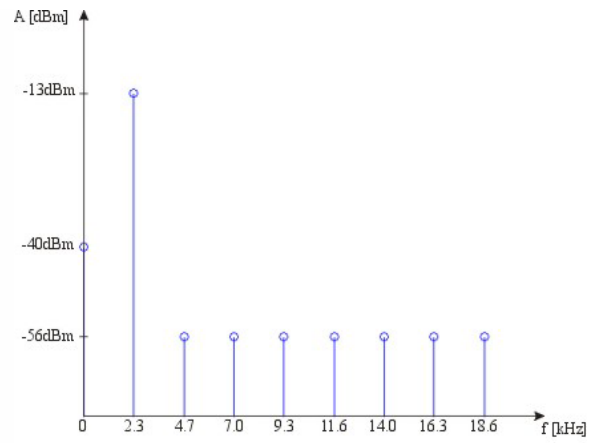
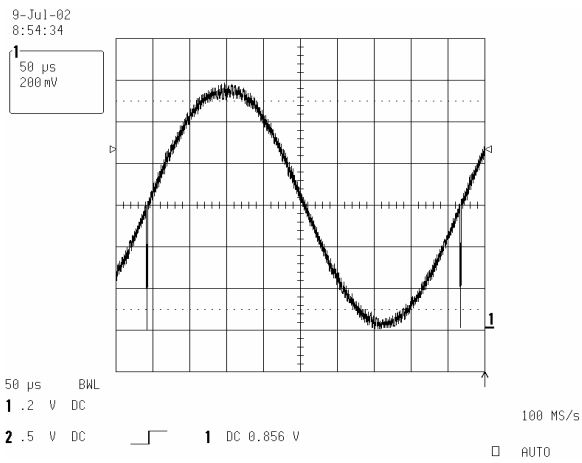
Iz teorije vemo, da ima sinusni signal, v frekvenčnem prostoru le eno frekvenčno komponento s frekvenco tvorjenega signala. Ker je to mogoče le v idealnih razmerah pričakujemo širši potek spektralne komponente. S širine bi lahko sklepali o kvaliteti »nihanja«.

Vendar pa se pri vsakem preizkušanju prototipnega vezja, najde veliko pomanjkljivosti, ki jih tudi v tem vezju ni manjkalo. Najprej so nam ponagajale skupne ozemljitve in predstavile celotne meritve skoraj za štirinajst dni. Ko je bil »skurjeni« čip zamenjan pa se je odkrila programska napaka, ki pa je v tistem času nisem znal poiskati. Tako je nastal glavni problem, da je ob koncu periode signala, ki je bil narisana na zaslonu vrednost skočila na nič, nato pa nazaj na normalno vrednost. To se vidi tudi iz podanih slik rezultatov. Bolj ko sem gostil periode na ekranu, manjkrat se je le ta »špica« pojavila. To se opazi na frekvenčnem spektru signala.

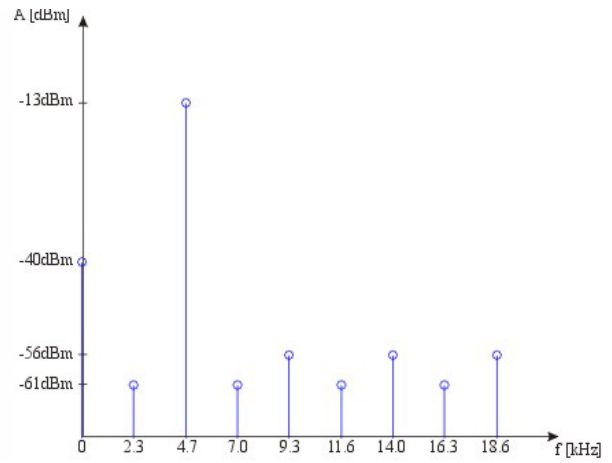
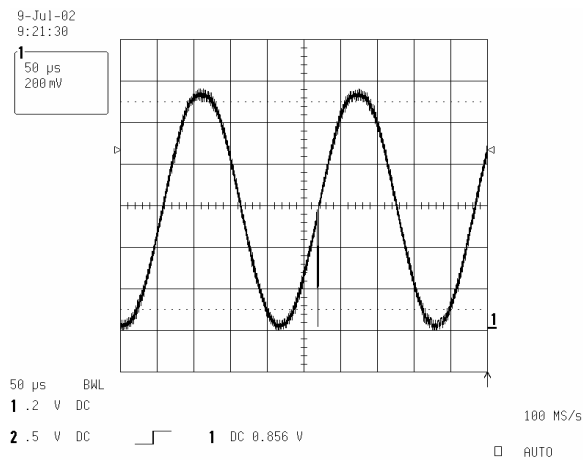
Seveda, pa tudi ni šlo brez problemov pri merilni opremi. Slike spektra, ki je bil zelo zanimiv, sem si želel dobiti za vsako meritev posebej. Vendar se je na koncu izkazalo, da je spektralni analizator prestar, da bi imel normalen PC format disketne enote, zato bi bila potrebna neka pretvorba, s posebnim priloženim programom, ki pa na novejših operacijskih sistemih ne deluje. Tako sem se risanja lotil ročno in kmalu ugotovil, da se oblike ponavljajo pri istih večkratnikih frekvence. Zato sem se lahko lotil risanja slik v programu Matlab, kjer sem jih tudi do konca obdelal. Vse kar manjka na slikah frekvenčnega spektra je širina spektralnih črt in pa gosti šum, ki se je nahajal pod -70dBm.

V naslednjem odstavku se nahajajo slike, kjer imamo v levem oziroma zgornjem delu sliko signala posneto z osciloskopom, na desnem oziroma spodnjem delu pa sliko frekvenčnega spektra signala. Opisna številka npr.: 5T pove, koliko period sinusnega signala smo narisali v programu in s tem povečali frekvenco signala.

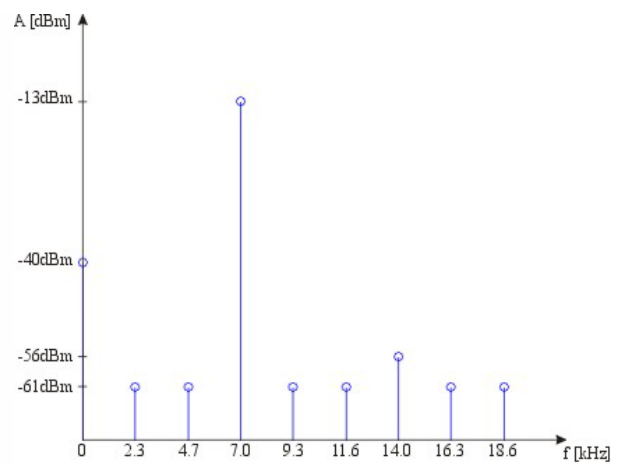
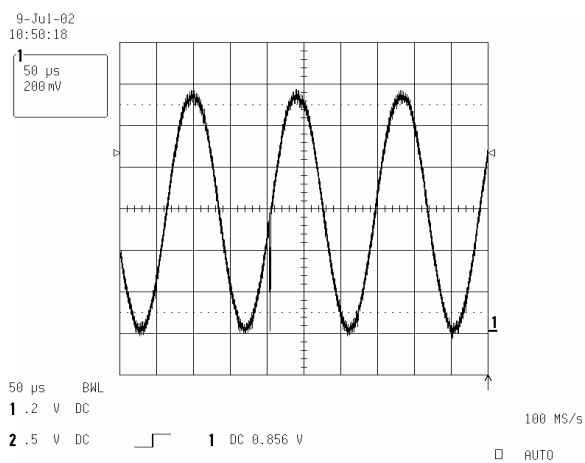
### 1T:



