



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET

Grafitni klavir

Završni rad I ciklusa studija

Mentor:

Red. prof. dr. Melita Ahić-Đokić

Kandidat:

Bešić Almir

Sarajevo, juli 2016. g.

Tema: Grafitni klavir

Sažetak:

U okviru rada je potrebno dizajnirati i implementirati tzv. "grafitni" klavir sa najmanje jednom oktavom. Grafitni klavir je klavir kod kojeg se tipke crtaju grafitnom olovkom na papiru. Dodir nacrtane tipke rukom rezultira promjenom odgovarajućih električnih signala.

Potrebno je dizajnirati i implementirati hardver koji omogućava pojačavanje i interpretaciju odgovarajućih ulaznih signala, i vrši generisanje izlaznih signala (muzičkih tonova/nota).

Polazna literatura:

1. T. Brodić, Analogna integrirana elektronika, Sarajevo: Svjetlost, 1986.
2. B. P. Lathi, Modern Digital and Analog Communication systems, New York: Oxford University Press, 1998.
3. G. Daryanani, Principles of active network synthesis and design, New York: Bell Telephone Laboratories, 1976.
4. B. Carter i L. Huelsman, Handbook Of Operation Amplifier Active RC Networks, Texas Instruments, 2001.
5. M. McRoberts, Arduino starter kit manual, London: Earthshine Electronics, 2009.
6. M. Ahić-Đokić, Signali i sistemi, Sarajevo: Elektrotehnički fakultet, 2010.
7. A. V. Oppenheim i A. S. Willsky, Signals and systems, Prentice-Hall, 1997.
8. D. Lancaster, Active-Filter Cookbook, indianapolis: Howard W. Sams & Co. Inc, 1975.
9. M. Rafiquzzaman, Fundamentals of Digital Logic and Microcomputer Design, Hoboken: John Wiley & Sons, Inc., 2005.
10. D. A. Neamen, Microelectronics: Circuit Analysis and Design, New York: McGraw-Hill, 2010.

Potpis mentora:

Red. prof. dr Melita Ahić-Đokić

SADRŽAJ

Izjava o autentičnosti.....	5
1. Uvod	6
2. Zvuk i osnovni pojmovi u muzici	7
2.1. Karakteristike zvuka.....	7
2.2. Osnovni muzički pojmovi	8
3. Diskretizacija.....	10
3.1. Analogni i digitalni signal	10
3.2. A/D konverzija	11
3.3. Kvantizacija i uzorkovanje signala.....	13
4. Princip kapacitivne detekcije	15
4.1. Kapacitivni senzori i senzor blizine.....	15
4.2. Detekcija dodira korištenjem <i>Arduino</i> mikrokontrolera.....	16
5. Dizajn osnovnih elemenata grafitnog klavira.....	19
5.1. Izbor izvora napajanja	19
5.2. Generisanje nota	20
5.3. Testiranje interfejsa za SD karticu.....	21
5.4. Testiranje brzine D/A konverzije	22
5.5. Pojačalo zvuka	23
6. Popis korištenih komponenti, sheme i PCB dizajn	24
7. Finalni eksperimentalni rezultati	29
Zaključak	30
Prilog A	31
Bibliografija	37

POPIS SLIKA

Slika 2.1 Transverzalni talas	7
Slika 2.2 Longitudinalni talas.....	7
Slika 2.3 Ilustracija prostiranja zvuka u prostoru (talasni front)	8
Slika 2.4 Spektrogram note E4.....	8
Slika 2.5 C-skala u notnom zapisu	9
Slika 2.6 Korespondencija nota i njima odgovarajućih frekvencija.....	9
Slika 3.1 Analogni i ekvivalentni digitalni signal	10
Slika 3.2 A/D konvertor	11
Slika 3.3 Primjer A/D konverzije	12
Slika 4.1 Struktura i karakteristika prvog tipa senzora blizine.....	15
Slika 4.2 RC krug	16
Slika 4.3 Simulink model RC kruga.....	17
Slika 4.4 Odzivi RC kruga	17
Slika 4.5 RC krug u spoju sa Arduinom: a) prije dodira tipke b) nakon dodira tipke	18
Slika 4.6 Prikaz odziva na senzorskom pin-u (nakon dodira tipke vrijedi $n_2 > n_1$).....	18
Slika 5.1 Prikaz radnog područja mikroprocesora ATMEL ATmega 328p.....	19
Slika 5.2 Primjerak originalne note nakon uzorkovanja i note sa skaliranim amplitudom	20
Slika 5.3 Shema spajanja mikrokontrolera sa D/A konvertorom i modulom za SD karticu	22
Slika 5.4 Pojačalo zvuka	23
Slika 6.1 Dio sheme br. 1 (Arduino)	24
Slika 6.2 Dio sheme br. 2 (D/A konvertor)	25
Slika 6.3 Dio sheme br. 3 (punjač).....	26
Slika 6.4 Dio sheme br. 4 (pojačalo zvuka).....	27
Slika 6.5 Fizički izgled razvijene štampane pločice (dimenzije 100x160 mm)	28
Slika 6.6 Potpuna shema grafitnog klavira.....	28
Slika 7.1 Grafitni klavir.....	29

Izjava o autentičnosti

1. Uvod

Predmet istraživanja ovog završnog rada jeste način realizacije klavira čije su tipke nacrtane grafitnom olovkom na papiru. Dodirom grafitnih tipki vršit će se generisanje odgovarajućeg tona u skladu sa notama i njihovim korespondentnim frekvencijama. Cilj je postići što realniji model pravog klavira koristeći se *Arduino Nano* ugradbenim sistemom i principom kapacitivne detekcije. Potrebno je izvršiti analizu zvučnih signala te objasniti na koji način će mikrokontroler obrađivati te signale da bi se napravio što vjerodostojniji model klavira.

U prvom poglavlju se govori o osnovnim muzičkim konceptima. Da bi razumjeli određene pojmove, koji se pojavljuju u radu, potrebno je objasniti karakteristike zvuka kao i razliku između običnog i muzičkog signala. Govorit će se o notama i pojmu oktave.

U drugom poglavlju je objašnjen princip diskretizacije signala, konkretno konverzije iz analognog u digitalni signal (A/D konverzija). Obrnuti proces se naziva D/A konverzija. *Arduino* može vršiti samo 10-bitnu A/D konverziju, ali uz dodavanje određenih komponenti moguća je i 8-bitna D/A konverzija. Sa tim u vezi potrebno je objasniti princip zadrške i uzorkovanja (sempliranja) signala. Za ovaj završni rad koristit će se D/A konverzija kojom će biti omogućeno pretvaranje digitalnog signala, kojeg generiše mikrokontroler, u analogni signal.

Treće poglavlje je orijentisano ka objašnjavanju kapacitivne detekcije i mehanizama koji omogućavaju da se dodirom prsta mijenja kapacitet električnog kruga. Ovaj rad je utemeljen na detekciji promjene kapaciteta i taj princip se mora razumjeti kako bi bio jasan način rada grafitnog klavira.

U četvrtom poglavlju je predstavljen eksperimentalni dio rada. Sva istraživanja i testiranja, koja su doprinijela pojednostavljenju i rješavanju određenih problema, su objašnjena u ovom poglavlju.

Preposljednje poglavlje se odnosi na dizajn sheme i izradu štampane ploče. Uključena je lista komponenti uz kratko objašnjenje kako se pojedini dijelovi integrišu u funkcionalne cjeline. Shema je razdvojena na četiri dijela: *Arduino* mikrokontrolersko okruženje, D/A konvertor, punjač i pojačalo zvuka.

U posljednjem poglavlju su uključeni konačni eksperimentalni rezultati i zaključci nakon testiranja. Opisane su sve prednosti, nedostaci te mogućnosti unapređenja grafitnog klavira.

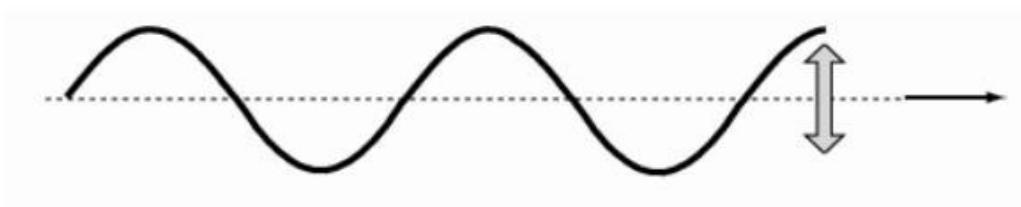
2. Zvuk i osnovni pojmovi u muzici

2.1. Karakteristike zvuka

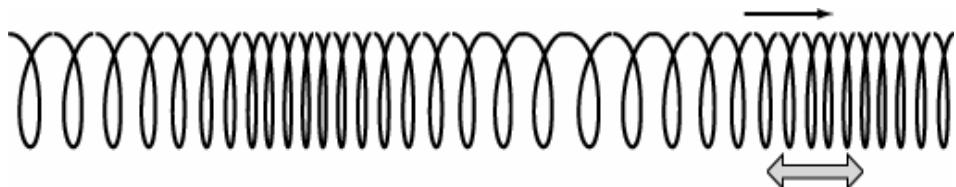
Prije navođenja definicije zvuka, potrebno je reći šta je talas. Talas se definiše kao poremećaj koji se širi kroz neki medij brzinom koja ovisi o karakteristikama tog medija. [1]

Postoje dvije vrste talasa:

- transverzalni – oscilovanje čestica medija i smjer kretanja talasa su okomiti,
- longitudinalni – oscilovanje čestica medija i kretanje talasa se odvija u istom smjeru.



Slika 2.1 Transverzalni talas



Slika 2.2 Longitudinalni talas

Zvuk je longitudinalni talas koji ima svoju frekvenciju i širi se kroz medij. Da bi nastao zvuk, tijelo koje ga proizvodi mora oscilovati i biti u direktnom kontaktu sa prenosnim medijem (zrak, voda, ...).

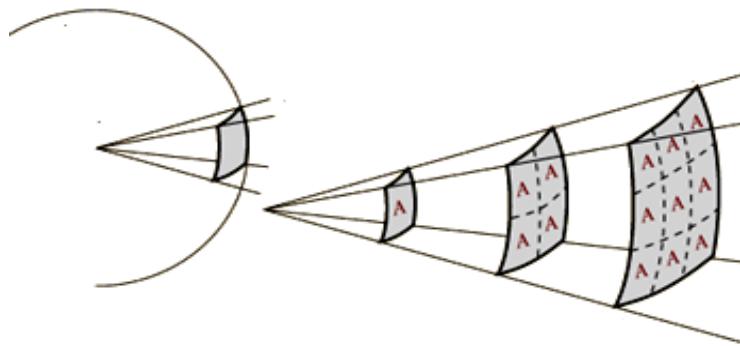
Zvučni talasi obuhvataju frekvencije od 16 Hz (donji prag čujnosti) do 20 kHz (gornji prag čulnosti). Zvuk frekvencije manje od 16 Hz naziva se infravez, dok se zvuk frekvencije više od 20 kHz naziva ultravez.

