Izrada vibracijske štimalice za ugađanje akustičkih instrumenata

Zetović, Marko

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:671254

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni diplomski studij

IZRADA VIBRACIJSKE ŠTIMALICE ZA UGAĐANJE AKUSTIČNIH INSTRUMENATA

Diplomski rad

Marko Zetović

Osijek, 2016.

SADRŽAJ

1. UVOD 3
1.1. Zadatak diplomskog rada: 3
2. PRIMJENJENE TEHNOLOGIJE I ALATI 4
2.1. Arduino programski paket 4
2.2. Povezivanje Arduina i računala 5
2.3. Timer 6
2.4. Piezo senzor 7
2.5. Magnetni senzor vibracije 9
3. RALIZACIJA SUSTAVA ZA DETEKCIJU FREKVENCIJE 11
3.1. Analogno-digitalna pretvorba 11
3.2. Analogno-digitalni signali 11
3.3. Pripremanje audio signala za Arduino 12
3.4. Pojačalo i offset 12
3.5. Programska realizacija sustava 14
3.5.1. Određivanje frekvencije 14
3.6. Tehnička realizacija sustava 15
3.6.1. Idejno rješenje sustava 15
3.6.2. Proizvodna dokumentacija 15
3.6.3. Metode rješenja sustava i rezultati 19
4. ZAKLJUČAK 21
LITERATURA 22
SAŽETAK 24
ŽIVOTOPIS 26

1. UVOD

Cilj ovog diplomskog rada je izrada vibracijske štimalice za ugađanje akustičnih instrumenata. Opisano je projektiranje, izrada i programiranje sustava temeljenog na mikroupravljaču Arduino uno. Pomoću mikroupravljača Arduino Uno omogućena je kontrola pojedinih elektroničkih komponenti i senzora. U radu su opisani, sustav štimalice i obrada snimljenih podataka. Korišteni su piezo i magnetni senzori te će se uspoređivati prednosti i nedostaci, odnosno usporedit će se njihov rad. U drugom poglavlju opisuje se primjena tehnologije i alati, sustav za detekciju frekvencije. U trećem poglavlju opisana je realizacija sustava za detekciju frekvencije te je opisana detekcija pomoću piezo senzora i magnetskog senzora.

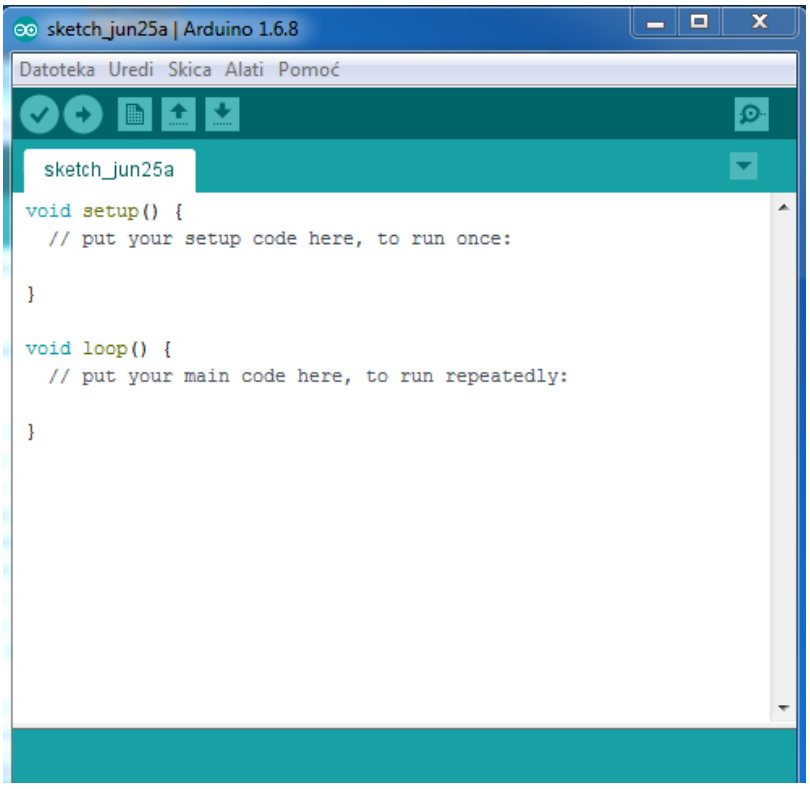
1.1. Zadatak diplomskog rada:

U ovom diplomskom radu potrebno je izraditi uređaj za uštimavanje akustičkih instrumenata s žicom. Uređaj treba mjeriti vibracije pomoću elektromagnetnog senzora koji pretvaraju mehaničke vibracije u električni signal. Mjerenjem vibracije žice omogućava se uštimavanje instrumenata u glasnoj okolini.

2. PRIMJENJENE TEHNOLOGIJE I ALATI

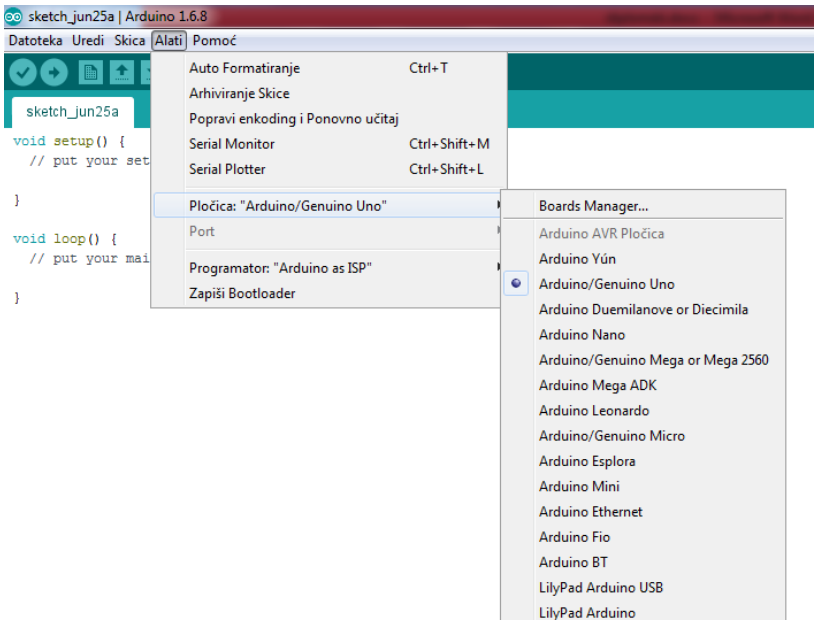
2.1. Arduino programski paket

Arduino je napravio svoj programski paket koji korisnicima omogućuje unos programa, prikazano na Slika 2.1. (dolje), njihovo jednostavno kompajliranje i prebacivanje na mikroupravljačku pločicu. Razvojno okruženje daje korisniku povratnu informaciju od računala (slučaj ako program sadrži greške), također može slati i prikazivati podatke preko serijske veze (npr. USB) te time olakšava testiranje uređaja jednom kada je isprogramiran. Programski paket je moguće preuzeti besplatno sa Arduino web stranica. [1]



Slika 2.1. Arduino software 1.6.8 [Izvor: autor]

Nakon što smo instalirali Arduino paket odabirom na TOOLS podesit BORD koji se koristi. U ovom slučaju treba odabrati Arduino/Genuino Uno i odabrati SERIAL PORT na koji je priključen, prikazano na Slika 2.2. (dolje).