### 2T:

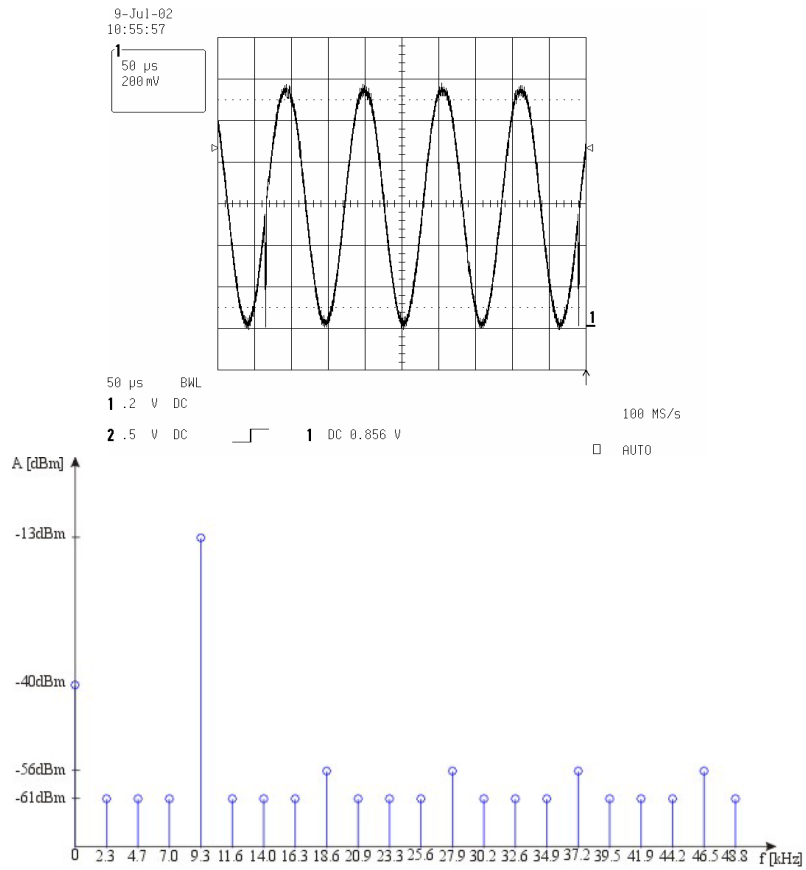


### 3T:

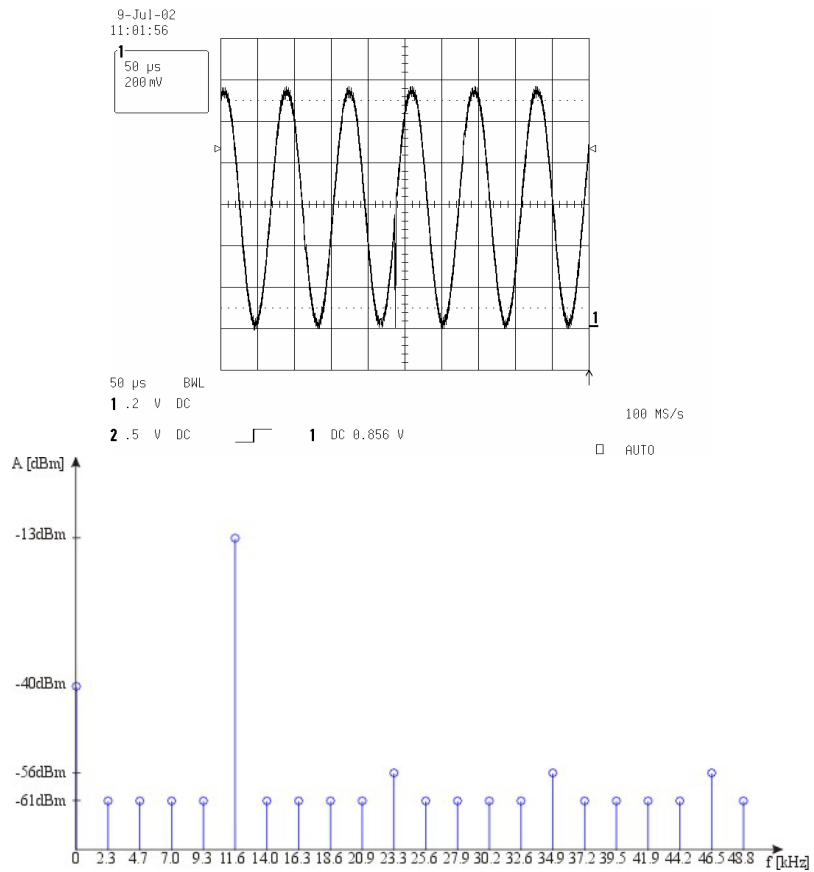




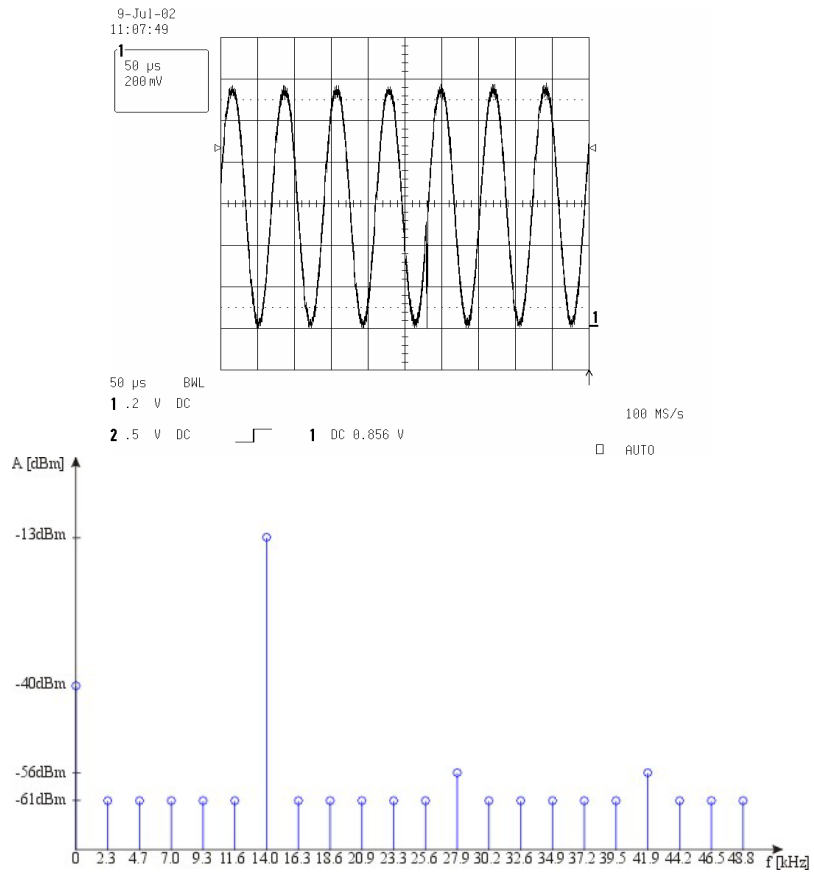
4T:



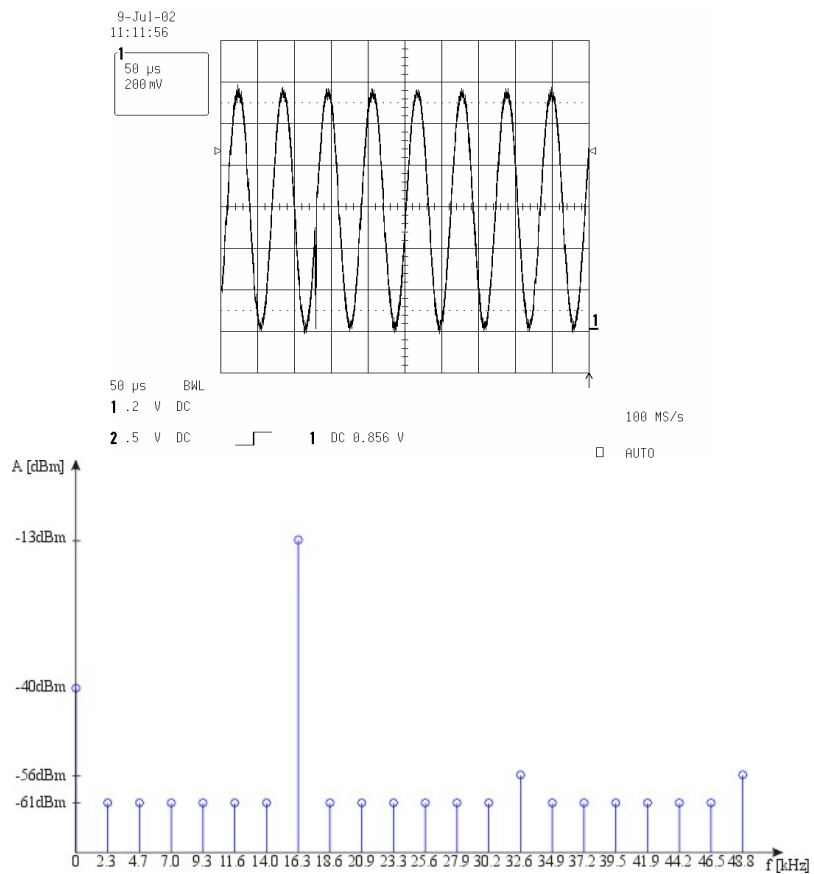
5T:



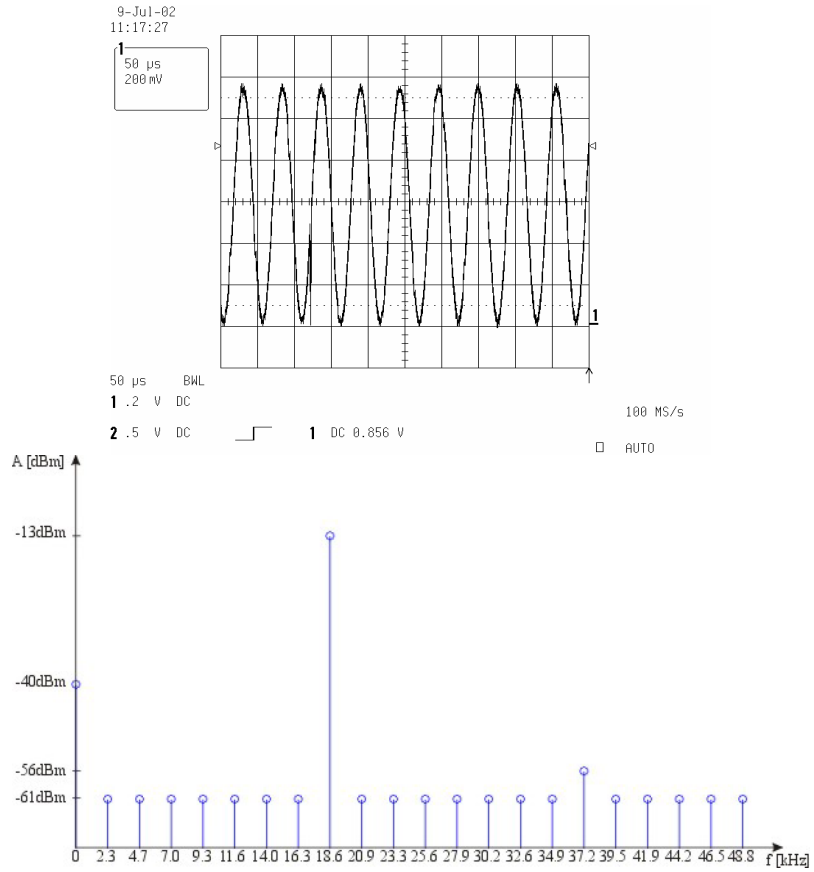
6T:



