

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD BR. 46

**Trokanalni elektrokardiograf s memorijom i  
bežičnim prijenosom signala**

KREŠIMIR TUŠEK

Zagreb, lipanj 2010.



## Sadržaj

1	Uvod .....	7
2	Opća arhitektura uređaja .....	8
3	Analogni dio uređaja .....	10
3.1	EKG signal i smetnje .....	10
	Smetnje zbog struja koje teku kroz tijelo .....	12
	Smetnje zbog struja koje teku u pojačalo .....	12
	Smetnje zbog struja koje teku u mjerne kablove .....	12
	Magnetski inducirane smetnje .....	13
	Polarizacijski napon elektroda .....	14
	Utjecaj istofaznog napona .....	14
3.2	Izvedba pojačala .....	16
	Ulagani stupanj i realizacija donje granične frekvencije .....	16
	Drugi stupanj pojačala .....	20
	Realizacija gornje granične frekvencije i treći stupanj pojačanja .....	21
4	Memorija .....	24
4.1	Izbor vrste memorije .....	24
	DRAM memorija .....	24
	SRAM memorija .....	24
	EEPROM memorija .....	25
	FRAM memorija .....	25
	NAND FLASH memorija .....	25
4.2	Hynix 512Mb NAND FLASH .....	26
5	Bluetooth veza .....	29
6	Mikrokontroler .....	33
6.1	Arhitektura porodice mikrokontrolera LPC13xx .....	34
6.2	Upravljanje potrošnjom mikrokontrolera LPC1313 .....	35
7	Baterijsko napajanje i potrošnja .....	42
7.1	Procjena potrošnje uređaja .....	43
7.2	Mogućnosti optimiranja potrošnje uređaja .....	45
7.3	Izvedba punjača baterija .....	47
8	Ugrađena programska podrška ( <i>firmware</i> ) .....	49
8.1	CMSIS .....	49
	<i>LPC13xx.h</i> .....	49
	core_cm3.c i core_cm3.h .....	50
	startup_lpc13.c .....	51
	system_lpc13xx.c i system_lpc13xx.h .....	51
8.2	Struktura ugrađene programske podrške .....	51
8.3	Upravljanje AD pretvornikom .....	56
8.4	Pristup FLASH memoriji .....	57
	Čitanje stranice iz memorije .....	58
	Pisanje stranice u memoriju .....	61
	Brisanje bloka .....	64
	Čitanje statusnog registra .....	66
8.5	Slanje podataka preko Bluetooth veze .....	67
9	Rezultati .....	69
10	Zaključak .....	71

Literatura .....	72
Sažetak .....	73
Trokanalni elektrokardiograf s memorijom i bežičnim prijenosom signala .....	73
Summary .....	73
3-channel electrocardiograph with memory and wireless data transfer .....	73
A. Privitak: Tehnička dokumentacija.....	74
A.1 Električka shema.....	74
A.2 Položajni nacrt .....	78
A.3 Tiskana pločica .....	80

## Popis slika

Slika 1. – Blok shema uređaja.....	9
Slika 2. – Položaj elektroda za mjerjenje standardnih Einthovenovih odvoda .....	10
Slika 3. – Model za opisivanje smetnji pri mjerenuju EKG signala .....	11
Slika 4. – Izvedba sklopa za pogon desne noge .....	16
Slika 5. – Shema instrumentacijskog pojačala INA333 .....	17
Slika 6. – Shema ulaznog stupnja pojačala .....	19
Slika 7. – Virtualna masa pojačala .....	20
Slika 8. – Shema drugog stupnja pojačala.....	21
Slika 9. – Amplitudno frekvencijska i fazno frekvencijska karakteristika niskopropusnog filtra .....	22
Slika 10. – Shema niskopropusnog filtra .....	23
Slika 11. – Priključci Hynix FLASH memorije .....	27
Slika 12. – Dimenzije modula WT-12 .....	30
Slika 13. – Blok shema porodice mikrokontrolera LPC13xx .....	35
Slika 14. – Blok shema jedinice za generiranje <i>clock</i> signala porodice mikrokontrolera LPC13xx.....	37
Slika 15. – Potrošnja mikrokontrolera LPC1313 u normalnom načinu rada u ovisnosti o temperaturi i frekvenciji sistemskog <i>clock</i> signala .....	38
Slika 16. – Potrošnja mikrokontrolera LPC1313 u <i>sleep</i> načinu rada u ovisnosti o temperaturi i frekvenciji sistemskog <i>clock</i> signala.....	39
Slika 17. – Potrošnja mikrokontrolera LPC1313 u <i>deep-sleep</i> načinu rada u ovisnosti o temperaturi i naponu napajanja .....	40
Slika 18. – Potrošnja mikrokontrolera LPC1313 u <i>deep power-down</i> načinu rada u ovisnosti o temperaturi i naponu napajanja .....	41
Slika 19. – Shema punjača li-ion akumulatora.....	47
Slika 20. – Blok-dijagram prekidne rutine vremenskog sklopa <i>SysTick</i> .....	53
Slika 21. – Blok-dijagram funkcije <i>main()</i> .....	55
Slika 22. – Tijek čitanja jedne stranice iz FLASH memorije.....	58

Slika 23. – Tijek pisanja stranice u FLASH memoriju .....	61
Slika 24. – Vremenski tijek brisanja bloka FLASH memorije .....	64
Slika 25. – Izgled gotovog elektrokardiografa .....	69
Slika 26. - Primjer EKG signala snimljenog razvijenim elektrokardiografom (bez digitalnog filtriranja) .....	70
Slika 27. – Primjer EKG signala snimljenog razvijenim elektrokardiografom (uz digitalno filtriranje) .....	70
Slika 28. – Shema napajanja .....	75
Slika 29. – Shema analognog dijela .....	76
Slika 30. – Shema digitalnog dijela.....	77
Slika 31. – Položajni nacrt gornjeg sloja.....	78
Slika 32. – Položajni nacrt donjeg sloja .....	79
Slika 33. – Izgled gornjeg sloja tiskane pločice .....	80
Slika 34. – Izgled donjeg sloja tiskane pločice .....	81

# 1 Uvod

Elektrokardiogram (EKG) je bioelektrički signal nastao aktivnošću srca. Predstavlja najbolji način dijagnosticiranja abnormalnosti u srčanom ritmu i njihovog mjerena. Zbog toga je mogućnost dugotrajnijeg snimanja EKG-a od neizmjerne važnosti u kardiološkoj dijagnostici i praćenju stanja pacijenta.

Minijaturizacija elektroničkih sklopova omogućava izradu elektrokardiografa (uređaja za mjerjenje i prikaz EKG-a) koji su malih dimenzija, prenosivi i lagani. Zbog takvih karakteristika uređaja moguće je snimati EKG i izvan bolničkog okruženja, tijekom svakodnevnih aktivnosti pacijenta.

Cilj ovog rada je projektirati i izraditi upravo takav uređaj. On mora biti malen, baterijski napajan, s vlastitom memorijom za pohranu signala. Osim toga, mora imati mogućnost bežične komunikacije (putem Bluetooth veze) s drugim uređajima, poput osobnog računala ili mobitela, koji mogu pohranjeni signal prikazati.

U sklopu rada opisane su komponente prijenosnog elektrokardiografa, zahtjevi na njihove karakteristike te njihova sklopovska realizacija. Opisana je i ugrađena programska podrška koja obavlja zadatke uzorkovanja i spremanja signala kao i bežičnu komunikaciju s drugim uređajem.

Opisani uređaj je i realiziran na dvoslojnoj tiskanoj pločici smještenoj u kutiju dimenzija 88x63x27 mm. Izrađena je i *Matlab* skripta koja uspostavlja bežičnu vezu s uređajem u svrhu preuzimanja podataka i njihovog prijenosa na računalo. Rezultat izvođenja skripte je grafički prikaz uzorkovanih signala čime je potvrđena ispravnost uređaja.

## 2 Opća arhitektura uređaja

Za bežični prijenosni elektrokardiograf definirana je sljedeća lista zahtjeva:

- prenosivost
- autonomno napajanje
- trokanalno mjerjenje (Einthoven-ovi odvodi) uz pojačanje 1000, donju graničnu frekvenciju 2Hz i gornju graničnu frekvenciju 150Hz
- 8-bitna razlučivost uzorkovanja
- memoriranje barem jedne minute signala (što više, to bolje)
- prijenos preko Bluetooth veze na računalo

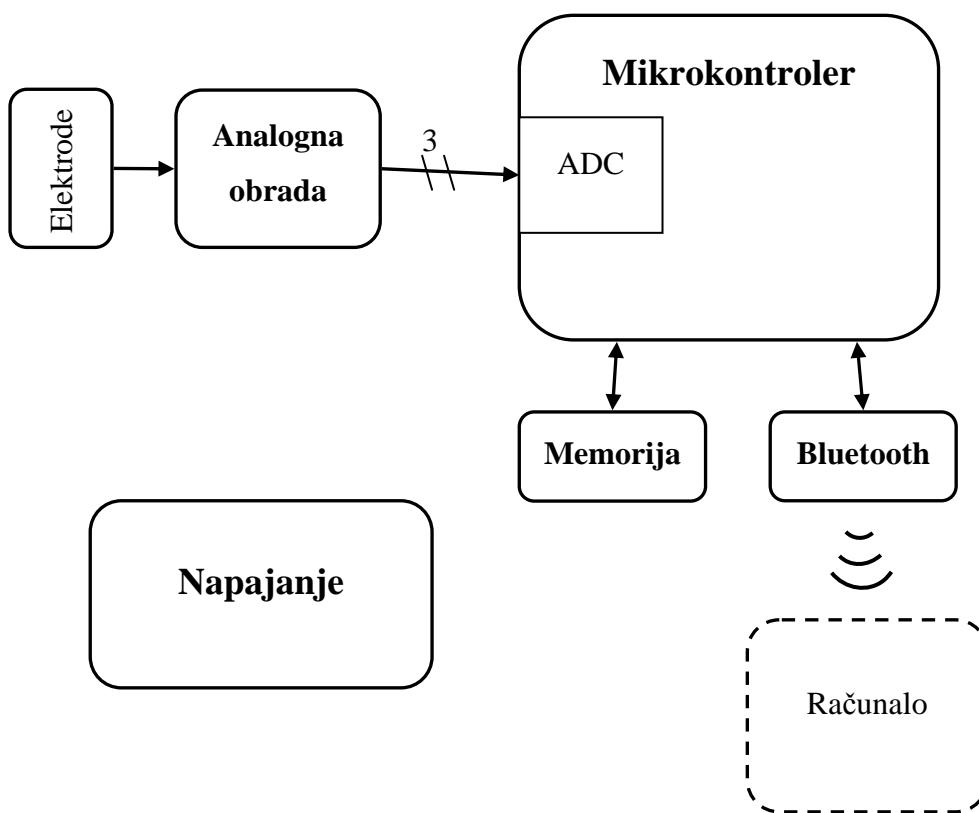
Osnovna značajka uređaja je njegova prenosivost što znači da je nužno napajati ga baterijskim napajanjem. Kod baterijski napajanih uređaja kritična karakteristika je potrošnja, što će utjecati na izbor svih komponenata. Osim toga bit će potrebno zadržati što manje dimenzije uređaja odnosno što manji broj komponenata. Mala potrebna razlučivost signala (8 bita) otvara mogućnost korištenja mikrokontrolera s ugrađenim Analogno-digitalnim (AD) pretvornikom, čime se štedi na broju komponenata (nema potrebe za vanjskim AD pretvornikom). Što se tiče memoriranja, već u prvotnom razmišljanju o konceptu uređaja postalo je jasno da memorija tipičnog mikrokontrolera neće biti dovoljna za pohranjivanje signala. Uz jedan bajt po uzorku, 500-tinjak uzoraka u sekundi na tri kanala ispada da je potrebno oko 90kB memorije za jednu minutu signala. Takva količina memorije je uglavnom izvan domašaja većine mikrokontrolera niske cijene i niske potrošnje. Osim toga, mikrokontroleri koji i imaju dovoljno memorije, imaju je u obliku radne memorije koja ne može zadržati podatke bez prisutnog napajanja što može biti nepraktično za ovakav uređaj. Prema tome, trebao bi sadržavati neki oblik trajne memorije. Količina takve memorije mogla bi biti i veća nego što je potrebno za minimalno traženih jednu minutu signala, čime bi se funkcionalnost uređaja povećala.

Sukladno prethodnom razmatranju uređaj će sadržavati sljedeće cjeline:

- analogni dio (pojačala i filtri EKG signala)

- mikrokontroler (uzorkovanje signala, upravljanje memorijom i komunikacijom)
- bluetooth modul (komunikacija s računalom)
- memorija
- napajanje

Odnos pojedinih dijelova uređaja prikazan je na slici 1.



**Slika 1. – Blok shema uređaja**

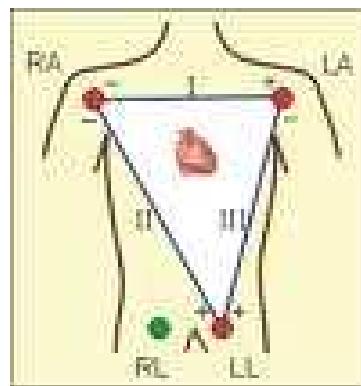
U narednim poglavljima bit će detaljnije opisana izvedba svakog od navedenih dijelova.

## 3 Analogni dio uređaja

### 3.1 EKG signal i smetnje

Zadaća analognog dijela uređaja je izdvojiti i pojačati EKG signal. Prije nego što se pristupi opisu konstruiranog pojačala, potrebno je opisati sam EKG signal i smetnje koje otežavaju njegovo pojačavanje.

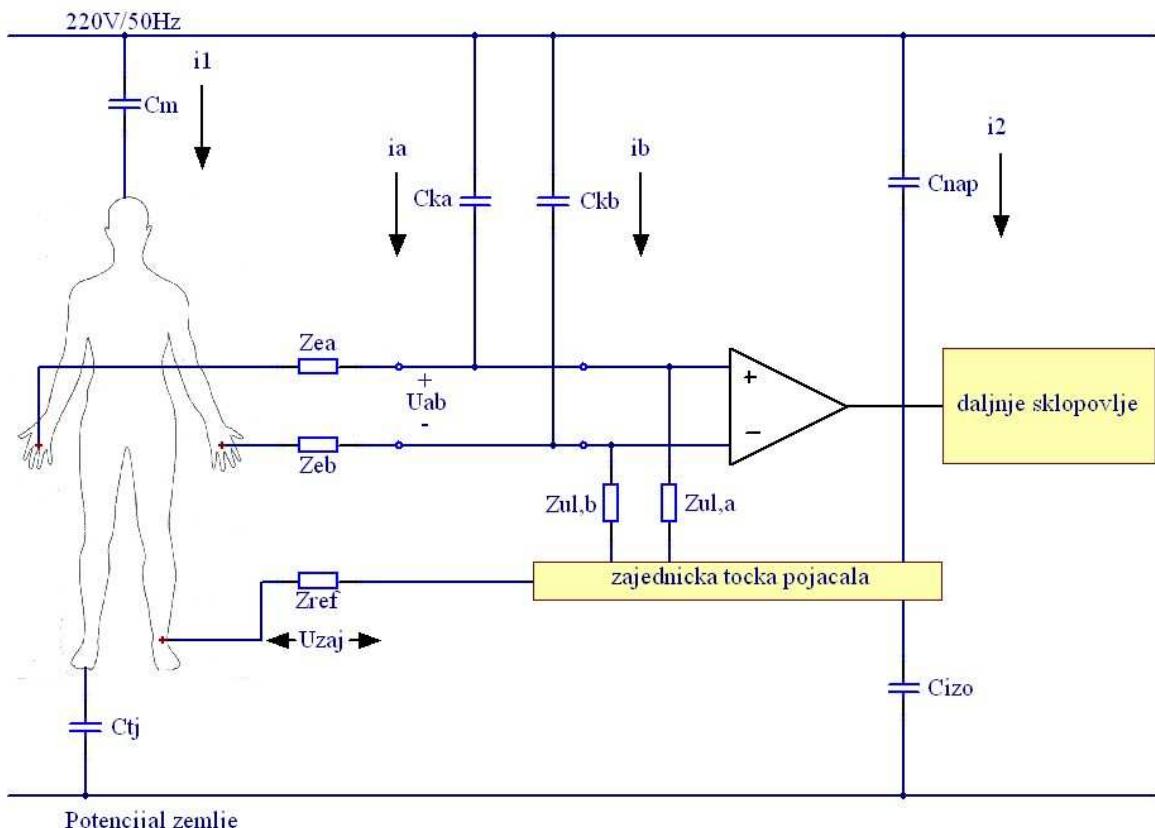
EKG signal je male amplitude (reda veličine nekoliko mV) i u ovom slučaju se mjeri sa standardnih Einthovenovih odvoda (lijeva i desna ruka te lijeva nogu). Signal se mjeri diferencijalno između dvije elektrode. Na tri elektrode moguće je izabrati tri takva para elektroda, pa zato postoje tri Einthovenova odvoda. Raspored elektroda u tom slučaju prikazan je na slici 2. Zeleno označena elektroda na slici je referentna i spaja se na masu pojačala.



**Slika 2. – Položaj elektroda za mjerjenje standardnih Einthovenovih odvoda**

EKG signal potrebno je pojačati oko 1000 puta da bi mu se amplituda dovela na razinu volta tj. unutar dinamičkog područja AD pretvornika. Tu zadaću otežava prisutnost različitih smetnji. Najčešće se radi o smetnjama nastalim zbog blizine vodova gradske mreže koje se prenose elektrostatskim putem. Osim toga prisutne su i magnetski inducirane smetnje te napon polarizacije elektroda. Prijenos smetnji elektrostatskim putem bit će opisan na modelu mjernog sustava sa slike 3 (Metting van Rijn, 1990). Model opisuje mjerjenje jednog kanala EKG-a uz referentnu elektrodu koja se kod mjerjenja EKG-a tipično postavlja na desnu nogu. Moguće je mjeriti i bez referentne elektrode, ali takvo

mjerenje donosi drugačiju problematiku koja ovdje neće biti obrađena. Impedancije  $Z_{ea}$ ,  $Z_{eb}$  i  $Z_{ref}$  predstavljaju impedancije pojedinih elektroda. Kapacitet  $C_{tj}$  opisuje ukupan utjecaj svih parazitnih kapaciteta između tijela i zemlje dok kapacitet  $C_m$  opisuje ukupan utjecaj svih parazitnih kapaciteta između tijela i vodova gradske mreže. Tipični iznosi su 300pF za  $C_{tj}$  i 3pF za  $C_m$  (Metting van Rijn, 1990), ali treba računati na to da njihove vrijednosti mogu varirati i za red veličine. Kapaciteti  $C_{ka}$  i  $C_{kb}$  opisuju parazitne kapacitete mjernih kablova prema vodovima gradske mreže. Kapaciteti  $C_{nap}$  i  $C_{izo}$  opisuju parazitne kapacitete između zajedničke plivajuće točke pojačala i vodova gradske mreže te između zajedničke plivajuće točke i zemlje. Impedancije  $Z_{ul,a}$  i  $Z_{ul,b}$  predstavljaju ulazne impedancije pojačala.



**Slika 3. – Model za opisivanje smetnji pri mjerenu EKG signala**

Uz opisanom modelu mjernog sustava mogu se izdvojiti tri izvora elektrostatskih smetnji:

- smetnje zbog struja koje teku kroz tijelo
- smetnje zbog struja koje teku u pojačalo
- smetnje zbog struja koje teku u mjerne kablove

### **Smetnje zbog struja koje teku kroz tijelo**

Kroz parazitne kapacitete  $C_m$  i  $C_{tj}$  teče struja  $i_1$  između vodova gradske mreže i zemlje. Kada se na pacijenta spoji pojačalo dio struje će se zatvoriti prema zemlji i preko impedancije zajedničke elektrode  $Z_{ref}$  kroz parazitni kapacitet  $C_{izo}$ . Ta struja će na impedanciji  $Z_{ref}$  uzrokovati razliku potencijala između srednjeg potencijala tijela i plivajuće referentne točke pojačala. Ta će se razlika efektivno vidjeti kao istofazna naponska smetnja na ulazu pojačala. Iznos tog napona bit će (Krois, 2001):

$$U_{zaj} = i_1 \frac{Z_{tj}}{Z_{tj} + Z_{nap} || Z_{izo}} Z_{ref} \quad (1)$$

### **Smetnje zbog struja koje teku u pojačalo**

Kroz kapacitete  $C_{nap}$  i  $C_{izo}$  teče struja  $i_2$  između vodova gradske mreže i zemlje. Dok je pacijent spojen na pojačalo struja smetnje se prema zemlji ne zatvara samo kroz  $C_{izo}$  nego i preko impedancije referentne elektrode kroz paralelnu kombinaciju kapaciteta  $C_{tj}$  i  $C_m$ . Ta struja će kao i u prethodnom slučaju izazvati pad napona na impedanciji referentne elektrode koji će se na ulazu pojačala vidjeti kao istofazna naponska smetnja. Iznos tog napona bit će (Krois, 2001):

$$U_{zaj} = i_2 \frac{Z_{izo}}{Z_{izo} + Z_{tj}} Z_{ref} \quad (2)$$

Valja primijetiti da će se zbog različitih smjerova struja naponi dviju opisanih oduzimati za dani slučaj (ako je izvor smetnji samo gradska mreža), ali to općenito ne mora biti tako pa se za najgori slučaj utjecaj ovih dviju smetnji treba zbrojiti (Krois, 2001).

### **Smetnje zbog struja koje teku u mjerne kablove**

Dva prethodno opisana načina unosa smetnji u mjerjenje rezultiraju istofaznim naponom smetnje na ulazu pojačala. Takva smetnja ne predstavlja velik problem ukoliko se koristi pojačalo s dovoljno visokim faktorom potiskivanja istofaznog signala (engl. *common mode rejection ratio - CMRR*), npr. instrumentacijsko pojačalo. S druge strane, smetnje izazvane

strujama koje teku u mjerne kablove rezultirat će diferencijalnim naponima smetnje na ulazu pojačala i na taj način imati veliki utjecaj na mjerjenje EKG signala. Struje  $i_a$  i  $i_b$  koje teku preko parazitnih kapaciteta  $C_{ka}$  i  $C_{kb}$  zatvarat će se prema zemlji kroz elektrode, tijelo, parazitni kapacitet  $C_{iso}$  te serijsku kombinaciju impedancije referentne elektrode  $Z_{ref}$  i parazitnog kapaciteta  $C_{tj}$ . Pošto se te dvije struje po iznosu uglavnom razlikuju (zbog različitih parazitnih kapaciteta kablova prema vodovima gradske mreže) i pošto se impedancije elektroda razlikuju, smetnja će se na pojačalu vidjeti kao diferencijalni napon na slici 3 označen sa  $U_{ab}$ . Iznos ovog napona smetnje dan je izrazom (Metting van Rijn, 1990):

$$V_{ab} = i_a Z_{ea} - i_b Z_{eb} = i Z_e \left( \frac{\Delta Z_e}{Z_e} + \frac{\Delta i}{i} \right), \quad (3)$$

gdje je  $i = 1/2(i_a + i_b)$  i  $Z_e = 1/2(Z_{ea} + Z_{eb})$

Uzmimo npr. srednju struju kroz kablove iznosa 10nA, srednju impedanciju elektroda iznosa  $20\text{k}\Omega$  i relativnu razliku u strujama kroz kablove i impedancijama elektroda od 50%. U tom će slučaju iznos diferencijalne smetnje na ulazu u pojačalo iznositi  $200\mu\text{V}_{pp}$  (Metting van Rijn, 1990), što je samo nekoliko puta manje od korisnog signala (EKG). Pošto je ova smetnja diferencijalna, bit će pojačana zajedno s korisnim signalom. Iz izraza (3) vidljivo je da će za smanjenje ove smetnje biti potrebno što više ujednačiti impedancije elektroda i parazitne kapacitete kablova prema vodovima gradske mreže. Smanjenje razlike između tih kapaciteta općenito se može postići tako da se mjerni kablovi približe jedan drugome što je više moguće. Osim toga smetnja se može smanjiti i udaljavanjem cijelog mjernog sustava i pacijenta od vodova gradske mreže, ali to nije uvijek praktično.