Slika 2.2. Odabiranje Arduino/Genuino Uno te odgovarajući port [Izvor: autor]

2.2. Povezivanje Arduina i računala

Za povezivanje mikroupravljačke pločice i računala potreban je USB kabel. Može se koristiti i za uspostavljanje serijske veze s drugim uređajima te prijenos podataka, a i kao napajanje. Nakon odabira porta i BORD-a na koji je Arduino priključen te završenog programa potrebno je učitati program u Arduino. Način povezivanja komponenata u cjelinu:

1. preuzeti Arduinov programski paket sa web stranica i onda ga instalirati,
2. priključiti USB kabel na računalo s jedne strane i s druge strane na mikroupravljačku pločicu,
3. upisati kod - pokrenuti programski paket,
4. odabrati Sketch -> Verify/Compile te ako se ona uspješno izvršila, File -> Upload to I/O Bord,
5. testirati sustav. [5]

2.3. Timer

Tajmer ili preciznije brojač je komad *hardware*-a ugrađen u Arduino. Kao sat, može se koristiti za mjerenje vremena događaja. Tajmerom je moguće programirati neke posebne registre, konfigurirati *prescaler* za tajmer, odnosno način rada i druge stvari. Kontroler od Arduina je Atmel AVR Atmega 168 ili Atmega328. Ovi čipovi su kompatibilni i razlikuju se po veličini interne memorije. Oba imaju 3 tajmera pod nazivom: timer0, timer1, timer2. Timer0 i timer2 su 8-bitni, a tajmer1 je 16-bitni. Razlika između 8 bit i 16 bit je rezolucija tajmera. 8-bit označuje 256 vrijednosti , a 16 bit 65536 vrijednosti te je on za veće rezolucije. Kod 8-bitnih prekidi se obavljaju 256/16000000 s približno 13us, dok za 16-bitne brojače 65536/16000000s približno 4ms. Svi tajmeri ovise o satu sustava Arduina. Normalni sistem sata je 16Mhz, dok je za Arduino Pro 3,3V, 8Mhz. Brzinu tajmer brojača možemo kontrolirati pomoću *prescaler*-a. *Prescaler* određuje brzinu tajmera na osnovu jednadžbe:

Brzina tajmera (Hz) = 16MHz / prescaler

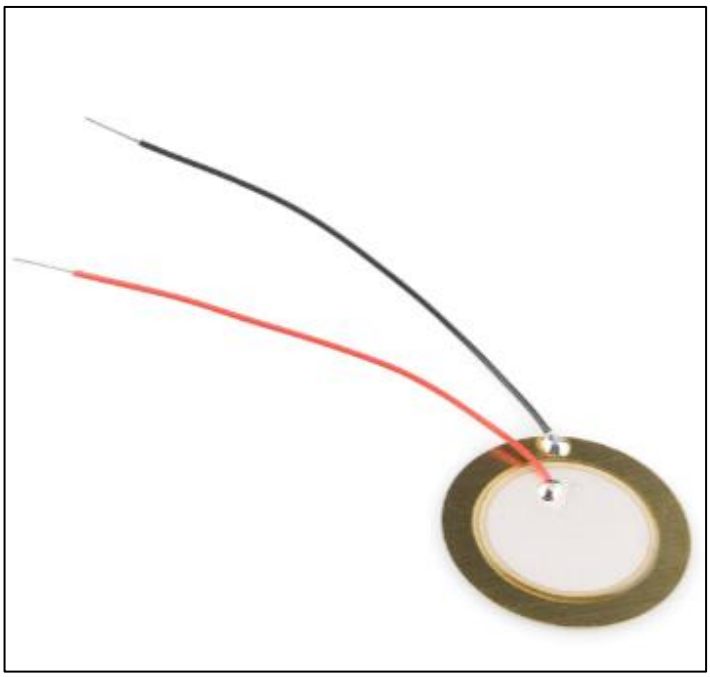
Vrijednosti *Prescaler*-a isključivo mogu biti: 1,8,64,256 i 1024. U većini situacija 16MHz je prebrzo, stoga koristimo vrijednosti Prescalera kako bi prilagodili brzinu tajmera. Potrebno je biti oprezan pri pisanju funkcije tajmera. Hardverski tajmer može biti konfiguriran s nekim posebnim tajmer registrima. [2]

Tablica 2.1. Odabiranje sata [Izvor: https://arduino-info.wikispaces.com/Timers-Arduino]

CS12	CS11	CS10	Opis
0	0	0	Tajmer/brojac zaustavljeni
0	0	1	Clk/1
0	1	0	Clk/8
0	1	1	Clk/64
1	0	0	Clk/256
1	0	1	Clk/1024
1	1	0	Vanjski izvor sata na T1 pin-u. Sata na rubu pada
1	1	1	Vanjski izvor sata na T1 pin-u.

2.4. Piezo senzor

Riječ piezo dolazi od grčke riječi *piezei*, što znači stisnuti ili pretisnuti. Piezo elektricitet se odnosi na proizvodnju električne energije ili električnog polariteta u dielektričnim kristalima što je prikazano Slikom 2.4. (dolje**)**. Kada se podvrgnu mehaničkom stresu, naprezanje u tim kristalima odgovara na apliciranu voltažu. Danas imamo mnogo piezoeletričnih svojstava u stotinama keramičkih i plastičnih materijala. Većina tih piezo materijala pokazuju električne efekte zbog promjene radijacija i temperature. Na molekularnoj razini struktura piezo materijala su ionski vezani kristali.

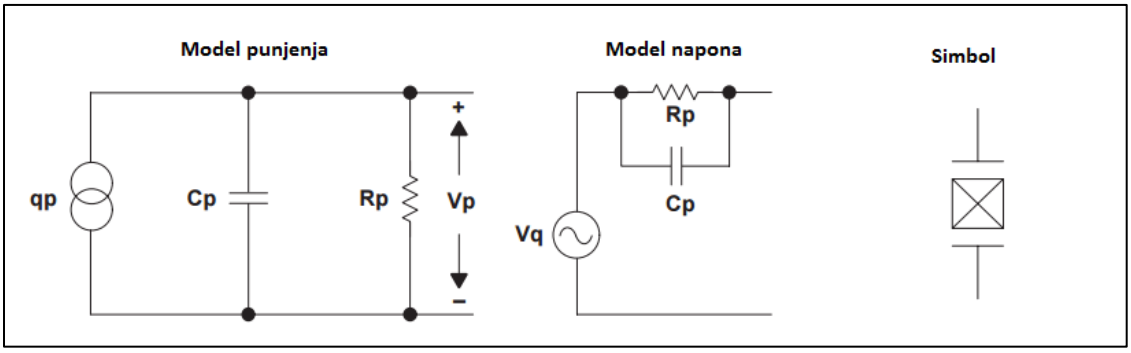


Slika 2.4. Piezo senzor, slika preuzeta iz [16]

Kada je u mirovanju pozitivni i negativni ioni se zbog simetrije kristala ne promatraju. Kada nije u stanju mirovanja tada se kristali izobliče te se izgubi simetrija. Piezo senzori nisu pogodni za statičke ili DC aplikacije, odnosno pogodni su za dinamičke ili AC aplikacije. Piezo senzor je modeliran kao izvor napona s nizom otpornika i kondenzatora, ovaj model je prikazan Slikom 2.5. (dolje**)** sa *Schematic* simbolom. Naboj punjenja ovisi o piezo konstantnom uređaju. Kapacitet određuje širinu, prostor i dielektrične konstante materijala. Izlazni naponi od piezo senzora mogu biti od mikrovolti do stotina volti te uvjeti krugova mogu puno varirati. Prije projektiranja pojačala potrebno je razmotriti:

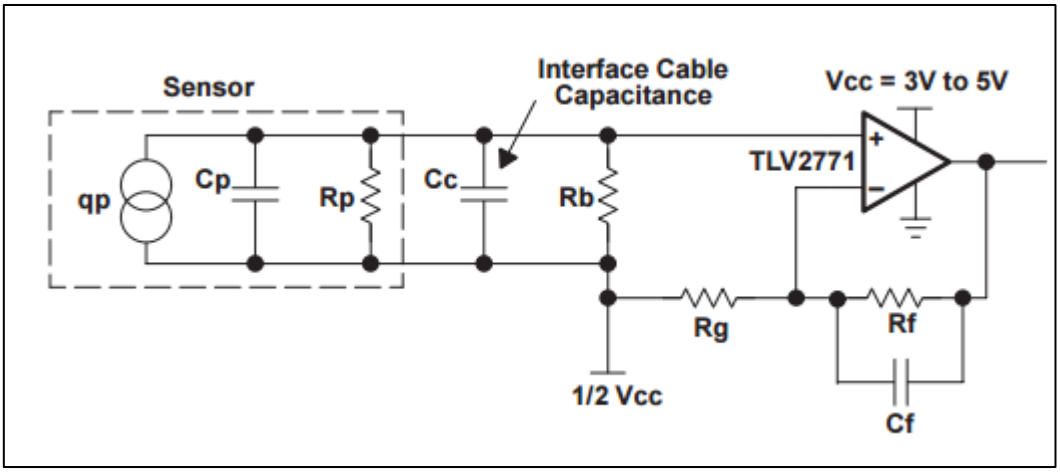
- amplitudu signala,
- frekvenciju rada,

- impedanciju ulaza,
- način rada.