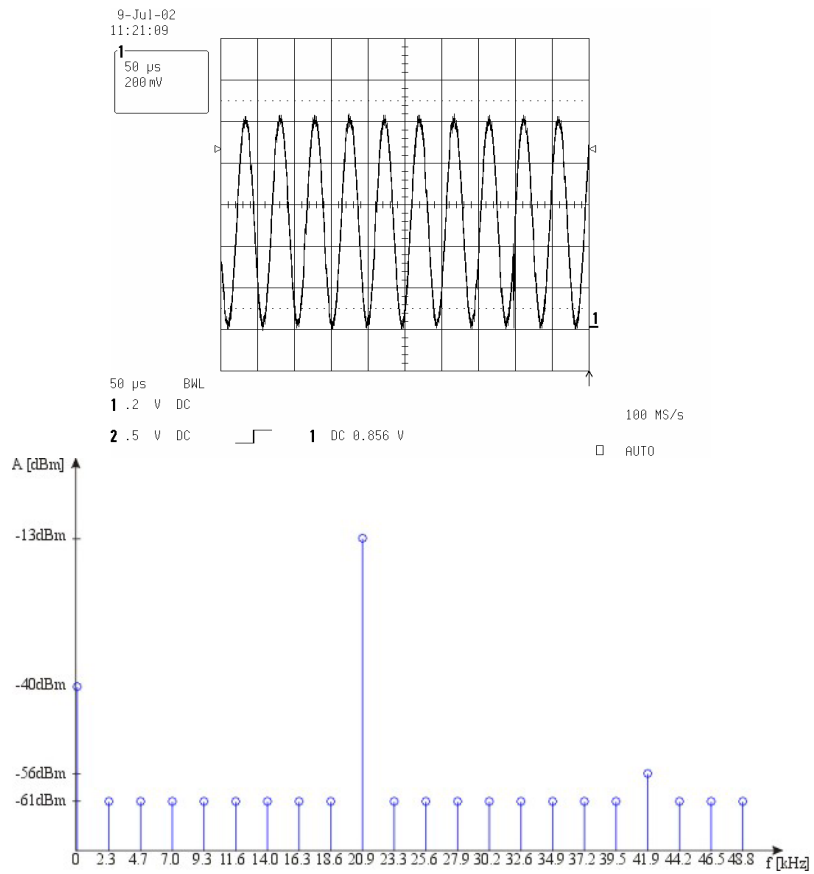
7T:



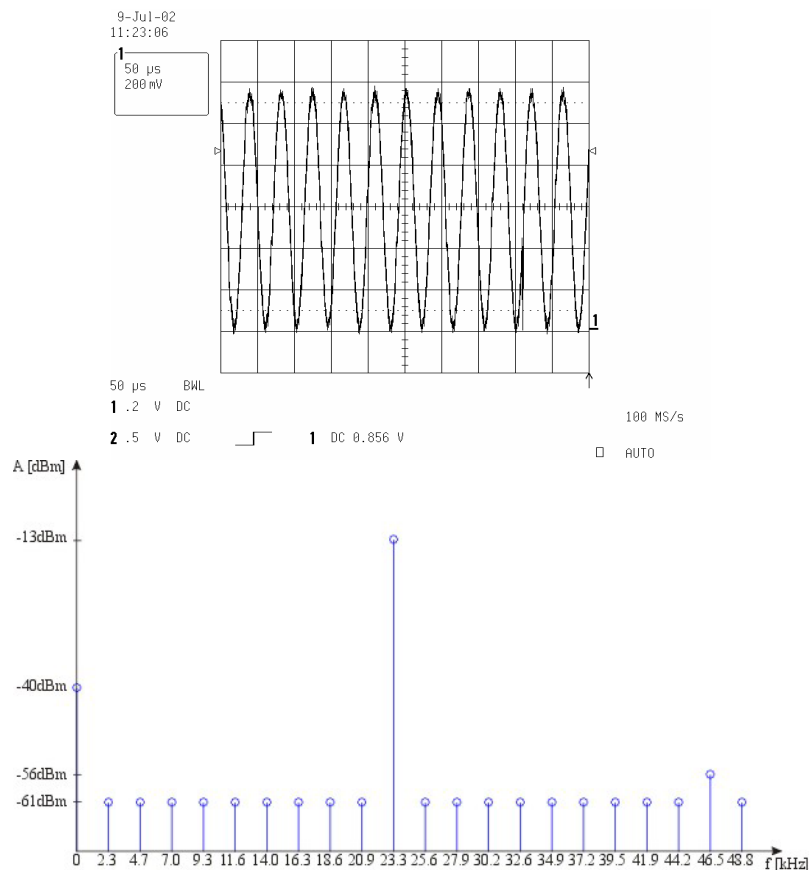
8T:



9T:



## 10T:

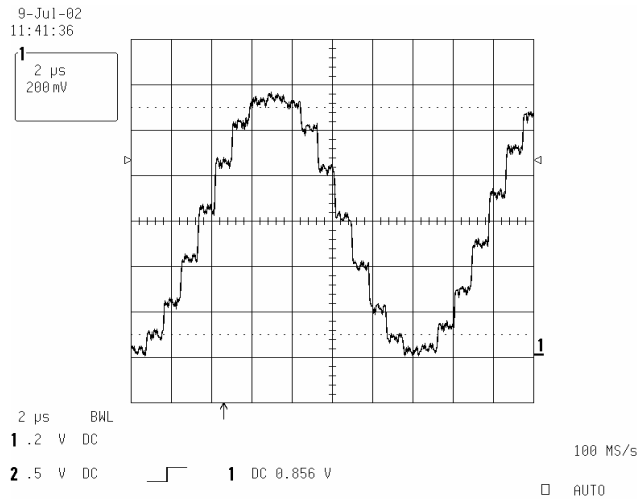


Iz rezultatov lahko opazimo, da imamo štiri tipe frekvenčnih komponent na vseh slikah. Največja komponenta  $-13\text{dBm}$  je osnovna komponenta signala in se nahaja pri frekvenci signala. Druga komponenta je pri frekvenci nič z velikostjo  $-40\text{dBm}$ . Ta predstavlja enosmerno komponento signala. Pojavlja se zaradi tvorjenja signala v monopolarne načinu, kjer smo morali signal dvigniti za polovico celotne možne amplitude kot kaže enačba (3). Tretjega tipa je komponenta z  $-56\text{dBm}$ , ki predstavlja špico pri koncu izrisanega signala. Pri opazovanju spektra opazimo, da se le ta s povečevanjem števila period na zaslonu, špica pojavlja manj krat, v frekvenčnem spektru pa se oddaljuje od osnovne frekvence in to za večkratnik periode, katerega signal tvorimo. Zadnja komponenta pri  $-61\text{dBm}$ , pa se nahaja na vsakem večkratniku frekvence. Če podrobneje razmislimo, so to komponente, ki nastanejo zaradi preslikavanja okoli vzorčne frekvence, nastanejo pa zaradi kvantizacije signala. Signal, če ga pogledamo поблиže, je v svoji odnovni obliki zaradi digitalno – analognega pretvornika bolj pravokoten kot siusen (slika 12).

Šum, ki se še nahaja v signalu, pa v narisanim spektru ni narisano, ker bi slika postala nepregledna, nahaja pa se okoli vrednosti  $-70\text{dBm}$ .

Marsikatero pomanjkljivost bi bilo potrebno odpraviti, da bi lahko dejali, da je to kvaliteten signalni generator. Prva in najvažnejša je popraviti napako »špice«. Po daljšem razhroščevanju obeh programov, tako na osebni računalnik kot mikroprocesorju, sem ugotovil, sem ugotovil, da je to sistemska napaka. Zgodi se, ko procesor prestavi svoj vektor na začetek položaja v pomnilniku, kjer ponovno tvori isti signal. Ta premik vzame nekaj ciklov, kar pa pomeni, da signal pade na digitalno – analognem pretvorniku na nič. Ob prikazovanju bipolarne signala te »špice« sploh ne vidimo, vendar pa se v frekvenčnem spektru vseeno pojavlja kot motnja, ker signal ni ravno sinusen, ampak prekinjen za nekaj

mikro sekund. Težavo bi lahko odpravil le s preprogramiranjem mikroprocesorja, tako da bi celoten sistem delal s pomočjo večopravnosti. To bi nam pomagalo pri tem, da bi imeli dodaten predpomnilnik FIFO strukture, ki bi ga polnil poseben podprogramček, ki bi se izvajal istočasno s tvorjenjem signala. Na drugi strani predpolnilnika pa bi vrednosti pošiljali na izhod. Tako preskok na začetek vektorja signala ne bi bil potreben in tudi problema »špice« ne bi bilo. Vendar pa se pri elektroporaciji lipidnih dvoslojev ponavadi uporablja enkratno proženje, kjer pa tega problema ni in je bil zato odkrit nekoliko pozno.



Slika 12. Pravokotnost signala zaradi digitalno – analognega pretvornika.