Dvije osnovne karakteristike zvuka su:

- Intenzitet zvuka [1] – energija koju prenese zvuk po jedinici površine talasnog fronta. Smanjuje se udaljavanjem od izvora (obrnuto proporcionalno površini talasnog fronta). Obično se prikazuje u logaritamskoj skali i računa po formuli:

$$I_{dB} = \log_{10} \left(\frac{I}{I_0} \right) [dB] \quad (2.1)$$

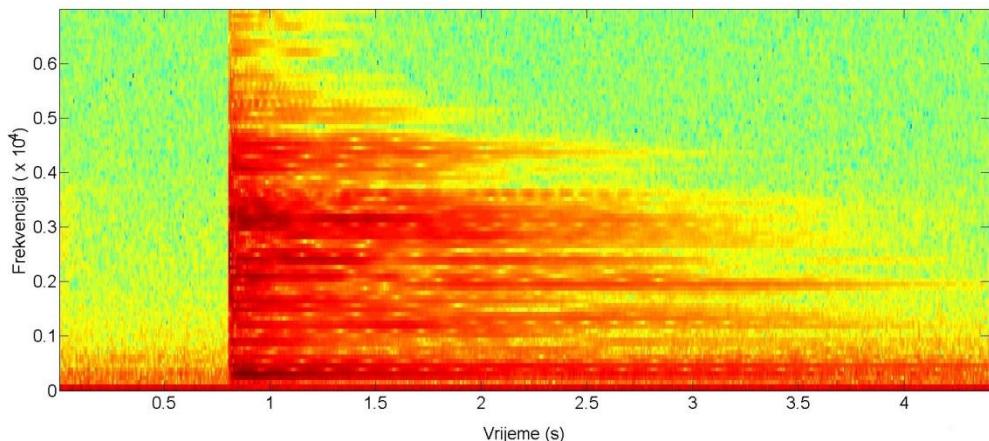
gdje je I trenutna jačina zvuka u $\left[\frac{W}{m^2} \right]$, a $I_0 = 10^{-12} \left[\frac{W}{m^2} \right]$ (ekvivalentno sa 0 [dB]) i predstavlja najmanji intenzitet zvuka koje ljudsko uho može registrovati.



Slika 2.3 Ilustracija prostiranja zvuka u prostoru (talasni front)

- Boja zvuka [1] – određena je djelovanjem viših harmonika koji se pojavljuju uz osnovni harmonik i različito se manifestuju u svakom zvučnom uzorku. Omogućava razlikovanje svih vrsta zvukova (ljudskih glasova, instrumenata, itd.).

Na slici 2.4. prikazan je spektrogram note E4 odsvirane na električnoj gitari. Vidimo da, pored osnovne frekvencije (približno 329 Hz), se javlja i mnoštvo drugih frekvencija koje se nazivaju višim harmonicima. Na spektrogramu se vidi da pojedini viši harmonici dosežu frekvencije od 6 kHz.



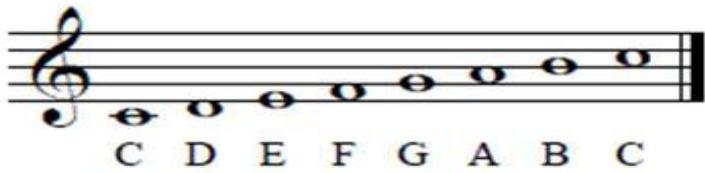
Slika 2.4 Spektrogram note E4

2.2. Osnovni muzički pojmovi

Muzika, za razliku od običnih zvučnih signala, se sastoji od pažljivo kombinovanih tonova u cilju stvaranja skladne i harmonične forme, koju će ljudski mozak percipirati kao privlačnu ili odbojnu, ovisno o subjektivnom mišljenju. Ono što treba znati jeste da se svaki signal ne može smatrati muzičkim. Postoje određena pravila kojih se treba pridržavati prilikom stvaranja muzike, koja će biti navedena u nastavku.

Da bi nastao muzički signal potrebne su note. Nota predstavlja zapis određene frekvencije iz spektra signala koje ljudsko uho može registrovati. Nizanjem nota od najmanje ka najvećoj (u

smislu frekvencije) nastaju skale. Postoje različite vrste skala, ovisno o tome kojom notom je započeta skala, tom istom notom se mora i završiti. Npr. C skala je prikazana na slici 2.5.

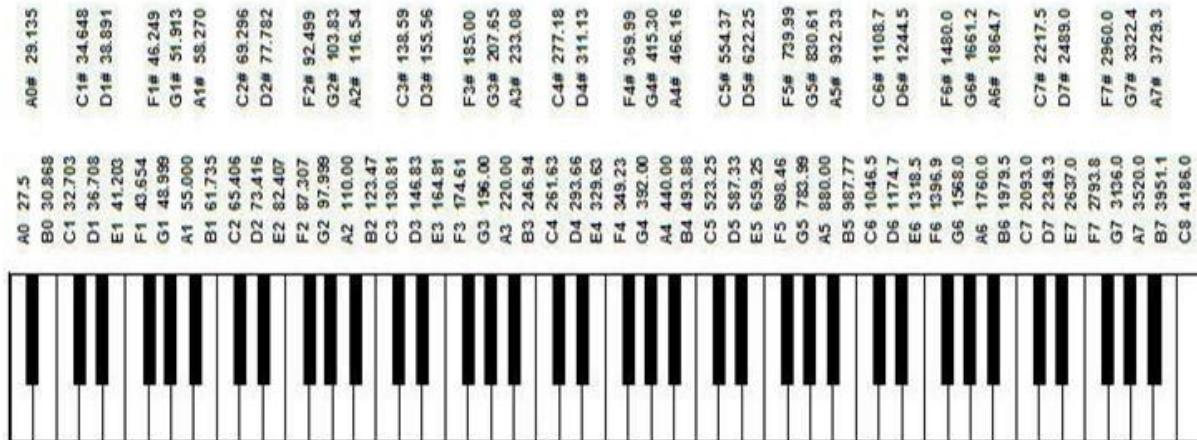


Slika 2.5 C-skala u notnom zapisu

Zanimljivo je da svaka od ovih nota ima odgovarajuću frekvenciju u spektru signala. Te frekvencije nisu nasumično odabrane, već postoji određena zakonitost po kojoj se vrši inkrementiranje.

Na slici 2.5. je prikazana jedna oktava. Mužičla oktava se može rastaviti na 12 jednakih koraka u logaritamskoj skali na takav način da se sačuva harmoničnost mužičkih signala.

Svaka nota je veća od prethodne za $2^{\frac{1}{12}} = 1.059463094$ puta (po frekvenciji) i taj interval je poznat kao polu-ton (polu-korak). Tako se note sortiraju od A0 pa do C8 (prikazano na slici 2.6.). [2]



Slika 2.6 Korespondencija nota i njima odgovarajućih frekvencija

Primijetimo da vrijedi:

$$f_{A0\#} = 1.059463094 \cdot f_{A0} = 1.059463094 \cdot 27.5 \text{ Hz} = 29.135 \text{ Hz}$$

i tako za svake dvije susjedne note.

Ova osobina može biti jako korisna prilikom stvaranja određenih tonova digitalnim putem. Moguće je generisati skale tonova koji se mogu kompjuterski obraditi. Na ovaj način se pravi štimjer, kojim se provjerava da li su ispravno podešeni neki instrumenti. Konkrentno, u vezi sa ovim završnim radom, mogu se izabratiti note potrebne za jednu oktavu grafitnog klavira. To

će biti oktava C3-C4 (vidi sliku 2.6.) jer predstavlja srednji dio opsega čiji tonovi ne zvuče previše „piskavo“ niti previše „duboko“. Frekvencije nota te oktave dovoljnu su niske da ih je moguće obraditi na *Arduino* ugradbenom sistemu.

3. Diskretizacija

3.1. Analogni i digitalni signal

Signal se definiše kao fizička veličina koja zavisi od jedne ili više nezavisno promjenjivih varijabli (često je to vrijeme) i nosi informaciju koja je sadržana u promjenama tog signala. Da bi se omogućio prenos, obrada ili čuvanje informacije, ona se mora pretvoriti u signal. Okvirno, signale možemo podijeliti u dvije grupe:

- 1) deterministički signali,
- 2) slučajni signali.

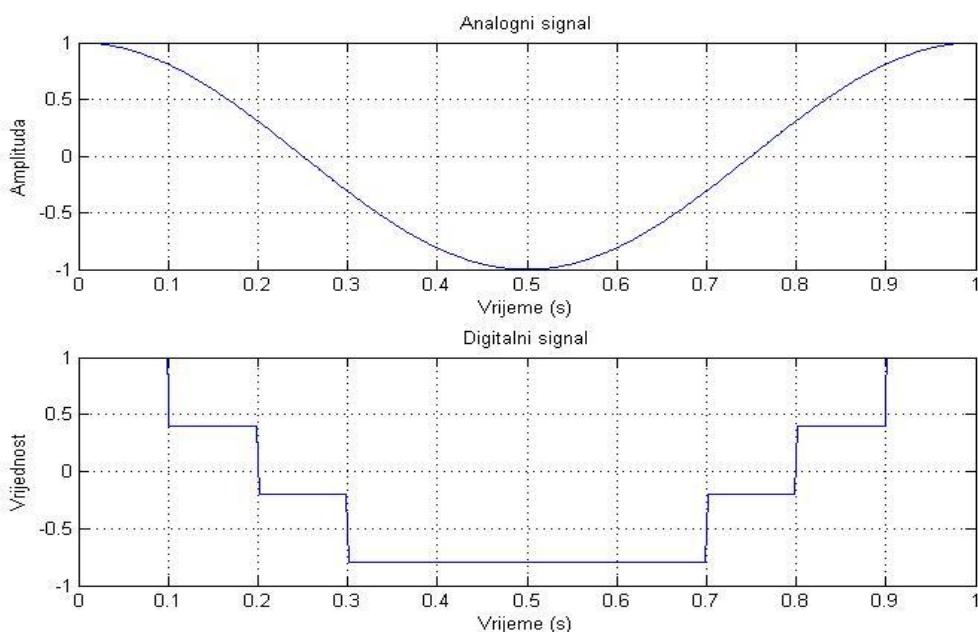
U prvu grupu spadaju signali čiji je fizički opis potpuno poznat, dok drugu grupu čine signali čija se buduća vrijednost ne može odrediti, već samo pretpostaviti [3]. Primjer prirodnog, determinističkog i jednodimenzionalnog signala, koji nosi informaciju, je muzička nota.

U cilju boljeg razumijevanja procesa diskretizacije, objasnit ćemo pojmove analognog i digitalnog signala.

- Signal čija je vrijednost definirana u svakom trenutku vremena, u određenom vremenskom intervalu, naziva se analogni signal.
- Digitalni signal je signal kod kojeg su amplituda i vrijeme diskretni, tj. ima definirane vrijednosti samo u određenim vremenskim trenucima.

Obje vrste signala imaju svoje prednosti i nedostatke, koje će biti analizirane u ovom radu.

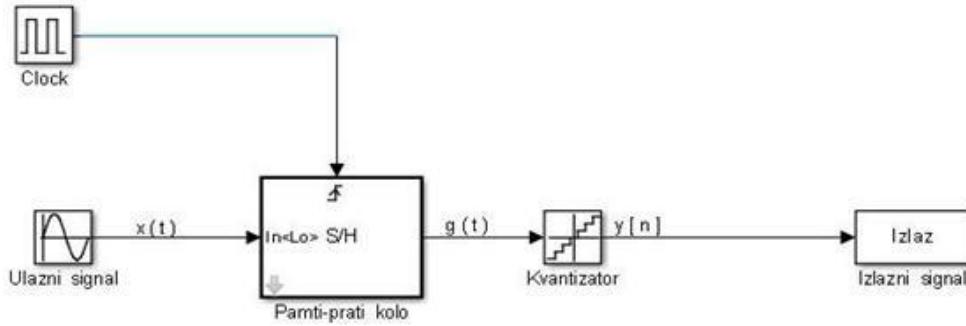
Dati su primjeri analognog i digitalnog signala na slici 3.1.