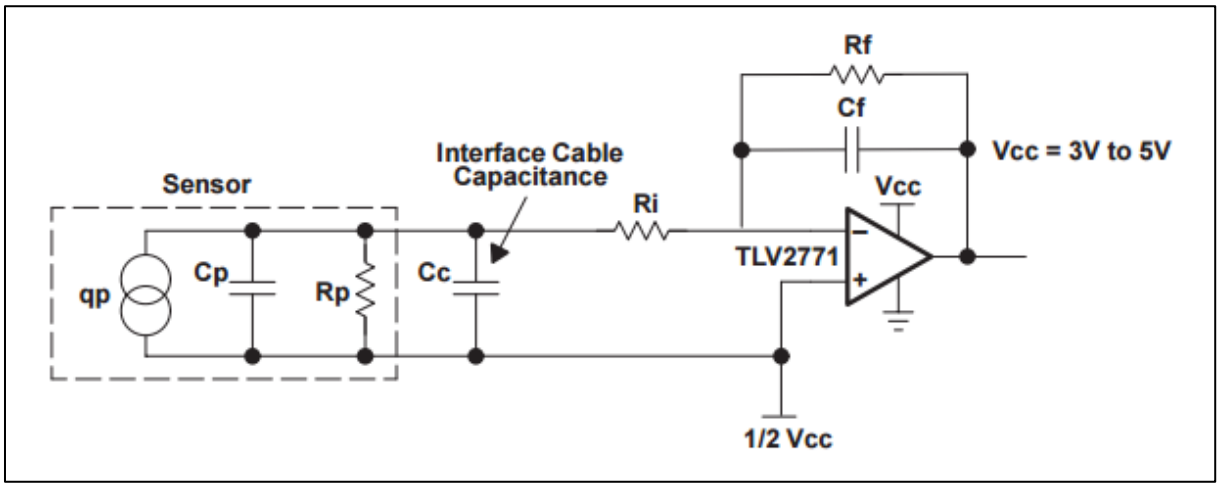
Slika 2.5. Senzor model, slika preuzeta iz [9]

Ako senzor treba umjerenu količinu pojačanja, željene razine signala od 3V do 5V raspona, visoka će impedancija zahtjevati pojačalo s visokim ulaznim impedancijama. Normalan izbor je CMOS ili JFET kao ulaz pojačala kao i TLV2771. Dva kruga se koriste za kondiciranje signala. Kada je pojačalo blizu senzora koristi se način „napon pojačanja kruga“ Slika 2.6. (dolje). Izlaz će ovisiti o količini kapaciteta, također, varijacija u Cc može uzrokovati probleme. Odabiranje Rf i Cf nam služi za postavljanje gornje kritične frekvencije. Kada je pojačalo daljinski senzor, koristi se način punjenja pojačanja. [9]



Slika 2.6. Napon pojačanja kruga Senzor model, slika preuzeta iz [9]

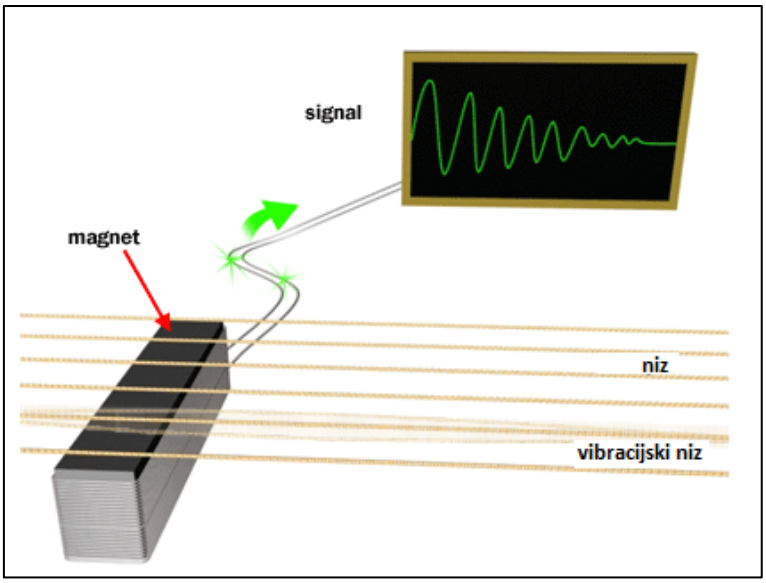
Kada je pojačalo daljinski senzor, koristi se način punjenja pojačanja kruga Slika 2.7. (dolje). Na ovaj način će pojačalo uravnotežiti naboj preko kondenzatora Cf. Vrijednosti Cf i Rf će postaviti nisku graničnu frekvenciju pojačala. S Otpornikom Ri dobivamo ESD zaštitu.



Slika 2.7. Punjenje pojačanja kruga, slika preuzeta iz [9]

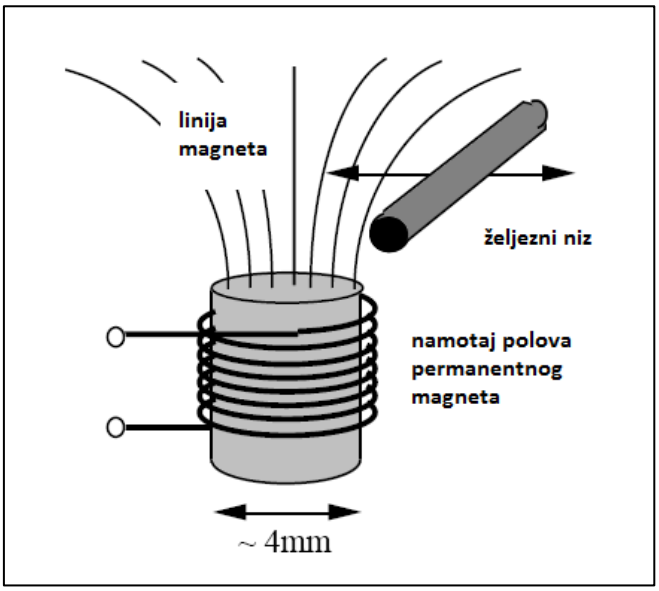
2.5. Magnetni senzor vibracije

Magnetni senzor vibracije ili *pickup,* Slika 2.8. (dolje), je sonda koja snima ili osjeti mehaničke vibracije od trzalačkih instrumenata kao što su električne bas gitare, električne gitare ili električne violine te ih pretvara u električni signal koji se pojačava električnim pojačalom. Signal iz pickap-a se može izravno snimiti koristeći *DI box* (npr. električni bas) ili emisiju na radiju ili na televiziji. Većina električnih gitara ili pedala koristi magnetske *pickup*-ove. Akustične gitare koriste piezoelektrične *pickup*-ove.