Druga nevšečnost pri frekvenčnem spektru pa je »glavnikasta« oblika komponent, ki nastanejo zaradi stopničastega signala. Za to bi lahko poskrbeli z integratorjem na izhodu digitalno - analognega pretvornika, oziroma z nizkofrekvenčnim sitom. Sam sinusni signal pa bi lahko izboljšali še z rekonstrukcijskim filtrom. Razlog zakaj prototipno vezje ne vsebuje teh nujno potrebnih filtrov je v tako obsežnem frekvenčnem področju. Če bi spreminjali frekvenco tvorjenja signala še v večjem obsegu, kot smo to naredili pri meritvah v tej seminarski nalogi, bi bilo potrebno spreminjati lego boka nizkofrekvenčnega sita glede na vzorčno frekvenco izhoda. Poleg tega bi vključili dodatne rekonstrukcijske filtre le za posebne tipe signalov. Pri poljubnih signalih pa tega ne moremo narediti.

## 4. ZAKLJUČEK

V seminarski nalogi sem poizkušal zajeti izdelavo modula za tvorjenje poljubnih signalov z vsemi dodatki, ki so bili potrebni za delovanje tega modula v praksi, ter preizkusom modula. S preizkusom sem spoznal, da je potrebno pri načrtovanju upoštevati širok spekter dejavnikov, ki bodo kasneje ob delovanju vplivali na vezje. Vendar pa se je pri potrebno pri zastavljanju ciljev omejiti in razdeliti problem na več prototipov, kjer postopoma odkrivamo in odpravljamo napake.

Osnovni namen modula za tvorjenje signalov je elektroporacija lipidnih dvoslojev. Principi elektroporacije zahtevajo napetosti do  $\pm 1V$ . Razširitev napetosti do  $\pm 8V$ , pomeni možnost uporabe tudi v splošne namene.

V prihodnosti bo potrebno narediti še bolj prijazno obliko programa v mikroprocesorju, potrebno bo napraviti tudi nekaj izboljšav pri doseganju še višjih frekvenc signalov. Velja pa tudi razmisliti o samostojnosti signalnega generatorja, brez nenehne odvisnosti od osebnega računalnika. To bi lahko zagotovili z dodatnim flash spominom, ki bi bil lahko vgrajen na dodatno vodilo mikroprocesorja. Pragtako bi bilo potrebno razmišljati o že vgrajenih signalih kot so sinusna, žagasta, trikotna in kvadratna napetost.

Vse to je izvedljivo, vendar pa je potrebno še veliko trdega dela, ki pa ga seminarska naloga ne more v celoti zavzeti.

## 5. LITERATURA

- [1] A. Maček-Lebar, G. Serša, M. Čemažar, D. Miklavčič, Elektroporacija, *Medicinski razgledi*, 37:339-354, 1998.
- [2] M. P. Rols, C. Delteil, M. Golzio, P. Dumond, S. Cros, J. Teissie, *In vivo* electrically mediated protein and gene transfer in murine melanoma, *Nature Biotechnology*, 16:168-171, 1998.
- [3] V. Sharma, Poloxamer 188 Decreases Susceptibility of Artificial Lipid Membranes to Electroporation, *Biophys. J.*, 71:3229-3241, 1996.
- [4] P. Kramar, *Programska oprema za sistem za generiranje pulzov poljubnih oblik, zajem podatkov in krmiljenje merilnega sistema za poskuse na lipidnih dvoslojih*, Prešernova naloga, Univerza v Ljubljani, Fakulteta za elektrotehniko, 2000.
- [5] *AD9750, 10-Bit, 125 MSPS High Performance TxDAC D/A Converter*, Data Sheet, Analog Devices, 1999.
- [6] *X9C102/103/104/503 Digitally Controlled Potentiometer (XDCCP)*, Data Sheet, Xicor 2000.
- [7] *MAX4053 Low-Voltage, CMOS Analog Multiplexers/Switches*, Data Sheet, MAXIM, 1996.
- [8] P. Kramar, D. Miklavčič *Modul za tvorjenje signalov za elektroporacijo lipidnih dvoslojev*, ERK, 2001.

## 6. DODATEK

### 6.1. Izvorna koda mikroprocesorja

```
;
; *****
; **      Operating system for Signal      *
; **      Generator Module                 *
; **      DEUKALION      2.0               *
; **      SEMINAR Elektronska Vezja       *
; *****
$INCLUDE "c:\cpu\MCF5206.HDR"
;
;
;          ORG      $0004
;          DB       00,00,04,00
;          ORG      $0400
;
; *****
; **      Initialization                    **
; *****
INIT      MOVE.L   #20000001,D0
          MOVEC   D0,MBAR
          MOVE.L   #10000001,D0
          MOVEC   D0,RAMBAR
          MOVEA.L #RAMBASE+SRAM_SIZE,A7
;
FLASH    MOVE.L   #00,D0
          MOVE.L   D0,CSAR0.L
          MOVE.L   #0001FF00,D0
          MOVE.L   D0,CSMR0.L
          MOVE.L   #00001D41,D0
          MOVE.L   D0,CSCR0.L
;
RAM      MOVE.L   #400000,D0
          MOVE.L   D0,CSAR1.L
          MOVE.L   #7FE00,D0
          MOVE.L   D0,CSMR1.L
          MOVE.L   #0501,D0
          MOVE.L   D0,CSAR1.L
;
DA       MOVE.B   #01,D0
          MOVE.B   D0,PAR.L
          MOVE.B   D0,PADDR.L
          MOVE.B   #00,D0
          MOVE.B   D0,PADAT.L
;
          MOVE.L   #30000,D0
          MOVE.L   D0,CSAR3.L
          MOVE.L   #00,D0
          MOVE.L   D0,CSMR3.L
          MOVE.L   #0101,D0
          MOVE.L   D0,CSCR3.L
;
RSINIT   MOVE.B   #%00100000,D0
          MOVE.B   D0,UCR1.L
;
          MOVE.B   #%00110000,D0
          MOVE.B   D0,UCR1.L
;
          MOVE.B   #%00010000,D0
          MOVE.B   D0,UCR1.L
;
          MOVE.B   #$F3,D0
          MOVE.B   D0,UMR11.L
;
          MOVE.B   #$07,D0
          MOVE.B   D0,UMR21.L
;
DISPLAY MOVE.L   #50000,D0
          MOVE.L   D0,CSAR2.L
          MOVE.L   #0,D0
          MOVE.L   D0,CSMR2.L
          MOVE.L   #00001D41,D0
          MOVE.L   D0,CSCR2.L
;
TIMER    MOVE.B   #$DD,D0
          MOVE.B   D0,UCSR1.L
;
          MOVE.B   #$00,D0
          MOVE.B   D0,UACR1.L
;
          MOVE.B   #$00,D0
          MOVE.B   D0,UIMR1.L
;
BAUDI    MOVE.B   #$00,D0
          MOVE.B   D0,UBG11.L
          MOVE.B   #$41,D0      ;ZA 20MHz
          MOVE.B   D0,UBG21.L
;
; *****
; **      Variables                        **
; *****
;
IRW      EQU      50000 ;instruk. reg.
IRR      EQU      50002
DRW      EQU      50001 ;data register
DRR      EQU      50003
;
REGDATA EQU      400000
R1DATA  EQU      400002
R2DATA  EQU      400004
R1SET   EQU      400006
R2SET   EQU      400008
STATUS  EQU      40000A
TRIG    EQU      40000C
NUM     EQU      40000E
DELAY   EQU      400010
REGIST  EQU      400012
;
; *****
; *      BOOTING                          *
; *****
;
BOOT     BSR      LCD_INI
          BSR      INTRO
          BSR      WAITL
          BSR      DEUKTXT
          BSR      WAITL
          BSR      SETTXT
          BSR      RINIT
          CLR.L   D0
          MOVE.W  D0,DELAY.L
;
; *****
; **      Program Beginning                **
; **      MAIN                            **
; *****
;
MAIN     BSR      COMMTXT
;
; *****
; **      System Control                    **
; *****
;
KONTRL   CLR.L   D0
          CLR.L   D1
          CLR.L   D2
          CLR.L   D3
          MOVE.B  #%00001001,D1
          MOVE.B  D1,UCR1.L
CAKK      MOVE.B  USR1.L,D1
          BTST   #0,D1
          NOP
          BEQ    CAKK
          MOVE.B  UR1.L,D0
;
CAKK1    MOVE.B  USR1.L,D1
          BTST   #0,D1
          NOP
          BEQ    CAKK1
          MOVE.B  UR1.L,D1
```



```

;premik
ASL #8,D0
ADD D1,D0

; pogled in izbira ukaza:
CMP.L #A005,D0
BEQ SIGNALS
CMP.L #A018,D0
BEQ REGSND
CMP.L #A019,D0
BEQ R1REC
CMP.L #A01A,D0
BEQ R2REC
CMP.L #A00B,D0
BEQ RUN
CMP.L #A009,D0
BEQ CONT
CMP.L #A00A,D0
BEQ SING
CMP.L #A01B,D0
BEQ NTRIG
CMP.L #A01C,D0
BEQ TDELAY
; e ni ni., nazaj êakat kontrolo
MOVE.W STATUS.L,D3
CMP.L #FFFF,D3
BEQ NASTAVI
BRA KONTRL

;
;
; *****
; *****
; *** Application Functions ***
; *****
; *****
;
; *****
; ** Signal Recive **
; *****

SIGNALS BSR RET
BSR RCRGTX
BSR REGREC
BSR SRTOMTS
BSR COMMTXT
BSR RET
BRA KONTRL

; -----
; sprejem signala
;
SPREJEM MOVEA.L #410000,A1
CLR.L D0
CLR.L D1
CLR.L D2
MOVE.B #00001001,D1
MOVE.B D1,UCR1.L
CAK MOVE.B USR1.L,D1
BTST #0,D1
NOP
BEQ CAK
MOVE.B UR1.L,D0

CAK1 MOVE.B USR1.L,D1
BTST #0,D1
NOP
BEQ CAK1
MOVE.B UR1.L,D1

ASL #8,D0
ADD D1,D0

MOVE.W D0,(A1)+
ADDI.L #02,D2
CMP.L #03E8,D2
BEQ KONEC
BRA CAK
KONEC RTS

; *****
; ** Register Recive **
; *****

REGSND BSR RET

; *****
; ** R1 Value Recive **
; *****

R1REC BSR RET
BSR R1STXT
BSR R12REC
MOVE.W D0,R1DATA.L

CLR.L D0
MOVE.W STATUS.L,D0
ADDI.L #00FF,D0
MOVE.W D0,STATUS.L

BSR COMMTXT
BSR RET
BRA KONTRL

; *****
; ** R2 Value Recive **
; *****

R2REC BSR RET
BSR R2STXT
BSR R12REC
MOVE.W D0,R2DATA.L

CLR.L D0

BSR RCRGTX
BSR REGREC
BSR SRTOMTS
BSR COMMTXT
BSR RET
BRA KONTRL

; -----
; sprejem registra
;
REGREC CLR.L D0
CLR.L D1
CLR.L D2
MOVE.B #00001001,D1
MOVE.B D1,UCR1.L
CH5 MOVE.B USR1.L,D1
BTST #0,D1
NOP
BEQ CH5
MOVE.B UR1.L,D0

;
CH6 MOVE.B USR1.L,D1
BTST #0,D1
NOP
BEQ CH6
MOVE.B UR1.L,D1

; premik
ASL #8,D0
ADD D1,D0

;
MOVE.W D0,REGDATA.L
RTS

; -----
; pošiljanje registra na modul
; za tvorjenje signalov
;
SRTOMTS CLR.L D4
CLR.L D5
MOVE.L REGDATA.L,D4
MOVE.L REGIST.L,D5
ADD D5,D4
MOVE.B #01,D0
MOVE.B D0,PADAT.L
MOVE.W D4,($30000).L
MOVE.B #00,D0
MOVE.B D0,PADAT.L
RTS

; *****
; ** R1 Value Recive **
; *****

R1REC BSR RET
BSR R1STXT
BSR R12REC
MOVE.W D0,R1DATA.L

CLR.L D0
MOVE.W STATUS.L,D0
ADDI.L #00FF,D0
MOVE.W D0,STATUS.L

BSR COMMTXT
BSR RET
BRA KONTRL

; *****
; ** R2 Value Recive **
; *****

R2REC BSR RET
BSR R2STXT
BSR R12REC
MOVE.W D0,R2DATA.L

CLR.L D0