Slika 3.1 Analogni i ekvivalentni digitalni signal

3.2. A/D konverzija

Muzički tonovi su po prirodi analogni signali. Da bi ih bilo moguće računarski obraditi, potrebno je izvršiti diskretizaciju analognog signala. Na taj se način vrijednosti signala, najjednostavnije rečeno, pretvaraju u niz brojeva nad kojima računar može vršiti razne operacije. Ta konverzija se obavlja pomoću tzv. analogno-digitalnih konvertora (A/D konvertori). Struktura jednog takvog konvertora je data na slici 3.2.



Slika 3.2 A/D konvertor

Primijetimo da postoje dvije osnovne komponente:

- Pamti-prati (eng. /sample and hold/) kolo – uzorkuje signal u trenucima nT_s ($n \in Z$), gdje je T_s period clock signala, i vrši zadršku te vrijednosti sve do sljedećeg trenutka uzorkovanja.
- Kvantizator – aproksimira signal na izlazu pamti-prati kola, konačnim vrijednostima iz odabranog skupa. Aproksimiranje se može vršiti na više načina i to zaokruživanjem na viši, niži ili bliži nivo.

Neka je analogni signal (ulaz) označen sa $x(t)$. Tada će izlaz iz pamti-prati kola biti:

$$g(t) = \sum_{n=-\infty}^{+\infty} x(nT_s) \{u(t - nT_s) - u[t - (n + 1)T_s]\} \quad (3.1)$$

gdje je:

$$u(t - nT_s) - u[t - (n + 1)T_s] = \begin{cases} 1, & t \in [nT_s, (n + 1)T_s) \\ 0, & t \notin [nT_s, (n + 1)T_s) \end{cases} \quad (3.2)$$

a $u(t)$ jedinična step funkcija. [4]

Kvantizator će, na osnovu konačnog skupa mogućih diskretnih nivoa i zaokruživanjem na gornji ili donji nivo, aproksimirati vrijednosti funkcije $g(nT_s)$. Na taj način dobijamo izlaz $y[n]$ koji je digitalni signal (diskretiziran po amplitudi i po vremenu).

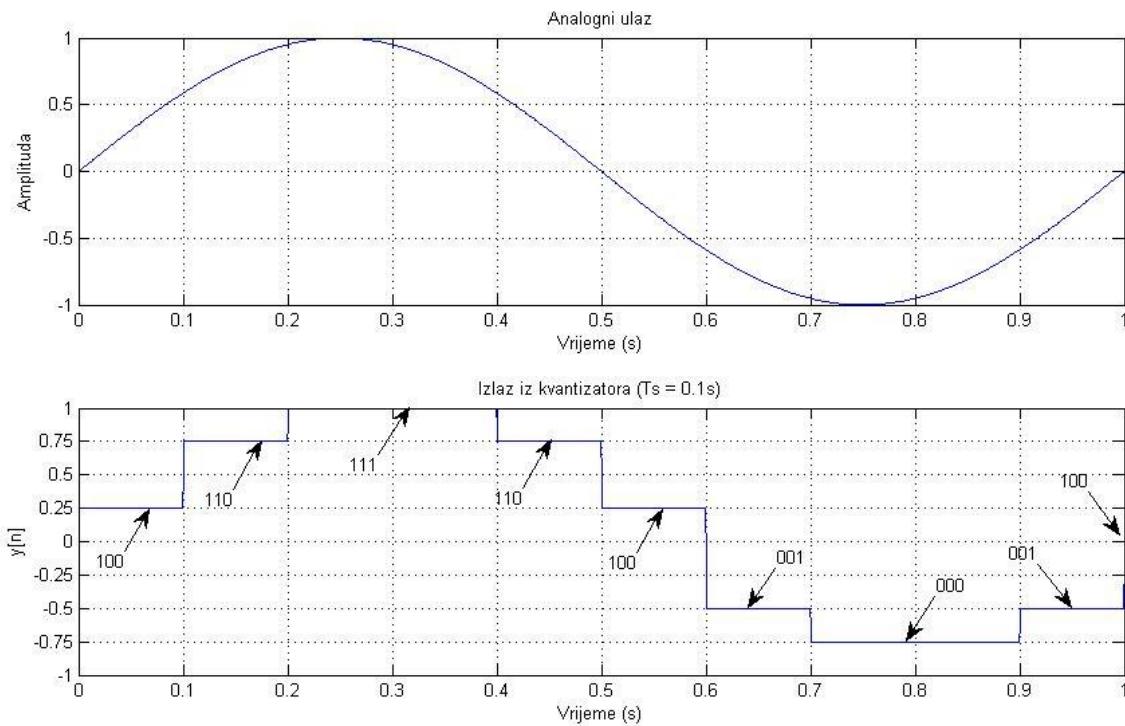
Pretpostavimo da je na ulaz u A/D konvertor doveden signal oblika $\sin(2\pi t)$. Neka se sempliriranje vrši svakih 100 ms i neka je konverzija 8-bitna. Korak diskretizacije će biti:

$$\frac{y_{MAX} - y_{MIN}}{\text{rezolucija}} = \frac{1 - (-1)}{8} = 0.25$$

Diskretne vrijednosti amplitude	Kodirane 8-bitne vrijednosti
1	111
0.75	110
0.5	101
0.25	100
0	011
-0.25	010
-0.5	001
-0.75	000

Tabela 3.1. Kodiranje diskretnih vrijednosti amplitude signala

Možemo primijetiti da mala rezolucija uzrokuje gubitak svih vrijednosti signala za amplitude od -0.75 do -1 . Zaokruživanje vrijednosti se vrši na gornji nivo. Rezultat A/D konverzije, koristeći već pomenute parametre, je prikazan na slici 3.3.



Slika 3.3 Primjer A/D konverzije

3.3. Kvantizacija i uzorkovanje signala

Da bi rezultat A/D konverzije bio ispravan, potrebno je pravilno kvantizirati i uzorkovati signal koji konvertujemo. U nastavku su objašnjeni osnovni principi pravilnog odabiranja parametara A/D konverzije.

Kvantizacija je uzorkovanje signala po amplitudi i vrši se pomoću već pomenutog kvantizatora. Da bi se signal što bolje rekonstruisao, potrebno je odabrati što veći broj kvantnih nivoa. Međutim, veliki broj kvantnih nivoa znači da je digitalna riječ, na izlazu kvantizatora, duža. To predstavlja ograničenje koje se odnosi na memoriju i u nekim slučajevima, gdje su memorijski resursi mali, treba biti vrlo pažljiv sa odabirom broja kvantizacionih nivoa. Potrebno je naći kompromis između željene kvalitete konverzije i memorije sa kojom raspolažemo. Dodatnu prepreku predstavlja i brzina konverzije, jer što imamo veći broj bita konverzija će biti sporija.

Uzorkovanje je kompleksnija operacija od kvantovanja. Teoretski, uzorkovanje se može objasniti kao množenje signala povorkom Diracovih impulsa $\delta_T(t)$ (češljasta funkcija).

$$\delta_T(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT_s) \quad (3.3)$$

gdje T_s predstavlja period povorke impulsa, a $\delta(t)$ Diracovu impulsnu funkciju za koju vrijedi:

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0 \end{cases} \quad (3.4)$$

$$\int_{-\infty}^{+\infty} \delta(t) dt = 1 \quad (3.5)$$

Množenjem signala $x(t)$ češljastom funkcijom dobija se tzv. zvjezdasta funkcija $x^*(t)$:

$$x^*(t) = x(t) \cdot \sum_{n=-\infty}^{+\infty} \delta(t - nT_s)$$

Koristeći osobinu (3.4) $x^*(t)$ postaje:

$$x^*(t) = \sum_{n=-\infty}^{+\infty} x(nT_s) \cdot \delta(t - nT_s) \quad (3.6)$$

Uzorkovanjem se gubi dio signala. Postavlje se pitanje kako uzorkovati signal a da se ne izgubi informacija koju on nosi?

Ako se signal brzo mijenja onda ga moramo često uzorkovati (manji period diskretizacije), a ako se signal sporo mijenja onda se može i sporije uzorkovati (veći period diskretizacije). Problem predstavlja činjenica da istu povorku impulsa može generisati beskonačno mnogo uzorkovanih signala. To se naziva *aliasing efekat* i javlja se kao posljedica niske frekvencije uzorkovanja. Taj fenomen se može izbjegći ako se ograniči spektar uzorkovanog signala.

Teorema (Nyquistova teorema o uzorkovanju): [1] Neka je $x(t)$ signal koji ima ograničen spektar tj. :

$$X(j\omega) = \begin{cases} X(j\omega), & |\omega| \leq \omega_M \\ 0, & |\omega| > \omega_M \end{cases} \quad (3.7)$$

Dovoljan uslov, da bi signal bio jedinstveno određen svojim uzorcima $x(nT_s)$ ($n \in Z$), je:

$$\omega_s \geq 2\omega_M \quad ili \quad f_s \geq 2f_M \quad (3.8)$$

gdje je:

$$\omega_s = \frac{2\pi}{T_s}$$

a f_M maksimalna frekvencija spektralno ograničenog signala.

Kod uzorkovanja muzičkih signala, frekvencije ne prelaze vrijednost od 20 kHz (prag čulnosti), pa se kao f_s obično uzima 44.1 kHz.

4. Princip kapacitivne detekcije

Princip kapacitivne detekcije je koristan u primjenama gdje je potrebno detektovati prisustvo ljudskog tijela. Ovaj fenomen će biti iskorišten za realizaciju grafitnih tipki klavira.

Kapacitivna detekcija se realizuje korištenjem kapacitivnih senzora koji će ukratko biti opisani u nastavku.

4.1. Kapacitivni senzori i senzor blizine

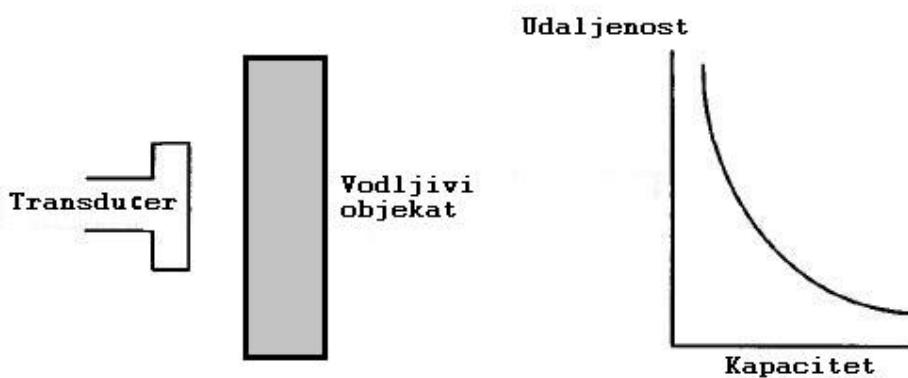
Kapacitivni senzori imaju široku primjenu u industriji i znanstvenim eksperimentima. Baziraju se na promjeni kapaciteta kao reakciji na vanjski fizički poremećaj. Postoji više vrsta ovih senzora za različite primjene. Neki od njih su:

- kapacitivni senzori sa promjenjivom udaljenošću ploča,
- kapacitivni senzori sa promjenjivom površinom ploča,
- kapacitivni senzori sa promjenjivim dielektrikom,
- diferencijalni kapacitivni senzori,
- kapacitivni akcelerometri,
- kapacitivni senzor blizine. [5]

Ovo poglavlje će se bazirati na pojašnjenju kapacitivnog senzora blizine.

Princip rada senzora blizine se zasniva na promjeni električnog polja koju izazivaju provodni ili dielektrični materijali. Postoje dva tipa ovih senzora.