### Magnetski inducirane smetnje

Magnetski inducirane smetnje nastaju na principu elektromagnetske indukcije napona na vodljivoj petlji. Mjerni kablovi tvore petlju o čijoj površini ovisi amplituda smetnji. Smetnje se mogu smanjiti smanjivanjem površine petlje (npr. uplitanjem kablova), udaljavanjem od izvora smetnji te oklapanjem pacijenta ili izvora smetnji slojem materijala s visokom magnetskom permeabilnošću (mu-metal).

## Polarizacijski napon elektroda

Razlike između polarizacijskih napona pojedinih elektroda mogu doseći i do 200mV. Taj napon se na ulazu pojačala vidi kao diferencijalni i bit će pojačan zajedno s korisnim signalom. Polarizacijski napon elektroda postavlja bitno ograničenje na dopušteno pojačanje ulaznog diferencijalnog stupnja pojačala jer ga već uz mala pojačanja može odvesti u zasićenje. Pošto polarizacijski napon predstavlja DC smetnju, bit će ga moguće ukloniti visokopropusnim filtrom. Filter se može smjestiti na različita mjesta unutar pojačala o čemu će biti riječi kasnije.

## Utjecaj istofaznog napona

Već je pokazano da se putem različitih mehanizama na ulazu pojačala može pojaviti istofazni napon. Postoje dva načina na koje istofazni napon može postati smetnja. Prvi način je nedovoljnim potiskivanjem istofaznog napona u pojačalu. Međutim taj mehanizam uglavnom nije problematičan pošto današnja diferencijalna pojačala nerijetko imaju faktore potiskivanja istofaznog signala preko 100dB. Drugi način na koji istofazni napon može postati smetnja je pretvaranje istofaznog napona u diferencijalni zbog razlika u impedancijama elektroda i u ulaznim impedancijama. Ovaj efekt naziva se efektom naponskog djelitelja. Amplituda diferencijalnog napona smetnje koji nastaje na ovaj način, na modelu sa slike 3, iznosi:

$$U_{ab} = U_{zaj} \left( \frac{Z_{ul,a}}{Z_{ul,a} + Z_{ea}} - \frac{Z_{ul,b}}{Z_{ul,b} + Z_{eb}} \right) \quad (4)$$

Uz pretpostavku da su ulazne impedancije puno veće od impedancija elektroda, izraz (4) se može zapisati kao (Metting van Rijn, 1990):

$$U_{ab} = U_{zaj} \frac{Z_e}{Z_i} \left( \frac{\Delta Z_e}{Z_e} + \frac{\Delta Z_{ul}}{Z_{ul}} \right), \quad (5)$$

gdje je  $Z_e = 1/2(Z_{ea} + Z_{eb})$  i  $Z_{ul} = 1/2(Z_{ul,a} + Z_{ul,b})$ .

Iz izraza (5) vidljivo je da će razina diferencijalne smetnje proizašle iz istofaznog napona može smanjiti na sljedeće načine:

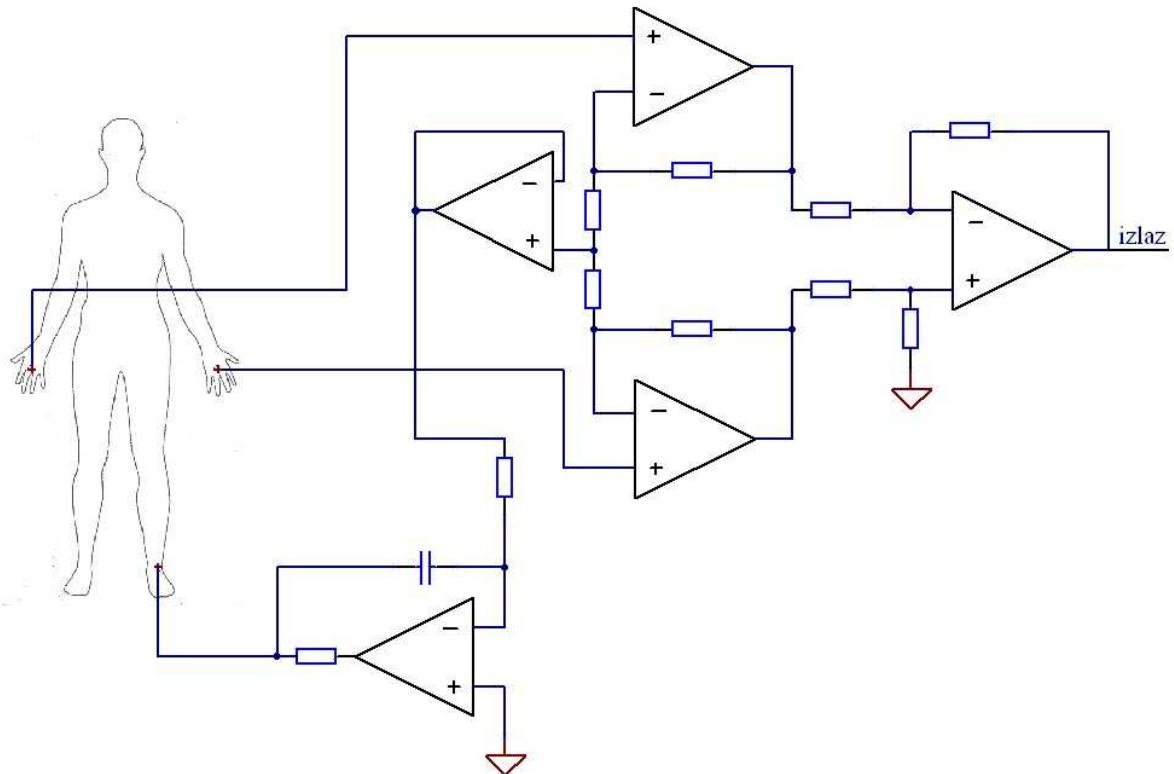
- povećanjem omjera iznosa ulaznih impedancija naspram impedancija elektroda
- smanjenjem razlika među impedancijama elektroda

- smanjenjem razlika među ulaznim impedancijama

Povećanje omjera ulaznih impedancija i impedancija elektroda može se postići korištenjem pojačala s velikom ulaznom impedancijom. Osim toga, pripremom kože pacijenta uz korištenje vodljivog gela mogu se impedancije elektroda smanjiti i ujednačiti. Razlike u ulaznim impedancijama uglavnom su posljedica različitih parazitnih kapaciteta na ulazima. Razlike se mogu smanjiti pažljivim dizajnom ulaznog stupnja i korištenjem oklopljenih kablova iste duljine (ako se koriste oklopljeni kablovi).

Osim smanjivanja efekta naponskog djelitelja moguće je i poduzeti korake da se smanji amplituda istofaznog napona, a time i amplituda diferencijalnog napona koji iz njega nastaje putem efekta naponskog djelitelja. Pošto istofazni napon na ulazu pojačala nastaje zbog parazitnih kapaciteta tijela i pojačala prema vodovima gradske mreže i zemlji, može ga se smanjiti smanjivanjem iznosa tih kapaciteta. Korištenjem baterijski napajanog plivajućeg pojačala (kao što se koristi u ovom radu) utjecaj struja koje teku u pojačalo postaje zanemariv jer je iznos kapaciteta  $C_{\text{nap}}$  za takvo pojačalo manji od 1pF (Metting van Rijn, 1990). U tom slučaju parazitni kapaciteti tijela prema vodovima gradske mreže tipično generiraju oko 1mV istofaznog napona što ne predstavlja velik problem.

Ako se smanjivanjem parazitnih kapaciteta ne može postići dovoljno mala razina smetnji, istofazni napon smetnje može se smanjiti na aktivan način, korištenjem sklopa za pogon desne noge. Radi se o integratoru u petlji negativne povratne veze koja nastoji održati što manju razliku između srednjeg potencijala tijela i referentne točke pojačala. Slika 4 prikazuje izvedbu sklopa za pogon desne noge ako se u ulaznom stupnju koristi instrumentacijsko pojačalo izvedeno s tri operacijska pojačala. Potiskivanje istofaznog napona koje se postiže ovim sklopom iznosi između 10 i 50 dB (Metting van Rijn, 1990). Pri projektiranju ovakvog sklopa treba pronaći optimum između pojačanja petlje povratne veze i stabilnosti pojačala. Veće pojačanje donosi bolje praćenje potencijala referentne točke pojačala, ali sklop može postati nestabilan.



Slika 4. – Izvedba sklopa za pogon desne noge

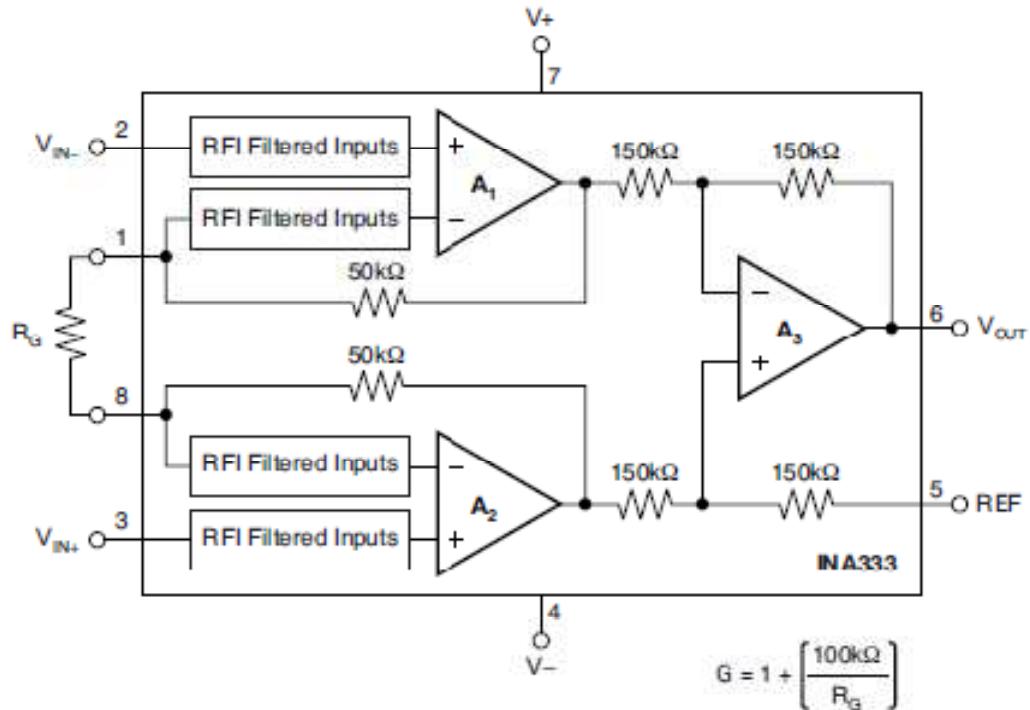
### 3.2 Izvedba pojačala

#### Ulagni stupanj i realizacija donje granične frekvencije

Kao što je već rečeno, za ulagni stupanj je potrebno koristiti diferencijalno pojačalo s dovoljno velikim faktorom potiskivanja istofaznog signala (npr. instrumentacijsko pojačalo). Odlučeno je da će se koristiti monolitno instrumentacijsko pojačalo zbog kompaktnosti i male potrošnje koju donosi monolitna izvedba. Na njega su postavljeni sljedeći zahtjevi:

- što veći faktor potiskivanja istofaznog signala na frekvencijama oko 10Hz
- što veća ulazna impedancija
- što manja potrošnja
- mogućnost rada s malim naponom napajanja (zbog baterijskog napajanja, poželjno 3V ili manje)

Odobrano je pojačalo INA333 proizvođača Texas Instruments. Radi se o izvedbi instrumentacijskog pojačala s tri operacijska pojačala (slika 5) u MSOP-8 kućištu. Osim samog pojačala na istom čipu je izvedena i zaštita od RF smetnji u vidu pasivnog RC-filtra granične frekvencije 8MHz na svakom od ulaza. Takva zaštita je korisna kada se uzme u obzir da uređaj može biti u blizini razne komunikacijske opreme, prvenstveno mobitela.



**Slika 5. – Shema instrumentacijskog pojačala INA333**

Karakteristike pojačala INA333 bitne za razvijani uređaj su:

- minimalni CMRR iznosa 80dB od 0 do 60Hz pri pojačanju 1
  - ulazna impedancija iznosa  $100G\Omega \parallel 3pF$
  - napon napajanja 1.8V do 5.5V
  - maksimalna potrošnja  $80\mu A$
  - pakirano u kućište MSOP-8 malih dimenzija

Već je bilo rečeno da će bitno ograničenje pri dizajnu ulaznog stupnja predstavljati razlika između polarizacijskih napona pojedinih elektroda koja se na ulazu pojačala pojavljuje kao istosmjerna diferencijalna smetnja iznosa do 200mV. Tu smetnju će biti potrebno ukloniti

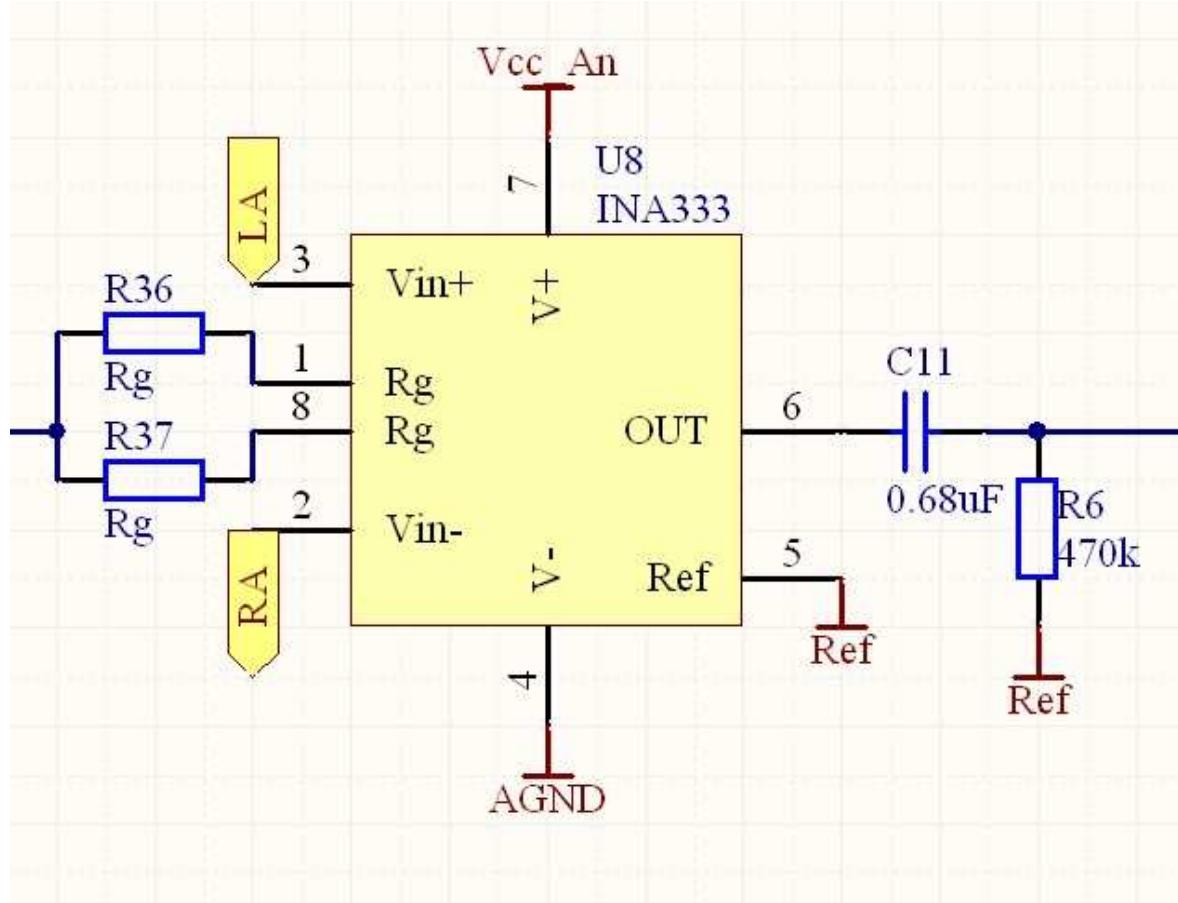
visokopropusni filtrom jer bi inače mogla odvesti daljnje stupnjeve pojačala u zasićenje. Pritom je visokopropusni filter moguće smjestiti na dva načina u odnosu na instrumentacijsko pojačalo.

Prvi način je smještaj visokopropusnog filtra prije instrumentacijskog pojačala. Takvo rješenje podrazumijeva po jedan visokopropusni CR član za svaki ulaz pojačala. Prednost ovakvog pristupa je uklanjanje polarizacijskog napona prije instrumentacijskog pojačala, tako da ono može imati veliko pojačanje bez da ode u zasićenje. Time instrumentacijsko pojačalo ostvaruje veći faktor potiskivanja istofaznog signala. Osim toga pojačalo u narednom stupnju može imati veći šum (jeftinije pojačalo) jer je signal puno veće amplitude nego na ulaznom stupnju. Ovakav pristup ima i dva ozbiljna nedostatka. Prvi nedostatak je značajno smanjenje ulazne impedancije pojačala čime se narušava njegov faktor potiskivanja istofaznog signala. Otpornici koji se mogu pronaći u uobičajenim izvedbama su iznosa do nekoliko  $M\Omega$  što je nekoliko redova veličine manje od ulazne impedancije pojačala INA333 ( $100G\Omega||3pF$ ). Drugi nedostatak je nepostojanje relativno velikih iznosa otpornika i kondenzatora, kakvi su potrebni za nisku donju graničnu frekvenciju, u dovoljno uskim tolerancijama. Tolerancije otpornika i kondenzatora visokopropusnog CR člana unose dodatno razgođenje putova signala čime se pojačava prije opisani efekt naponskog djelitelja tj. više istofazne smetnje se na ulazu pojačala pojavljuje kao diferencijalna smetnja.

Drugi način smještaja je pozicioniranje CR člana nakon instrumentacijskog pojačala, prije drugog stupnja pojačala. Glavna prednost ovakvog pristupa je ta što CR član ne utječe na faktor potiskivanja pojačala. Sada je moguće uzeti kondenzatore i otpornike širokih tolerancija (npr. 5% za otpornik i 20% za kondenzator), što pojednostavljuje izbor. Nedostatak ovog pristupa je postojanje velikog (do 200mV) diferencijalnog napona polarizacije elektroda na ulazu instrumentacijskog pojačala zbog čega ulazni stupanj ne može imati veliko pojačanje. Uz pretpostavku baterijskog napajanja može se prepostaviti da će maksimalni hod izlaznog napona instrumentacijskog pojačala biti oko  $+/-1V$  što znači da je maksimalno pojačanje 5. Zbog toga će signal na drugi stupanj stići s manjom amplitudom nego u prvom načinu zbog čega će i pojačala drugog stupnja morati biti kvalitetnija (s manje šuma).

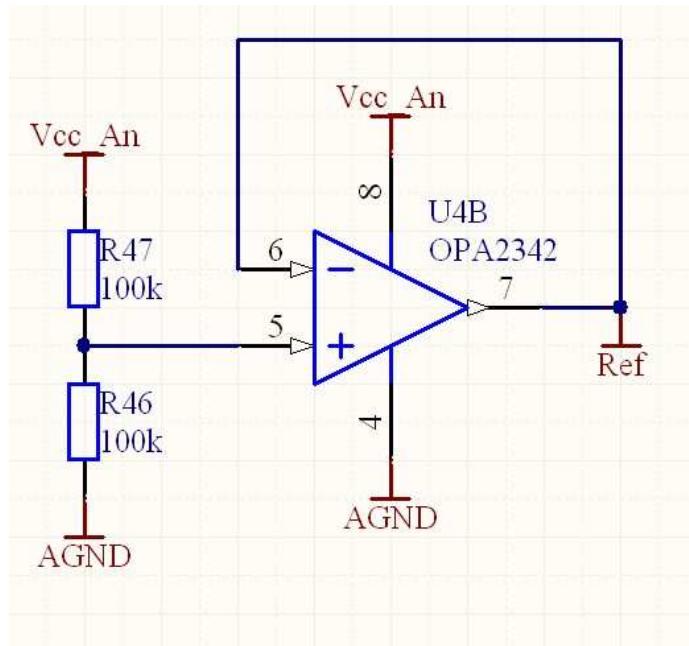
Zbog teškoće nabave i teškoće uparivanja otpornika i kondenzatora izabran je drugi način tj. pozicioniranje visokopropusnog CR člana nakon instrumentacijskog pojačala. CR član realiziran je otpornikom iznosa otpora  $470\text{k}\Omega$  i tolerancije 1% te kondenzatorom iznosa kapaciteta  $680\text{nF}$  i tolerancije 10% čime je donja granična frekvencija postavljena na  $2\text{Hz}$ .

Shema kompletног ulaznog stupnja pojačala dana je na slici 6.



**Slika 6. – Shema ulaznog stupnja pojačala**

Točka koja se nalazi između dva otpornika  $R_g$  spaja se na naponsko slijedilo koje dalje prenosi signal na sklop za pogon desne noge. Točka iza visokopropusnog CR člana vodi na sljedeći stupanj pojačala. Referentna točka (Ref) predstavlja virtualnu masu pojačala pošto je napajanje unipolarno. Virtualna masa dobiva se kombinacijom otporničkog dijelila i naponskog slijedila, kako prikazuje slika 7.



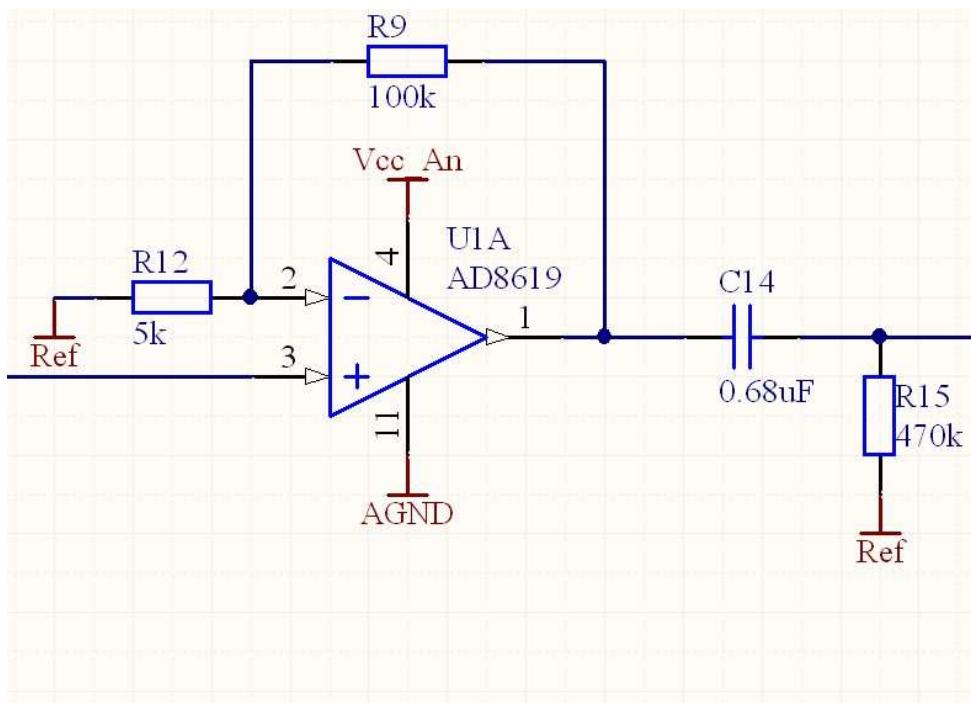
**Slika 7. – Virtualna masa pojačala**

### Drugi stupanj pojačala

Drugi stupanj pojačala izведен je operacijskim pojačalom u konfiguraciji neinvertirajućeg pojačala. Pojačanje drugog stupnja je 20 puta. U drugom stupnju pojačala koristi se operacijsko pojačalo AD8619, proizvođača *Analog Devices*. U jednom čipu su pakirana po četiri operacijska pojačala čime se štedi na prostoru. Jedno pojačalo koristi se za drugi stupanj pojačanja, jedno se koristi za naponsko slijedilo u sklopu za pogon desne noge i preostala dva koriste se u filtru koji slijedi nakon drugog stupnja pojačanja. Operacijsko pojačalo AS8619 odabrano je zbog sljedećih karakteristika:

- mala struja pomaka: maksimalno 110pA na rasponu -45°C do +85°C, tipično 1pA na 25°C
- mala potrošnja: 50 $\mu$ A po pojačalu
- napajanje od 1.8V do 5V
- male dimenzije: 4 pojačala u TSSOP-14 kućištu
- *Rail-to-Rail* ulaz i izlaz

Za struju pomaka je bitno da je malog iznosa jer se zatvara kroz otpornik visokopropusnog CR člana koji prethodi drugom stupnju pojačanja i na taj način stvara napon pomaka na ulazu. U najgorem slučaju će, uz korištenje pojačala AD819, taj napon pomaka iznositi  $110pA * 470k\Omega = 51.7\mu V$ . Taj napon se može zanemariti naspram napona pomaka pojačala koji iznosi  $2.2mV$  u najgorem slučaju. Taj napon pomaka bit će pojačan 20 puta te će na izlazu drugog stupnja iznositi u najgorem slučaju oko  $45mV$ . Pošto će treći stupanj (filtrar s pojačanjem) morati pojačati signal još desetak puta da bi ukupno pojačanje bilo 1000, napon pomaka pojačala drugog stupnja bi na izlazu filtra mogao doseći  $450mV$  što je neprihvatljivo. Zbog toga je na izlaz drugog stupnja dodan visokopropusni CR član kao i na izlazu instrumentacijskog pojačala. Shema drugog stupnja pojačala dana je na slici 8. Izlaz visokopropusnog člana sa slike 8 spojen je na ulaz trećeg stupnja pojačala.