```

```

MOVE.W STATUS.L,D0
ADDI.L #$FF00,D0
MOVE.W D0,STATUS.L

BSR COMMTXT
BSR RET
BRA KONTRL

;-----
; sprejem uporov
;
R12REC CLR.L D0
CLR.L D1
CLR.L D2
MOVE.B %#00001001,D1
MOVE.B D1,UCR1.L
RR5 MOVE.B USR1.L,D1
BTST #0,D1
NOP
BEQ RR5
MOVE.B URB1.L,D0
;
RR6 MOVE.B USR1.L,D1
BTST #0,D1
NOP
BEQ RR6
MOVE.B URB1.L,D1
; premik
ASL #8,D0
ADD D1,D0
;
RTS

;-----
; nastavitev uporov
;
NASTAVI BSR R1STXT
CLR.L D3
CLR.L D4
CLR.L D5
MOVE.L REGIST.L,D4
MOVE.L #$8,D5
SUB D4,D5
MOVE.W D4,REGIST.L
ADDI.L #$40,D4
BSR VNOS
AGATN MOVE.L REGIST.L,D4
BSR VNOS
ADDI.L #$40,D4
BSR VNOS
ADDI.L #$01,D3
CMP.L R1DATA.L,D5
BEQ SHRANN
BRA AGATN

SHRANN MOVE.W REGIST.L,D4
ADDI.L #$8,D4
MOVE.W D4,REGIST.L
BSR VNOS

BSR R2STXT
CLR.L D3
CLR.L D4
CLR.L D5
MOVE.L REGIST.L,D4
MOVE.L #$10,D5
SUB D4,D5
MOVE.W D4,REGIST.L
ADDI.L #$40,D4
BSR VNOS
AGATM MOVE.L REGIST.L,D4
BSR VNOS
ADDI.L #$40,D4
BSR VNOS
ADDI.L #$01,D3
CMP.L R2DATA.L,D5
BEQ SHRANN
BRA AGATM

SHRANN MOVE.W REGIST.L,D4
ADDI.L #$10,D4
MOVE.W D4,REGIST.L

BSR VNOS
CONT BSR RET
CLR.L D0
MOVE.W #$F000,D0
MOVE.W D0,TRIG.L
BRA KONTRL

; *****
; ** Set Continuous Trig **
; *****
;
CONT BSR RET
CLR.L D0
MOVE.W #$F000,D0
MOVE.W D0,TRIG.L
BRA KONTRL

; *****
; ** Set Single Trig **
; *****
;
SING BSR RET
CLR.L D0
MOVE.W #$0F00,D0
MOVE.W D0,TRIG.L
BRA KONTRL

; *****
; ** Set Numbered Trig **
; *****
;
NTRIG BSR RET
BSR R12REC
MOVE.W D0,NUM.L
CLR.L D0
MOVE.W #$00F0,D0
MOVE.W D0,TRIG.L
BSR RET
BRA KONTRL

; *****
; ** Set Time Delay **
; *****
;
TDELAY BSR RET
BSR R12REC
MOVE.W D0,DELAY.L
BSR RET
BRA KONTRL

; *****
; ** RUN Function **
; *****
;
RUN BSR RET
CLR.L D0
MOVE.W DELAY.L,D0
CMP.L #$0000,D0
BEQ NODELR
BRA DELRUN

;-----
; nastavitev proženja brez zakasnitev
;
NODELR CLR.L D0
MOVE.W TRIG.L,D0
CMP.L #$F000,D0
BEQ CONTND
CMP.L #$0F00,D0
BEQ SINGND
CMP.L #$00F0,D0
BEQ NTRGND
BRA KONTRL

;-----
; Continious proženje brez zakasnitev
;
CONTND BSR GENSX

```

```

        MOVEA.L #GEN, A1
        MOVEA.L #10000000, A2
NASL1  MOVE.W  (A1)+, (A2)+
        CMPA   #END, A1
        BNE   NASL1
        MOVE.W (A1), (A2)
GEN     JMP     (10000000).L
        CLR.L  D2
        MOVEA.L #410000, A1
        MOVEA.L #30000, A2
NASL    MOVE.W  (A1)+, (A2)
        ADDI.L #02, D2
        CMP.L  #03E8, D2
        BEQ   GEN
        BRA   NASL
END     BRA   $

;-----
;
; Single proženje brez zakasnitev
;
SINGND BSR     GENSX
        MOVEA.L #GEN1, A1
        MOVEA.L #10000000, A2
NASL3  MOVE.W  (A1)+, (A2)+
        CMPA   #END1, A1
        BNE   NASL3
        MOVE.W (A1), (A2)
GEN1   JMP     (10000000).L
        CLR.L  D2
        MOVEA.L #410000, A1
        MOVEA.L #30000, A2
NASL2  MOVE.W  (A1)+, (A2)
        ADDI.L #02, D2
        CMP.L  #03E8, D2
        BEQ   DOMU1
        BRA   NASL2
DOMU1  JMP     (MAIN).L
END1   BRA   $

;-----
;
; Števno proženje brez zakasnitev
;
NTRGND BSR     GENSX
        CLR.L  D3
        MOVEA.L #GEN2, A1
        MOVEA.L #10000000, A2
NASL5  MOVE.W  (A1)+, (A2)+
        CMPA   #END2, A1
        BNE   NASL5
        MOVE.W (A1), (A2)
GEN2   JMP     (10000000).L
        CLR.L  D2
        MOVEA.L #410000, A1
        MOVEA.L #30000, A2
NASL4  MOVE.W  (A1)+, (A2)
        ADDI.L #02, D2
        CMP.L  #03E8, D2
        BEQ   DELTEST
        BRA   NASL4
DELTEST CLR.L  D4
        MOVE.W NTRIG.L, D4
        CMP.L  D3, D4
        BEQ   DOMU2
        ADDI.L #01, D3
        BRA   GEN2
DOMU2  JMP     (MAIN).L
END2   BRA   $

;-----
;
; nastavitev proženja z zakasnitvami
;
DELRUN CLR.L  D0
        MOVE.W TRIG.L, D0
        CMP.L  #$F000, D0
        BEQ   CONTDE
        CMP.L  #$0F00, D0
        BEQ   SINGDE
        CMP.L  #$00F0, D0
        BEQ   NTRGDE
        BRA   KONTRL

;-----
;
; Countinuous proženje z zakasnitvami
;
CONTDE BSR     GENSX
        MOVEA.L #GEN3, A1
        MOVEA.L #10000000, A2
NASL7  MOVE.W  (A1)+, (A2)+
        CMPA   #END3, A1
        BNE   NASL7
        MOVE.W (A1), (A2)
GEN3   JMP     (10000000).L
        CLR.L  D2
        MOVEA.L #410000, A1
        MOVEA.L #30000, A2
NASL6  MOVE.W  (A1)+, (A2)
        JSR   CAKAJ.L
        ADDI.L #02, D2
        CMP.L  #03E8, D2
        BEQ   GEN3
        BRA   NASL6
END3   BRA   $

;-----
;
; Single proženje z zakasnitvami
;
SINGDE BSR     GENSX
        MOVEA.L #GEN4, A1
        MOVEA.L #10000000, A2
NASL9  MOVE.W  (A1)+, (A2)+
        CMPA   #END4, A1
        BNE   NASL9
        MOVE.W (A1), (A2)
GEN4   JMP     (10000000).L
        CLR.L  D2
        MOVEA.L #410000, A1
        MOVEA.L #30000, A2
NASL8  MOVE.W  (A1)+, (A2)
        JSR   CAKAJ.L
        ADDI.L #02, D2
        CMP.L  #03E8, D2
        BEQ   DOMU4
        BRA   NASL8
DOMU4  JMP     (MAIN).L
END4   BRA   $

;-----
;
; Števno proženje z zakasnitvami
;
NTRGDE BSR     GENSX
        CLR.L  D3
        MOVEA.L #GEN5, A1
        MOVEA.L #10000000, A2
NASL11 MOVE.W  (A1)+, (A2)+
        CMPA   #END5, A1
        BNE   NASL11
        MOVE.W (A1), (A2)
GEN5   JMP     (10000000).L
        CLR.L  D2
        MOVEA.L #410000, A1
        MOVEA.L #30000, A2
NASL10 MOVE.W  (A1)+, (A2)
        JSR   CAKAJ.L
        ADDI.L #02, D2
        CMP.L  #03E8, D2
        BEQ   DELTES1
        BRA   NASL10
DELTES1 CLR.L  D4
        MOVE.W NTRIG.L, D4
        CMP.L  D3, D4
        BEQ   DOMU5
        ADDI.L #01, D3
        BRA   GEN5
DOMU5  JMP     (MAIN).L
END5   BRA   $

;-----
;
; Delay zanka za zakasnitev
;
CAKAJ  MOVE.L  DELAY.L, D1