Prvi tip koristi kondenzator sa paralelnim pločama. Jedna ploča predstavlja pretvarač energije (eng. /transducer/) dok je druga ploča objekat čiju blizinu mjerimo. Promjenom udaljenosti između ove dvije ploče mijenja se i ukupni kapacitet. Nedostatak ovakvog tipa senzora je što mjereni objekat mora biti napravljen od provodnog materijala da bi mogao činiti drugu ploču kondenzatora. Struktura i karakteristika jednog takvog senzora su prikazani na slici 4.1. [5]

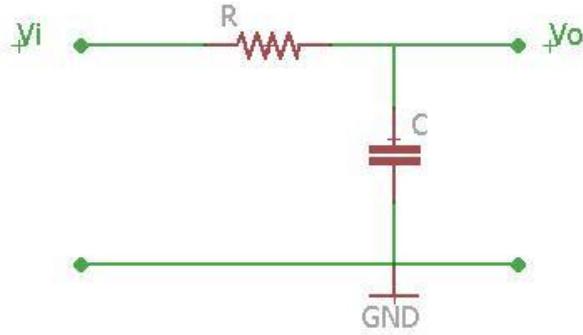


Slika 4.1 Struktura i karakteristika prvog tipa senzora blizine

Drugi tip senzora blizine se zasniva na prelaznom procesu RC kruga. Ovaj senzor ima dvije elektrode i samim time objekat, čiju udaljenost mjerimo, ne mora činiti jednu od elektroda. Dovoljno je da ima određenu kapacitivnost. Slična postavka je ugrađena na *Arduino* ugradbenom sistemu i može se iskoristiti za realizaciju tipki grafitnog klavira. Ovo će biti detaljnije objašnjeno u nastavku.

4.2. Detekcija dodira korištenjem *Arduino* mikrokontrolera

Prvo će biti objašnjen odziv jednostavnog RC kruga prikazanog na slici 4.2.



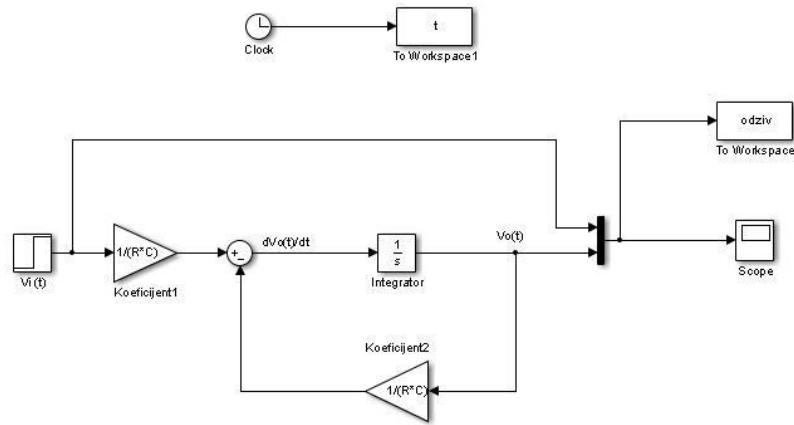
Slika 4.2 RC krug

gdje je $V_i(t)$ ulazni napon (pobuda) a $V_o(t)$ izlazni napon (odziv).

Da bismo saznali nešto više o odzivu ovog sistema, izvodi se diferencijalna jednačina koja ga opisuje:

$$V_i(t) = RC \frac{dV_o(t)}{dt} + V_o(t) \quad (4.1)$$

Jednačina 4.1 se može modelirati u MATLAB-u. Taj model će nam koristiti da simuliramo odziv koji je bitan za razumijevanje principa kapacitivne detekcije na *Arduinu*.



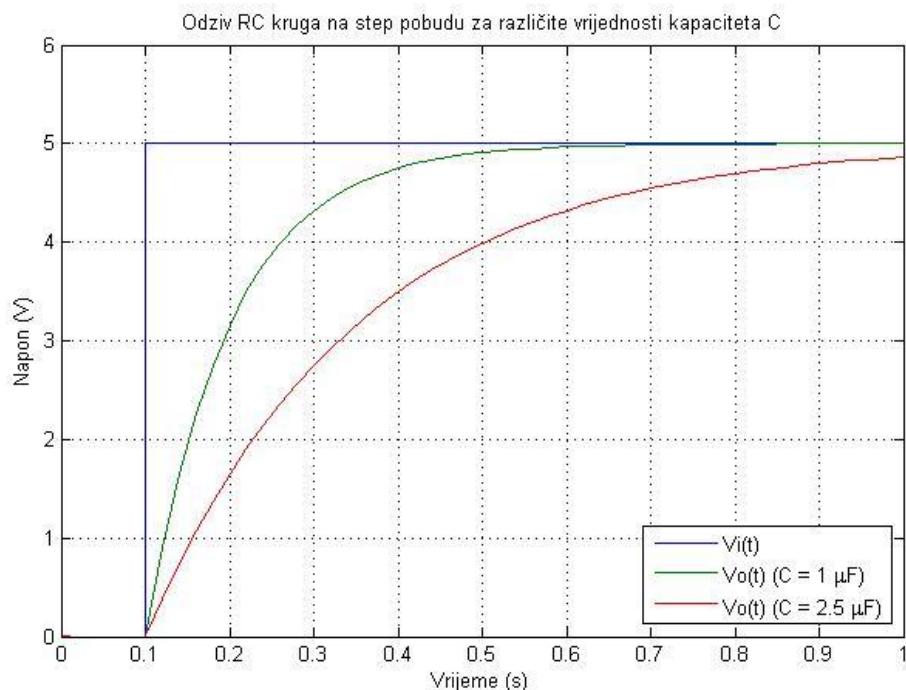
Slika 4.3 Simulink model RC kruga

Pobudu će predstavljati step amplitude 5V koji se javlja u 0.1s simulacije.

Za parametre se pretpostavljaju dva slučaja:

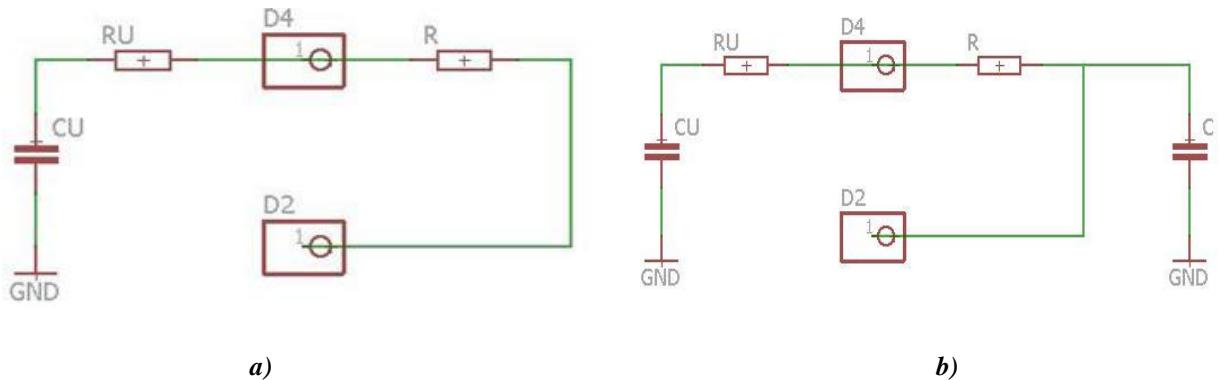
- 1) $C = 1 \mu F$ i $R = 100\ 000 \Omega$
- 2) $C = 2.5 \mu F$ i $R = 100\ 000 \Omega$

Pokretanjem simulacije dobiveni su odzivi prikazani na slici 4.4.



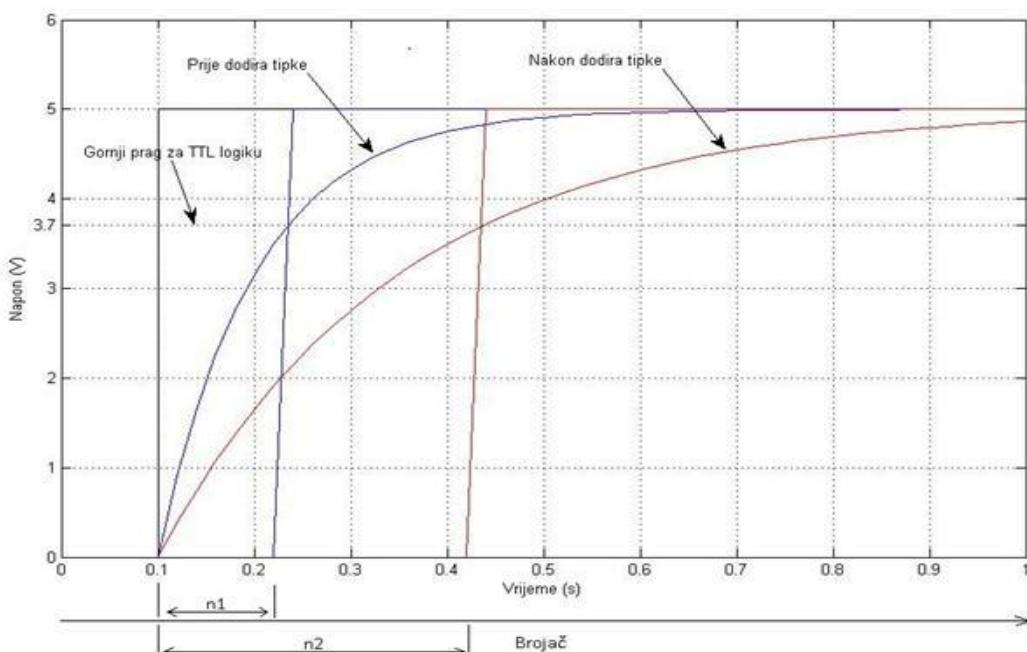
Slika 4.4 Odzivi RC kruga

Primijetimo da je sa povećanjem kapaciteta došlo i do povećanja vremenske konstante sistema. To je bilo očekivano jer je vremenska konstanta proporcionalna količini kapaciteta ($\tau = RC$). Zahvaljujući spomenutoj razlici u vremenskim konstantama napisana je biblioteka u *Arduino* programskom okruženju koja omogućava implementaciju kapacitivne detekcije i koja će biti korištena u ovom radu. Radi boljeg objašnjenja, napravljene su sheme prikazane na slici 4.5.



Slika 4.5 RC krug u spoju sa Arduinom: a) prije dodira tipke b) nakon dodira tipke

Digitalni pin D4 generiše step napon sa 0V na 5V. Digitalni pin D2 predstavlja senzorski pin koji računa vrijeme, potrebno da napon dostigne vrijednost logičke jedinice za TTL integracijsku logiku (približno 3.7 V), tako što inkrementira određenu varijablu unutar petlje (brojač). Ako prstom dodirnemo senzorski pin povećava se kapacitet RC kruga i samim time vremenska konstanta. Na osnovu toga mikrokontroler računa razliku inkrementiranih brojača i vraća tu vrijednost koja se može koristiti prilikom programiranja funkcionalnosti grafitnih tipki. [6]



Slika 4.6 Prikaz odziva na senzorskom pin-u (nakon dodira tipke vrijedi $n_2 > n_1$)

5. Dizajn osnovnih elemenata grafitnog klavira

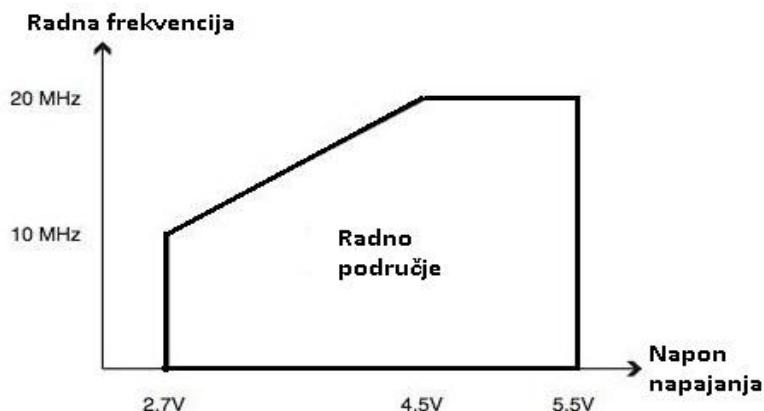
Testiranje pojedinih dijelova konačnog rješenja grafitnog klavira je vrlo bitno u svrhu jednostavnije identifikacije i lokalizacije mogućih problema. Da bi se napravila uspješna reprodukcija zvuka, prvo se moraju napraviti određene provjere koje će osigurati da model klavira bude što bolji.