Slika 2.8. Pickup, slika preuzeta iz [17]

Magnetski pickup se sastoji od permanentnog magneta s jezgrom kao što je ferit omotan s spiralom od bakrene žice. Žice za gitaru od magnetske legure vibriraju unutar magnetskog polja i stvaraju električnu struju u žici na istoj frekvenciji. Pickup se najčešće montira na tijelo instrumenta, ali može i na most instrumenta. Permanentni magnet Slika 2.9. (dolje) stvara stalno magnetsko polje, a gibanje vibrirajućih žica remeti to polje, mijenja magnetski tok i inducira napon u svitku. [10]

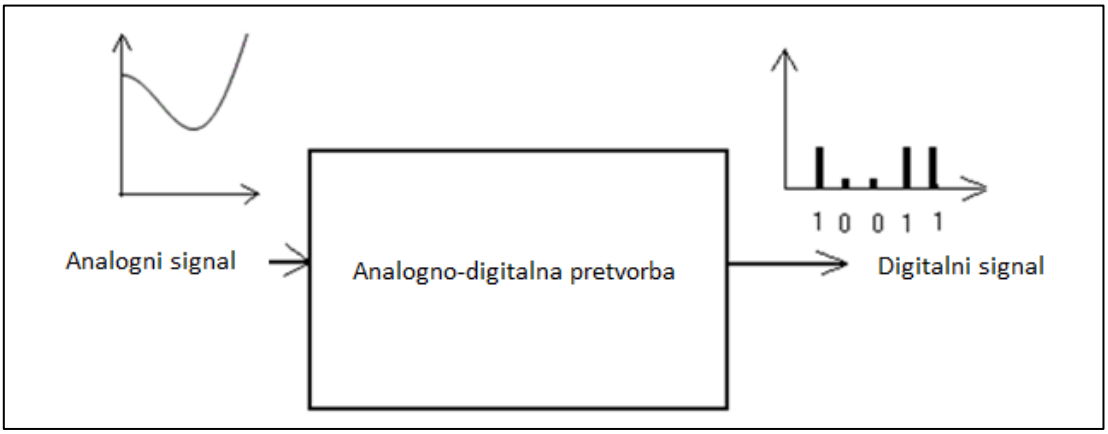


Slika 2.9. Shematski dijagram jednostavnog pickup-a, slika preuzeta iz [18]

3. RALIZACIJA SUSTAVA ZA DETEKCIJU FREKVENCIJE

3.1. Analogno-digitalna pretvorba

Analogno-digitalni pretvarač elektronički uređaj koji pretvara analogni signal iz našeg svijeta u digitalni oblik. Za to je potrebna komponenta koja može pretvoriti analogne signale u naponske signale. Taj napon se dovodi u analogno digitalnu pretvorbu i to rezultira binarnim znamenkama koje se dalje po zahtjevima obrađuju. Navedeno predstavlja jedan od najvažnijih tema mikroupravljačkog programiranja. [3]



Slika 3.1. Analogno-digitalna pretvorba, slika preuzeta iz [18]

3.2. Analogno-digitalni signali

Prije svega, potrebno je razjasniti tri pojma: analogni signal, signal i digitalni signal. Signali su postavljene vrijednosti koje prenose informacije. U području komunikacije, odnosno, obradi signala, signal je definiran kao bilo kakva prostorna ili vremenska promjena stanja medija kroz koji se rasprostire. U računalnoj tehnici, u promjenama električnih veličina koje se prenose kroz računalni sustav, možemo razlikovati signale kao što su analogni i digitalni. Analogni signali su kontinuirani signali, koji se razlikuju s vremenom. Analogni signal je promjenjiv napon ili struja u elektronici. Amplituda može imati bilo koju vrijednost. Digitalni

signal vremenski diskretan signal, dobiva se ako se uzme analogni signal i postupkom uzorkovanja (*sampling*) zapišu stanja analognog signala u određenim vremenskim koracima, dužina ovisi od frekvenciji uzorkovanja (*sampling rate*). Digitalni signal čija se vrijednost može mijenjati samo u određenom broju koraka i stanja. Amplituda može imati samo neki ograničeni broj vrijednosti, za razliku od analognog. Primjer digitalnog signala sklopka uključena/ isključena. Sve informacije su digitalne ili analogne. Moderna računala sada rade isključivo s digitalnim informacijama.[3]

3.3. Pripremanje audio signala za Arduino

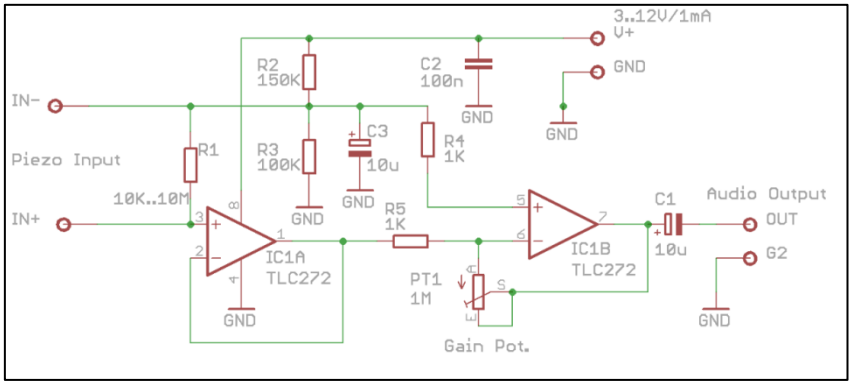
Ako snimimo audio signal na računalo, vidjet ćemo valni oblik koji kada uvećamo će se sastojati od tisuća sitnih oscilacija naprijed i natrag. To se zove zvučni signal, a u elektronici te oscilacije predstavljaju osciliranje napona tijekom vremena. Kada gledamo audio signal s osciloskopom vidjet ćemo kako audio signal oscilira oko napona 0V što je i normalno za audio signal. Udaljenost između centra napona i njegovog niskog i visokog vrha se naziva amplituda zvučnog signala. Ako želimo izmjeriti audio signal s jedim od analognih ulaza Arduino može mjeriti samo napone između 0 i 5V. Ako želimo mjeriti negativne napone u signalu, Arduino će nam to prikazivati kao 0V.

3.4. Pojačalo i offset

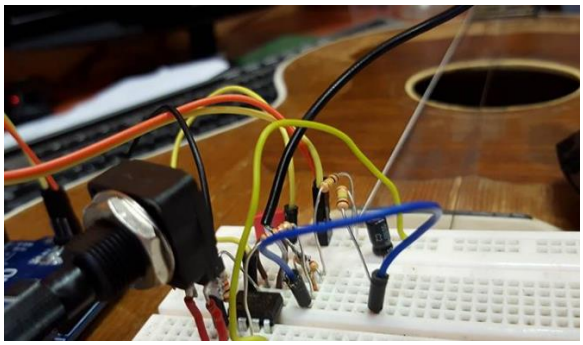
Pojačavanje *offseta* signala potrebno je postaviti u rasponu od 0 do 5V. Idealni slučaj je kada želimo signal s amplitudom 2.5V, koji oscilira oko 2.5V, postaviti tako da je njegov minimalni napon 0V i maksimalni napon 5V. U nastavku je vidljiv izračun:

Minimalni napon = centar za napon – amplituda
Minimalni napon = 2.5V – 2.5V = 0V
Maksimalni napon = centar za napon + amplituda
Maksimalni napon = 2.5V + 2.5V = 5V

Signal koji ide ravno iz instrumenta na osciloskop je vrlo mali. Amplituda toga signala je vrlo mala oko 200Mv. Kako bi dobili željene amplitude npr. 2.5V, signali moraju biti pojačani. Povećanje se odnosi na povećanje udaljenost između središnje i maksimalne, središnje i minimalne točke. Kada je sve namješteno, signal je spreman ići na Arduino analogni ulazni pin. Audio signal koji dolazi iz električne gitare treba biti pojačan na 5V od vrha do vrha, a *offset* centriran oko 2.5V, za razliku 0V. Signal mora biti između 0V i 5V kako bi se mogao čitati s Arduino analognog pina. Potrebno je imati najveću moguću amplitudu, bez *clipping*-a, kako bi dobili točnije izračune frekvencije. Slika 3.2 (dolje) prikazuje shematski krug pojačala koji se mora izvesti, izvedba je prikazana Slikom 3.3. (dolje). Test je izvršen s pomoću osciloskopa. Audio ulaz čini žica odnosno *audio jack*, dok je druga spojena na masu. Potrebno je spojiti gitaru na *audio jack* te provjeriti je li je centriran oko 2.5V te je li u neposrednoj blizini, ali da ne prelazi 5V od vrha do vrha.