```

```

CAKAJ1 SUBI #1,D1 VB MOVE.B #%01000010,D0 ;znak B
NOP BSR DISPC
BNE CAKAJ1 MOVE.B D0,DRW.L
RTS RTS
;
; *****
; ** Potentiometer Initialization **
; *****
;
RINIT CLR.L D4 VC MOVE.B #%01000011,D0 ;znak C
CLR.L D5 BSR DISPC
MOVE.B D0,DRW.L
RTS
;postavi vrednosti upora na ni• VE MOVE.B #%01000101,D0 ;znak E
MOVE.W D4,R1SET.L BSR DISPC
MOVE.W D4,R2SET.L MOVE.B D0,DRW.L
RTS
;nadaljuje z nastavitvijo VF MOVE.B #%01000110,D0 ;znak F
MOVE.L #$40,D4 BSR DISPC
BSR VNOS MOVE.B D0,DRW.L
RTS
AGAT MOVE.L #$00,D4 VG MOVE.B #%01000111,D0 ;znak G
BSR VNOS BSR DISPC
MOVE.L #$40,D4 MOVE.B D0,DRW.L
BSR VNOS RTS
ADDI.L #$01,D5 VH MOVE.B #%01001000,D0 ;znak H
CMP.L #$64,D5 BSR DISPC
BEQ SHRANIR MOVE.B D0,DRW.L
BRA AGAT RTS
VNOS MOVE.B #01,D0 VI MOVE.B #%01001001,D0 ;znak I
MOVE.B D0,PADAT.L BSR DISPC
MOVE.W D4,($30000).L MOVE.B D0,DRW.L
MOVE.B #00,D0 RTS
MOVE.B D0,PADAT.L VJ MOVE.B #%01001010,D0 ;znak J
BSR WAITRS BSR DISPC
RTS MOVE.B D0,DRW.L
RTS RTS
WAITRS CLR.L D2 VK MOVE.B #%01001011,D0 ;znak K
WAITRS1 ADDI.L #01,D2 BSR DISPC
CMP.L #$1FFF,D2 MOVE.B D0,DRW.L
NOP RTS
BNE WAITRS1 VL MOVE.B #%01001100,D0 ;znak L
RTS BSR DISPC
SHRANIR ADDI.L #$18,D4 VM MOVE.B D0,DRW.L
BSR VNOS MOVE.B D0,DRW.L
RTS RTS
; *****
; ** Potrditev **
; *****
RET RTS VO MOVE.B #%01001111,D0 ;znak O
MOVE.B #%00000110,D0 BSR DISPC
MOVE.B D0,UCR1.L MOVE.B D0,DRW.L
MOVE.W #$1D,D0 RTS
MOVE.W #$A0,D1 VP MOVE.B #%01010000,D0 ;znak P
MOVE.B D1,UTB1.L BSR DISPC
RT2 MOVE.B USR1.L,D2 MOVE.B D0,DRW.L
BTST #03,D2 RTS
BEQ RT2 VQ MOVE.B #%01010001,D0 ;znak Q
MOVE.B D0,UTB1.L BSR DISPC
RT3 MOVE.B USR1.L,D2 MOVE.B D0,DRW.L
BTST #03,D2 RTS
BEQ RT3 VR MOVE.B #%01010010,D0 ;znak R
RTS BSR DISPC
MOVE.B D0,DRW.L
RTS RTS
; VS MOVE.B #%01010011,D0 ;znak S
; ***** BSR DISPC
; ** Display Functions ** MOVE.B D0,DRW.L
; ***** RTS
;
; Waiting for display capture
; VT MOVE.B #%01010100,D0 ;znak T
DISPC MOVE.B #80,D1 BSR DISPC
MOVE.B IRR.L,D2 MOVE.B D0,DRW.L
AND D2,D1 RTS
BNE DISPC
RTS
;
; Chartype of display great letters
;
VA MOVE.B #%01000001,D0 ;znak A
BSR DISPC
MOVE.B D0,DRW.L
RTS
VW MOVE.B #01010111,D0 ;znak W
BSR DISPC
MOVE.B D0,DRW.L

```

```

VX      RTS
        MOVE.B  #01011000,D0 ;znak X
        BSR     DISPC
        MOVE.B  D0,DRW.L
VY      RTS
        MOVE.B  #01011001,D0 ;znak Y
        BSR     DISPC
        MOVE.B  D0,DRW.L
VZ      RTS
        MOVE.B  #01011010,D0 ;znak Z
        BSR     DISPC
        MOVE.B  D0,DRW.L
;
; Chartype of display, small letters
;
A      MOVE.B  #01100001,D0 ;znak a
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
B      MOVE.B  #01100010,D0 ;znak b
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
C      MOVE.B  #01100011,D0 ;znak c
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
D      MOVE.B  #01100100,D0 ;znak d
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
E      MOVE.B  #01100101,D0 ;znak e
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
F      MOVE.B  #01100110,D0 ;znak f
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
G      MOVE.B  #01100111,D0 ;znak g
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
H      MOVE.B  #01101000,D0 ;znak h
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
I      MOVE.B  #01101001,D0 ;znak i
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
J      MOVE.B  #01101010,D0 ;znak j
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
K      MOVE.B  #01101011,D0 ;znak k
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
L      MOVE.B  #01101100,D0 ;znak l
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
M      MOVE.B  #01101101,D0 ;znak m
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
N      MOVE.B  #01101110,D0 ;znak n
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
O      MOVE.B  #01101111,D0 ;znak o
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
P      MOVE.B  #01110000,D0 ;znak p
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
Q      MOVE.B  #01110001,D0 ;znak q
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
R      MOVE.B  #01110010,D0 ;znak r
        BSR     DISPC
        MOVE.B  D0,DRW.L
;
S      RTS
        MOVE.B  #01110011,D0 ;znak s
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
T      MOVE.B  #01110100,D0 ;znak t
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
U      MOVE.B  #01110101,D0 ;znak u
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
V      MOVE.B  #01110110,D0 ;znak v
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
W      MOVE.B  #01110111,D0 ;znak w
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
X      MOVE.B  #01111000,D0 ;znak x
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
Y      MOVE.B  #01111001,D0 ;znak y
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
Z      MOVE.B  #01111010,D0 ;znak z
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
;
; Special Char
;
SPACE  MOVE.B  #01111110,D0 ;presledek
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
DOT    MOVE.B  #00101110,D0 ;pika
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
VEC    MOVE.B  #00111110,D0 ;ve•je >
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
;
; Numerus
;
ZERO   MOVE.B  #00110000,D0 ;nula
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
ONE    MOVE.B  #00110001,D0 ;enka
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
TWO    MOVE.B  #00110010,D0 ;dvojka
        BSR     DISPC
        MOVE.B  D0,DRW.L
        RTS
;
; *****
; *   Display Setting   *
; *****
;
; Boot of display
;
LCD_INI MOVE.B  #1,D0
        BSR     DISPC
        MOVE.B  D0,IRW.L ;clr. disp.
        MOVE.B  #38,D0
        BSR     DISPC
        MOVE.B  D0,IRW.L
        MOVE.B  #0C,D0 ;Display ON
        BSR     DISPC
        MOVE.B  D0,IRW.L