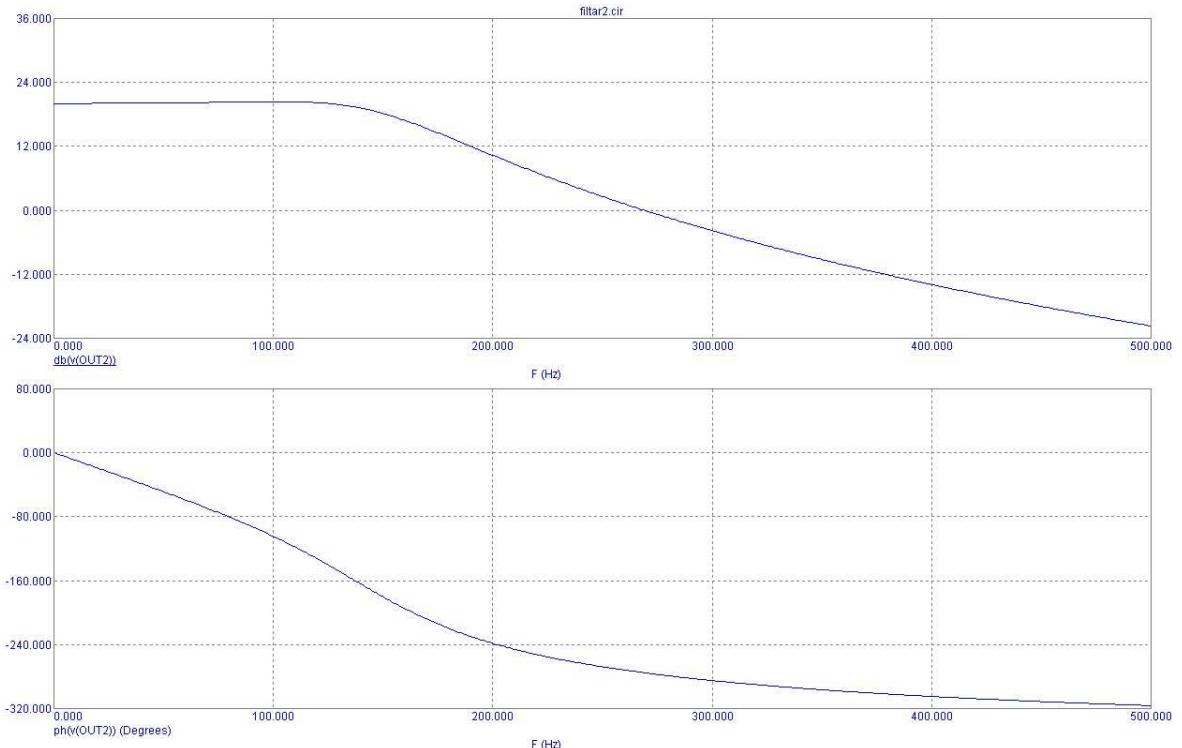
Slika 8. – Shema drugog stupnja pojačala

### Realizacija gornje granične frekvencije i treći stupanj pojačanja

Treći stupanj pojačala obavlja dvije funkcije: definira gornju graničnu frekvenciju pojačala (150Hz) i pojačava signal 10 puta. Filter je dizajniran koristeći Butterworthovu

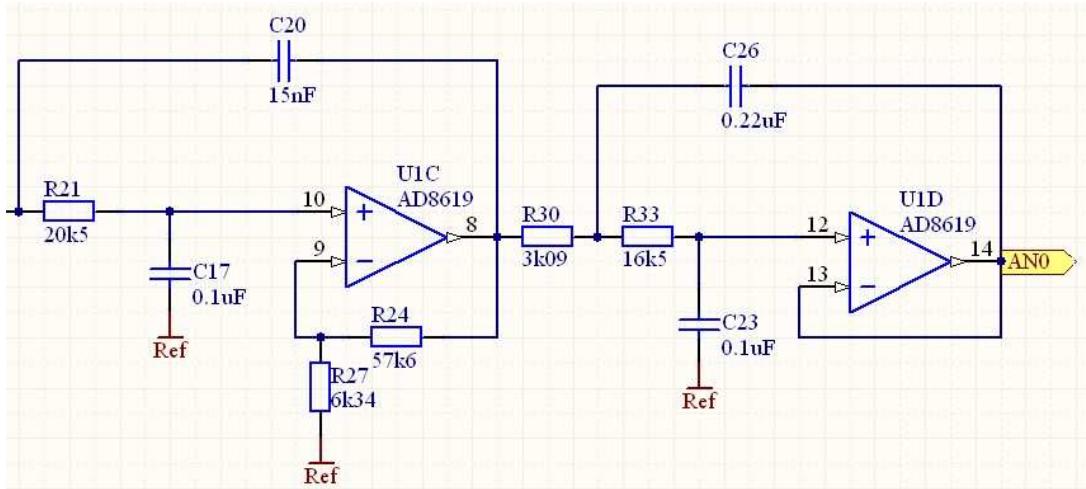
aproksimaciju idealne prijenosne funkcije uz uvjet da filter bude 4-tog reda (da se koriste dva operacijska pojačala). Filter je projektiran pomoću programskog alata *FilterLab2.0*.

Amplitudna i fazna frekvencijska karakteristika prikazane su na slici 9.



**Slika 9. – Amplitudno frekvencijska i fazno frekvencijska karakteristika niskopropusnog filtra**

Shema filtra prikazana je na slici 10. Izlaz filtra spojen je na AD pretvornik mikrokontrolera.



Slika 10. – Shema niskopropusnog filtra

## 4 Memorija

### 4.1 Izbor vrste memorije

Već je rečeno da za minutu EKG signala uređaj treba imati oko 90kB memorije. Pošto na tržištu postoji više različitih tipova memorije, potrebno je odabrati onu koja najbolje zadovoljava potrebe uređaja. Smjernice u odabiru su sljedeće:

- minimalno 90kB tj. minimalno 720kb kapaciteta (zapravo 1 Mb jer je to prvi veći kapacitet od 720kb, a u kojem se memorije proizvode), što više to bolje
- poželjna je što manja potrošnja
- vrlo je poželjno da memorija može zadržati podatke bez prisustva napajanja
- brzina upisa nije važna pošto se upisuje svega 1.5kB u sekundi
- nije previše važno piše li se i čita iz memorije bajt po bajt ili blok po blok jer je pristup podacima uvijek sekvencijalan (uzorci signalata se čitaju i pišu redom)
- poželjna je što manja cijena

#### DRAM memorija

Tipično se koristi kao radna memorija u sustavu pa svojim karakteristikama vjerojatno najmanje od svih odgovara prije opisanoj primjeni. Odlikuje ju velika brzina pisanja/čitanja (koja za ovaj uređaj nije bitna), ali i velika potrošnja do nekoliko stotina mA. Osim toga, sadržaj memorije je potrebno periodički osvježavati da bi se podaci održali. Prema tome, jasno je da bez napajanja ne može zadržati podatke. Svojim velikim kapacitetom (256Mb na više po čipu) bi itekako odgovarala, ali iz navedenih nedostataka jasno je da korištenje DRAM-a u ovakve svrhe nema smisla.

#### SRAM memorija

Od DRAM memorije se razlikuje u tome što ju nije potrebno osvježavati. Osim toga brzina pristupa je manja, ali još uvijek višestruko veća od potrebne. Potrošnja je prihvatljivih nekoliko mA pri pristupu i svega nekoliko  $\mu$ A u periodima neaktivnosti. Kapacitet od desetak Mb je standardno dobavljen što bi zadovoljilo potrebe. Međutim glavni nedostatak

još uvijek ostaje nemogućnost zadržavanja podataka bez prisustva napajanja, jer je SRAM, kao i DRAM, ipak zamišljen kao radna memorija, a ne za dugotrajniju pohranu podataka.

### **EEPROM memorija**

EEPROM (engl. *Electrically Erasable Programmable Read-Only Memory*) memoriju odlikuje vrlo mala potrošnja od nekoliko stotina  $\mu\text{A}$  pri pristupu i nekoliko stotina nA u neaktivnom periodu čime dobro odgovara baterijski napajanom uređaju. Brzina pristupa (naročito upisa) je mnogo manja nego kod SRAM/DRAM memorija, ali zadovoljava potrebe uređaja (premda relativno tjesno). Lako dobavljive izvedbe su kapaciteta do 1Mb što je dovoljno za nešto više od minute signala. Za razliku od SRAM/DRAM memorija broj upisa u EEPROM je ograničen na 1 do 10 milijuna upisa, ovisno o proizvođaču. Ta činjenica ne predstavlja velik problem pošto uređaj ne koristi memoriju intenzivno nego samo za pohranu signala. Prema tome, uređaj bi mogao nekoliko milijuna puta obaviti svoju funkciju što je više nego dovoljno. Velika prednost u odnosu na SRAM/DRAM memorije je zadržavanje podataka bez prisustva napajanja.

### **FRAM memorija**

FRAM (engl. *Ferroelectric Random Access Memory*) je kapacitetom i potrošnjom slična EEPROM memoriji. Brzina pristupa je puno veća u odnosu na EEPROM memoriju. Broj upisa je ograničen, ali puno veći nego kod EEPROM memorije – do  $10^{12}$  puta što je praktički neograničeno u većini primjena. Bolje performanse dolaze uz veću cijenu koja nekoliko puta viša u odnosu na EEPROM. Za ovdje promatrani uređaj korištenje FRAM-a u odnosu na EEPROM ne bi donijelo nikakve značajne prednosti, samo bi povećalo cijenu.

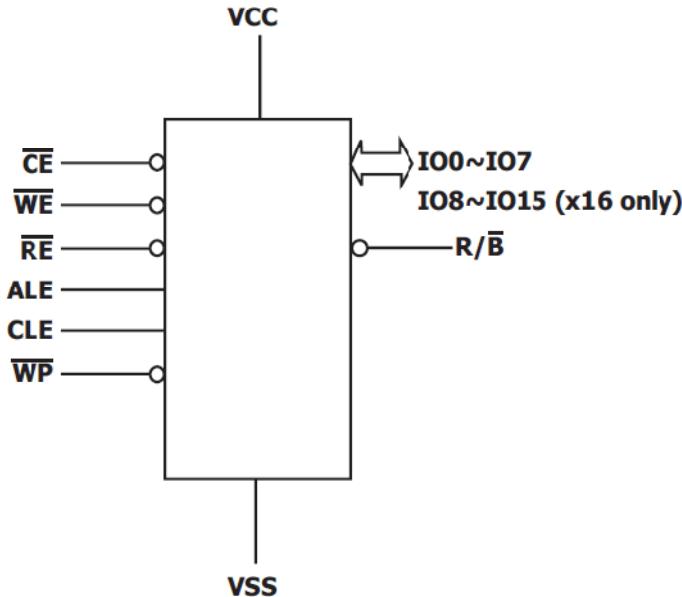
### **NAND FLASH memorija**

FLASH memorija kapacitetom nadaleko premašuje dosad opisane tipove memorije. Lako su dobavljni čipovi sa kapacitetom od 512Mb za cijenu za koju bi se moglo dobiti tek 256kb FRAM-a ili nekoliko Mb EEPROM-a. Brzina pristupa je više nego dovoljna za potrebe uređaja. 1,5kB podataka koliko treba spremiti svake sekunde može se upisati u FLASH memoriju za nekoliko ms. Zbog toga memorija većinu vremena može provesti u *standby* modu pritom trošeci 50-ak  $\mu\text{A}$  struje. Potrošnja tijekom pisanja u memoriju (nekoliko desetaka mA) nije toliko bitna jer bi razvijani sklop u ovaku memoriju pisao

manje od 1% vremena. Nedostatak FLASH memorije je mogućnost pristupa samo blokovima podataka, a ne bajt po bajt. No ta činjenica ne predstavlja problem opisanom uređaju pošto se podaci zapisuju sekvencijalno pa ih nije teško podijeliti u blokove bilo koje veličine. Još jedan od nedostataka FLASH memorije je mogućnost kvarenja blokova tijekom rada sklopa zbog čega je potrebno u ugrađenoj programskoj podršci predvidjeti mehanizam označavanja i preskakanja takvih blokova. Sve u svemu, FLASH memorija, uz određene komplikacije u ugrađenoj programskoj podršci, nudi drugim memorijama nedostižnu količinu prostora koja je dovoljna za 6 i više sati signala. Uz toliko memorije uređaju se može drastično povećati funkcionalnost jer se tada može koristiti za dugotrajno praćenje rada srca ili može u memoriji držati veći broj zapisa (a ne samo jedan) prije prebacivanja bežičnom vezom na računalo. Prema tome FLASH memorija predstavlja dobar izbor za razvijani elektrokardiograf.

## 4.2 Hynix 512Mb NAND FLASH

Odabrana memorija je H27US08121B-TPCB, kapaciteta 512Mb, proizvođača Hynix Semiconductor. Memorija je podijeljena u 4096 blokova. Svaki blok podijeljen je u 32 stranice (engl. *Page*). Svaka stranica sadrži 512 bajta osnovnog prostora i još 16 bajta dodatnog prostora. Dodatni prostor se može koristiti za iste namjene kao i glavni, ali ga se ipak najčešće koristi za potrebe kodova za ispravljanje grešaka ili ujednačavanje broja pisanja po blokovima (engl. *Wear Leveling*). Memoriji se pristupa paralelno, sabirnicom širokom 8 bita tj. jedan bajt. Priklučci memorije prikazani su na slici 11, a njihov je opis dan u tablici 1.



Slika 11. – Priključci Hynix FLASH memorije

Tablica 1. – Ulszno/izlazni priključci Hynix FLASH memorije

Priključak	Opis
IO0~IO7	Ulaszno/izlazni priključci za prijenos podataka, adrese i instrukcija memoriji
CE	U niskom stanju selektira čip. Dok je CE u visokom stanju, čip je u <i>standby</i> načinu rada i troši 50-ak $\mu$ A
RE	Kontrolira ciklus čitanja podataka iz memorije
WE	Kontrolira ciklus upisa podataka/adrese/instrukcija u memoriju
ALE	Određuju hoće li se vrijednost na podatkovnim priključcima interpretirati kao adresa, podatak ili instrukcija
CLE	
WP	Kontrolira sklopošku zaštitu protiv neželjenog pisanja i brisanja

R/B	Izlazni priključak koji signalizira stanje čipa (slobodan/zauzet)
VCC	Napajanje čipa
VSS	Masa čipa

Najjednostavniji način spajanja ovakve memorije na mikrokontroler je korištenje po jednog ulazno/izlaznog priključka opće namjene za svaki priključak memorije. Prema tome, bit će potrebno 15 slobodnih priključaka na mikrokontroleru.

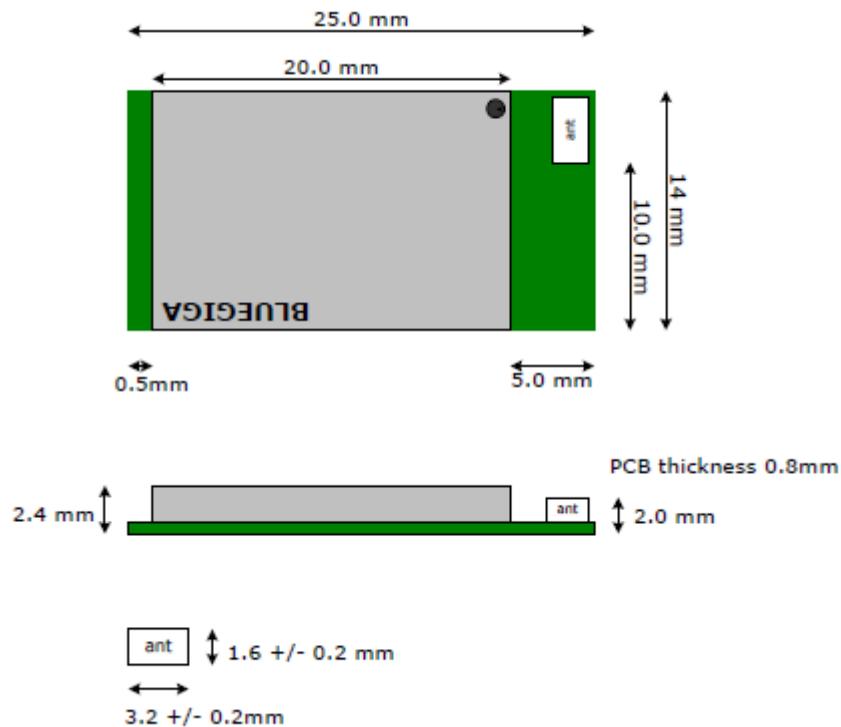
Već je rečeno da se određeni broj blokova (do 2% od ukupnog broja) može pokvariti tijekom korištenja memorije. No i potpuno novi čip može sadržavati neispravne blokove. Zbog toga svi NAND FLASH čipovi prolaze tvornički test ispravnosti. Blokovi koji su ispravni se brišu tako da se na svakom bajtu nalazi vrijednost 0xFF. Blokovi koji su neispravni se najčešće označuju tako da na 517. bajtu nulte stranice (ili prve, ako nulta nije isprana) imaju neku vrijednost različitu od 0xFF. Na taj način se korisniku omogućuje da na novom čipu ispita ispravnost svih blokova, zabilježi one koji su neispravni i ne pristupa im više.

## 5 Bluetooth veza

Cilj ovog rada nije opisivanje Bluetootha kao protokola komunikacije. Zbog toga će ovo poglavlje biti usmjereni ka opisivanju konkretnе implementacije Bluetooth bežične veze između razvijanog uređaja i računala ili nekog drugog Bluetooth kompatibilnog uređaja.

Bluetooth protokol je višenamjenski i podržava više različitih načina korištenja. Zbog toga su definirani različiti Bluetooth profili koji opisuju na koji način dva uređaja trebaju komunicirati u određenom scenariju primjene bežične veze. Profili opisuju sučelje između dva uređaja na najvišoj razini protokola. Da bi dva uređaja mogla komunicirati moraju podržavati isti profil. Razvijani uređaj pronosit će jedan kontinuirani blok podataka bez posebnog formata koji sadržava prikupljene uzorke EKG signala. Ovakvoj primjeni najviše odgovara Bluetooth profil naziva *Serial port profile (SPP)*. Uređaji koji podržavaju ovaj profil „vidjet“ će Bluetooth vezu kao virtualni serijski port (engl. *Virtual COM Port - VCP*). Na taj način Bluetooth veza se emulira kabelska RS232 veza. Ovakav način komunikacije jednostavan je za implementaciju i sa strane mikrokontrolera i sa strane PC-a (ili drugog uređaja koji podržava Bluetooth i SPP).

Za implementaciju Bluetooth SPP veze odabran je modul WT-12 proizvođača *Bluegiga Technologies*. Radi se o modulu koji ovisno o izabranoj verziji ugrađene programske podrške (engl. *firmware*) implementira dio ili cijeli Bluetooth protokol. Odabrana je *iWRAP* verzija ugrađene programske podrške koje implementira cijeli Bluetooth protokol i podržava SPP. Implementacija protokola skrivena je od korisnika tako da se modulu pristupa pomoću jednog od ponuđenih sučelja (USB, UART, PCM). Pošto je odlučeno da će se koristiti SPP, logično je modulu pristupati putem UART-a. Osim mikrokontrolera koji realizira programski dio protokola, modul sadrži i svo ostalo sklopljje potrebno za ostvarivanje Bluetooth veze (primopredajnik i antena). Zbog toga nema potrebe za eksternim komponentama što u kombinaciji s malim dimenzijama (slika 12) čine ovaj modul pogodnim za korištenje u prijenosnim uređajima.



Slika 12. – Dimenziije modula WT-12

*iWRAP firmware* radi u jednom od dva načina rada: podatkovnom ili komandnom. Inicijalno se modul nalazi u komandnom načinu rada u kojem je moguće upravljati njegovim radom šaljući mu ASCII komande putem serijskog sučelja. Raspoloživih komandi ima mnogo, ali način korištenja Bluetooth veze u razvijanom uređaju je jednostavan pa će ih trebati samo nekoliko. Komande koje će se koristiti su sljedeće:

- **SET BT AUTH \* {pin}** – Definira sigurnosnu zaporku (pin) koja mora biti poznata drugom uređaju prilikom uparivanja
- **SET CONTROL BAUD {baud\_rate},8{parity}{stop\_bits}** – Mijenja postavke lokalnog UART sučelja. Pritom *baud\_rate* označava brzinu prijenosa podataka u bps, *parity* određuje paritet (n – nema pariteta, o – odd, e - even), a *stop\_bits* određuje broj stop bitova (1 – jedan stop bit, 2 – dva stop bita).

U podatkovnom načinu rada svi podaci poslani na UART se transparentno prosljeđuju na Bluetooth vezu. Metode prelaska iz jednog u drugi način opisuje tablica 2.

**Tablica 2. – Metode prelaska između komandnog i podatkovnog načina rada modula WT-12**

Trenutni način rada	Način u koji se prelazi	Metoda prelaska
<b>Komandni način (bez aktivnih Bluetooth veza)</b>  Modul se nalazi u ovom načinu kad se uključi.  U ovom načinu se modulu mogu slati ASCII komande	Podatkovni način	Modul uspješno inicira Bluetooth vezu s drugim uređajem pomoću CALL komande.  Drugi uređaj uspješno inicira Bluetooth vezu s modulom.
<b>Podatkovni način</b>  U ovom načinu rada podaci se transparentno šalju s UART-a na Bluetooth vezu	Komandni način	Slanje posebne sekvene (sekunda bez slanja komandi pa tri ASCII znaka „+“ zaredom i na kraju opet sekunda bez slanja komandi).  Prekid veze (veza prekinuta ili na inicijativu drugog uređaja ili zbog gubitka signala).
<b>Komandni način (postoje aktivne Bluetooth veze)</b>  U ovom načinu se modulu mogu slati ASCII komande	Podatkovni način	Ili slanjem već opisane posebne sekvene ili pomoću SELECT komande.

Kao što je već rečeno, modul se inicijalno nalazi u komandnom načinu rada. Redoslijed događaja u uspostavi veze s računalom će biti sljedeći:

1. modul WT-12 se uključuje kada dođe vrijeme da se uzorci signala prenesu na drugi uređaj

2. modul je u komandnom načinu rada pa mu se šalju komande kojima ga se konfigurira na željeni način
3. računalo (ili neki drugi Bluetooth uređaj koji podržava SPP) pronalazi uređaj, uparuje se s njime i registrira ga kao VCP (*Virtual COM Port*)
4. Bluetooth veza je uspostavljena i svi podaci se transparentno šalju između VCP-a na računalu i UART-a na modulu. Sada računalo može poslati komandu koja će preko UART-a modula stići na mikrokontroler i pokrenuti slanje podataka
5. mikrokontroler čita podatke iz FLASH memorije i šalje ih preko UART-a modula na VCP računala.
6. kada su poslani svi podaci mikrokontroler prebacuje modul u komandni način rada i prekida Bluetooth vezu.