Kao što je već pomenuto, centralni dio grafitnog klavira predstavlja *Arduino Nano* modul. Odabran je prvenstveno zbog jednostavnog načina upotrebe i programiranja. Podržan je velikim brojem biblioteka koje dodatno olakšavaju rad. *Nano* modul je vrlo kompaktan i lahko ga je smjestiti na štampanu ploču.

5.1. Izbor izvora napajanja

Cilj je da se sve elektroničke komponente na štampanoj ploči napajaju na osnovu baterijskog izvora od 3.7 V. Razlog tome je što se želi postići prenosivi sistem koji je neovisan o napajaju.

Iz tehničke dokumentacije mikroprocesora ATMEL ATmega328p (mikroprocesor koji je ugrađen na *Arduino Nano* modulu), izdvojena je sljedeća slika:



Slika 5.1 Prikaz radnog područja mikroprocesora ATMEL ATmega 328p

Potrebno je ispitati da li će frekvencija oscilatora biti dovoljna za ispravan rad mikrokontrolera na naponu od 3.7 V (poželjno je da ta vrijednost bude 16 MHz). Računa se jednačina prave prikazane na opsegu od 2.7 V do 4.5 V:

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

gdje su: $(x_1, y_1) = (2.7, 10)$ i $(x_2, y_2) = (4.5, 20)$

Jednačina prave: $y = 5.55 \cdot x - 4.985$ [MHz]

Ako umjesto varijable x uvrstimo željeni napon od 3.7V, kao rezultat se dobije frekvencija od 15.55 MHz što je povoljna brzina za svrhe ovog rada.

Pošto je napajanje baterijsko, u konačni dizajn sistema uključen je i punjač čiji princip rada neće biti detaljno objašnjen, jer to nije fokus ove teme. [7]

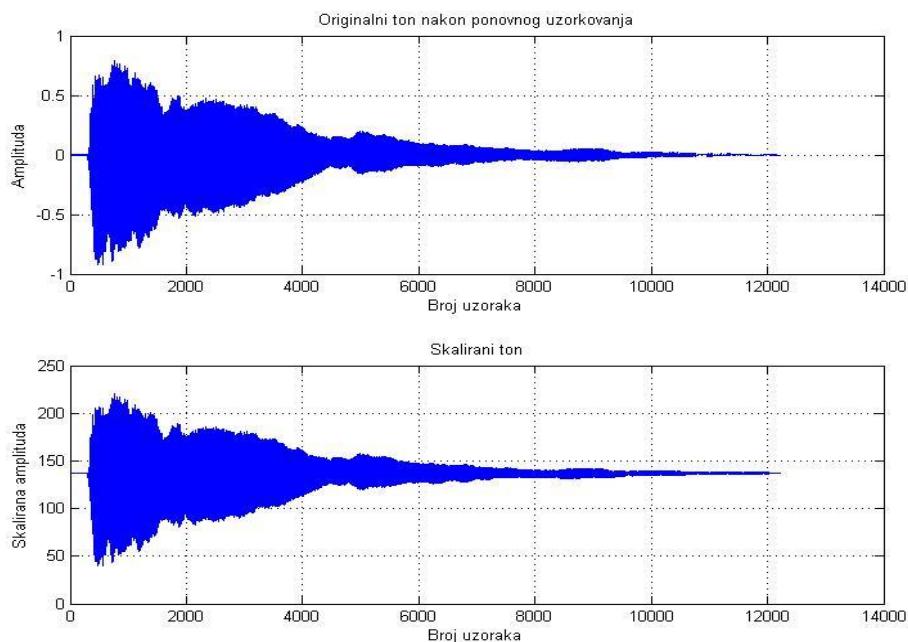
5.2. Generisanje nota

Potrebno je snimiti 8 nota (c, D, E, F, G, A, H, C) jedne od oktava. Mikrokontroler može generisati određene diskretne nivoe koje će 8-bitni D/A konvertor pretvoriti u analogni signal ekvivalentan snimljenim notama.

Prvo se snimaju note pravog klavira, jer na taj način je moguće sačuvati informaciju o boji zvuka. Snimljeni uzorci su u .wav (eng. /waveform audio file/) formatu, jer MATLAB omogućava njihovu jednostavnu obradu. Predstavlja valni oblik zvuka i zbog toga je vrlo praktičan za primjenu u ovom radu. Može se uzorkovati i lako konvertovati u oblik koji mikrokontroler procesira za dalje korištenje.

Snimljene note uzorkovane su frekvencijom od 48.000 Hz. Za *Arduino* ugradbeni sistem takva frekvencija je previsoka za reprodukciju. Zbog toga se vrši ponovno uzorkovanje snimaka frekvencijom od 12 kHz (vidjeti poglavlje 5.4. za razlog izbora ove frekvencije). Time nije narušen Nyquistov uslov naveden u poglavljiju 3.3.

Vrijednosti amplituda originalnih nota .wav formata se nalaze u opsegu od -1 do 1, zbog čega bi D/A konvertor morao koristiti float vrijednosti. Nad takvim vrijednostima je nemoguće vršiti konverziju, jer je D/A konvertor 8-bitni i prepoznaje samo cijelobrojne vrijednosti od 0 do 255. Zato se amplituda svakog uzorka skalira na opseg od 40 do 220. Ne uzima se cijeli opseg D/A konvertora da bi se izbjegao prelazak signala u zasićenje (iznad 3.7V). Nakon skaliranja, dobivene vrijednosti upisujemo kao ASCII karaktere u .txt file. ASCII kodiranje je pogodno zato što cijeli broj pohranjuje kao jedan karakter. To nam omogućava jednostavno očitavanja karaktera iz tekstualnih datoteka i njihovu konverziju u cijelobrojnu vrijednost.



Slika 5.2 Primjerak originalne note nakon uzorkovanja i note sa skaliranim amplitudom

Nakon obavljenog eksperimenta može se zaključiti da je kvaliteta novog zvučnog uzorka lošija zbog frekvencije uzorkovanja koja je četiri puta niža od originalne. Veći broj uzoraka bi se mogao postići korištenjem kvalitetnijeg mikrokontrolera, ali za svrhe ovog završnog rada izloženo rješenje je zadovoljavajuće.

Za generisanje spomenutih tekstualnih datoteka, korišten je MATLAB kôd dat u okviru priloga (A.1)

5.3. Testiranje interfejsa za SD karticu

Da bi očitavanje nota iz tekstualnih datoteka bilo efikasno, brzina ekstrakcije podataka sa SD kartice mora biti dovoljno visoka. Pošto SD imaju nisku brzinu očitavanja, neophodno je izvršiti eksperiment koji je opisan u nastavku.

U *.txt* file, koristeći MATLAB, upisano je 100.000 nula u jednoj koloni. Taj file se prebacuje na SD karticu gdje će biti iskorišten za čitanje. Važno je napomenuti da se u samom file-u, pored 100.000 *nula*, nalazi još 100.000 znakova za novi red i 100.000 znakova za bit prenosa koji su skriveni. To znači da ukupno postoji 300.000 karaktera. Kôd, koji omogućava testiranje brzine očitavanja, dat je u prilogu (A.2).

Nakon izvršenja programa ispisuje se vrijeme očitavanja svih 300.000 karaktera. Program prikazuje da to vrijeme iznosi 13.03 s. Frekvencija očitavanja se računa kao:

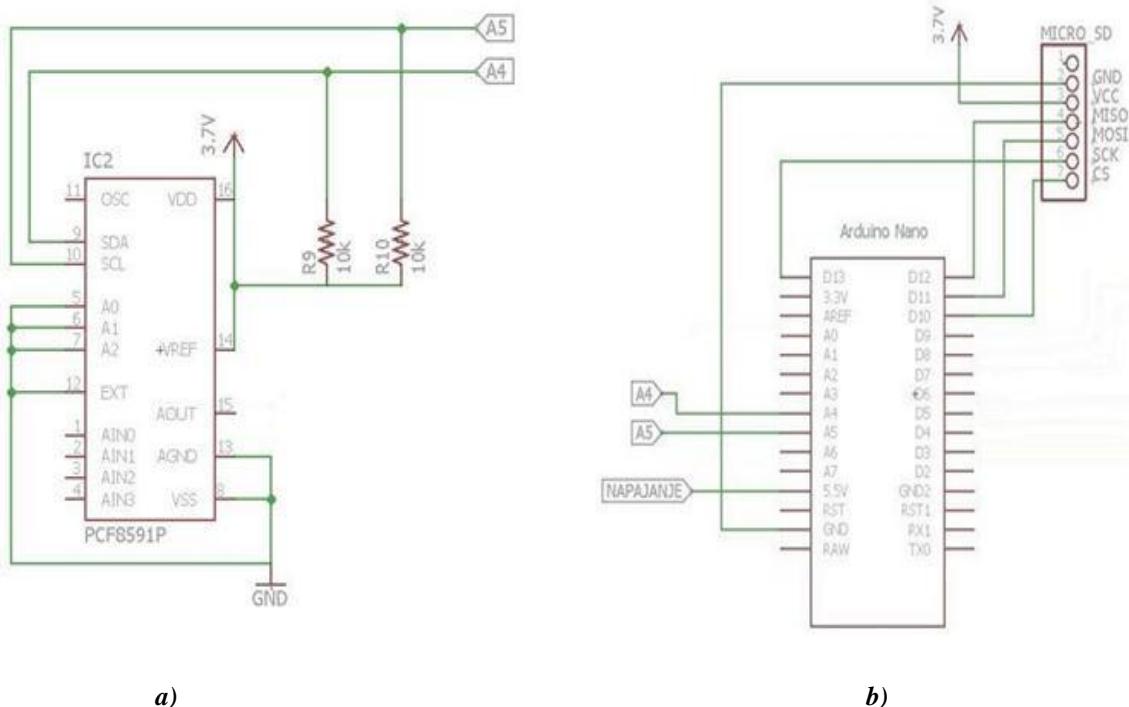
$$f_{\text{očitavanja}} = \frac{300.000}{13.03\text{s}} = 23.023\text{ kHz}$$

Time je pokazano da mikrokontroler može očitati približno 23.000 karaktera po sekundi što je zadovoljavajuća brzina obzirom da su note uzorkovane brzinom od 12.000 Hz. To znači da je kartica u mogućnosti da čita 12.000 ASCII karaktera u jednoj sekundi.

Shema spajanja je prikazana na slici 5.3. b).