```

```

;
;*****
;**      Text for Display      **
;*****
;
; Intro Text
;
INTRO  MOVE.B  #$C2,D0 ;2 vrstica
        BSR    DISPC
        MOVE.B  D0,IRW.L
        BSR    VS
        BSR    I
        BSR    G
        BSR    N
        BSR    A
        BSR    L
        BSR    SPACE
        BSR    VG
        BSR    E
        BSR    N
        BSR    E
        BSR    R
        BSR    A
        BSR    T
        BSR    O
        BSR    R
        MOVE.B  #$98,D0
        BSR    ISPC
        MOVE.B  D0,IRW.L
        BSR    VT
        BSR    E
        BSR    S
        BSR    T
        BSR    SPACE
        BSR    VV
        BSR    E
        BSR    R
        BSR    S
        BSR    I
        BSR    O
        BSR    N
        RTS

;-----
;      Deukalion intro text
;
DEUKTXT MOVE.B  #1,D0
        BSR    DISPC
MOVE.B  D0,IRW.L ;clear display

        MOVE.B  #$C3,D0 ;2 vrstica
        BSR    DISPC
        MOVE.B  D0,IRW.L
        BSR    VD
        BSR    VE
        BSR    VU
        BSR    VK
        BSR    VA
        BSR    VL
        BSR    VI
        BSR    VO
        BSR    VN
        BSR    SPACE
        BSR    SPACE
        BSR    TWO
        BSR    DOT
        BSR    ZERO
        RTS

;
;-----
;      comand line
;
COMMTXT MOVE.B  #1,D0
        BSR    DISPC
        MOVE.B  D0,IRW.L ;clear display

        MOVE.B  #$C2,D0 ;2 vrstica
        BSR    DISPC
        MOVE.B  D0,IRW.L
        BSR    VC
        BSR    O
        BSR    M
        BSR    M
        BSR    A
        BSR    N
        BSR    D
        BSR    VEC
        BSR    SPACE

```

```

        RTS
;
;-----
;      reciving signal txt to command
;
RCSGTX MOVE.B  #$96,D0 ;3 vrstica
        BSR    DISPC
        MOVE.B  D0,IRW.L
        BSR    VR
        BSR    E
        BSR    C
        BSR    I
        BSR    V
        BSR    I
        BSR    N
        BSR    G
        BSR    SPACE
        BSR    SPACE
        BSR    VS
        BSR    I
        BSR    G
        BSR    N
        BSR    A
        BSR    L
        RTS

;
;-----
;      reciving register txt to command
;
RCRGTX MOVE.B  #$96,D0 ;3 vrstica
        BSR    DISPC
        MOVE.B  D0,IRW.L
        BSR    SPACE
        BSR    VD
        BSR    E
        BSR    V
        BSR    I
        BSR    C
        BSR    E
        BSR    SPACE
        BSR    SPACE
        BSR    VS
        BSR    E
        BSR    T
        BSR    U
        BSR    P
        RTS

;
;-----
;      reciving R1 txt to command
;
R1STXT MOVE.B  #$96,D0 ;3 vrstica
        BSR    DISPC
        MOVE.B  D0,IRW.L
        BSR    SPACE
        BSR    SPACE
        BSR    SPACE
        BSR    VS
        BSR    E
        BSR    T
        BSR    U
        BSR    P
        BSR    SPACE
        BSR    SPACE
        BSR    SPACE
        BSR    VR
        BSR    ONE
        RTS

;
;-----
;      reciving R2 txt to command
;
R2STXT MOVE.B  #$96,D0 ;3 vrstica
        BSR    DISPC
        MOVE.B  D0,IRW.L
        BSR    SPACE
        BSR    SPACE
        BSR    SPACE
        BSR    VS
        BSR    E
        BSR    T

```

```

BSR    U
BSR    P
BSR    SPACE
BSR    SPACE
BSR    SPACE
BSR    VR
BSR    TWO
RTS

;
;-----
;
;   generating signal text
;
GENSX  MOVE.B #1,D0
        BSR    DISPC
        MOVE.B D0,IRW.L ;clear display

        MOVE.B #$C4,D0 ;2 vrstica
        BSR    DISPC
        MOVE.B D0,IRW.L
        BSR    VG
        BSR    E
        BSR    N
        BSR    E
        BSR    R
        BSR    A
        BSR    T
        BSR    I
        BSR    N
        BSR    G
        MOVE.B #$9A,D0 ;2 vrstica
        BSR    DISPC
        MOVE.B D0,IRW.L
        BSR    VS
        BSR    I
        BSR    G
        BSR    N
        BSR    A
        BSR    L
        RTS

;
;-----
;
;-----

```

```

;
;   generating signal text
;
SETTXT MOVE.B #1,D0
        BSR    DISPC
        MOVE.B D0,IRW.L ;clear display

        MOVE.B #$C4,D0 ;2 vrstica
        BSR    DISPC
        MOVE.B D0,IRW.L
        BSR    VI
        BSR    N
        BSR    I
        BSR    T
        BSR    I
        BSR    A
        BSR    L
        BSR    I
        BSR    Z
        BSR    E
        MOVE.B #$9A,D0 ;2 vrstica
        BSR    DISPC
        MOVE.B D0,IRW.L
        BSR    VD
        BSR    E
        BSR    V
        BSR    I
        BSR    C
        BSR    E
        RTS

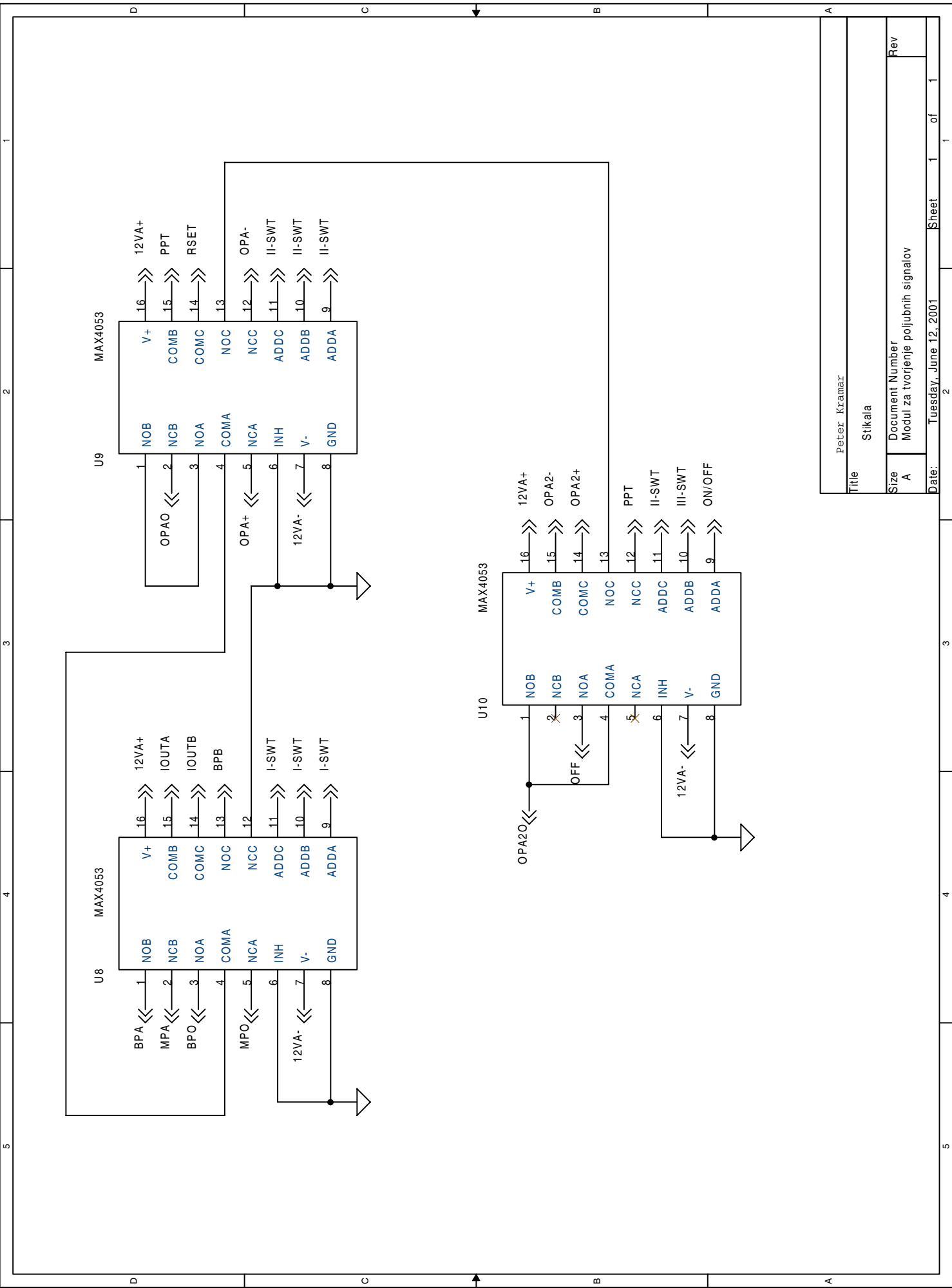
;
;-----
;
;*****
;**          Delay Functions          **
;*****
;
WAITL  CLR.L  D2
WAITL1 ADDI.L #01,D2
        CMP.L #2FFFF,D2
        NOP
        BNE  WAITL1
        RTS

;

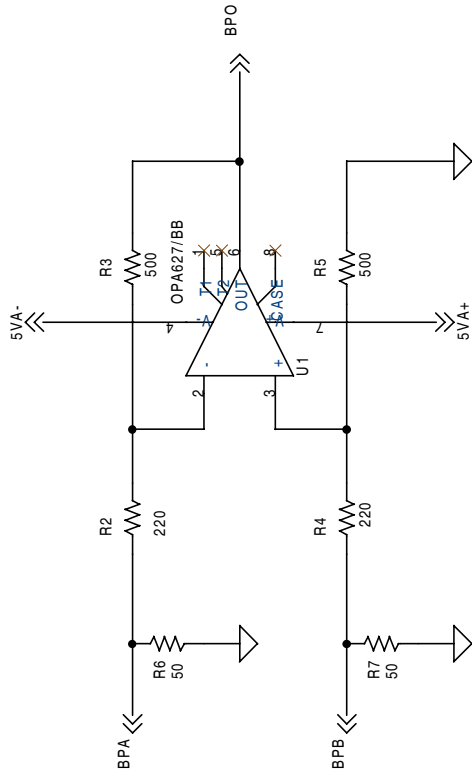
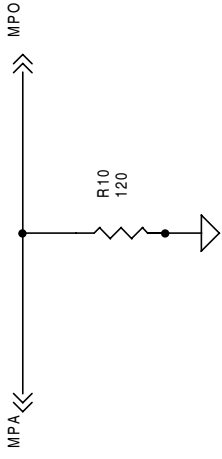
```