Potrošnja modula varira ovisno o tome što trenutno radi. Za razvijani uređaj bitni su iznosi potrošnje tijekom prijenosa podataka (za brzinu od 115200 kbps srednja vrijednost struje je 30-ak mA, a vršna 70mA) i minimalna potrošnja u periodima neaktivnosti od  $400\mu\text{A}$ . Potrošnja u periodima neaktivnosti je dosta visoka za baterijski napajan uređaj, osam puta veća od potrošnje jednog operacijskog pojačala i pet puta veća od potrošnje jednog instrumentacijskog pojačala. Bluetooth modul zapravo nema posla tijekom snimanja signala, nego tek kad snimanje završi. Zbog toga bi bilo praktično osmisiliti način da se modul uopće ni ne napaja sve dok snimanje ne završi i ne dođe vrijeme za prijenos podataka. O tome će biti govora u poglavljiju o baterijskom napajanju.

## 6 Mikrokontroler

Najvažniji zahtjev postavljen na mikrokontroler je niska potrošnja energije. Procesna moć i nije toliko važna stavka jer mikrokontroler ne radi nikakve računski intenzivne operacije.

Ukupna lista zahtjeva postavljenih na mikrokontroler je sljedeća:

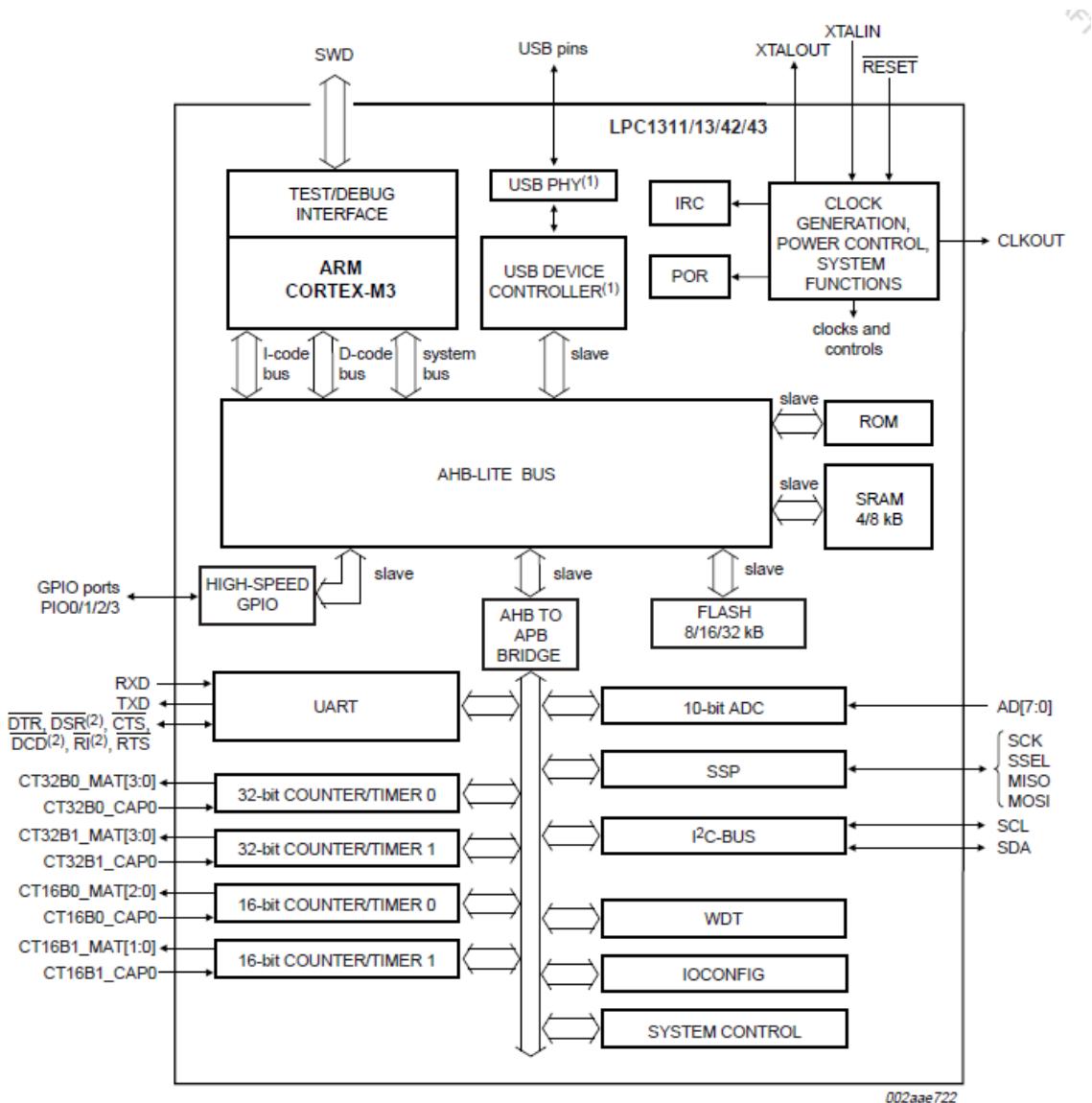
- niska potrošnja
- barem 15 ulazno/izlaznih priključaka opće namjene za memoriju + još 5 ili više za druge namjene (tipke, LE diode i sl.)
- UART sučelje za komunikaciju s Bluetooth modulom
- AD pretvornik razlučivosti 8 bita ili više, s tri ili više kanala
- što manje dimenzije
- mogućnost rada s malim naponom napajanja (zbog baterijskog napajanja, poželjno 3V ili manje)

Prema navedenim zahtjevima i cijeni/dostupnosti razvojnog sustava odabrana je porodica LPC13xx porodica mikrokontrolera proizvođača *NXP Semiconductors*. Porodica je temeljena na 32-bitnoj ARM Cortex-M3 arhitekturi koja je pogodna za korištenje u sklopovima niske potrošnje. Iz porodice je izabran mikrokontroler LPC1313. Njegove karakteristike bitne za razvijani uređaj su:

- niska potrošnja (bit će pobliže opisana kasnije)
- do 40 ulazno/izlaznih priključaka opće namjene (broj ovisi o korištenim perifernim sklopovima na mikrokontroleru)
- UART sučelje
- 8kB radne memorije
- napon napajanja od 2.0V do 3.6V
- 48-pinsko LQFP kućište malih dimenzija

## 6.1 Arhitektura porodice mikrokontrolera LPC13xx

Porodica je bazirana na ARM Cortex-M3 arhitekturi optimiranoj za nisku potrošnju. Slika 13 prikazuje blok shemu porodice mikrokontrolera LPC13xx. Blok označen s *ARM CORTEX-M3* predstavlja jezgru mikrokontrolera. Jezgra komunicira s ostatkom mikrokontrolera preko AHB (engl *Advanced High-Speed Bus*) sabirnice. Na tu sabirnicu spojene su i memorije (programska memorija, radna memorija i *boot ROM* memorija), USB kontroler (nije na svim mikrokontrolerima iz porodice) i brzi periferni sklopovi. Kod porodice LPC13xx jedini brzi periferni sklop je kontroler ulazno/izlaznih priključaka opće namjene. Na AHB sabirnicu spojen je i „most“ (engl. *Bridge*) prema APB (engl. *Advanced Peripheral Bus*) sabirnici. Na APB sabirnici nalaze se sporiji sklopovi od kojih će se koristiti AD pretvornik (*ADC*), serijsko sučelje (*UART*) te blokovi za konfiguraciju ulazno/izlaznih priključaka i cjelokupnog mikrokontrolera (*IOCONFIG* i *SYSTEM CONTROL*).



Slika 13. – Blok shema porodice mikrokontrolera LPC13xx

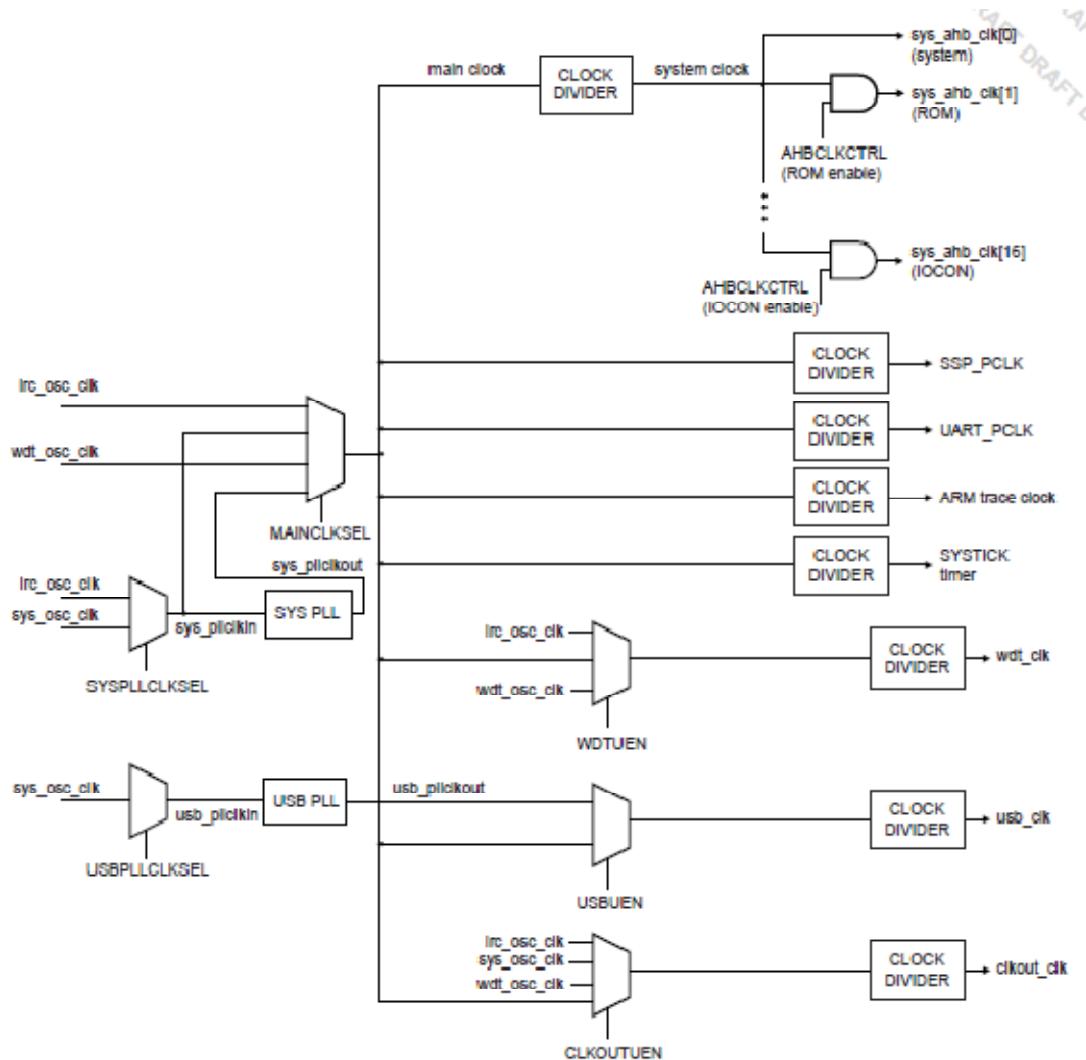
## 6.2 Upravljanje potrošnjom mikrokontrolera LPC1313

LPC1313 podržava širok raspon mehanizama za kontrolu potrošnje. U normalnom načinu rada moguće je mijenjati frekvenciju sistemskog (onog koji vodi jezgru i memorije) signala vremenskog vođenja (engl. *clock*) i na taj način smanjiti potrošnju na račun smanjivanja procesne moći. Osim toga moguće je i spriječiti dovođenje *clock* signala pojedinim perifernim sklopovima na čipu koji se ne koriste i na taj način ih zaustaviti uz smanjivanje potrošnje. Neki periferni skloovi imaju i svoje djelitelje frekvencije *clock* signala kojima

je moguće usporiti njihov rad i smanjiti im potrošnju. Slika 14 prikazuje blok shemu jedinice za generiranje *clock* signala (engl. *Clock Generation Unit - CGU*) na kojoj su vidljivi opisani mehanizmi upravljanja *clock* signalima. Osim toga vidljivi su i mogući izvori glavnog *clock* signala (iz kojeg se dijeljenjem dobiva sistemski *clock* signal i pojedini periferni *clock* signali), a to su:

- sistemski oscilator (koji koristi vanjski kristal kvarca)
- integrirani RC oscilator
- integrirani *WATCHDOG* oscilator
- sistemski oscilator pomnožen ili podijeljen pomoću PLL-a (engl. *Phase Locked Loop*)
- integrirani RC oscilator pomnožen ili podijeljen pomoću PLL-a

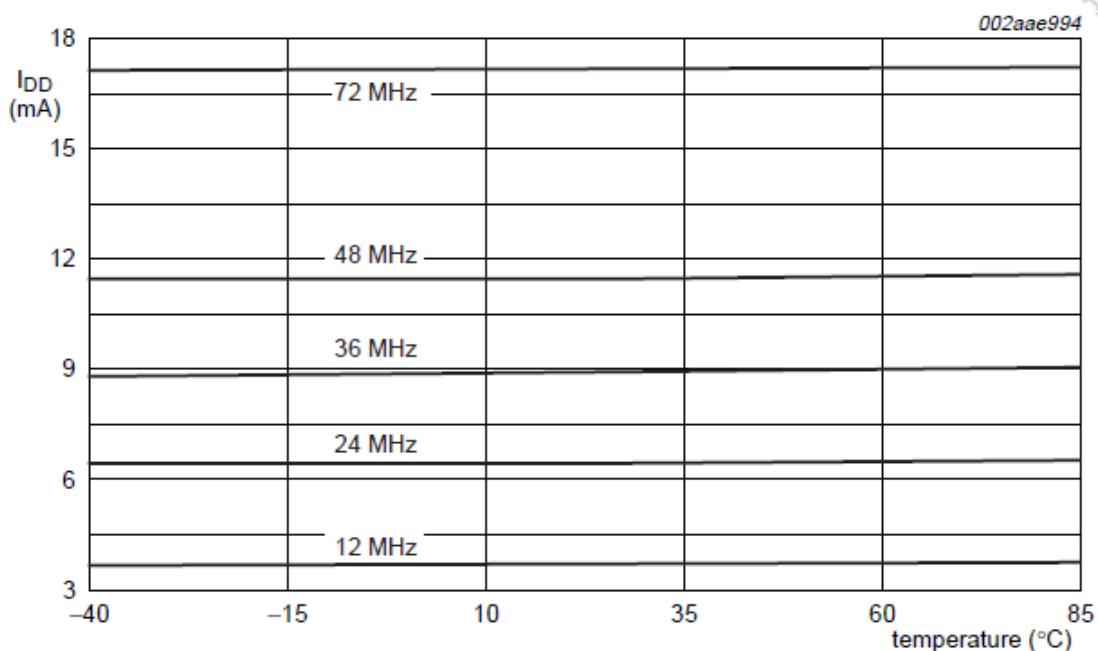
Od navedenih koristit će se integrirani RC oscilator zato što ne zahtjeva eksterne komponente. RC oscilator je manje precizan (odstupanje apsolutnog iznosa frekvencije od maksimalno 1%) od kvarcnog oscilatora (odstupanje apsolutnog iznosa frekvencije od nekoliko desetaka ppm), ali to ne predstavlja problem za ovu aplikaciju.



**Slika 14. – Blok shema jedinice za generiranje *clock* signala porodice mikrokontrolera LPC13xx**

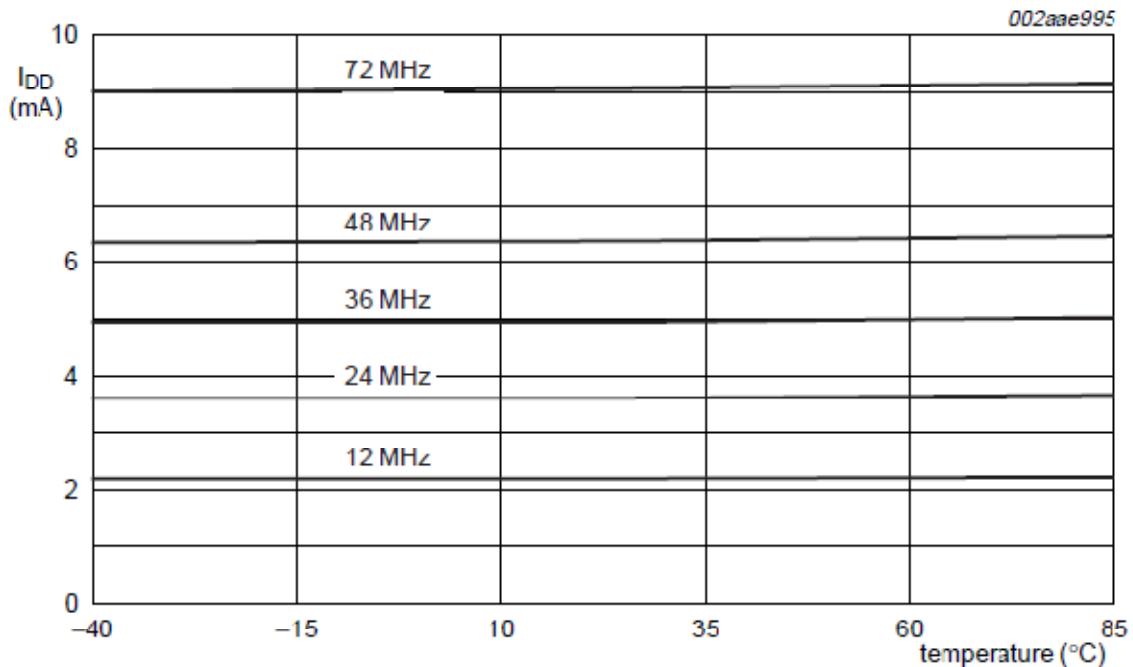
Osim upravljanja brzinom rada pojedinih dijelova mikrokontrolera, potrošnju je moguće smanjiti i prelaskom u neki od načina rada smanjene potrošnje. Načina rada, brojeći i normalni ima četiri.

Krivulja ovisnosti tipične potrošnje o temperaturi i frekvenciji u normalnom načinu rada dana je na slici 15. Krivulja prikazuje potrošnju kada se ne koriste pritezni otpornici ulazno/izlaznih priključaka, kada su svi periferni sklopovi ugašeni i uz napon napajanja iznosa 3.3V. Izvor sistemskog *clock* signala je sistemski oscilator s vanjskim kristalom kvarca propušten kroz PLL.



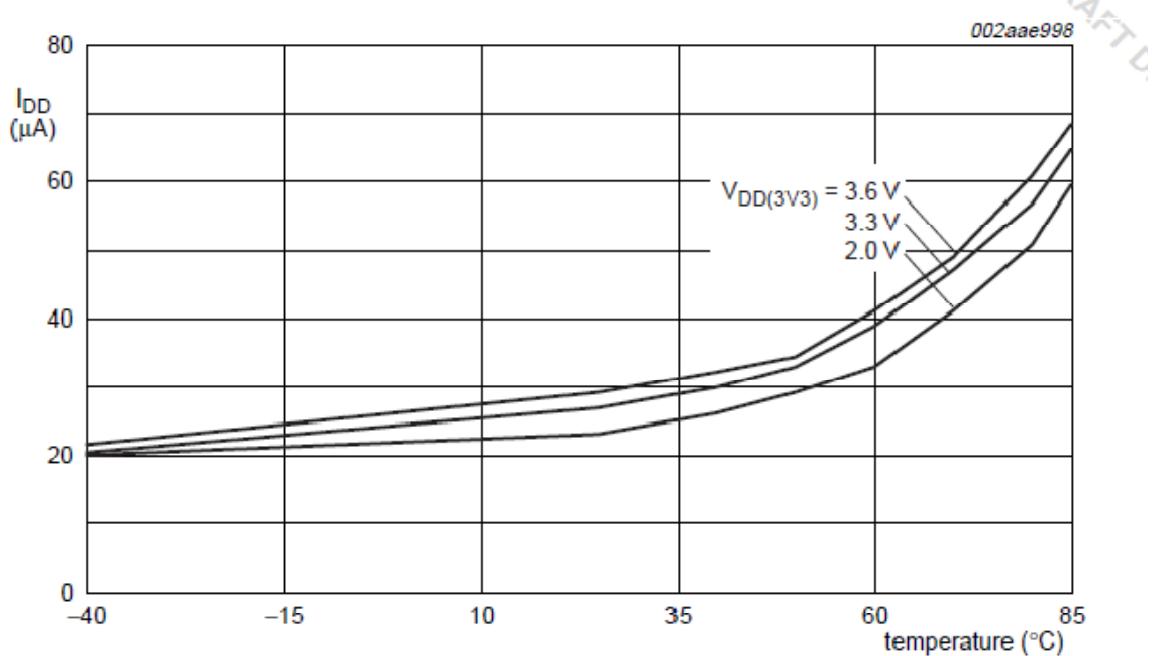
**Slika 15. – Potrošnja mikrokontrolera LPC1313 u normalnom načinu rada u ovisnosti o temperaturi i frekvenciji sistemskog *clock* signala**

Osim normalnog načina rada postoje i tri načina rada sa smanjenom potrošnjom. Prvi od njih naziva se *sleep* način rada. U *sleep* načinu sistemski *clock* signal se ne dovodi do jezgre zbog čega mikrokontroler prestaje izvoditi instrukcije. Na taj način se uklanja dinamička potrošnja procesora, memorija i sistemskih sabirnica. U *sleep* načinu rada registri procesora, registri perifernih sklopova i stanje radne memorije su očuvani. Logičke razine izlaznih priključaka ostaju nepromijenjene. Pri ulasku u *sleep* način rada moguće je odabrati periferne sklopove koji će nastaviti raditi, dok se svi ostali gase. Mikrokontroler ostaje u *Sleep* načinu rada sve dok se ne dogodi *reset* ili prekid od strane nekog perifernog sklopa. Krivulja ovisnosti tipične potrošnje o temperaturi i frekvenciji u *sleep* načinu rada dana je na slici 16. Krivulja prikazuje potrošnju kada se ne koriste pritezni otpornici ulazno/izlaznih priključaka, kada su svi periferni sklopovi ugašeni i uz napon napajanja iznosa 3.3V. Sistemski oscilator i PLL su aktivni. Vidljivo je da je potrošnja skoro dvostruko veća nego u normalnom načinu rada.



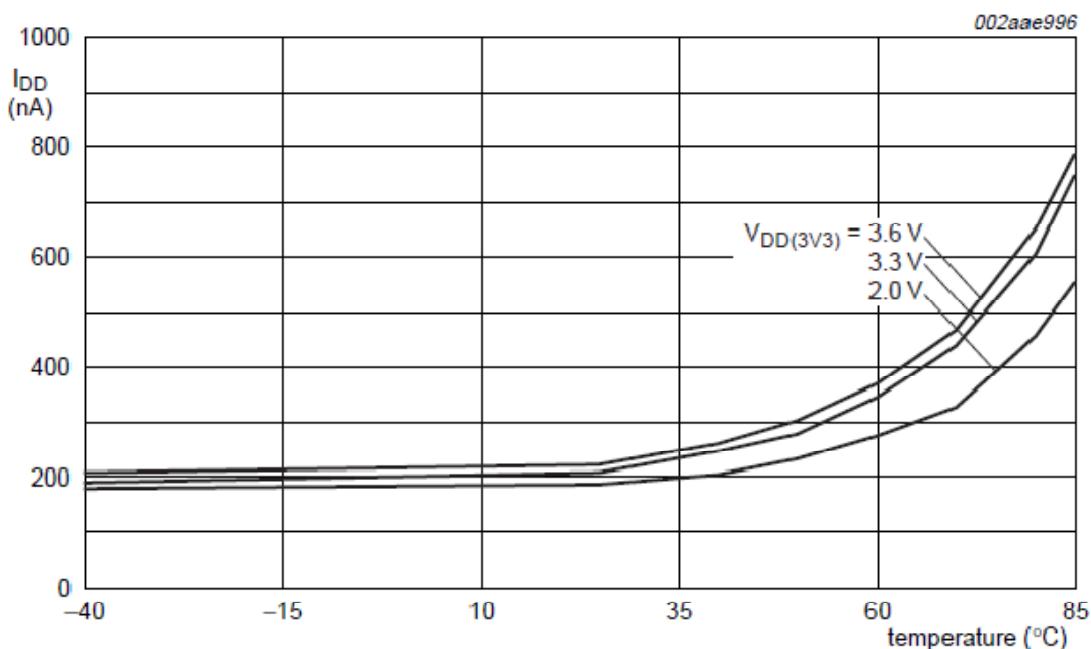
Slika 16. – Potrošnja mikrokontrolera LPC1313 u *sleep* načinu rada u ovisnosti o temperaturi i frekvenciji sistemskog *clock* signala

Drugi način rada smanjene potrošnje naziva se *deep-sleep* način. Za razliku od *sleep* načina rada u ovom načinu moguće je ugasiti i oscilatore i PLL sklopove čime se potrošnja uvelike smanjuje. Razlika je i u tome što se iz *deep-sleep* načina rada mikrokontroler može osim na prekide perifernih sklopova probuditi i na pojavu brida na ulaznim priključcima. Kao i u *sleep* načinu rada registri procesora, registri perifernih sklopova i stanje radne memorije su očuvani. Logičke razine izlaznih priključaka također ostaju nepromijenjene. Krivulja na slici 17. prikazuje ovisnost tipične potrošnje u *deep-sleep* načinu rada o temperaturi i naponu napajanja. Krivulja prikazuje potrošnju kada su ugašeni svi oscilatori i PLL sklopovi te svi periferni sklopovi. Vidljivo je da je potrošnja mnogo manja nego u normalnom i *sleep* načinu.