Slika 3.2. Shema sklopa, slika preuzeta iz [11]



Slika 3.3. Realiziran sklop s potrebnim komponentama [Izvor: autor]

3.5. Programska realizacija sustava

Pošto veće frekvencije imaju manju amplitudu signala teže će nam biti otkriti frekvenciju. Imamo varijablu koja za minimalne amplitude signala preko Arduina računa frekvenciju nazvana je ampPrag. S njim dobivamo dovoljan raspon frekvencija koje su nam potrebne za štimer. Ta varijabla treba biti oko 25 kako bi dobro radila. Vrijednosti se mogu korigirati po potrebi, jer nije isto štimati bas gitaru ili pravu gitaru zbog različitog raspona frekvencija. Bas gitara ima manje frekvencije, odnosno, visinu tona te time imamo i veću amplitudu audio signala.

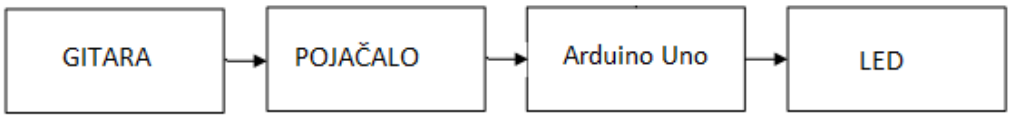
3.5.1. Određivanje frekvencije

Prilikom određivanje frekvencije imamo kod, koji koristi brzinu uzorkovanja 38.5kHz. Kada ulazni audio signal daje visoki signal tada ima pozitivan nagib, a negativan nagib kada izlazi niski signal. Vrhovi se uvijek nalaze između prijelaza iz visokog u niski i niskog u visoki. Promatrajući kod, sve se odvija u ADC prekidu. Prekid, i zatim ide nova analogna vrijednost spremna od A0. Tajmer prekida omogućava obavljanje zadatka u određenom vremenu, bez obzira na to što nam se još nalazi u istom kodu. Arduino biblioteka funkcija npr. analogRead i digitalWrite koje se sastoje od puno naredbi.[4] Arduino tajmer prekida će nam omogućiti preuzimanje, odnosno, na trenutak normalan slijed, koji se odvija u funkciji loop() u točno određenim intervalima. Kada se ove naredbe izvrše Arduino nastavlja gdje je bio u petlji loop(). Prekidi su nam potrebni kada mjerimo dolazni signal u jednakim razmacima tj. konstantnoj frekvenciji ili kada šaljemo signal određene frekvencije. Određivanje impulsa nepravilnog radnog ciklusa je puno složenije od samog sinusnog vala. Svaki puta kada val dosegne maksimum i minimum prebacuje se izlazni puls.

3.6. Tehnička realizacija sustava

3.6.1. Idejno rješenje sustava

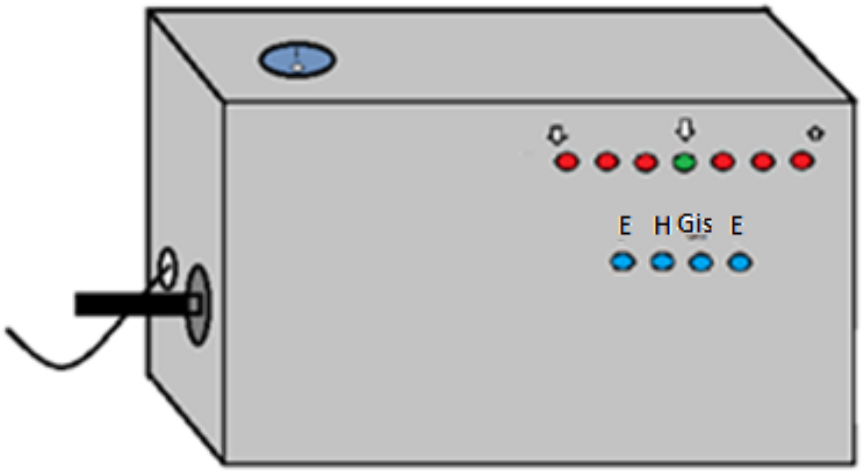
Idejno rješenje sustava čini: mikroupravljač Arduino Uno, gitara, pojačalo i LED, Slika 3.4. (dolje). Pri testiranju korišten je, *serial monitor* i *serial ploter*.



Slika 3.4. Idejno rješenje sustava [Izvor: Autor]

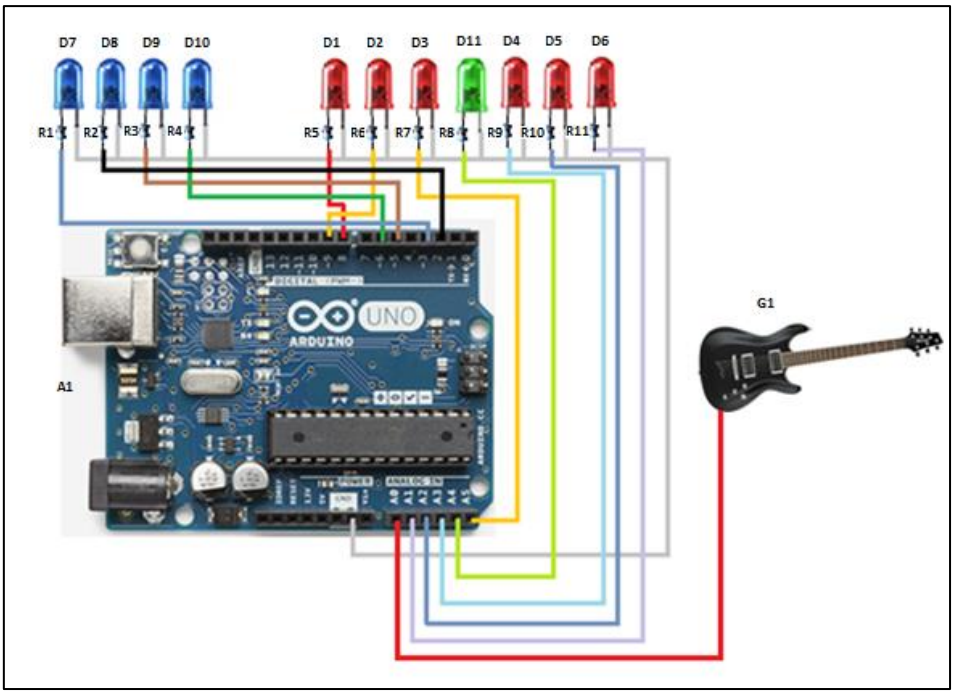
3.6.2. Proizvodna dokumentacija

Slikom 3.5. (dolje) prikazan je dizajn štimalice korištenjem ledica. Plave ledice označavaju žice na gitari e, h, Gis, e. Gornjih šest crvenih ledica označava odstupanje od ispravne frekvenicije. Vidljiv je prekidač pomoću kojega palimo štimalicu te iz pojačala audio ulaz koji spajamo na gitaru, također prikazan je i potenciometar pomoću kojega pojačavamo ili smanjujemo signal.