5.4. Testiranje brzine D/A konverzije

Do sada su svi uvjeti za kvalitetno reproduciranje zvuka ispunjeni. Preostaje još ispitati brzinu D/A konverzije tj. koliko brzo možemo mijenjati stanje digitalnog izlaza. To radimo tako što ćemo generisati četvrtasti signal na osnovu kojeg će D/A konvertor proslijediti vrijednosti logičke nule i logičke jedinice na izlaz. Shema spajanja je sljedeća:



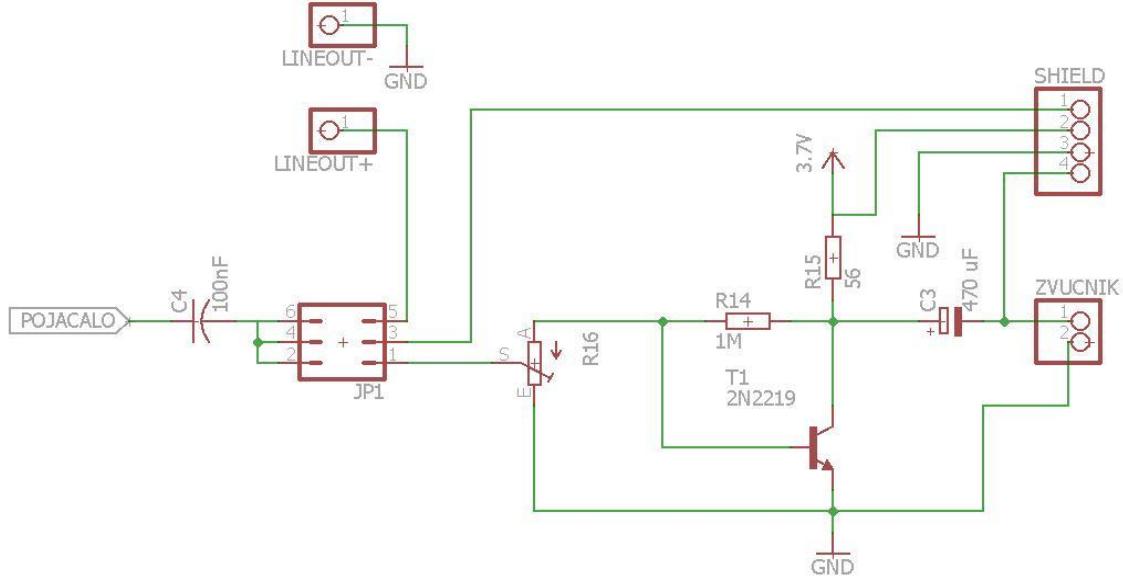
Slika 5.3 Shema spajanja mikrokontrolera sa D/A konvertorom i modulom za SD karticu

Na AOUT pin D/A konvertora se spaja osciloskop i mjeri se vrijeme promjene između nivoa logičke nule i logičke jedinice. Izmjereno vrijeme je 0.15 ms što znači da se može postići maksimalna frekvencija od približno 6.600 Hz . Ubrzavanjem clock impulsa u postavkama procesora moguće je postići frekvenciju od 13.000 Hz . Očitavanje i konverzija nota, pohranjenih na SD kartici, se odvija nesmetano, jer su uzorkovane frekvencijom od 12.000 Hz .

Kôd za generisanje četvrtastog signala dat je u prilogu (A.3).

5.5. Pojačalo zvuka

Spajanje niskoomskog zvučnika direktno na izlaz D/A konvertora rezultiralo bi slabim intenzitetom zvuka. Potrebno je pojačati signal dizajniranjem odgovarajućeg pojačala. Uključivanjem potenciometra u shemu postižemo efekat smanjivanja i pojačavanja zvuka.



Slika 5.4 Pojačalo zvuka

Na slici 5.4. je prikazano pojačalo koje će biti integrисано на elektronској плочи. Ако се појави потреба за коришћењем другог појачала, направљени су изводи за уградњу додатне штампане плаће (SHIELD). Када је потребно, изводи за слушалке (LINEOUT+ и LINEOUT-). Режим рада се mijenja једноставним пребациванијем преспојника (енг. /jumpera/) на одговарајуће место (JP1).

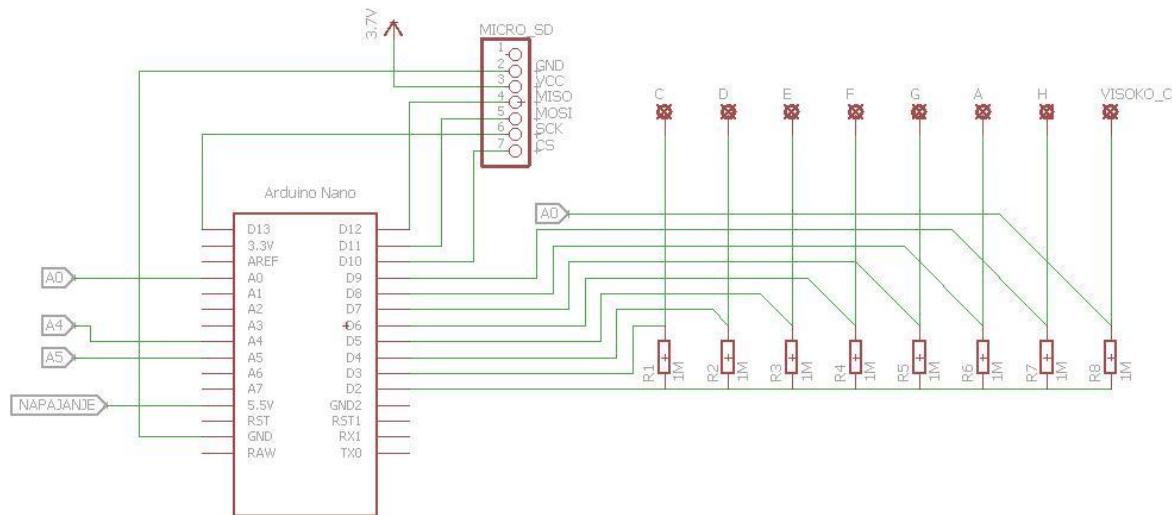
6. Popis korištenih komponenti, sheme i PCB dizajn

1) Arduino razvojno okruženje:

- Arduino Nano (x 1),
- Modul za čitanje SD kartice (x 1),
- Otpornici ($8 \times 1 \text{ M}\Omega$),
- Female pin headeri.

Dodatne napomene:

- Modul za SD karticu se spaja na odgovarajuće pin-ove Arduina (MISO-D12, MOSI-D11, SCK-D13, CS-D10).
- Otpornici su spojeni kao što je to objašnjeno u poglavlju 4. da bi realizovali tipke klavira.



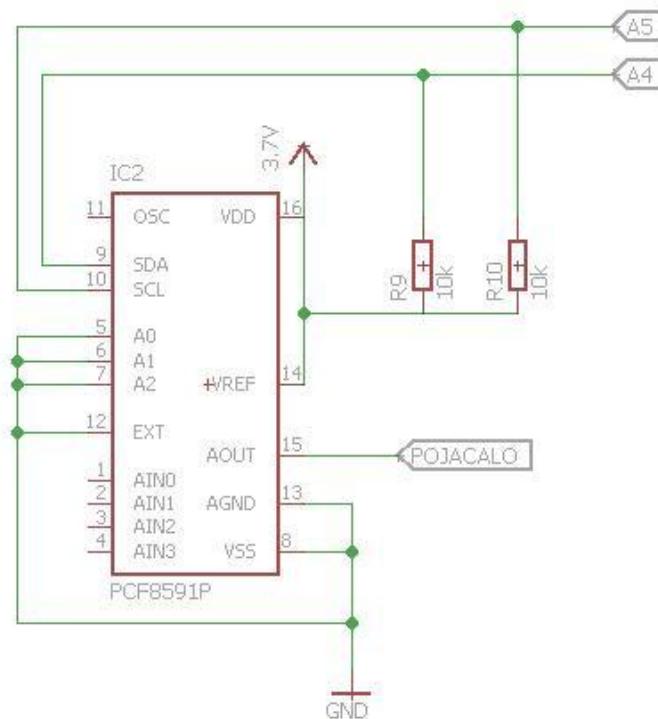
Slika 6.1 Dio sheme br. 1 (Arduino)

2) D/A konvertor:

- 8-bitni D/A konvertor PCF8591P (x 1),
- Otpornici ($2 \times 10 \text{ k}\Omega$).

Dodatne napomene:

- PCF8591P je CMOS uređaj koji se bazira na I2C komunikaciji. Ima 4 analogna ulaza i jedan analogni izlaz. Tri adresna pin-a omogućavaju priključivanje 8 dodatnih uređaja na I2C interfejs. I2C komunikacija je dvolinijska i dvosmjerna. Funkcije ovog uređaja predstavljaju multipleksiranje ulaza, zadrška te 8-bitna A/D i D/A konverzija.
- SDA i SCL pin-ovi D/A konvertora su spojeni na analogne pin-ove A4 i A5 Arduina da bi se omogućila D/A konverzija.
- AOUT pin D/A konvertora se spaja na pojačalo zvuka radi dobivanja kvalitetnijeg signala.



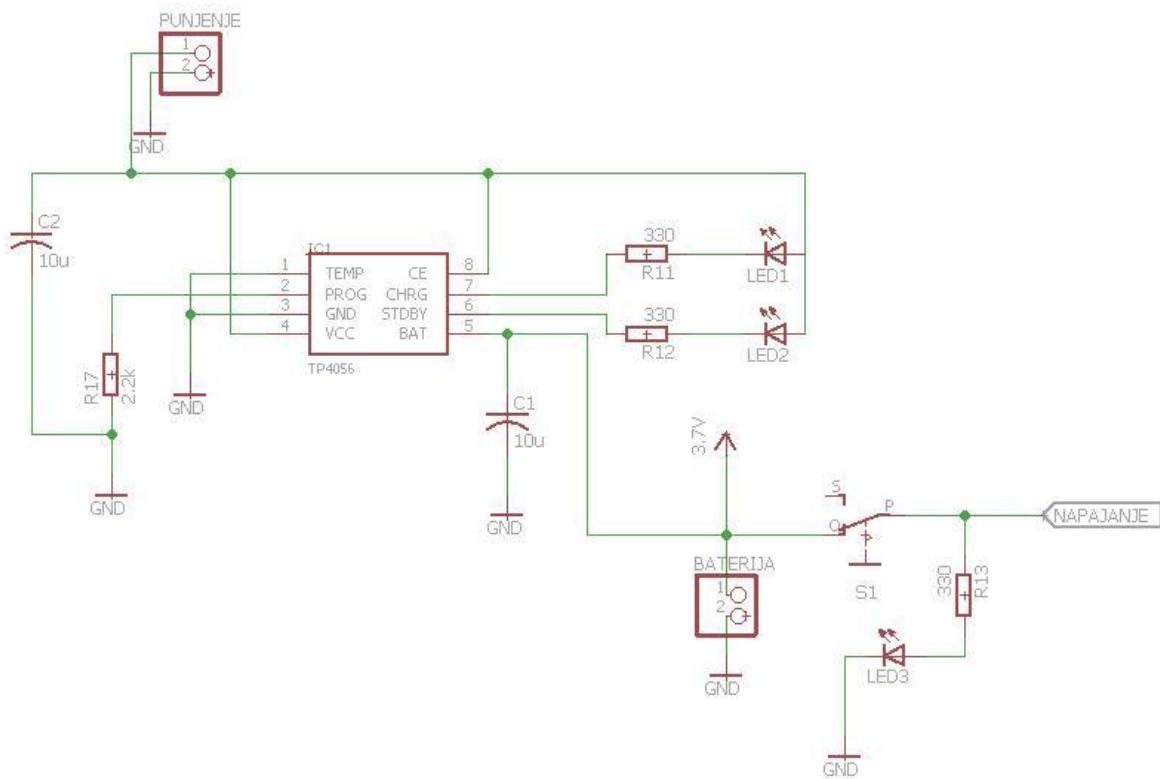
Slika 6.2 Dio sheme br. 2 (D/A konvertor)

3) Punjač:

- TP4056 linearni punjač konstantne struje i napona za Li-Ion baterije (x 1),
 - Otpornici (3 x 330 Ω , 1 x 2k2 Ω),
 - Kondenzatori (2 x 10 μF),
 - LED (x 3),
 - Li-Ion baterija (x 1),
 - Prekidač,
 - Female pin headeri.