Slika 17. – Potrošnja mikrokontrolera LPC1313 u *deep-sleep* načinu rada u ovisnosti o temperaturi i naponu napajanja

Treći način rada smanjene potrošnje naziva se *deep power-down* način. U tom načinu prekida se napajanje svim izvorima *clock* signala i ostatku čipa osim *WAKEUP* priključku. Procesor se može probuditi iz *deep power-down* načina samo na impuls na *WAKEUP* priključku. U ovom načinu rada gube se stanja svih registara i radne memorije. Samo četiri registra posebne namjene čuvaju svoje stanje. Potrošnja mikrokontrolera je drastično smanjena kao što je vidljivo na slici 18. Krivulja na slici 18 prikazuje ovisnost tipične potrošnje u *deep power-down* načinu rada o temperaturi i naponu napajanja.



Slika 18. – Potrošnja mikrokontrolera LPC1313 u *deep power-down* načinu rada u ovisnosti o temperaturi i naponu napajanja

## 7 Baterijsko napajanje i potrošnja

Donju granicu na iznos napona napajanja postavljaju FLASH memorija i Bluetooth modul koji zahtijevaju minimalno 2.7V.

Zbog lakoće dobavljivosti izbor je ograničen na dvije vrste akumulatora: litij-ionske (Li-ion) i nikal-metal hidridne (NiMH). NiMH akumulatori su nominalnog napona 1.2V u punom stanju i prazne se do oko 0.9 ili 1V. Prema tome, za postizanje dovoljnog napona potrebno je ili koristiti DC-DC pretvarač uz jedan akumulator ili koristiti tri akumulatora u seriji. Korištenje DC-DC pretvarača u osjetljivim biomedicinskim pojačalima u najmanju ruku komplicira njihov dizajn, pa je odlučeno da se takav pristup izbjegne. Zbog toga ostaje korištenje triju akumulatora u seriji. Pritom je napon kombinacije akumulatora 3.6V u punom stanju i prazni se do neke vrijednosti između 2.7V i 3V ovisno o struji pražnjenja i samim akumulatorima. Takav opseg napona dobro odgovara razvijanom uređaju.

Drugi tip akumulatora, litij-ionski, je nominalnog iznosa napona 3.6V i prazni se do 3V. Takav raspon napona vrlo dobro odgovara razvijanom uređaju jer se može uz korištenje linearног regulatora s malim padom napona (engl. *Low Dropout - LDO*) stabilizirati na npr. 2.8V što je dovoljno za sve komponente (i nije baš na donjoj granici napona napajanja memorije i Bluetooth modula), a još uvijek iskorištava potpuni kapacitet akumulatora. Regulirani izvor napajanja će uvelike smanjiti efekt velikih tranzijenata u potrošnji na analogna pojačala (tranzijenti će nastajati prvenstveno zbog pisanja u FLASH memoriju). Osim toga, pomoću linearnih regulatora napajanja s funkcijom gašenja (engl. *Shutdown*) bit će moguće smanjiti potrošnju uređaja o čemu će biti riječi kasnije u ovom poglavlju.

Iz ovog razmatranja vidljivo je da obje vrste akumulatora naponskim iznosima zadovoljavaju zahtjeve aplikacije. Pritom li-ion nešto bolje odgovaraju jer se pražnjenjem do 3V koristi cijeli njihov kapacitet. Kod NiMH akumulatora to ne mora biti slučaj u svim situacijama. Još jedan od potencijalnih problema NiMH akumulatora je korištenje tri akumulatora u seriji. Karakteristike pojedinih akumulatora se mogu malo razlikovati što će dovesti do različitog stanja napunjenošti na kraju ciklusa punjenja i posredno uzrokovati slabije iskorištenje kapaciteta akumulatora. Osim navedenog, prednost li-ion akumulatora

je i u većoj gustoći energije (lakši su i manji za isti iznos energije) što je pogodno za bilo koji prijenosni uređaj. Još jedna prednost li-ion akumulatora je činjenica da ih ne treba povremeno potpuno isprazniti i puniti do kraja kao NiMH akumulatora, što je praktičnije za korisnika.

Zbog navedenih razloga odlučeno je da će se izvor napajanja realizirati pomoću jednog li-ion akumulatora i LDO regulatora izlaznog napona 2.8V, s padom napona manjim od 200mV.

## 7.1 Procjena potrošnje uređaja

Sada kada su identificirane sve komponente uređaja i njihove maksimalne potrošnje, moguće je aproksimirati potrošnju uređaja. Tablica 3 prikazuje komponente, njihov broj i potrošnje.

**Tablica 3. – Potrošnje pojedinih komponenata uređaja**

Komponenta	Broj	Potrošnja po komponenti
Instrumentacijsko pojačalo INA333	3	80 $\mu$ A
Operacijsko pojačalo AD8619	12	40 $\mu$ A
Operacijsko pojačalo OPA2342	2	250 $\mu$ A
Mikrokontroler LPC1313	1	4mA (u prvoj aproksimaciji) uzimimo da mikrokontroler radi na 12MHz u normalnom načinu rada
FLASH memorija	1	30mA tijekom pristupa, inače 40 $\mu$ A
Bluetooth modul	1	30mA tijekom prijenosa, inače 400 $\mu$ A

Promotrimo prosječnu potrošnju uređaja tijekom perioda unutar kojeg se potpuno napuni memorija (6 sati) i pošalje njezin sadržaj preko Bluetootha. Slanje 512Mb podataka maksimalnom brzinom od 600kbps traje približno 850 sekundi, što možemo zaokružiti na četvrtinu sata. Sada je prosječna potrošnja neke komponente tijekom cijelog perioda:

$$I_p = \frac{6*I_1 + 0.25*I_2}{6+0.25} \quad (6)$$

pri čemu je  $I_1$  prosječna potrošnja tijekom prikupljanja signala, a  $I_2$  prosječna potrošnja tijekom slanja podataka.

Prepostavimo da se tijekom prikupljanja signala u memoriju zapisuje 1% vremena (konzervativno), a da je ostatak vremena neaktivna (*standby*). U tom slučaju prosječna potrošnja  $I_1$  memorije iznosi približno  $340\mu\text{A}$ .  $I_2$  iznosi  $30\text{mA}$ . Tada je ukupna prosječna potrošnja memorije  $1.53\text{mA}$ . Što se tiče Bluetooth modula, on se nalazi tijekom cijelog vremena prikupljanja podataka u *standby* načinu i troši  $400\mu\text{A}$ . Tijekom slanja troši prosječno  $30\text{ mA}$ . Tada prosječna potrošnja modula tijekom cijelog ciklusa iznosi  $1.58\text{mA}$ . Zbroje li se sada sve potrošnje po komponentama (tablica 4) dobiva se ukupna prosječna potrošnja uređaja od približno  $8.33\text{mA}$ . Prema tome, potrebna je baterija koja može 6 sati i 15 minuta davati  $8.33\text{mA}$  tj. baterija kapaciteta barem  $52\text{mAh}$ . Naravno da nikada nije dobro uzeti bateriju točno onog kapaciteta koji je potreban iz razloga poput samopražnjenja baterije, različitih gubitaka, pada kapaciteta baterije korištenjem i slično.

**Tablica 4. – Prosječne potrošnje komponenata**

Komponenta	$I_1$	$I_2$	$I_p$
<b>Bluetooth Modul</b>	$0.4\text{mA}$	$30\text{mA}$	$1.58\text{mA}$
<b>Memorija</b>	$30*1\% + 0.04*99\%$ ≈ $0.34\text{mA}$	$30\text{mA}$	$1.53\text{mA}$
<b>Analogni dio</b>	$1.22\text{mA}$	$1.22\text{mA}$	$1.22\text{mA}$
<b>Mikrokontroler</b>	$4\text{mA}$	$4\text{mA}$	$4\text{mA}$

## 7.2 Mogućnosti optimiranja potrošnje uređaja

Tijekom rada uređaja jasno je određeno kada koja komponenta ima nekog posla a kada naprsto troši energiju čekajući u *standby* modu. Razmatranjem takvih situacija dolazi se do tri mogućnosti smanjenja potrošnje:

1. Bluetooth modul ne mora biti napajan dok se prikuplja signal nego samo dok se podaci šalju preko Bluethootha
2. Analogni dio uređaja ne mora biti napajan kada prikupljanje signala završi i počne slanje preko Bluethootha
3. FLASH memorija ne mora biti napajana dok se u nju ne upisuju podaci

Prekidanje napajanja može se izvesti upotrebom linearnih regulatora napajanja koji imaju funkciju gašenja (engl. *Shutdown*). Takav regulator je TPS73028. Potrošnja se sada mijenja na način da se tijekom rada potrošnji komponente dodaje i vlastita potrošnja regulatora (tipično  $170\mu A$  za TPS73028), dok tijekom stanja isključenosti komponenta ne troši ništa, a regulator troši približno  $1\mu A$ .

Prva navedena mogućnost smanjenja potrošnje rezultirat će prosječnom potrošnjom Bluetooth modula od približno  $1.21mA$ , što je ušteda od  $370\mu A$ .

Druga mogućnost rezultirat će povećanjem (a ne smanjenjem) potrošnje sa  $1220\mu A$  na  $1334\mu A$ . Razlog je preveliki gubitak zbog vlastite potrošnje regulatora u odnosu na dobitak zbog kratkog vremena neaktivnosti analognih sklopova. No, ipak će se koristiti regulator napajanja za analogne sklopove zbog same stabilizacije napajanja, bez obzira na nešto povećanu potrošnju.

U trećoj situaciji može se koristiti regulator napajanja izrazito male vlastite potrošnje, koji zbog toga ima lošije ostale karakteristike (FLASH memoriji to neće smetati, za razliku od osjetljivih analognih sklopova ili RF primopredajnika Bluetooth modula). Takav regulator je TPS78228. Vlastita potrošnja mu je oko  $3\mu A$  u uvjetima u kojima će se koristiti. Potrošnju u ugašenom stanju može se zanemariti (stotinjak nA). Prosječna potrošnja memorije tijekom prikupljanja podataka je približno  $300\mu A$ . Tijekom slanja prosječna potrošnja je oko  $30mA$ . Ukupna prosječna potrošnja memorije je sada  $1.49mA$ , što je ušteda od  $40\mu A$ .

Tablica 5 prikazuje prosječne potrošnje komponenata uređaja uz dodavanje regulatora napajanja s mogućnošću gašenja.

**Tablica 5. – Prosječne potrošnje komponenata nakon optimizacije**

Komponenta	I <sub>1</sub>	I <sub>2</sub>	I <sub>p</sub>
<b>Bluetooth Modul</b>	0.001mA	30.17mA	1.21mA
<b>Memorija</b>	$30 * 1\% + 0.0001 * 99\%$ ≈ 0.30mA	30.003mA	1.49mA
<b>Analogni dio</b>	1.39mA	0.001mA	1.33mA
<b>Mikrokontroler</b>	4mA	4mA	4mA

Tablica 6 sumira promjenu prosječne potrošnje postignutu dodavanjem regulatora napajanja.

**Tablica 6. – Promjena potrošnje zbog optimizacije**

Komponenta	Potrošnja bez regulatora	Potrošnja s regulatorom	Promjena
<b>Bluetooth modul</b>	1.58mA	1.21mA	-0.37mA
<b>Memorija</b>	1.53mA	1.49mA	-0.04mA
<b>Analogni dio</b>	1.22mA	1.33mA	+0.11mA
<b>Mikrokontroler</b>	4mA	4mA	4mA
<b>Ukupno</b>	8.33mA	8.03mA	<b>-0.3mA</b>

Ukupno smanjenje potrošnje nije posebno veliko (3.75%), ali situacija se može promotriti i s druge strane. Za malo povišenje cijene (regulatori su jeftini) dobiva se rješenje koje nudi

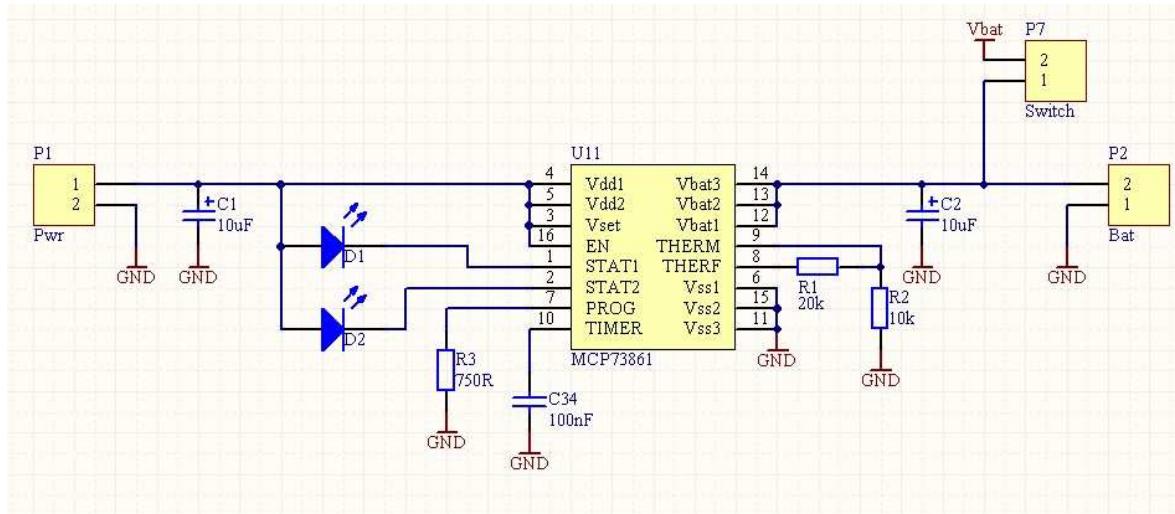
regulaciju napona napajanja i još uz to smanjuje potrošnju umjesto da je poveća zbog vlastite potrošnje regulatora. Puno veće mogućnosti optimizacije potrošnje nudit će prebacivanje mikrokontrolera u neki od načina rada smanjene potrošnje (jer mikrokontroler i troši oko pola ukupne energije ukoliko neprestano radi u normalnom načinu rada).

### 7.3 Izvedba punjača baterija

Punjjenje li-ion baterije odvija se u tri faze:

- polagano punjenje akumulatora konstantnom strujom ako je preduboko ispraznjen; faza se preskače ako nije
  - faza punjenja konstantnom strujom: akumulator se puni dok ne dosegne napon od 4.1 ili 4.2 V (ovisno o tipu anode akumulatora)
  - faza punjenja konstantnim naponom: akumulator se puni uz konstantan napon sve dok struja punjenja ne padne na 3% početne vrijednosti kada se akumulator smatra punim

Regulaciju napona i struje punjenja održuje integrirani punjač li-ion akumulatora MCP73861 proizvođača Microchip Technology Inc. Izvedba punjača prikazana je na slici



**Slika 19.** – Shema punjača li-ion akumulatora

Na konektor oznake *Pwr* se dovodi vanjski izvor napona 4.5 – 12V koji predstavlja izvor energije za punjenje akumulatora. LE diode signaliziraju trenutno stanje punjača i njihovo

ponašanje opisano je u dokumentaciji čipa MCP73861. Priključak  $V_{set}$  se može pritegnuti na masu ili na napajanje i time odrediti napon punjenja na 4.1V ili 4.2V. Otpornik između priključka PROG i mase sklopa određuje struju punjenja akumulatora. Spajanjem kondenzatora na priključak TIMER određuje se maksimalno trajanje pojedinih faza punjenja. Nominalna vrijednost je 100nF pri čemu svaka faza može trajati 1h prije nego što punjač prekine punjenje iz sigurnosnih razloga. Priključci THERM i THERMF mogu se koristiti za praćenje temperature termistora ugrađenog u akumulator, ali se ovdje ne koriste pa su spojeni na fiksni iznos napona kako je preporučeno u dokumentaciji čipa. Na priključak označen *Bat* spaja se akumulator. Na priključak označen *Switch* spaja se prekidač koji spaja akumulator na ostatak uređaja i time mu daje napajanje.

## 8 Ugrađena programska podrška (*firmware*)

### 8.1 CMSIS

CMSIS je skraćenica za *Cortex Microcontroller Software Interface Standard* tj. standardizirano programsko sučelje za *Cortex* mikrokontrolere. Radi se o setu programskih komponenata koje definiraju način pristupa registrima, adresi prekidnih vektora i sl. Korištenjem CMSIS-a ubrzava se razvoj programske podrške i olakšava korištenje već postojećeg koda. Dio sučelja koji je zajednički svim mikrokontrolerima *Cortex* definira ARM Ltd., a dio koji je specifičan pojedinoj porodici definira njezin proizvođač. CMSIS jeписан u programskom jeziku C i distribuiran u obliku nekoliko datoteka. Opis koji slijedi bit će baziran na verziji CMSIS-a pisanoj za porodicu LPC13xx.

#### **LPC13xx.h**

Izraduje ju proizvođač. Centralno zaglavljeno je, koje se koristi u korisničkoj aplikaciji. Sadrži:

- **Definicije rednih brojeva za sve prekidne rutine.**
- **Konfiguracijske parametre za datoteku *core\_m3.h*** (koja konfigurira jezgru mikrokontrolera). Konkretno, za porodicu LPC13xx definira se postojanje MPU-a (engl. *memory protection unit*) i broj bitova prioriteta prekida. Definira se i koja će se konfiguracijska funkcija koristiti za vremensku jedinicu *SysTick*, ona definirana od strane ARM-a ili ona definirana od strane proizvođača (ukoliko takva uopće postoji).
- **Sloj za pristup perifernim jedinicama.** Definiraju se strukture za svaku perifernu jedinicu koje rasporedom varijabli odgovaraju rasporedu registara periferne jedinice. Definirane strukture se mapiraju na odgovarajuća mesta u memoriji.
- **Pomoćne funkcije za pristup perifernim jedinicama.** Pomoćne funkcije obavljaju razne zadaće poput inicijalizacije perifernih sklopova, promjene njihove funkcionalnosti i slično. Ovaj dio je optionalan i nema ga u datoteci *LPC13xx.h*

Jedna od struktura za pristup perifernim jedinicama izgleda ovako (radi se o strukturi za pristup AD pretvorniku):

```
typedef struct
```

```
{
    __IO uint32_t CR;
    __IO uint32_t GDR;
    uint32_t RESERVED0;
    __IO uint32_t INTEN;
    __I  uint32_t DR0;
    __I  uint32_t DR1;
    __I  uint32_t DR2;
    __I  uint32_t DR3;
    __I  uint32_t DR4;
    __I  uint32_t DR5;
    __I  uint32_t DR6;
    __I  uint32_t DR7;
    __I  uint32_t STAT;
} LPC_ADC_TypeDef;
```

Struktura se mapira u memoriji pomoću sljedećih definicija:

```
#define LPC_APB0_BASE      (0x40000000UL)
* * *
#define LPC_ADC_BASE        (LPC_APB0_BASE + 0x1C000)
* * *
#define LPC_ADC      ((LPC_ADC_TypeDef *) LPC_ADC_BASE)
```

Na isti način, samo uz druge adrese, mapiraju se i ostale strukture za pristup perifernim jedinicama.

### **core\_cm3.c i core\_cm3.h**

Izrađuje ih ARM Ltd. zajedno čine sloj za pristup registrima jezgre mikrokontrolera. U *core\_cm3.h* definirane su strukture za pristup registrima jezgre na način kao i strukture za pristup perifernim jedinicama u datoteci *LPC13xx.h*. Osim toga definiraju se i maske za pristup pojedinim bitovima registara jezgre, prototipi funkcija za konfiguriranje raznih aspekata funkcioniranja jezgre (npr. podešavanje prioriteta prekida) i statičke *inline* (ne poziva ih se nego se ugrađuju na mjesto poziva – vrlo efikasne u smislu brzine) funkcije za pristup nekim registrima. Datoteka *core\_cm3.c* sadrži implementacije funkcija čiji su prototipi dani u *core\_cm3.h*.

### **startup\_lpc13.c**

Izrađuje je ARM Ltd. Proizvođač ju preuređuje da bi odgovarala konkretnoj porodici mikrokontrolera. Sadrži prekidne rutine za sve prekide mikrokontrolera. Prekidne rutine sadrže samo beskonačnu petlju i definirane su kao *weak* (engl. *weak* - slab) funkcije. To znači da korisnički kod može definirati funkciju istog imena koja obavlja željeni posao. Kada postoje funkcije istog imena, a jedna je *weak*, *compiler* će koristiti uvijek onu funkciju koja nije *weak*. Prema tome, može se reći da *weak* funkcije iz datoteke *startup\_lpc13.c* „čuvaju mjesto pravim prekidnim rutinama“ koje će definirati korisnički kod.

U datoteci *startup\_lpc13.c* definirana je i *reset* prekidna rutina koja se poziva nakon *reset-a* ili kod paljenja mikrokontrolera (jer se mikrokontroler najprije resetira kada mu se dovede napajanje). Ona inicijalizira mikrokontroler i na kraju poziva funkciju *main()*.

### **system\_lpc13xx.c i system\_lpc13xx.h**

Izrađuje ih ARM Ltd. Proizvođač ih preuređuje da bi odgovarale konkretnoj porodici mikrokontrolera. Datoteka *system\_lpc13xx.c* sadržava funkciju za inicijalizaciju mikrokontrolera koja postavlja izvor *clock* signala, njegovu frekvenciju, ostale oscilatore (*Watchdog, USB*) te konfigurira PLL po potrebi. Odabir željene konfiguracije vrši se izmjenom simboličkih konstanti koje su u datoteci definirane. Sadržava i globalnu varijablu koja odgovara frekvenciji sistemskog *clock* signala kao i funkciju za osvježavanje. Funkcija za inicijalizaciju mikrokontrolera poziva se iz *reset* prekidne rutine prije nogo što se uđe u funkciju *main()*.

Datoteka *system\_lpc13xx.h* sadržava prototipe dviju navedenih funkcija i deklaraciju globalne varijable za frekvenciju sistemskog *clock* signala.

## **8.2 Struktura ugrađene programske podrške**

Zamišljeno je da se uređajem se upravlja pomoću tipke koja je priključena na mikrokontroler. Informacije o svojem stanju uređaj daje pomoću jedne LE diode spojene na mikrokontroler. Čitanje stanja tipke i paljenje/gašenje LE diode bit će zadaće ugrađene programske podrške.