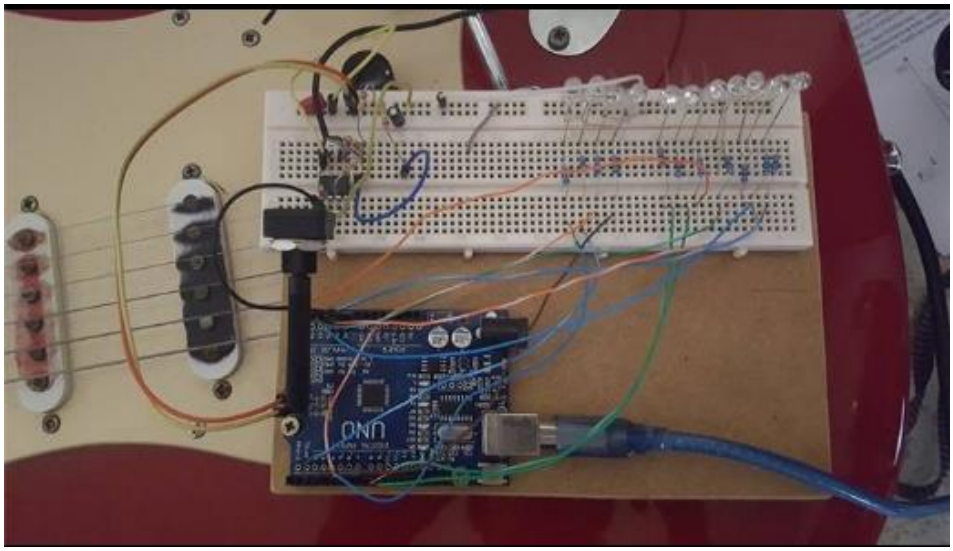


Slika 3.5. Dizajn štimalice [Izvor: autor]

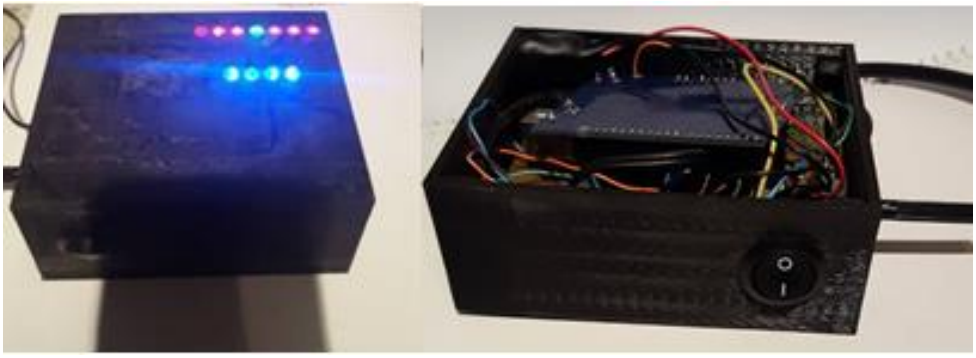
Slikom 3.6. prikazana je shema spajanja svih potrebnih sklopova i komponenti te su u Tablici 3.1. (dolje) prikazane potrebne komponente za realizaciju sustava, kao i oznake koje su predodređene za svaku komponentu posebno. Slika 3.7. (dolje) prikazuje kako izgleda realizirani navedeni sustav.



Slika 3.6. Shema spajanja svih potrebnih sklopova i komponenti, slika preuzeta iz [15]



Slika 3.7. Realizirani sustav sa svim potrebnim sklopovima i komponentama (protoboard) [Izvor: autor]



Slika 3.8. Štimalica [Izvor: autor]

Na Slici 3.9. prikazane su dvije gitare: jedna je električna kontra, druga je bas gitara. Navedeni glazbeni instrumenti korišteni su za testiranje zbog većega raspona frekvencija. Bas gitara ima duboke tonove te omogućuje testiranje malih frekvencija, veća je amplituda te su veće promjene i lakše je detektirati frekvenciju. Bas gitara ima dodatno pojačavanje u sebi što dovodi do toga kako se lakše regulira pojačanje, što nije slučaj kod električne kontre koja ima u sebi namješteno pojačanje.



Slika 3.9. Gitare za testiranje [Izvor: autor]

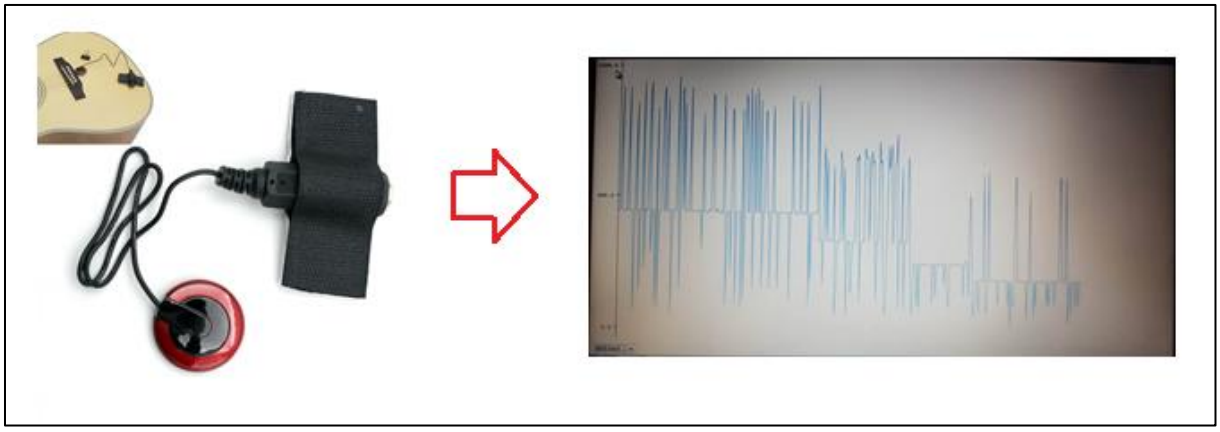
Tablica 3.1. Potrebne komponente za realizaciju sustava

r.br.	Naziv	oznaka	Tip i vrijednost	količina
1.	Arduino uno	A1	mikroupravljač	1
2.	Gitara	G1	Električna gitara	1
3.	otpornik 330Ω	R1-R2-R3-R4-R5-R6-R7-R8-R9-R10-R11	otpornik 330 Ω	3
4.	otpornik 10 kΩ	R12-R13	otpornik 10 kΩ	2
5.	otpornik 100 kΩ	R14-R15	otpornik 100 kΩ	2
6.	otpornik 150 kΩ	R16-R17	otpornik 150 kΩ	2
7.	otpornik 500 kΩ	R18-R19	otpornik 500 kΩ	2
8.	kondenzator 10uF	C1	Kondenzator elektrolitski 10uF	2
9.	kondenzator 1uF	C2	Kondezator kristalni 1uF	2
10.	kondenzator 10nF	C3	Kondezator kristalni 10nF	2
11.	potenciometar 1M	P1	Linearni 1M	1
12.	dioda LE crvena 5mm	D1-D2-D3-D4-D5-D6	LE dioda	6
13.	dioda LE plava 5mm	D7-D8-D9-D10	LE dioda	4
14.	dioda LE zelena 5mm	D11	LE dioda	1
15.	otpornik 1 kΩ	R20-R21	otpornik 1 kΩ	2

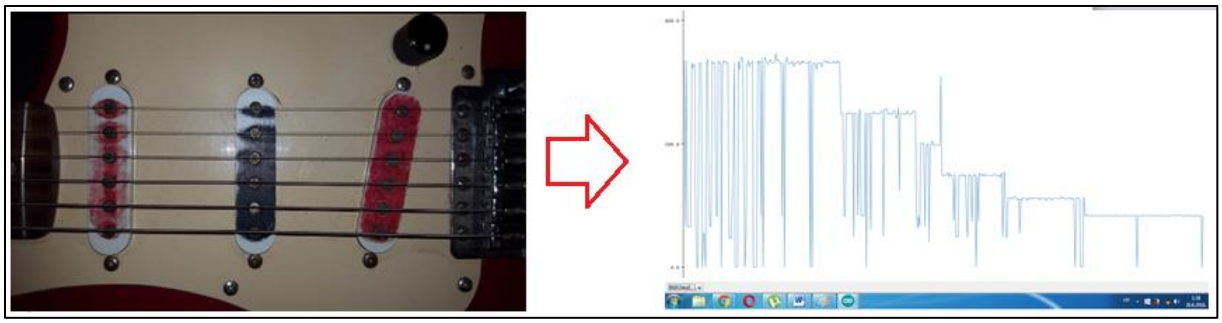
3.6.3. Metode rješenja sustava i rezultati

Tjekom ispitavanja korišten je piezo uređaj za detekciju vibracija. Korištenjem navedenog dobiveni je zaključak kako ne postoji mogućnost detektirati frekvencije zbog velike buke i prevelike osjetljivosti. Slikom 3.8. vidljiva je velika količina oscilacija, ali možemo ustanoviti koja je frekvencija. Prisutne su 4 različite frekvencije. Slika je dobivena preko Arduino (*serial ploter*). Moguće je uočiti kako su veće oscilacije na višim tonovima odnosno frekvencijama, što su manje frekvencije manje su i oscilacije. Tjekom ispitanja dobiveni je zaključak kako Piezo neće biti od koristi. U nastavku ispitivanja magneti su se pokazali puno boljim izborom. Magneti nisu osjetljivi na vanjske utjecaje kao što je buka. Vidljivo je kako na dubokim tonovima učinkovitost je gotovo savršena što je prikazano Slikom 3.9.