Dodatne napomene:

- Električni krug koji napaja cijelu pločicu.
 - Ugrađen prekidač za prekid napajanja.
 - LED diode indiciraju trenutna stanja kruga (ZELENA - napajanje uključeno, ŽUTA - baterija je napunjena, CRVENA - baterija se puni).



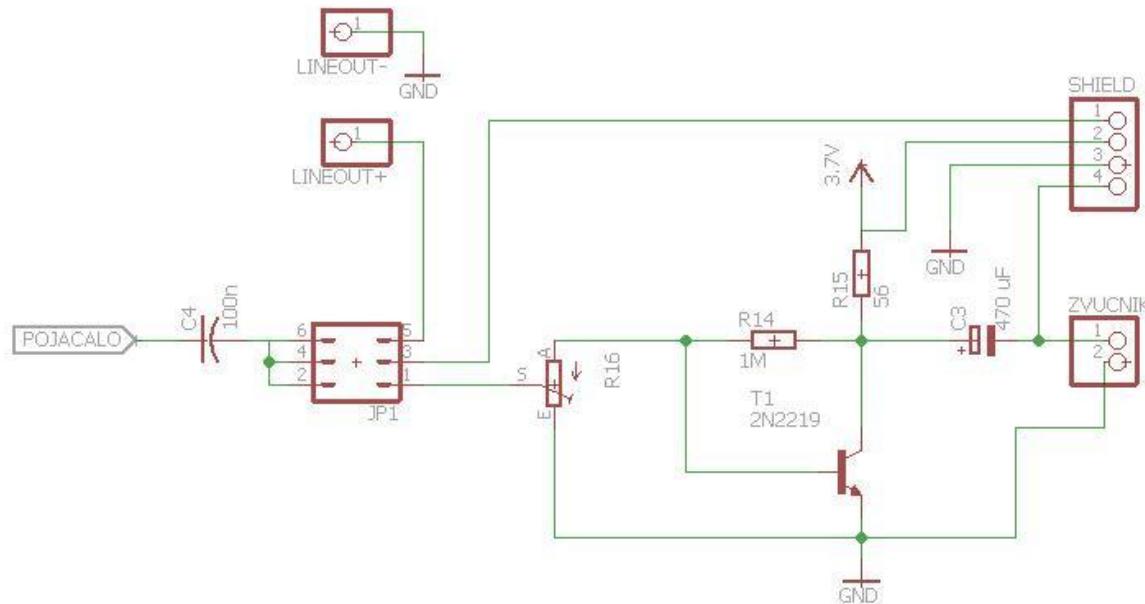
Slika 6.3 Dio sheme br. 3 (punjač)

4) Pojačalo zvuka:

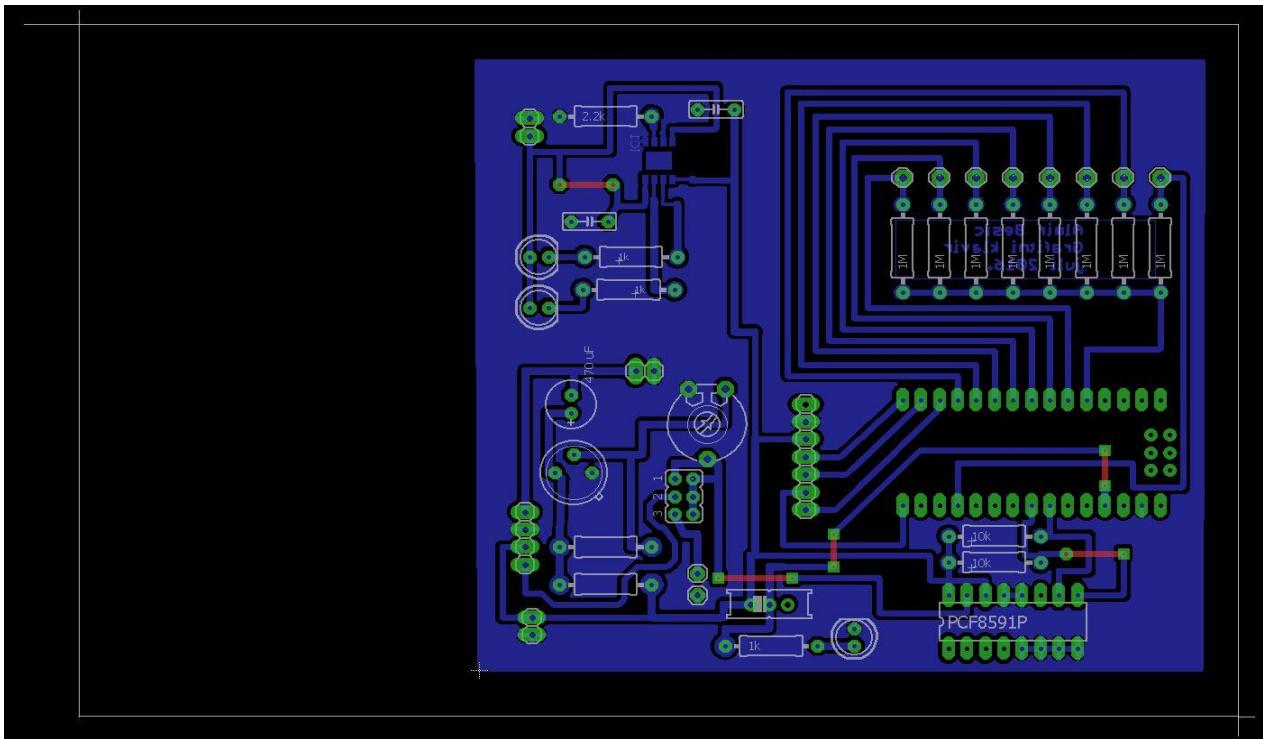
- BC140-16 NPN tranzistor (x 1),
- Potenciometar ($1 \times 10 \text{ k}\Omega$),
- Otpornici ($1 \times 1 \text{ M}\Omega$, $1 \times 56 \text{ }\Omega$),
- Elektrolitski kondenzator ($1 \times 470 \text{ }\mu\text{F}$),
- Kondenzator ($1 \times 100 \text{ nF}$),
- Niskoomski zvučnik ($x 1$),
- Female pin headeri,
- Jumper.

Dodatne napomene:

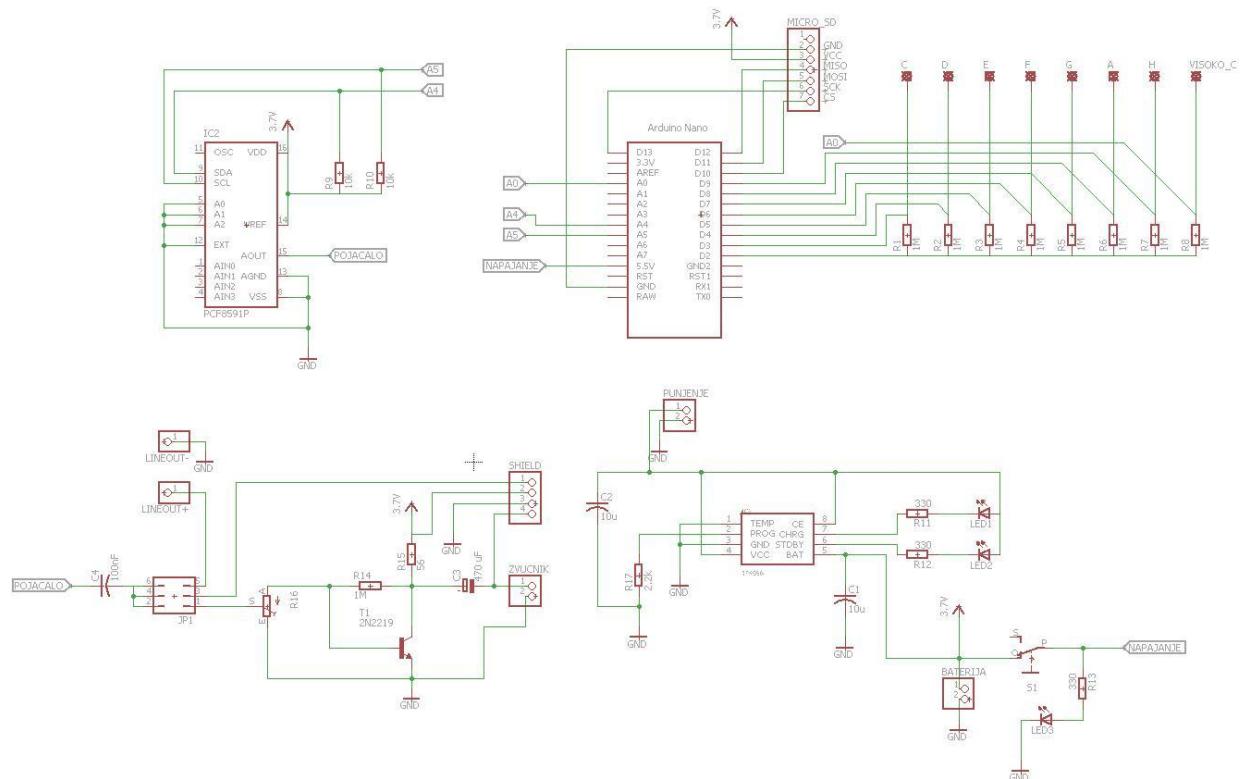
- Standardni pojačavački krug realizovan pomoću NPN tranzistora.
- Na ulazu se nalazi kondenzator koji filtrira ulazni signal kao i potenciometar koji služi za mijenjanje pojačanja.
- Izlaz za ugradnju dodatne štampane ploče (SHIELD).
- Izlaz za slušalice.



Slika 6.4 Dio sheme br. 4 (pojačalo zvuka)



Slika 6.5 Fizički izgled razvijene štampane pločice (dimenzije 100x160 mm)



Slika 6.6 Potpuna shema grafitnog klavira

Shema i PCB dizajn su izvedeni korištenjem programskog paketa Eagle.

7. Finalni eksperimentalni rezultati

Nakon izrade štampane ploče i odgovarajućeg kućišta grafitni klavir je postao funkcionalan. Pojavili su se određeni problemi sa punjačem koji su onemogućili njegovo korištenje. Ostali dijelovi funkcionišu onako kako je to predviđeno i opisano u radu.