Sve zadaće ugrađene programske podrške su:

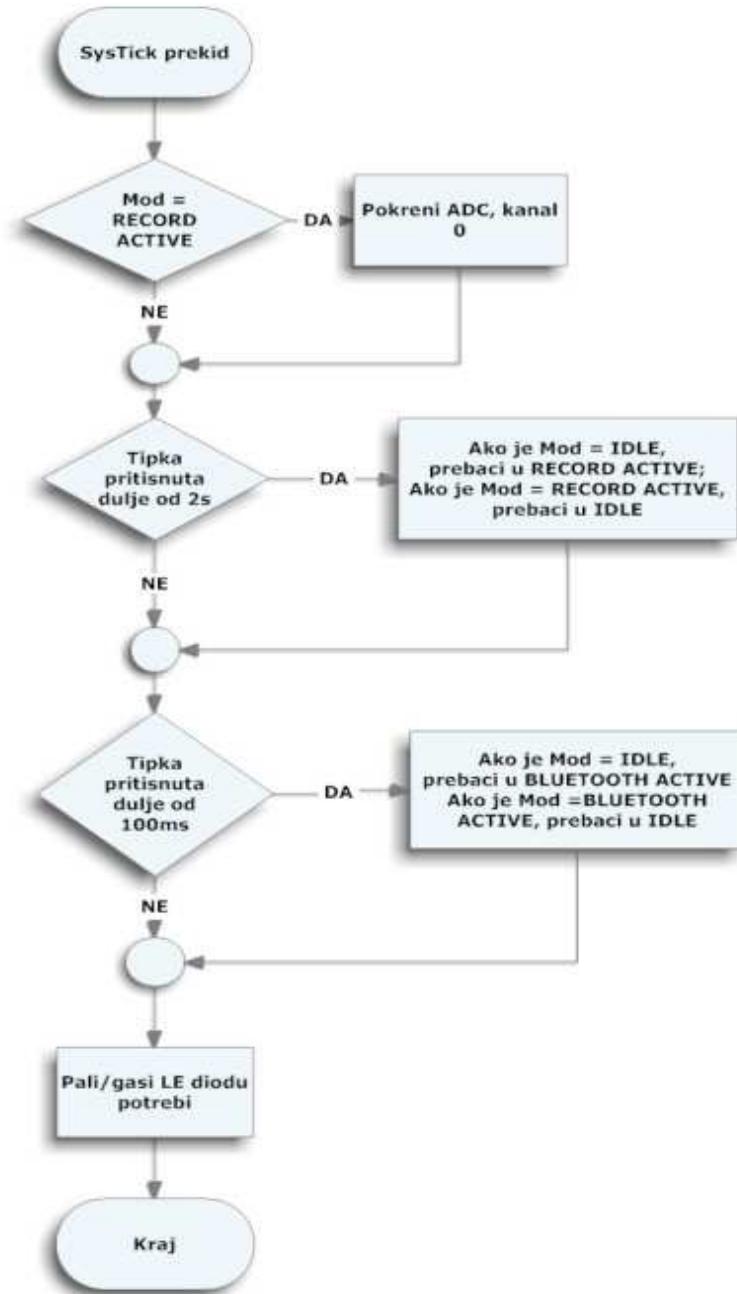
- upravljanje radom AD pretvornika
- pristup FLASH memoriji
- upravljanje Bluetooth modulom
- slanje podataka
- čitanje stanja tipke
- paljenje/gašenje LE diode

Rad ugrađene programske podrške može se podijeliti na tri načina (u svakom trenu je aktivan samo jedan):

- **IDLE:** Uredaj ne radi ništa osim čitanja stanja tipke. Ukoliko je tipka pritisnuta dulje od 2 sekunde, prelazi se u način RECORD ACTIVE. Ukoliko je tipka pritisnuta dulje od 100 ms, a manje od 2 sekunde, prelazi se u način BLUETOOTH ACTIVE. U IDLE načinu rada LE dioda ne svjetli.
- **RECORD ACTIVE:** Uredaj uzorkuje signale i spremi uzorke u memoriju. Maksimalno se može uzorkovati 6 sati signala. Bluetooth modul je ugašen. Ukoliko je tipka pritisnuta dulje od 2 sekunde prelazi se u način IDLE. U načinu rada RECORD ACTIVE LE dioda svijetli 200 ms unutar perioda od 1 sekunde.
- **BLUETOOTH ACTIVE:** Uredaj ne uzorkuje signale. Analogni dio je ugašen. Bluetooth modul je aktivan i čeka zahtjeve za uspostavom bežične veze od drugih uređaja. Kada se uspostavi veza preko virtualnog serijskog porta, uređaj čeka da drugi uređaj pošalje bajt vrijednosti 0x01, na što počinje slati podatke. Ukoliko je u ovom načinu tipka pritisnuta više od 100 ms, prelazi se u način IDLE. U načinu rada BLUETOOTH ACTIVE LE dioda konstantno svijetli.

Strukturalno se programska podrška može podijeliti na dva procesa: beskonačnu *while()* petlju unutar funkcije *main()* i prekidnu rutinu vremenskog sklopa *SysTick* koja se aktivira svake milisekunde.

Prekidna rutina vremenskog sklopa *SysTick* pokreće konverziju na AD pretvorniku, čita stanje tipke i, ovisno o načinu rada, pali ili gasi LE diodu. Osim toga unutar prekidne rutine se vrši prebacivanje iz jednog načina rada u drugi. Blok dijagram prekidne rutine vremenskog sklopa *SysTick* prikazan je na slici 20.

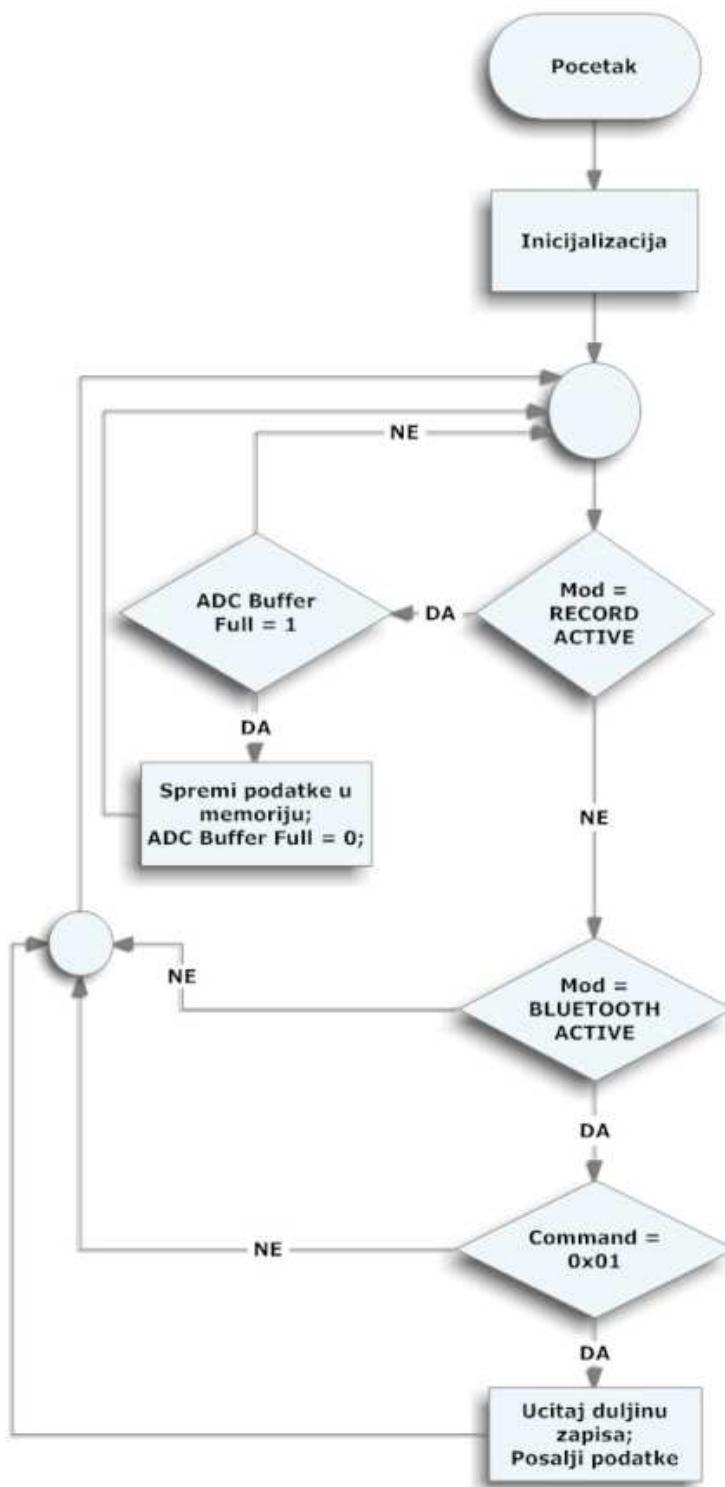


**Slika 20. – Blok-dijagram prekidne rutine vremenskog sklopa *SysTick***

Beskonačna petlja u funkciji *main()* se izvršava cijelo vrijeme dok nije aktivan neki prekid. Njezina zadaća je ovisno o načinu rada snimati ili slati podatke. Ako je trenutni način rada RECORD ACTIVE, u petlji će se neprestano ispitivati stanje varijable *ADC\_Buffer\_Full* koju prekidna rutina AD pretvornika postavlja u jedinicu kada se napuni polje uzorka (po 512 uzorka za svaki kanal). Kada se stanje varijable promjeni u jedinicu, pokreće se prebacivanje podataka u memoriju. Ako je trenutni način rada BLUETOOTH ACTIVE, u

petlji će se provjeravati je li drugi uređaj preko UART-a posao bajt 0x01 (zadnji primljeni bajt sa UART-a spremi se u varijablu *Command*) i ako je započet će prijenos podataka. Opisana struktura funkcije *main()* prikazana je na slici 21.

Osim opisana dva procesa koriste se još dvije prekidne rutine. Prva se pokreće kada preko UART-a stigne bajt podataka. Unutar rutine se bajt kopira u globalnu varijablu *Command*. Druga se pokreće kad AD pretvornik završi s pretvorbom. Ona će biti pobliže opisana u sljedećem potpoglavlju.

Slika 21. – Blok-dijagram funkcije *main()*

### 8.3 Upravljanje AD pretvornikom

AD pretvorba se pokreće iz prekidne rutine vremenskog sklopa *SysTick*. Period vremenskog sklopa podešen je na 1ms. Izbor kanala koji će se uzorkovati vrši se pomoću globalne varijable *ADC\_Channel*. *SysTick* prekidna rutina postavlja varijablu *ADC\_Channel* na vrijednost 0 i inicijalizira AD pretvornik na uzorkovanje nultog kanala (ukoliko se trenutno uzorkuje signal što se vidi iz varijable *Mode*). Tu zadaću obavlja sljedeći odsječak:

```
if (Mode == RECORD_MODE)
{
    //Configure ADC (ch0, clkdiv=2+1, start now)
    LPC_ADC->CR = (0x01) | (0x02<<8) | (0x01<<24);
    ADC_Channel = 0;
}
```

Kada uzorkovanja nultog kanala završi, AD pretvornik zahtjeva prekid i pokreće se AD prekidna rutina:

```
void ADC_IRQHandler(void)
{
    uint32_t RegVal;

    RegVal = LPC_ADC->STAT; /* Read ADC will clear the
interrupt */
    RegVal = LPC_ADC->DR0;
    RegVal = LPC_ADC->DR1;
    RegVal = LPC_ADC->DR2;
    RegVal = (LPC_ADC->GDR & 0xFFFF) >> 8;
    ADC_Result[ADC_Result_IDX + ADC_Channel*512] = RegVal;
    LPC_ADC->CR &= 0xF8FFFFFF; /* stop ADC now */

    if(ADC_Channel == 0)
    {
        ADC_Channel = 1;

        //Configure ADC (ch1, clkdiv=2+1, start now)
        LPC_ADC->CR = (0x02) | (0x02<<8) | (0x01<<24);
    }

    else if(ADC_Channel == 1)
    {
        ADC_Channel = 2;

        //Configure ADC (ch2, clkdiv=2+1, start now)
```

```

LPC_ADC->CR = (0x04) | (0x02<<8) | (0x01<<24);

}

else if(ADC_Channel == 2)
{
    ADC_Channel = 0;

    ADC_Result_IDX++;
    if(ADC_Result_IDX == 512)
    {
        ADC_Result_IDX = 0;
        ADC_Buffer_Full = 1;
    }
}
}

```

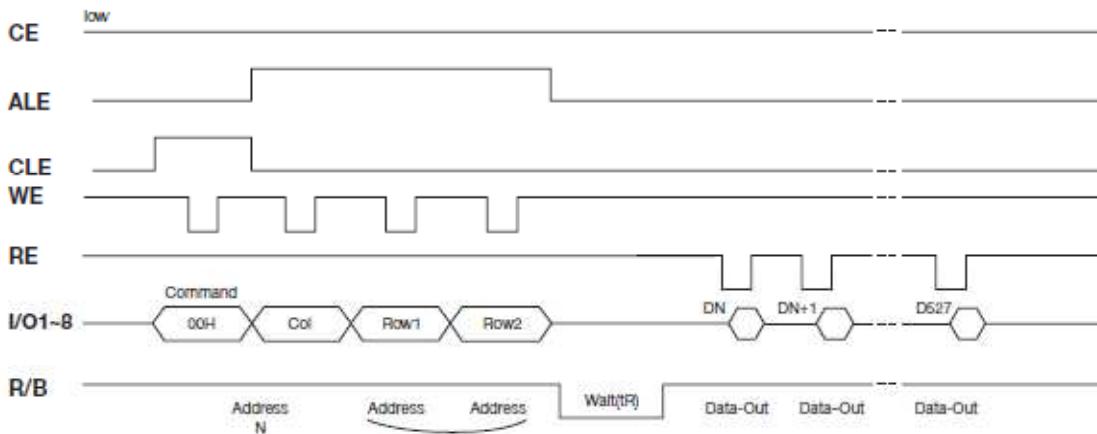
U prekidnoj rutini čitaju se registri s rezultatom pretvorbe da bi se poništo prekid. Čitaju se vrijednosti za sva tri kanala iako samo jedan sadrži rezultat i samo njega bi trebalo pročitati da se prekid poništi. No, brže je naprsto pročitati sva tri registra nego ispitivati varijablu *ADC\_Channel* i pročitati odgovarajući registar. Rezultat pretvorbe (gornjih 8 bita) se spremi u polje *ADC\_Result* koje sadrži po 512 rezultata za sva tri kanala. Nakon toga se varijabla *ADC\_Channel* cirkularno prebacuje na sljedeći kanal (ako je prethodna konverzija bila za kanal 0, sljedeća će biti za kanal 1). Zatim se AD pretvornik konfigurira za pretvorbu na tom kanalu i pokreće. U narednom prekidu AD pretvornika sve se ponavlja. Sekvenca traje dok se ne uzorkuje kanal 2 tj. posljednji kanal u sekvenci. Nakon toga se povećava indeks polja za rezultate i prebacuje vrijednost varijable *ADC\_Channel* natrag na vrijednost 0. Na kraju sekvence se još i provjerava je li indeks polja rezultata stigao do kraja i ako je, to stanje se signalizira funkciji *main()* postavljanjem varijable *ADC\_Buffer\_Full* na vrijednost 1.

## 8.4 Pristup FLASH memoriji

Pristup memoriji uvijek započinje postavljanjem priključka CE u nisko stanje. Nakon toga se pomoću priključaka CLE i ALE određuje pristupa li se adresnom, podatkovnom ili instrukcijskom registru memorije. Ostatak se razlikuje ovisno o vrsti pristupa (pisanje stranice u memoriju, čitanje stranice iz memorije, brisanje bloka ili čitanje statusnog registra).

## Čitanje stranice iz memorije

Slika 22 prikazuje vremenski tijek čitanja jedne stranice iz memorije.



**Slika 22. – Tijek čitanja jedne stranice iz FLASH memorije**

Najprije je potrebno postaviti priključak CE u nisko stanje da bi memorije prešla u aktivno stanje. Nakon toga se postavlja ALE u nisko stanje, a CLE u visoko stanje čime se selektira pristup instrukcijskom registru. Sada treba na sabirnicu postaviti komandu koja za čitanje iznosi 0x00, 0x01 ili 0x50. Instrukcija se prenosi memoriji na padajući brid signala WE (*write enable*). Svaki upis u memoriju (podataka, adresa ili instrukcija) vrši se postavljanjem podataka na sabirnicu i zatim spuštanjem signala WE u nisko stanje. Iduće na sabirnicu treba postaviti najniži adresni bajt. On predstavlja adresu bajta unutar stranice od kojega se želi započeti čitanje. Adresa je 8-bitna, znači može poprimiti vrijednosti od 0 do 255. No, bajtova u stranici ima više - 528. Sada je moguće objasniti zašto postoje tri različite komande za čitanje. Instrukcija 0x00 (naziva se i *READ1*) započinje čitanje na bajtu koji je zadan adresom, znači na bilo kojem bajtu iz raspona 0 do 255. Instrukcija 0x01 (također se naziva *READ1*) započinje čitanje na bajtu koji je zadan adresom uz pomak od 256, znači na bilo kojem bajtu iz raspona 256 do 511. Instrukcija 0x50 (naziva se i *READ2*) započinje čitanje na bajtu koji je zadan adresom uz pomak od 512, znači na bilo kojem bajtu iz raspona 512 do 528 (pritom se uzimaju samo niža četiri bita adresnog bajta). U većini slučajeva se želi pročitati cijela stranica od početka pa se koristi instrukcija 0x00 uz prvi adresni bajt 0x00. Nakon prvog adresnog bajta, šalje se drugi u kojemu najnižih pet bitova definira adresu stranice unutar bloka (0 - 31). Najviša tri bita su dio adrese bloka. Blokova ima 4096, znači potrebno je 12 bitova da ih se adresira. Već je

rečeno da najviša tri bita drugog adresnog bajta predstavljaju tri najniža bita adrese bloka. U trećem adresnom bajtu nalazi se dalnjih 8 bita adrese i u zadnjem, četvrtom, adresnom bajtu nalazi se zadnji, najviši, bit adrese bloka. Taj bit nalazi se najnižoj poziciji. Opisani sadržaj adresnih bajtova vidi se i u tablici 7. Pritom se oznake IO0 – IO7 odnose na ulazno/izlazne priključke memorije, a oznake A0 – A25 na bitove adrese. A0 – A7 čine adresu početnog bajta unutar stranice (zajedno sa pomakom koji definira instrukciju). A9 – A13 čine adresu stranice unutar bloka, a A14 – A25 čine adresu bloka.

**Tablica 7. – Adresiranje bajte unutar FLASH memorije**

	<b>IO7</b>	<b>IO6</b>	<b>IO5</b>	<b>IO4</b>	<b>IO3</b>	<b>IO2</b>	<b>IO1</b>	<b>IO0</b>
<b>1. adresni bajt</b>	A7	A6	A5	A4	A3	A2	A1	A0
<b>2. adresni bajt</b>	A16	A15	A14	A13	A12	A11	A10	A9
<b>3. adresni bajt</b>	A24	A23	A22	A21	A20	A19	A18	A17
<b>4. adresni bajt</b>	A25	Nisko						

Nakon adresiranja spušta se ALE u nisko stanje. Sada memorija počinje prebacivati sadržaj adresirane stranice u svoj memorijski međuspremnik. To traje 200-tinjak  $\mu$ s. Unutar tog vremena memorija je zauzeta i ne može joj se pristupiti. Za to vrijeme je priključak R/B u niskom stanju. Mikrokontroler treba čekati da se R/B podigne u visoko stanje što znači da je memorija prebacila stranicu u međuspremnik i da ju se može pročitati. Memorija uvijek prebacuje cijelu stranicu u međuspremnik, neovisno o početnom bajtu danom u adresi. Početni bajt samo definira od kojeg će se mjesto početi čitati međuspremnik. Podaci se čitaju bajt po bajt, na padajući brid signala RE (*Read Enable*). Kada se pročita zadnji bajt iz međuspremnika, memorija će automatski početi prebacivati sljedeću stranicu u međuspremnik. Ako to nije potrebno i želi se završiti ciklus čitanja, potrebno je postaviti CE u visoko stanje čime se čitanje prekida i memorija postavlja u neaktivno stanje (*standby*).

Čitanje stranice obavlja funkcija *FLASH\_Page\_Read*:

```
void FLASH_Page_Read(unsigned int Block, unsigned char Page)
```

```

{
    unsigned int i = 0;

    GPIOSetValue(CS_PORT, CS_BIT, 0);
    Set_Port2_As_Output();

    //Command
    Port2_Write(0x00);
    GPIOSetValue(CLE_PORT, CLE_BIT, 1);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);
    GPIOSetValue(CLE_PORT, CLE_BIT, 0);

    //Column
    Port2_Write(0x00);
    GPIOSetValue(ALE_PORT, ALE_BIT, 1);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);

    //Row 1
    Port2_Write((Page & 0x1F) | ((Block & 0x0007) << 5));
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);

    //Row 2
    Port2_Write((Block & 0x07F8) >> 3);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);

    //Row 3
    Port2_Write((Block & 0x0800) >> 11);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);

    GPIOSetValue(ALE_PORT, ALE_BIT, 0);

    //Wait (dok R/B ne bude jednak 1)
    while (!(LPC_GPIO2->MASKED_ACCESS[0x200]));

    Set_Port2_As_Input();
    for(i=0; i<528; i++)
    {
        GPIOSetValue(RE_PORT, RE_BIT, 0);
        FLASH_Page[i] = Port2_Read();
        GPIOSetValue(RE_PORT, RE_BIT, 1);
    }

    GPIOSetValue(CS_PORT, CS_BIT, 1);
}

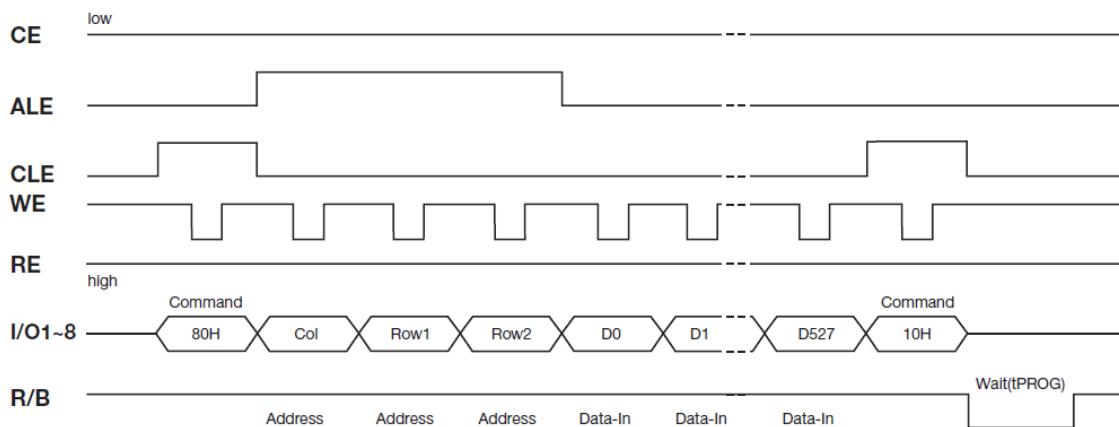
```

}

Funkcija prima dva parametra: redne brojeve stranice i bloka unutar kojeg se stranica nalazi. Pročitani podaci spremaju se u polje *FLASH\_Page* dugo 528 bajta. Za manipulaciju ulazno/izlaznim priključcima koristi se funkcija *GPIO\_Set\_Value* koja prima tri parametra: broj *porta* na kojem se priključak nalazi, redni broj priključka i vrijednost (nula ili jedan). Funkcija se nalazi u datoteci *gpio.c* koja se standardno isporučuje s razvojnim sustavom za LPC mikrokontrolere.

### Pisanje stranice u memoriju

Slika 23. prikazuje vremenski tijek pisanja stranice u memoriju.



**Slika 23. – Tijek pisanja stranice u FLASH memoriju**

Način slanja instrukcije i adresiranje su kod pisanja stranice isti kao i kod čitanja stranice i nije ih potrebno ponovno opisivati osim činjenice da je instrukcija za pisanje 0x80. Nakon adresiranja slijedi slanje do 528 bajtova koje memorija sprema u međuspremnik. U njemu su sve lokacije prije slanja postavljane na 0xFF. Kod prebacivanja sadržaja međuspremnika u memoriju, pišu se samo bitovi vrijednosti „0“, a bitovi vrijednosti „1“ se ignoriraju. Zbog toga se mora prije pisanja obrisati cijeli blok što postavlja sve bajtove memorije u 0xFF. Nakon prijenosa podataka potrebno je postaviti CLE u visoko i poslati instrukciju 0x10 koja će pokrenuti prebacivanje sadržaja međuspremnika u memoriju. Za vrijeme prebacivanja podataka memorija je zauzeta (signal R/B u niskom stanju). Kada prebacivanje završi (signal R/B se postavi u visoko stanje), trebalo bi pročitati statusni registar i provjeriti je li pisanje uspjelo.