.

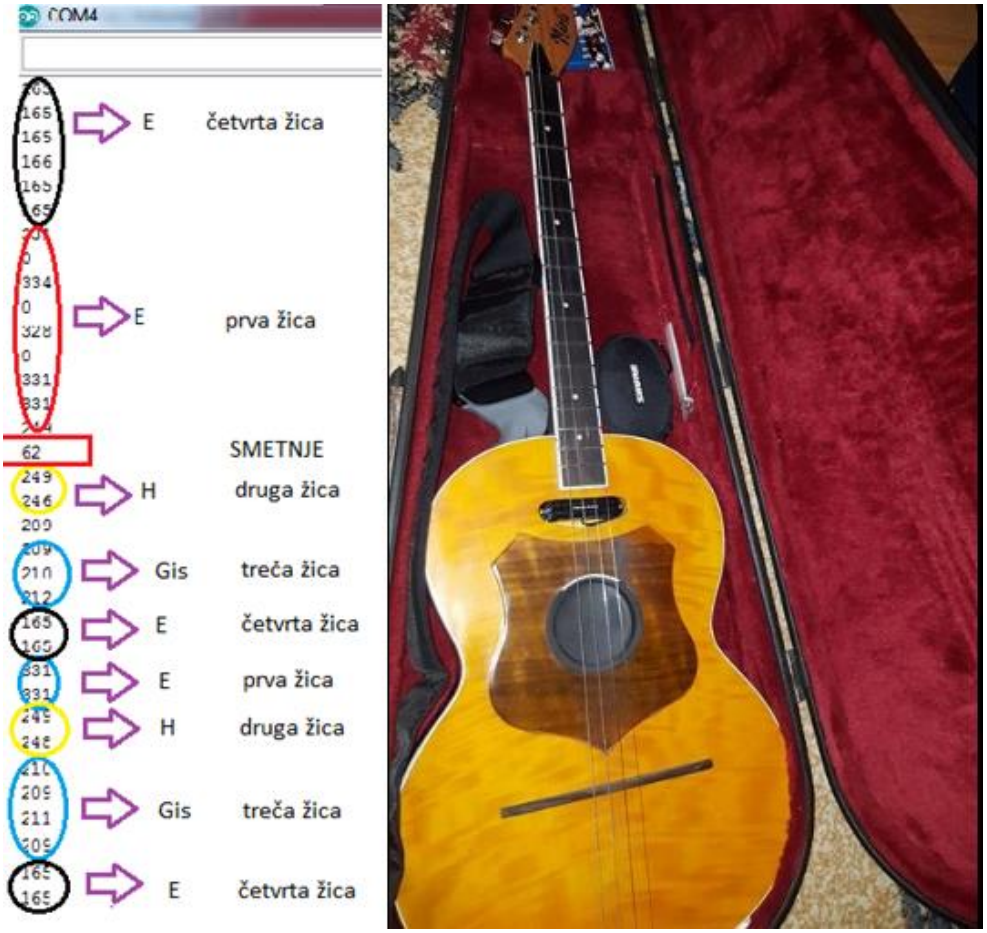


Slika 3.10. Dobiveni rezultati s piezo (*serial ploter*), slika preuzeta iz [15]



Slika 3.11. Dobiveni rezultati s magnetima (*serial ploter*) [Izvor: autor]

Na slici 3.10. prikazani su rezultati u (*serial monitor)*. Možemo primijetiti kako se sudaranjem različite žice pojavljuju različite frekvencije. Kada se vratimo na istu žicu vidljivo je kako je ista frekvencija kakva je bila i prvi puta, s malim odstupanjem, koje je uzrokovano različitim trzanjem po gitari, koje ujedno karakterizira visoka osjetljivost. U dobivenim rezultatima vidljiva je i frekvencija 62 Hz, koja je u ovom slučaju smetnja. Smetnje možemo smanjiti tako što smanjujemo pojačanje.



Slika 3.12. Dobiveni rezultati s magnetima (*serial monitor*) [Izvor: autor]

4. ZAKLJUČAK

U ovom radu izrađena je vibracijska štimalica za ugađanje akustičkih instrumenata. Štimalica se sastoji od: pojačala, Arduina, te ledica, koje nam prikazuju ton žice te točnost frekvencija. S obzirom kako veće frekvencije imaju manju amplitudu signala, teže se otkriva frekvencija, time manje frekvencije su bile jednostavnije za određivanje, jer veće amplitude imaju i veće promjene, čime je jednostavnije odrediti frekvenciju. U radu je korišten Piezo uređaj s kojim se određivala frekvencija, što se nije pokazalo uspješnim zbog prevelike osjetljivosti na buku. Drugi dio ispitivanja realizirao se uz pomoć magneta, što je pokazano boljom tehnikom, također, korišten je i alat za određivanje frekvencije i ugađanje instrumenata. Mana ovog ispitivanja je, ukoliko želimo bolje rezultate potrebna nam je bolja tehnologija i alati, što je istovremeno skuplje. Dobiveni zaključak je, kako određivanje frekvencije s Piezo uređajem daje maksimalan rezultat samo u idealnim uvjetima, također, rezultati bi bili učinkovitiji ukoliko bi se koristio kvalitetniji magnet, kao i sam kvalitetniji Piezo uređaj.

LITERATURA

[1] Arduino: Uno http://arduino.bloger.index.hr/default.aspx (5.3.2016.)

[2] Tajmer: https://arduino-info.wikispaces.com/Timers-Arduino (20.4.2016.)

[3] Analogno-digitalna pretvorba: http://www.robotplatform.com/knowledge/ADC/adc_tutorial.html (20.4.2016.)

[4] Arduino adc: http://www.microsmart.co.za/technical/2014/03/01/advanced-arduino-adc/ (4.5.2016.)

[5] Arduino: https://www.arduino.cc/en/Guide/HomePage (18.5.2016.)

[6] Arduino: https://www.arduino.cc/en/Guide/Introduction (6.6.2016.)

[7] EAGLE: http://www.house4hack.co.za/wordpress/wcontent/uploads/2012/02/Eagle_CAD_fill_isolate.png (14.6.2016.)

[8] Piezo: http://www.eng.hmc.edu/NewE80/PDFs/PiezoElectronics.pdf (2.6.2016)

[9] Piezo: http://www.ti.com/lit/an/sloa033a/sloa033a.pdf (24.6.2016)

[10] Magnetni senzor: http://audio-pro-central.com/WordPress/?p=28 (24.6.2016)

[11] Relizacija pojačala shema: http://interface.khm.de/index.php/lab-log/piezo-disk-preamplifier/ (25.6.2016)

[12] Eagle: http://www.cadsoft.de/eagle-pcb-design-software/ueber-eagle/ (26.6.2016)

[13] http://www.newtechnologysite.com/computers/analog_digital.html (22.9.2016)

[14] http://www.premierguitar.com/articles/Maxwells_Silver_Hammer_of_the_Gods (22.9.2016)

[15] https://www.arduino.cc/en/Main/Products (22.9.2016)

[16] http://lista.mercadolivre.com.br/instrumentos-musicais/captador-de-contato-para-cavaquinho (24.6.2016.)