## **6.2. Električna shema modula za tvorjenje poljubnih signalov**





Title		Peter Kramar	
Size		Stikala	
Document Number		Modul za tvorjenje poljubnih signalov	
Rev	A	Sheet	1 of 1
Date:	Tuesday, June 12, 2001	Sheet	1 of 1



Peter Kramar

Title  
Tokovno napetostni pretvornik

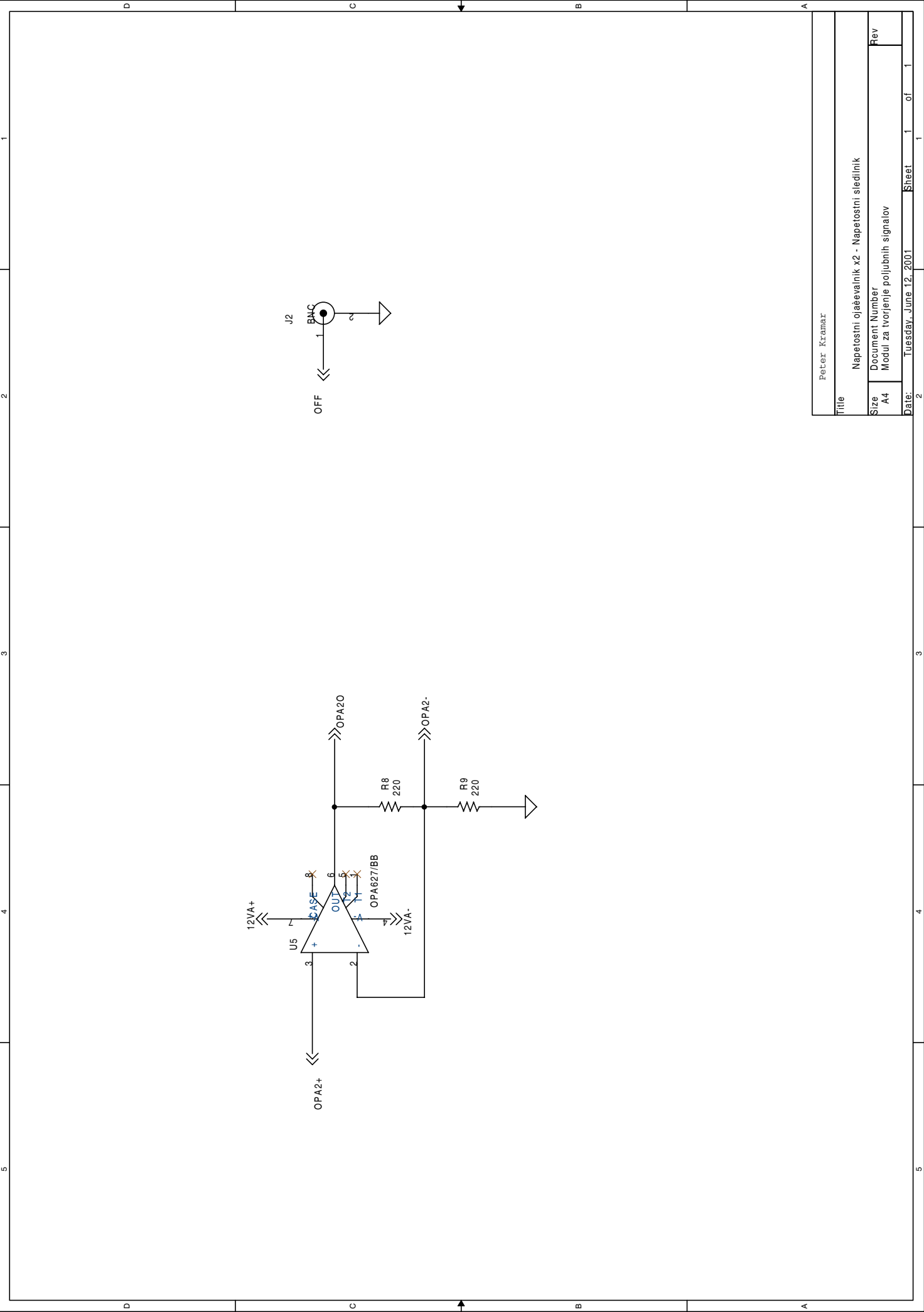
Size  
A4

Document Number  
Modul za tvorjenje poljubnih signalov

Date: Tuesday, June 12, 2001

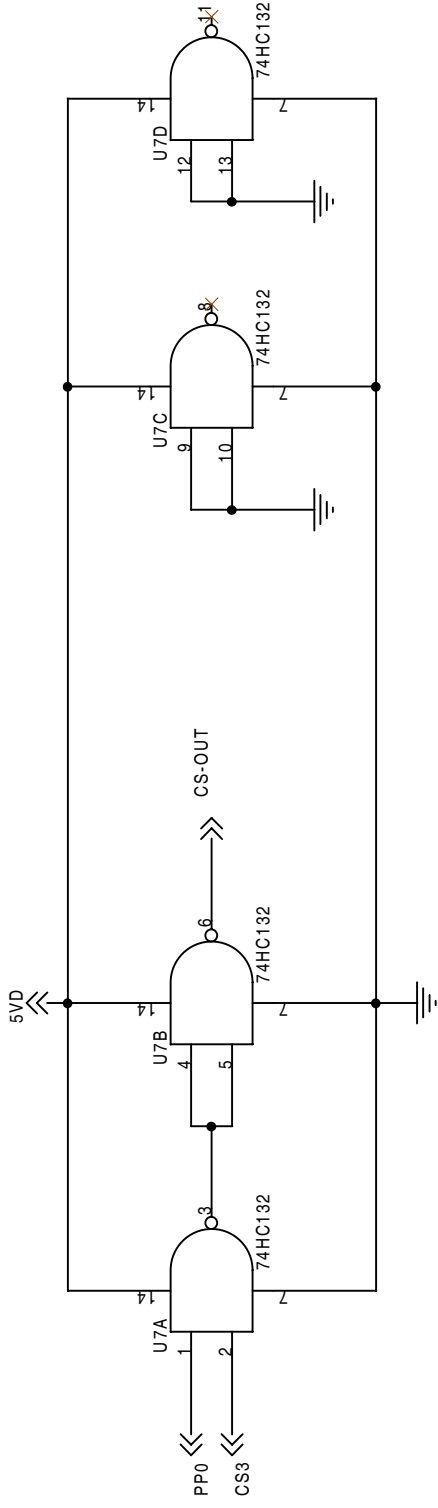
Sheet 1 of 1

Rev

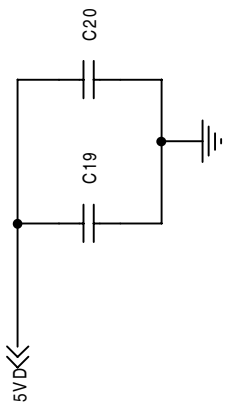
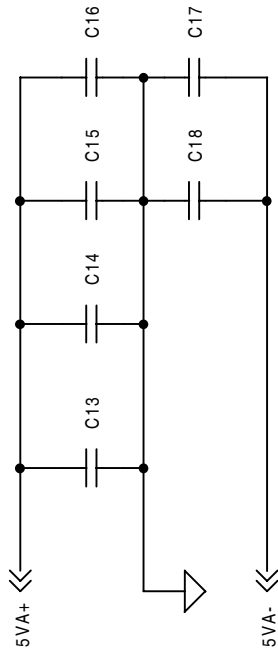
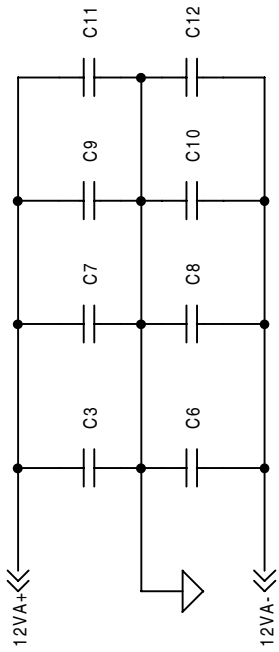


Peter Kramar

Title	
Napetostni ojaevalnik x2 - Napetostni sledilnik	
Size	Document Number
A4	Modul za tvorjenje poljubnih signalov
Date:	Rev
Tuesday, June 12, 2001	1
Sheet	of
1	1



Peter Kramar	
CS Logika	
Document Number	Rev
A	
Date: Tuesday, June 12, 2001	Sheet 1 of 1



Title		Blokirni kondenzatorji za integrirana vezja	
Size	Document Number	Rev	
A	Modul za tvorjenje poljubnih signalov		
Date:	Tuesday, June 12, 2001	Sheet	1 of 1