Pisanje stranice u memoriju obavlja funkcija FLASH\_Page\_Write:

```

unsigned char FLASH_Page_Write(unsigned char *Data,
unsigned int Len,                                     unsigned int Block, unsigned
char Page)
{
    unsigned int i = 0;

    //Ako je blok los vrati 1(pisanje nije uspjelo)
    if(FLASH_Check_Bad_Block(Block))
    {
        return 1;
    }

    GPIOSetValue(CS_PORT, CS_BIT, 0);
    Set_Port2_As_Output();

    //Command
    Port2_Write(0x80);
    GPIOSetValue(CLE_PORT, CLE_BIT, 1);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);
    GPIOSetValue(CLE_PORT, CLE_BIT, 0);

    //Column
    Port2_Write(0x00);
    GPIOSetValue(ALE_PORT, ALE_BIT, 1);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);

    //Row 1
    Port2_Write((Page & 0x1F) | ((Block & 0x0007) << 5));
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);

    //Row 2
    Port2_Write((Block & 0x07F8) >> 3);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);

    //Row 3
    Port2_Write((Block & 0x0800) >> 11);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);
    GPIOSetValue(ALE_PORT, ALE_BIT, 0);

    for(i=0; i<Len; i++)
    {

```

```

        GPIOSetValue(WE_PORT, WE_BIT, 0);
        Port2_Write(*Data++);
        GPIOSetValue(WE_PORT, WE_BIT, 1);
    }

    Port2_Write(0x10);
    GPIOSetValue(CLE_PORT, CLE_BIT, 1);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);
    GPIOSetValue(CLE_PORT, CLE_BIT, 0);

    GPIOSetValue(CS_PORT, CS_BIT, 1);

    //Wait (dok R/B ne bude jednak 1)
    while (! (LPC_GPIO2->MASKED_ACCESS[0x200]));

    i = FLASH_Status_Read();
    //Ako pisanje nije uspjelo, označi blok kao los
    if(i & 0x01)
    {
        FLASH_Tag_Bad_Block(Block);
        return 1;
    }

    return 0;
}

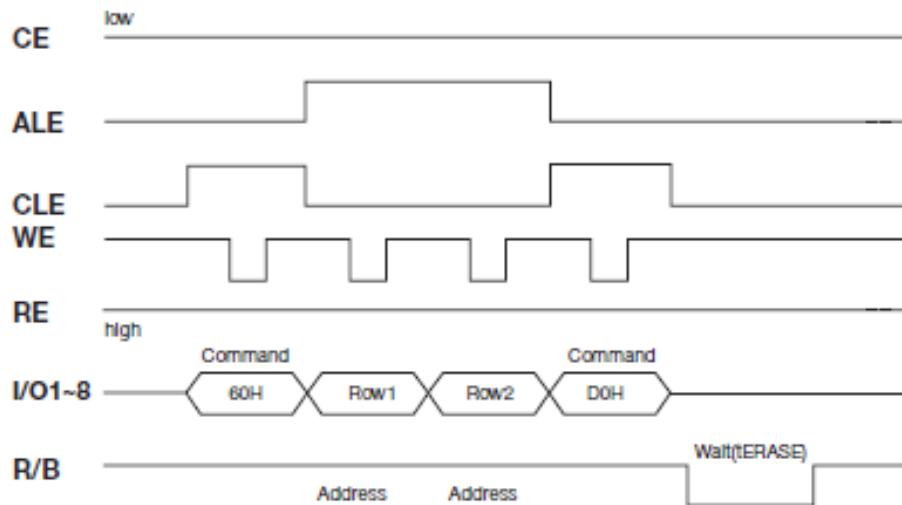
```

Funkcija prima četiri parametra: pokazivač na početak polja podataka koje će se pisati u memoriju, duljinu tog polja te redne brojeve stranice i bloka unutar kojeg se stranica nalazi. Izvedba funkcije slična je kao i u slučaju čitanja. Razlika je u provjeri ispravnosti bloka prije pisanja i ukoliko blok nije ispravan, funkcija ne piše ništa nego završava izvođenje uz povratnu vrijednost 1. Ispravnost se provjerava čitanjem vrijednosti iz tablice neispravnih tj. polja *FLASH\_Bad\_Blocks*. Polje je dugo 512 bajta tj. 4096 bita – po jedan bit za svaki blok. Ako je bit odgovarajućeg bloka postavljen u 1, blok je neispravan. Tablica neispravnih blokova puni se kada se prvi puta pristupi memoriji, jer se u memoriju nakon tvorničkog testa u svaki blok upiše je li ispravan ili ne. U tablicu se dodaju i blokovi koji se pokvare tijekom upotrebe. Ukoliko blok nije označen u tablici kao neispravan, pristupa se pisanju u memoriju. Po završetku pisanja provjerava se statusni registar memorije. Čitajući statusni registar moguće je provjeriti je li operacija pisanja uspjela. Ako operacija nije uspjela, blok se proglašava neispravnim, što se bilježi u tablici neispravnih blokova. U tom slučaju funkcija također završava izvođenje s povratnom vrijednošću 1. Ukoliko pisanje uspije funkcija završava izvođenje s povratnom vrijednošću 0. U dijelu

koda koji poziva ovu funkciju potrebno je osigurati mehanizam koji će prepoznati izlaznu vrijednost 1 kao znak neuspješnog pisanja iz bilo kojeg od dva opisana razloga. U tom slučaju potrebno je ponoviti pokušaj pisanja na sljedećem bloku.

### Brisanje bloka

Slika 24 prikazuje vremenski tijek brisanja bloka.



**Slika 24. – Vremenski tijek brisanja bloka FLASH memorije**

Način slanja instrukcije i adresiranje slični su kao i kod pisanja/čitanja. Instrukcija za brisanje bloka je 0x60. Adresiranje kod ove operacije se razlikuje u tome što se ne šalje prvi adresni bajt (nije potreban) i u tome što se adresa stranice ignorira (briše se cijeli blok). Nakon adresiranja potrebno je još poslati instrukciju 0xD0 kojom se potvrđuje brisanje bloka. Nakon toga počinje brisanje bloka i memorija postaje zauzeta (signal R/B u niskom stanju). Kada prebacivanje završi (signal R/B se postavi u visoko stanje), trebalo bi pročitati statusni registar i provjeriti je li pisanje uspjelo.

Brisanje bloka obavlja funkcija *FLASH\_Block\_Erase*:

```
unsigned char FLASH_Block_Erase(unsigned int Block)
{
    unsigned int i = 0;

    //Ako je blok los vrati 1(pisanje nije uspjelo)
    if(FLASH_Check_Bad_Block(Block))
    {
        return 1;
    }
    else
    {
        //Nakon uspješnog pisanja vrati 0
        return 0;
    }
}
```

```

}

GPIOSetValue(CS_PORT, CS_BIT, 0);
Set_Port2_As_Output();

//Command
Port2_Write(0x60);
GPIOSetValue(CLE_PORT, CLE_BIT, 1);
GPIOSetValue(WE_PORT, WE_BIT, 0);
GPIOSetValue(WE_PORT, WE_BIT, 1);
GPIOSetValue(CLE_PORT, CLE_BIT, 0);

//Row 1
GPIOSetValue(ALE_PORT, ALE_BIT, 1);
Port2_Write((Block & 0x0007) << 5);
GPIOSetValue(WE_PORT, WE_BIT, 0);
GPIOSetValue(WE_PORT, WE_BIT, 1);

//Row 2
Port2_Write((Block & 0x07F8) >> 3);
GPIOSetValue(WE_PORT, WE_BIT, 0);
GPIOSetValue(WE_PORT, WE_BIT, 1);

//Row 3
Port2_Write((Block & 0x0800) >> 11);
GPIOSetValue(WE_PORT, WE_BIT, 0);
GPIOSetValue(WE_PORT, WE_BIT, 1);
GPIOSetValue(ALE_PORT, ALE_BIT, 0);

Port2_Write(0xD0);
GPIOSetValue(CLE_PORT, CLE_BIT, 1);
GPIOSetValue(WE_PORT, WE_BIT, 0);
GPIOSetValue(WE_PORT, WE_BIT, 1);
GPIOSetValue(CLE_PORT, CLE_BIT, 0);

GPIOSetValue(CS_PORT, CS_BIT, 1);

//Wait (dok R/B ne bude jednak 1)
while (! (LPC_GPIO2->MASKED_ACCESS[0x200]));

i = FLASH_Status_Read();
//Ako pisanje nije uspjelo, označi blok kao los
if(i & 0x01)
{
    FLASH_Tag_Bad_Block(Block);
    return 1;
}

return 0;

```

```
}
```

Funkcija prima jedan parametar, redni broj bloka koji treba obrisati. Izvedba funkcije slična je kao i u slučaju pisanja, samo što u ovom slučaju nema prijenosa podataka. Funkcija provjerava ispravnost bloka na isti način kao i funkcija za pisanje stranice, uz iste povratne vrijednosti.

## Čitanje statusnog registra

Čitanje statusnog registra vrši se slanjem instrukcije 0x70. Nakon toga je na padajući brid signala RE moguće pročitati sadržaj statusnog registra (jedan bajt). Najniži bit statusnog registra govori je li prethodna operacija pisanja ili brisanja uspjela. Ukoliko je vrijednost bita „0“, operacija je uspjela. Ukoliko je vrijednost „1“ operacija nije uspjela i blok kojem se pristupalo može se smatrati nevaljanim.

Čitanje statusnog registra obavlja funkcija *FLASH\_Status\_Read*:

```
unsigned char FLASH_Status_Read(void)
{
    unsigned char Data = 0;

    GPIOSetValue(CS_PORT, CS_BIT, 0);
    Set_Port2_As_Output();

    //Command
    Port2_Write(0x70);
    GPIOSetValue(CLE_PORT, CLE_BIT, 1);
    GPIOSetValue(WE_PORT, WE_BIT, 0);
    GPIOSetValue(WE_PORT, WE_BIT, 1);
    GPIOSetValue(CLE_PORT, CLE_BIT, 0);

    Set_Port2_As_Input();
    GPIOSetValue(RE_PORT, RE_BIT, 0);
    Data = Port2_Read();
    GPIOSetValue(RE_PORT, RE_BIT, 1);

    GPIOSetValue(CS_PORT, CS_BIT, 1);

    return Data;
}
```

Funkcija ne prima parametre i vraća stanje statusnog registra FLASH memorije.

## 8.5 Slanje podataka preko Bluetooth veze

Već je rečeno kada se i kako pokreće prijenos podataka. Ostaje još opisati kako se podaci šalju. Taj dio obavlja funkcija *Send\_Log()*:

```
void Send_Log(void)
{
    unsigned int Blocks = 0;
    unsigned char Pages = 0;
    unsigned char Offset = 0;
    unsigned int i,j;

    //Ucitaj duljinu loga
    FLASH_Page_Read(0,0);

    Blocks = (FLASH_Page[0] << 8) | FLASH_Page[1];
    UARTSend((unsigned char *)&Blocks, 2);

    Pages = FLASH_Page[2];
    while(Command != SEND);
    Command = 0;
    UARTSend(&Pages, 1);

    for(i=START_BLOCK; i<(Blocks+START_BLOCK); i++)
    {
        for(j=0; j<30; j++)
        {
            while(FLASH_Check_Bad_Block(i+Offset))
            {
                Offset++;
            }

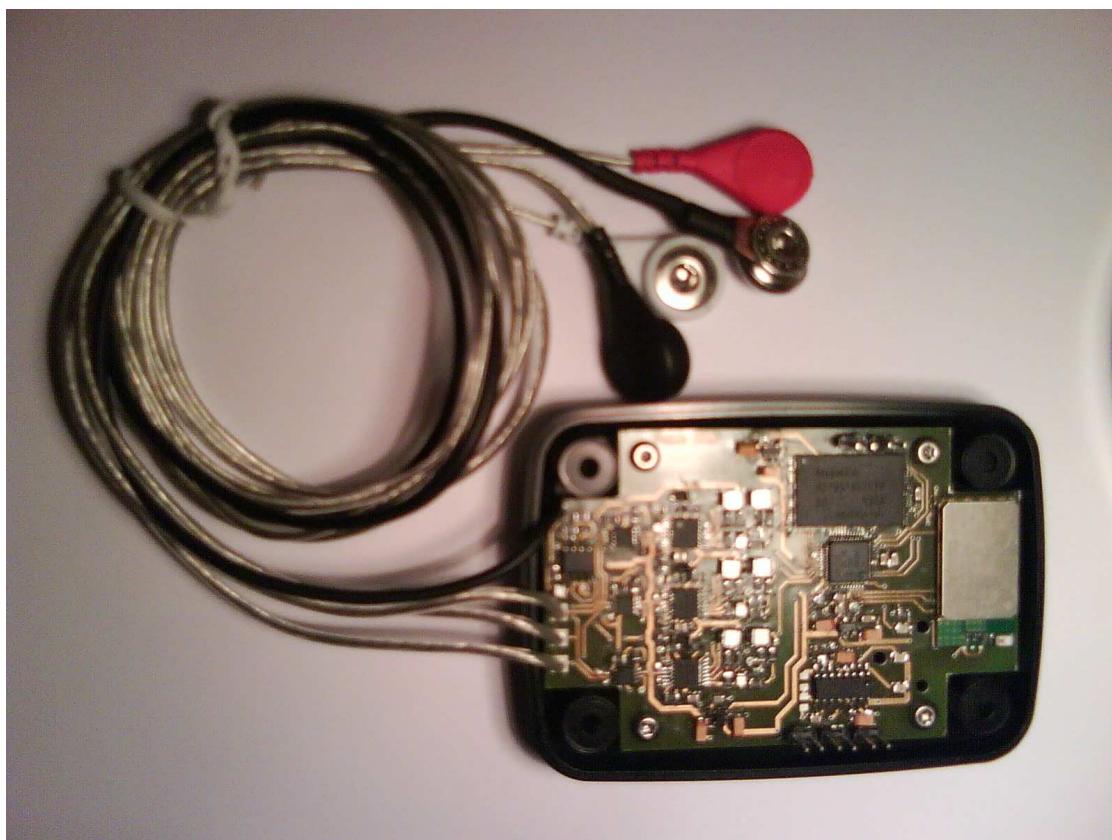
            while(Command != SEND);
            Command = 0;
            FLASH_Page_Read(i+Offset, j);
            UARTSend(&FLASH_Page[0], 512);
            if((i == (Blocks+START_BLOCK-1)) && (j == (Pages-1))) break;
        }
    }
}
```

Na početku se iz bloka 0, stranice 0 učitava duljina zapisa. Dujina je zapisana u obliku broja popunjениh blokova i stranica. Nakon toga se drugom uređaju šalju brojevi blokova i stranica. Nakon svakog slanja čeka se da drugi uređaj odgovori slanjem bajta 0x01.

Naposljetu se podaci šalju stranicu po stranicu tj. u blokovima od 512 bajta. Nakon svake poslane stranice drugi uređaj mora potvrditi primitak slanjem bajta 0x01. Pritom se, prije čitanja pojedinog bloka, provjerava je li ispravan i ukoliko nije, preskače ga se. U tom slučaju se uvećava varijabla *Offset* čime se osigurava da se pošalje dobar broj blokova (kad bi se umjesto varijable *Offset* uvećala varijabla *i* efektivno bi se poslao jedan blok manje). Slanje se prekida kada se pošalje zadnja stranica. Pritom valja primijetiti da posljednji blok ne mora biti pun do kraja, nego samo djelomično. Zbog toga se u zadnjoj iteraciji vanjske petlje provjerava je li poslana posljednja stranica i ako je, petlja se prekida (*break*).

## 9 Rezultati

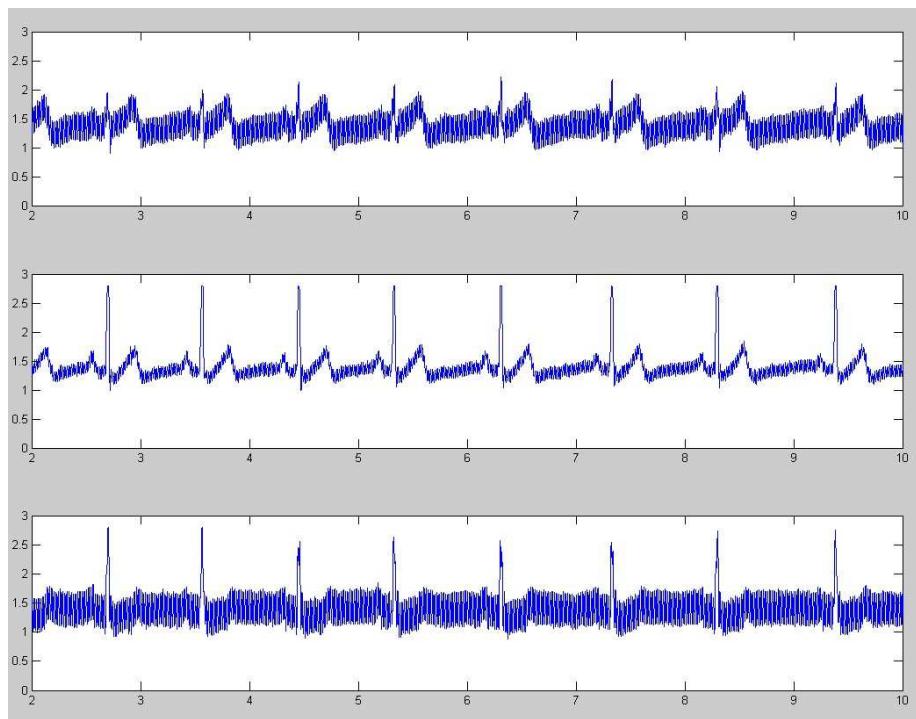
Uređaj je realiziran na jednoj dvoslojnoj tiskanoj pločici smještenoj u kutiju vanjskih dimenzija 88x63x27 mm (slika 25).



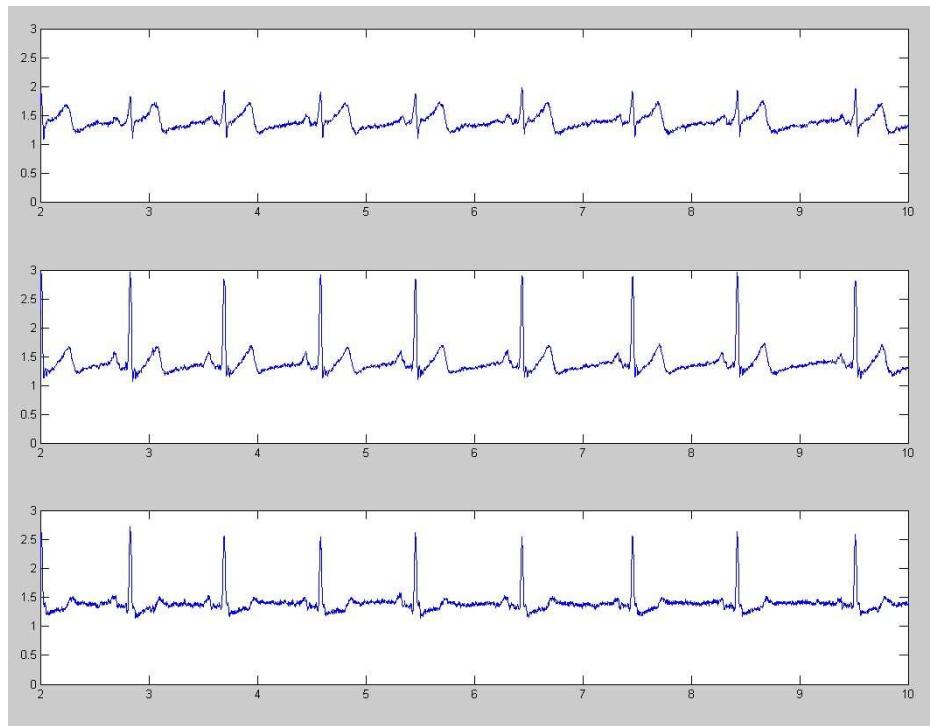
**Slika 25. – Izgled gotovog elektrokardiografa**

U svrhu isprobavanja funkcionalnosti uređaja napisana je *Matlab* skripta koja se preko virtualnog serijskog porta (ostvarenog Bluetooth vezom) povezuje s uređajem i preuzima s njega podatke. Rezultat skripte su tri grafa, svaki sa po jednim kanalom snimljenog EKG signala. Skripta sa signala dodatno uklanja smetnju izazvanu blizinom vodova gradske mreže (50 Hz) pomoću digitalnog FIR filtra. Dio signala u trajanju od nekoliko otkucaja srca prikazan je na slikama 26 i 27 (sa i bez korištenja digitalnog filtriranja).

Uređaj se napaja baterijom kapaciteta 750mAh koja je također smještena u kutiji. Uz izračunatu prosječnu potrošnju od oko 8mA uređaj može s jednim punjenjem baterije raditi preko 80 sati.



**Slika 26. - Primjer EKG signala snimljenog razvijenim elektrokardiografom  
(bez digitalnog filtriranja)**



**Slika 27. – Primjer EKG signala snimljenog razvijenim elektrokardiografom  
(uz digitalno filtriranje)**

## 10 Zaključak

U okviru ovog rada obradena je problematika registracije EKG signala. Identificirane su komponente prijenosnog, baterijski napajanog, bežičnog (Bluetooth) uređaja za registraciju EKG signala (elektrokardiografa). Za svaku komponentu su opisani zahtjevi i realizacija koja odgovara tim zahtjevima. Uređaj je realiziran na dvoslojnoj tiskanoj pločici i zajedno baterijskim napajanjem smješten u kutiju malih dimenzija. U kombinaciji sa sklopovljem izrađena je i ugrađena programska podrška za uređaj. Ukupni rezultat je baterijski napajan elektrokardiograf s tri kanala koji se uzorkuju frekvencijom od 1000Hz i vlastitom memorijom za pohranu do 6 sati signala. Rad uređaja isprobana je povezivanjem s osobnim računalom pomoću skripte napisane u programskom paketu Matlab. Rezultat skripte je grafički prikaz registriranog signala.

Mogućnosti povezivanja razvijenog elektrokardiografa su velike zbog raširenosti Bluetooth kompatibilnih uređaja. Jedini zahtjev na drugi uređaj je podržavanje *Serial port* Bluetooth profila (SPP). Ukoliko se razvijeni elektrokardiograf spoji s npr. mobitelom, moguće je razviti sustav za kontinuirano praćenje pacijenta, analizu EKG signala i upozoravanje hitne službe u slučaju potrebe. Moguće je i realizirati telemetrijski sustav koji bi liječniku po potrebi slao EKG signal pacijenta. Zbog toga je logični nastavak ovog rada izrada aplikacija za razne Bluetooth kompatibilne uređaje i za razne namjene.

## Literatura

- [1] Krois, I. Mjerni sustav za intraoperacijski nadzor neurofizioloških parametara. Doktorska disertacija. Fakultet elektrotehnike i računarstva, 2001.
- [2] Metting van Rijn A.C., Peper A., Grimbergen C. A. High-quality recording of bioelectric events. Part 1 interference reduction, theory and practice. Med.& Biol. Eng. & Comput., 28 (1990), 389-397.
- [3] Garcia-Niebla, J., Serra-Autonell, G. Effects of inadequate low-pass filter application. Journal of Electrocardiology, 42 (2009), 303-304.
- [4] NAND Flash Applications Design Guide, Ožujak 2004, <http://www.data-io.com/pdf/NAND/Toshiba/NandDesignGuide.pdf.pdf>, Travanj 2010.

## Sažetak

### **Trokanalni elektrokardiograf s memorijom i bežičnim prijenosom signala**

Rad se bavi problematikom registracije EKG signala i konstrukcijom prijenosnog, baterijski napajanog elektrokardiografa s bežičnim prijenosom prikupljenih signala prema drugim uređajima (Bluetooth). Opisane su komponente takvog uređaja, zahtjevi postavljeni na njih, kao i njihove realizacije. Izrađena je i ugrađena programska podrška za mikrokontroler koja obavlja zadaće prikupljanja podataka, njihovog spremanja u memoriju i prijenosa na druge bežične uređaje. Prikazan je primjer spajanja uređaja s osobnim računalom i vizualizacija prikupljenih podataka.