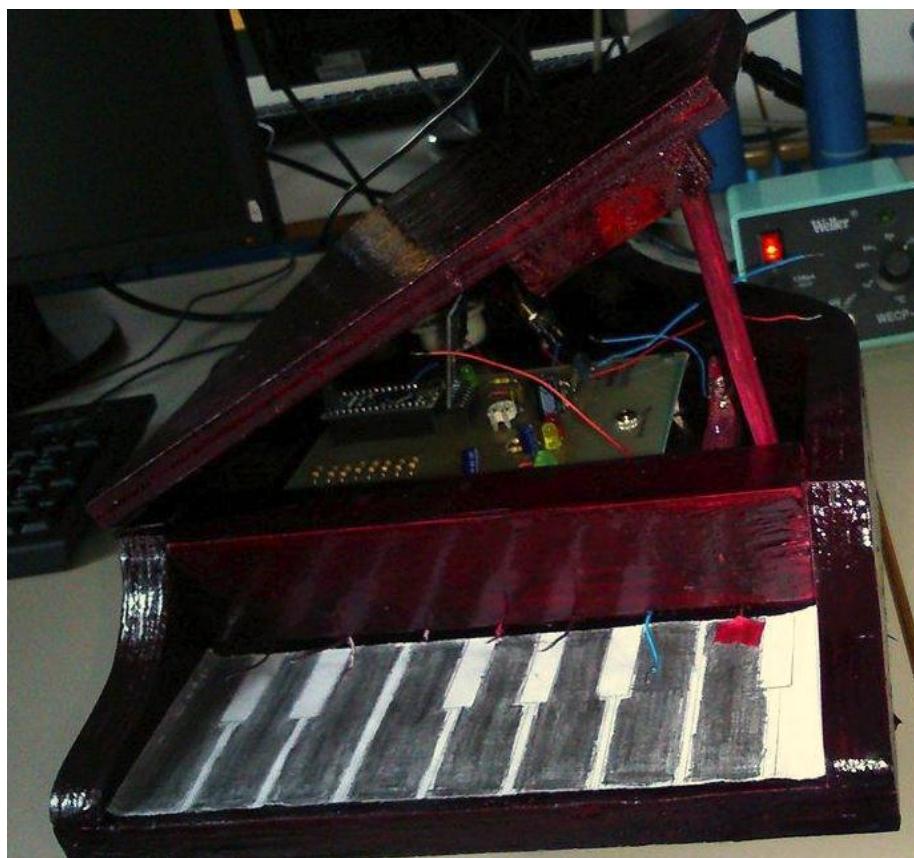
Greške u dizajnu su postale očite nakon spajanja kompletne sheme. Položaj modula za SD karticu i Arduina onemogućavao je istovremeno programiranje i testiranje klavira. Nije bilo moguće pristupiti Arduinu programerskim kablom bez uklanjanja modula za SD karticu.

Iz kvalitete reproduciranog zvuka zaključuje se da Arduino razvojno okruženje nije optimalan hardver za ovu vrstu zadatka, isključivo zbog brzine i nedostatka memorije. Manjak RAM memorije onemogućuje simultano sviranje nota i ograničava količinu kôda koji se može napisati. Niska rezolucija D/A konverzije dodatno umanjuje kvalitet odsviranih nota.

Dodatni napredak je moguć u pogledu iskorištenog prostora, jer se korištenjem SMD komponenti može razviti kompaktnija pločica.

Drveno i akustično kućište pojačava zvuk i daje izgled pravog koncertnog klavira.

Konačni kôd koji je implementiran dat je u prilogu (A.4).



Slika 7.1 Grafitni klavir

Zaključak

Kroz ovaj rad objašnjen je princip realizacije grafitnog klavira na bazi *Arduino* ugradbenog sistema. Najveći izazov je vjerna konverzija zvučnog signala iz diskretnog u analognu formu gdje su prepreke predstavljali ograničenje brzine SD kartice te brzina D/A konverzije. Ti problemi su, u dovoljnoj mjeri, ublaženi testiranjem i izvođenjem eksperimenata koji su omogućili realizaciju konačnog rješenja grafitnog klavira.

Napredak je postignut u pogledu napajanja koje je oboren sa 5V na 3.7V. Ovakva promjena ne utječe na brzinu ili kvalitet rada komponenti u bilo kojem pogledu. Time je osigurana dodatna ekonomičnost rješenja.

U svrhu postizanja kompaktnog i prenosivog klavira, napravljena je pločica koja integriše sve funkcionalnosti na jednom mjestu.

Integrисano pojačalo zvuka je dovoljno za demonstraciju rada klavira. Unapređenje se može postići korištenjem izvoda za dodatnu štampanu ploču koji se mogu priključiti na eksterno pojačalo.

Prilog A

A.1

```
%% učitavanje snimljenog zvuka klavira
[y, uzorci] = audioread('tone8.wav');

%% resample sa 48000/4=12000 (koristi se samo prva kolona
%% stereo vektora)
y1 = resample(y(:,1), 1, 4);

%% snimanje novog tona radi poređenja sa starim
audiowrite('Ton8_novo.wav',y1,12000);

%% skaliranje na opseg od 40 do 220
trenutni_opseg = max(y1) - min(y1);
novi_opseg = 220 - 40;
y2 = (y1 - min(y1)) / trenutni_opseg;
y2 = (y2 * novi_opseg) + 40;

%% upis int vrijednosti kao ASCII karaktera u .txt file
y2 = uint8(y2);
fid = fopen('Ton8_konacni.txt','wt');
fwrite(fid, y2);
fclose(fid);
```

A.2

```
#include <SD.h>
#include <SPI.h>
const int chipSelect = 4;

void setup()
{
    Serial.begin(9600);
    Serial.println("Inicijalizacija...");
    pinMode(4, OUTPUT);

    // Provjera da li je SD kartica ubačena u modul
    if (!SD.begin(chipSelect))
    {
        Serial.println("SD kartica nije očitana.");
        return;
    }
    Serial.println("SD kartica je uspješno očitana.");

    if (SD.exists("data.txt")) // Provjera da li je file dostupan
    {
        Serial.println("File postoji.");
    }
    else
    {
        Serial.println("File ne postoji.");
    }

    File dataFile = SD.open("data.txt", FILE_READ);
    unsigned long pocetak = millis();

    // Čitanje jednog po jednog karaktera sa SD kartice
    if (dataFile)
    {
        while (dataFile.available())
        {
            uint8_t data = dataFile.read();
        }

        dataFile.close();
    }
    else Serial.println("File nije dostupan.");
    unsigned long kraj = millis();
    Serial.print("Program završen. Trajanje programa: ");
    Serial.print(kraj-pocetak); //Ispis trajanja očitavanja u ms
    Serial.println("ms");
}
```

A.3

```
#include "Wire.h"
#include <SD.h>
#include <SPI.h>
#define PCF8591 (0x90 >> 1)

const int chipSelect = 4;

void AnalogOut(uint8_t vrijednost)
{
    sei(); //omogući prekide
    Wire.beginTransmission(PCF8591); // uključi PCF8591
    Wire.write(0x40); // kontrolni bajt
    Wire.write(vrijednost); // postavi vrijednost (0-255)
    Wire.endTransmission(); // završi slanje
}

void setup()
{
    Wire.begin();
    Serial.begin(9600);
}
void loop()
{
    AnalogOut(0); // nivo logičke nule
    AnalogOut(255); // nivo logičke jedinice
}
```

A. 4

```
#include "Wire.h"
#include <SD.h>
#include <SPI.h>
#include <CapacitiveSensor.h>
#define PCF8591 (0x90 >> 1)

const int chipSelect = 10;
File dataFile;
int pritisnut=0;

// Deklaracija pin-ova koji će se koristiti za kapacitivnu
detekciju
CapacitiveSensor cs_2_3 = CapacitiveSensor(2,3);
CapacitiveSensor cs_2_4 = CapacitiveSensor(2,4);
CapacitiveSensor cs_2_5 = CapacitiveSensor(2,5);
CapacitiveSensor cs_2_6 = CapacitiveSensor(2,6);
CapacitiveSensor cs_2_7 = CapacitiveSensor(2,7);
CapacitiveSensor cs_2_8 = CapacitiveSensor(2,8);
CapacitiveSensor cs_2_9 = CapacitiveSensor(2,9);
CapacitiveSensor cs_2_0 = CapacitiveSensor(2,A0);

long tipka1;
long tipka2;
long tipka3;
long tipka4;
long tipka5;
long tipka6;
long tipka7;
long tipka8;
uint8_t vrijednost;

void AnalogOut(uint8_t vrijednost)
{
    sei(); // omoguci prekide
    Wire.beginTransmission(PCF8591); // ukljuci PCF8591
    Wire.write(0x40); // kontrolni bajt
    Wire.write(vrijednost); // postavi vrijednost (0-255)
    Wire.endTransmission(); // zavrси slanje
}

void setup()
{
    Wire.begin();

    // Isključivanje rekalibracije funkcije CapacitiveSensor
    cs_2_3.set_CS_AutoCal_Millis(0xFFFFFFFF);
    cs_2_4.set_CS_AutoCal_Millis(0xFFFFFFFF);
    cs_2_5.set_CS_AutoCal_Millis(0xFFFFFFFF);
    cs_2_6.set_CS_AutoCal_Millis(0xFFFFFFFF);
```

```

cs_2_7.set_CS_AutoCal_Millis(0xFFFFFFFF);
cs_2_8.set_CS_AutoCal_Millis(0xFFFFFFFF);
cs_2_9.set_CS_AutoCal_Millis(0xFFFFFFFF);
cs_2_0.set_CS_AutoCal_Millis(0xFFFFFFFF);

if (!SD.begin(chipSelect))
{
    return;
}
}

void loop()
{
    // Pridruživanje vrijednosti na senzorskom pin-u
    // odgovarajućim varijablama
    tipka8 = cs_2_3.capacitiveSensor(20);
    tipka7 = cs_2_4.capacitiveSensor(20);
    tipka6 = cs_2_5.capacitiveSensor(20);
    tipka5 = cs_2_6.capacitiveSensor(20);
    tipka4 = cs_2_7.capacitiveSensor(20);
    tipka3 = cs_2_8.capacitiveSensor(20);
    tipka2 = cs_2_9.capacitiveSensor(20);
    tipka1 = cs_2_0.capacitiveSensor(20);

    // Ako se detektuje dodir, otvara se datoteka koja
    // korespondira pritisku na tipki
    if (tipka1 > 200)
    {
        dataFile = SD.open("Ton1.txt", FILE_READ);
        pritisnut=1;
    }

    if (tipka2 > 200)
    {
        dataFile = SD.open("Ton2.txt", FILE_READ);
        pritisnut=1;
    }

    if (tipka3 > 200)
    {
        dataFile = SD.open("Ton3.txt", FILE_READ);
        pritisnut=1;
    }

    if (tipka4 > 200)
    {
        dataFile = SD.open("Ton4.txt", FILE_READ);
        pritisnut=1;
    }
}

```

```

if (tipka5 > 200)
{
    dataFile = SD.open("Ton5.txt", FILE_READ);
    pritisnut=1;
}

if (tipka6 > 200)
{
    dataFile = SD.open("Ton6.txt", FILE_READ);
    pritisnut=1;
}

if (tipka7 > 200)
{
    dataFile = SD.open("Ton7.txt", FILE_READ);
    pritisnut=1;
}

if (tipka8 > 200)
{
    dataFile = SD.open("Ton8.txt", FILE_READ);
    pritisnut=1;
}

// U zavisnosti od pritisnute tipke očitava se odgovarajuća
nota
if (pritisnut)
{
    while (dataFile.available())
    {
        vrijednost = dataFile.read();
        AnalogOut(vrijednost);
    }
    dataFile.close();
    pritisnut=0;
}

```

Bibliografija

- [1] Lesić, I., 2006. Vizualizacija akustičnih pojava. *Sveučilište u Zagrebu*.
- [2] Muller, M., Ellis, D.P., Klapuri, A. and Richard, G., 2011. Signal processing for music analysis. *Selected Topics in Signal Processing, IEEE Journal of*, 5(6), pp.1088-1110.
- [3] Ahić-Đokić, M., Signali i sistemi. *Elektrotehnički fakultet Sarajevo*. 2010.
- [4] Stojić, M.R., Digitalni sistemi upravljanja. *Elektrotehnički fakultet Beograd*. 1998.
- [5] Webster, J.G. and Eren, H. eds., 2014. *Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement* (Vol. 1). CRC press.
- [6] Arduino Playground - CapacitiveSensor . 2016. *Arduino Playground - CapacitiveSensor* . [ONLINE] Available at: <http://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense>. [Accessed 03 July 2016].
- [7] https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf. 2012. *TP4056 Li-Ion linear charger*. [ONLINE] Available at: <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf>. [Accessed 9 July 2016].