Ključne riječi: EKG, elektrokardiograf, Bluetooth

## Summary

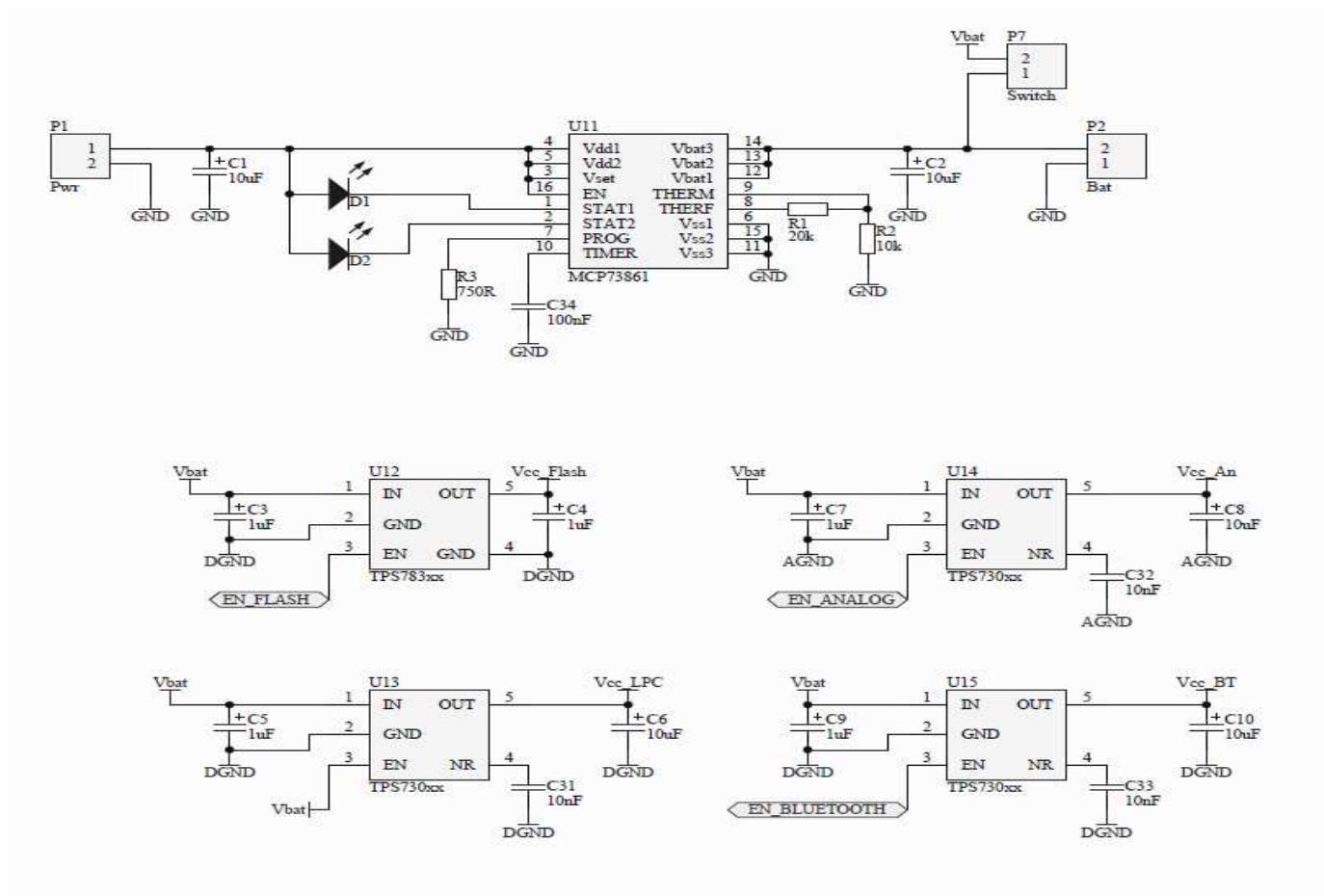
### **3-channel electrocardiograph with memory and wireless data transfer**

This thesis is based on identifying problems in ECG registration, as well as constructing a mobile, battery-powered electrocardiograph capable of wireless data transfer (Bluetooth) to other wireless capable devices. Main hardware components of such a device are identified and demands on their characteristics are discussed. Realization of each component, best suited for given application, is described. In addition to hardware design, firmware for a microcontroller embedded in the device is written and explained. Its main tasks are signal acquisition and storing in local memory as well as transfer of acquired data to other wireless devices. An example of interfacing the device to a personal computer is shown which results in visualization of acquired data.

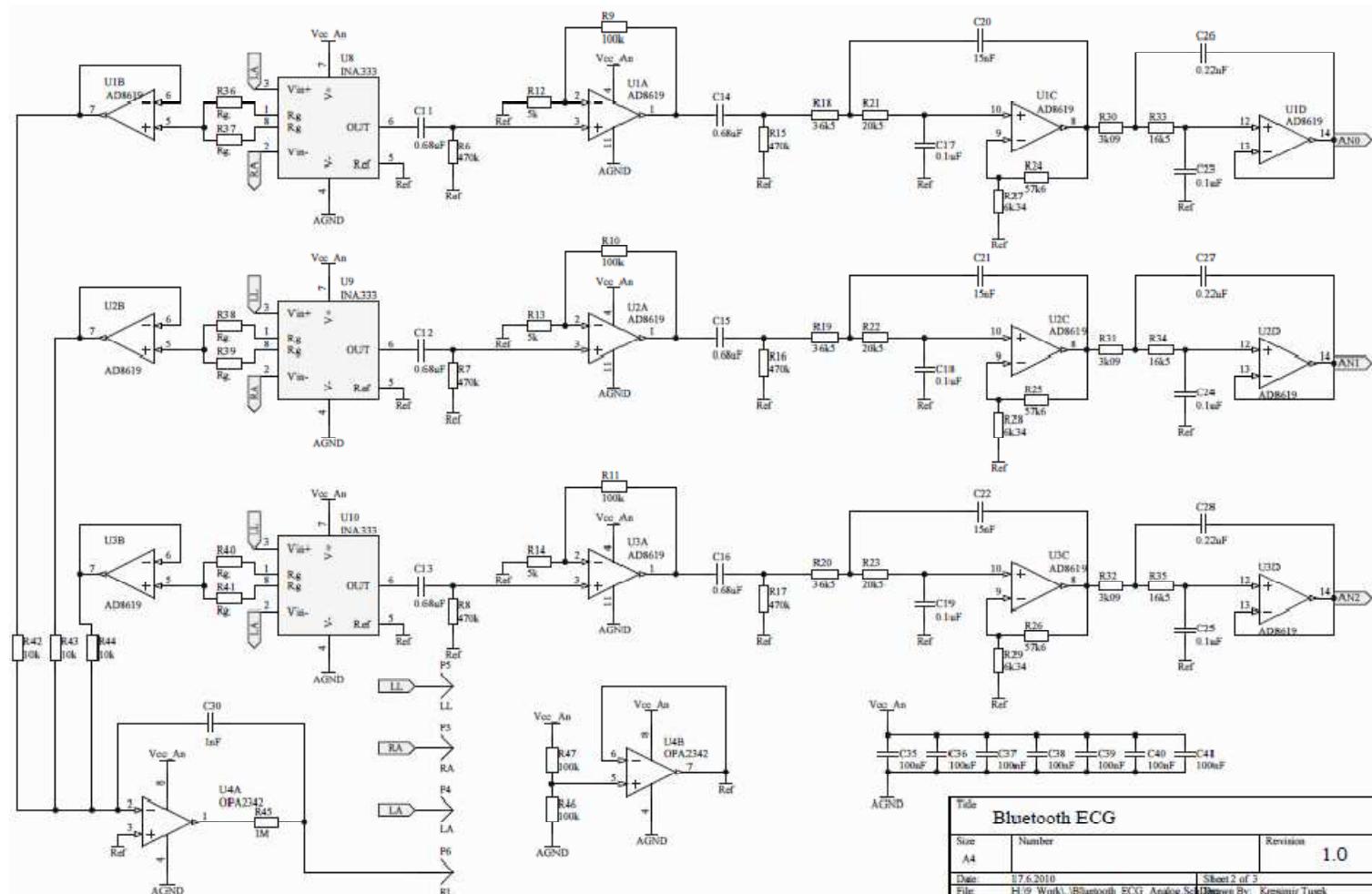
Keywords: ECG, electrocardiograph, Bluetooth

## A. Privitak: Tehnička dokumentacija

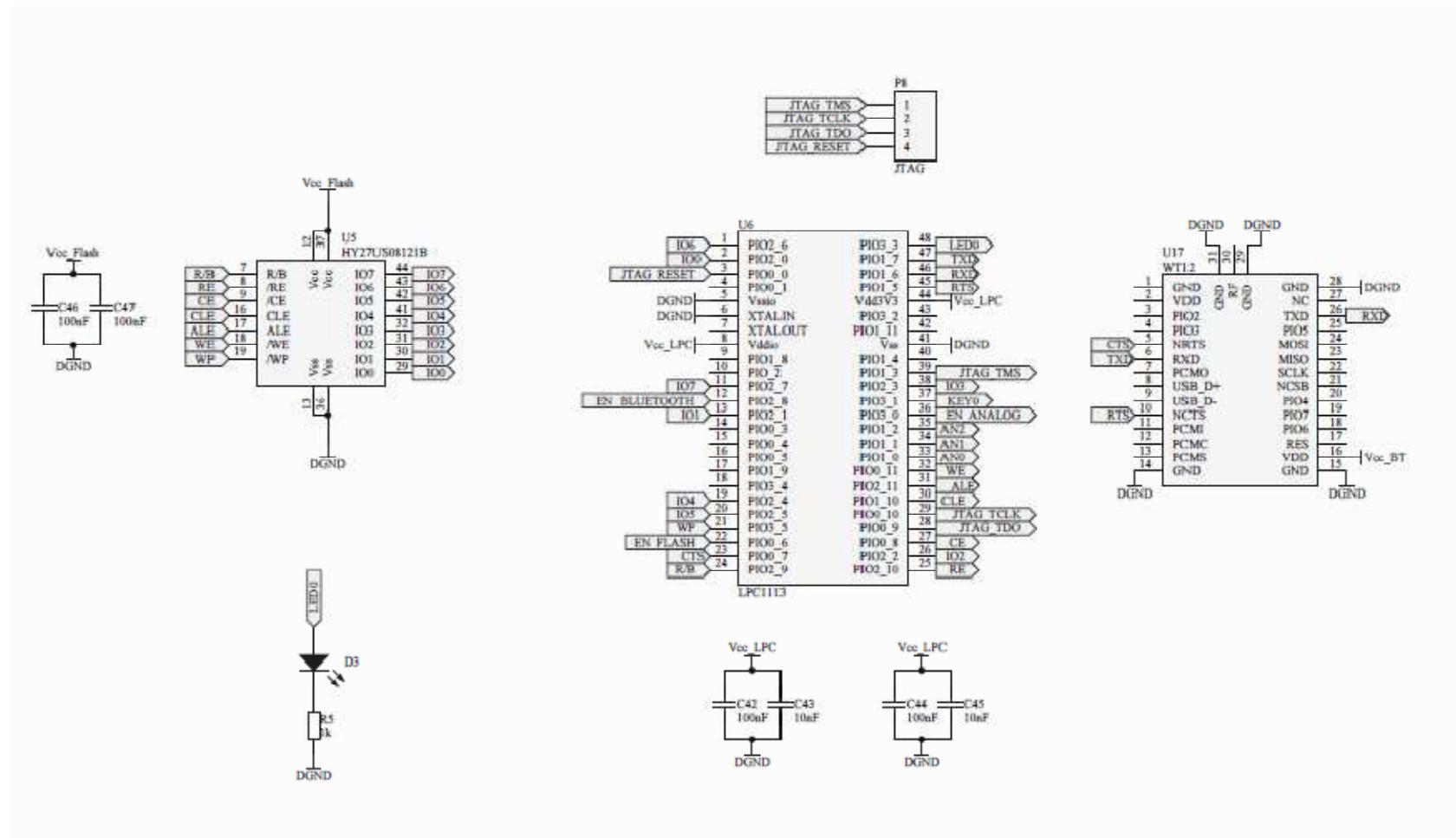
### A.1 Električka shema



Slika 28. – Shema napajanja

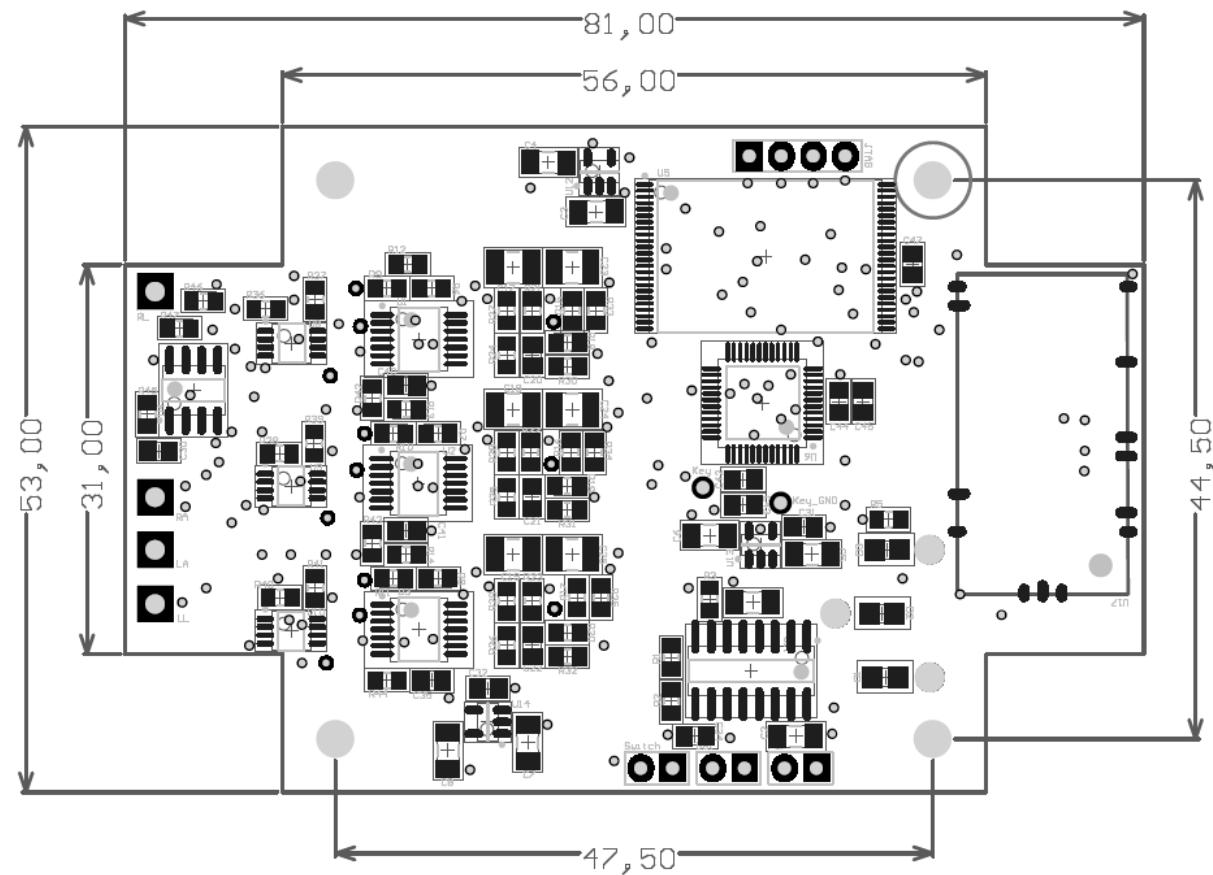


Slika 29. – Shema analognog dijela

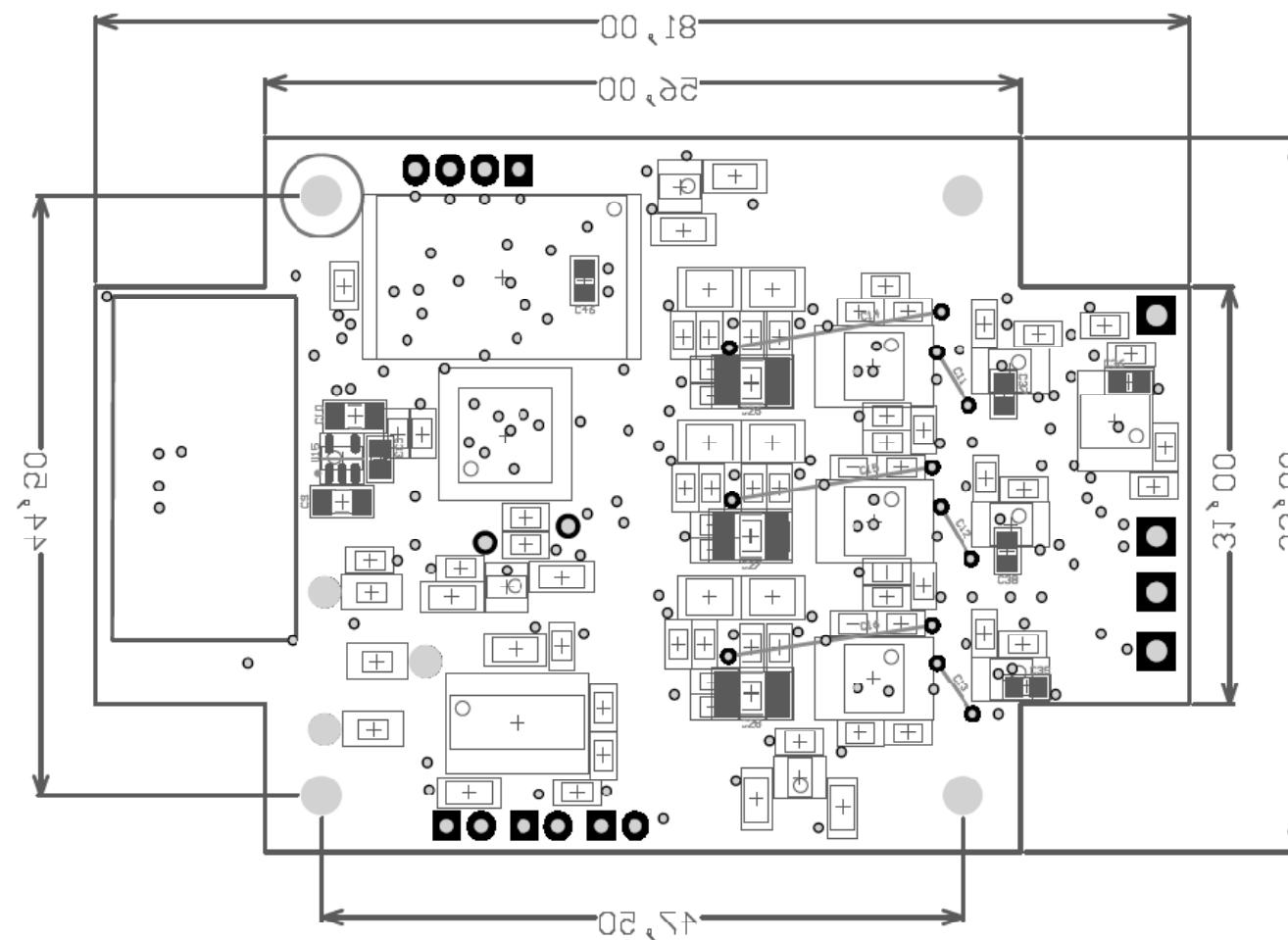


Slika 30. – Shema digitalnog dijela

## A.2 Položajni nacrt

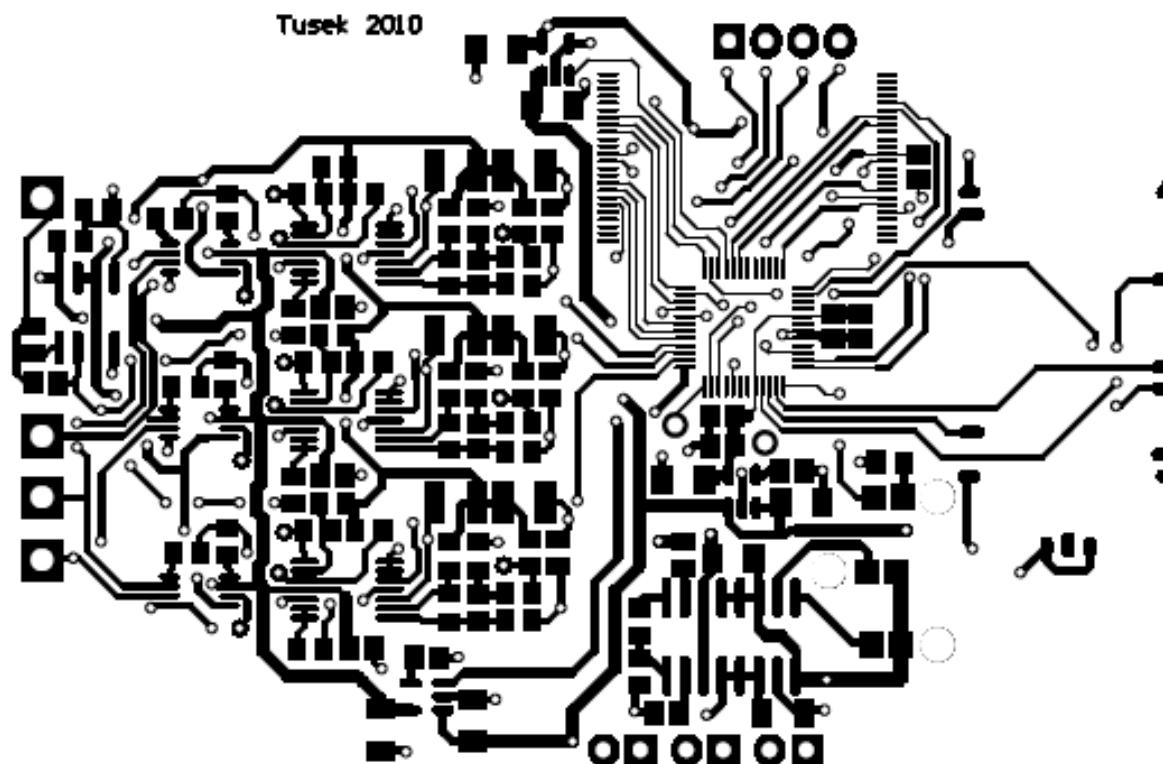


**Slika 31. – Položajni nacrt gornjeg sloja**

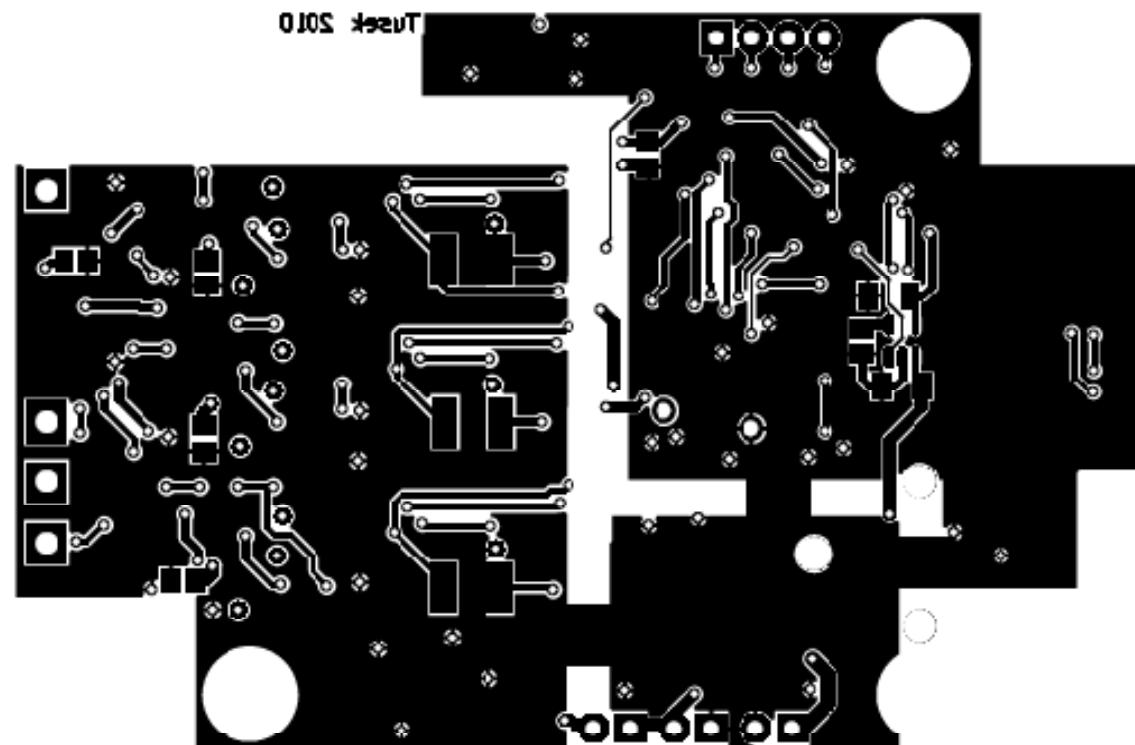


Slika 32. – Položajni nacrt donjeg sloja

### A.3 Tiskana pločica



Slika 33. – Izgled gornjeg sloja tiskane pločice



Slika 34. – Izgled donjeg sloja tiskane pločice