[17] http://s.hswstatic.com/gif/eg-magnet.gif (24.6.2016.)

[18] http://www.premierguitar.com/articles/Maxwells_ Silver_Hammer_of_the_Gods (22.9.2016)

[19] https://www.vdrums.com/forum/advanced/technical/1129958-yamaha-dtxpress-witch-piezo slika preuzeta iz (22.9.2016.)

SAŽETAK

U ovom radu izrađena je vibracijska štimalice za ugađanje akustičkih instrumenata. Šimalica se sastoji od pojačala, Arduina te ledica koje nam prikazuju ton žice te točnost frekvencije. Korištenjem Piezo uređaja pokušano je detektirati frekvenciju što se nije pokazalo uspješnim, zbog prevelike osjetljivosti na buku. Drugi dio ispitivanja vršio se magnetima, što se pokazala kao bolja tehnika i alat za određivanje frekvencije i ugađanje instrumenata.

Ključne riječi: Arduino, Piezo, magnetski senzor, mikroupravljač, štimalica.

ABSTRACT

Title: How to make a tuner to tune acoustic instruments

In this dissertation I tasted and described how to create a vibrational Tuner for acoustic instruments. The vibrational Tuner is consist of, an audio power amplifier, Arduino and LEDs which are showing to us the ccorrect tone of the wire and exactness of every frequency. Using Piezo device it has been found that it is impossible to detect ccorrectness of frequency because of sensitivity to noise. The second part of the test is realized using magnets as a tool of testing. Magnets proved to be a better choice in determination of ccorrectness of frequencies and tuning instruments.

Keywords: Arduino, Piezo, magnetic sensor, microcontroller, tuner.

ŽIVOTOPIS

Marko Zetović rođen je 27. studenog 1991. godine u Zagrebu, Republika Hrvatska. Osnovnu školu je pohađao u Budrovcima. Nakon završetka osnovne škole upisuje Srednju strukovnu školu Braće Radića Đakovo, smjer Računalni tehničar za strojarstvo. Istu završava sa vrlo dobrim uspjehom i potom upisuje Elektrotehnički fakultet, smjer Informatika koji redovito završava. Zatim upisuje razlikovnu godinu na Elektrotehničkom fakultetu te ju završava, svoj studij nastavlja na Diplomskom studiju, smjer Procesno računalstvo. Pored svojih fakultetskih obaveza, profesionalno se bavi glazbom svirajući u mladom, ali iskusnom tamburaškom sastavu Brace.

Marko Zetović

PRILOG A. Karakteristike Arduino Nano razvojne ploče

Tablica karakteristika Arduina

Mikroupravljač	Atmega328
Radna voltaža	5V
Ulazni napon	6-20V
Digitalni pinovi	14(od kojih je 6 PWM izlaz)
Analogni pinovi	6
Struja prema pinovima	40mADC
Struja prema 3.3V pinu	50mA
Flash memorija	32KB
SRAM	2KB
EEPROM	1KB
Takt	16MHz

PRILOG B. Program za Arduino Nano razvojnu ploču

```
byte noviPodaci = 0;
byte stariPodaci = 0;
unsigned int time = 0;
int tajmer[10];
int nagib[10];
unsigned int ukupanTajmer;
unsigned int period;
byte index = 0;
int frekvencija;
int maxNagib = 0;
int noviNagib;
int filtriranaFreq;


byte Match = 0;
byte tolerancijaNagiba = 10;
int tolerancijaTajmera = 50;


unsigned int ampTajmera = 0;
byte maxAmplituda = 0;
byte provjeriMaxAmp;
byte ampPrag = 126;//povecati ako je jak signal 127 ne radi

//variabla stimlice
int ispravnaFreq;//tocna frekvencija koju sviram

void setup(){

  Serial.begin(9600);

  //LED pinovi

  pinMode(6,OUTPUT);   //pin6
  pinMode(5,OUTPUT);   //pin5
  pinMode(2,OUTPUT);   //pin2
  pinMode(3,OUTPUT);   //pin3
  pinMode(A3,OUTPUT);  //pinA3
  pinMode(A1,OUTPUT);  //pinA1
  pinMode(A2,OUTPUT);  //pinA2
  pinMode(A4,OUTPUT);  //pinA4
  pinMode(A5,OUTPUT);  //pinA5
  pinMode(8,OUTPUT);   //pin8
  pinMode(9,OUTPUT);   //pin9


  //paljenje stimalice

  digitalWrite(3,1);
  delay(500);
  digitalWrite(2,1);
  delay(500);
  digitalWrite(5,1);
  delay(500);
  digitalWrite(6,1);
  delay(500);
  analogWrite(A1,255);
  delay(500);
```

```
  analogWrite(A2,255);
  delay(500);
  analogWrite(A3,255);
  delay(500);
  digitalWrite(8,1);
  delay(500);
  digitalWrite(9,1);
  delay(500);
  analogWrite(A5,255);
  delay(500);

  analogWrite(A4,255);
  delay(1000);




  cli();//onemoguci interapt

  //postavljeno kontinuirano uzorkovanje analognog pin 0 na 38.5kHz

  //obrisi ADCSRA i ADCSRB registre
  ADCSRB = 0; // obrisan B registar
  ADCSRA = 0; // obrisan A registar

  ADMUX |= (1 << REFS0); //postavio referentnu voltazu s S0
  ADMUX |= (1 << ADLAR);  //napusti adc velicinu da mogu citati samo 8 bita
sa adch registra
  ADCSRA |= (1 << ADPS2)|(1 << ADPS0) ; //postavi ADC sat sa 32 prescaler-
16mHz/32=500kHz
  ADCSRA |= (1 << ADATE); //onemoguci auto trigger
  ADCSRA |= (1 << ADIE); //onemoguci prekid kada je mjerenje kompletno
  ADCSRA |= (1 << ADEN); //omoguci ADC
  ADCSRA |= (1 << ADSC); //počmi ADC mjerenje

  sei();//omoguci interapt
}

ISR(ADC_vect) {//kada je novi ADC vrijednost je spremna

  PORTB &= B11101111;//postavi pin 12 low
  stariPodaci = noviPodaci;
  noviPodaci = ADCH;//uzmi vrijednost od A0
  if (stariPodaci < 127 && noviPodaci >=127){//ako je u porastu i prelaska
polovište
    noviNagib = noviPodaci - stariPodaci;
    if (abs(noviNagib-maxNagib)<tolerancijaNagiba){//ako je nagib jednak ==

      nagib[index] = noviNagib;
      tajmer[index] = time;
      time = 0;
      if (index == 0){//novi maksimalni nagib resetirat
        PORTB |= B00010000;// pin 12
        Match = 0;
        index++;
      }
      else    if    (abs(tajmer[0]-tajmer[index])<tolerancijaTajmera    &&
abs(nagib[0]-noviNagib)<tolerancijaNagiba){
        //zbroji vrijednsti od tajmera
        ukupanTajmer = 0;
```

```
        for (byte i=0;i<index;i++){
          ukupanTajmer+=tajmer[i];
        }
        period = ukupanTajmer;

        tajmer[0] = tajmer[index];
        nagib[0] = nagib[index];
        index = 1;
        PORTB |= B00010000;
        Match = 0;
      }
      else{
        index++;
        if (index > 9){
          reset();
        }
      }
    }
    else if (noviNagib>maxNagib){// ako je noviNagib veci onda ga zamjeni
      maxNagib = noviNagib;
      time = 0;//resetirat sat
      Match = 0;
      index = 0;//resetirat index
    }
    else{
      Match++;
      if (Match>9){
        reset();
      }
    }
  }

  time++;

  ampTajmera++;
  if (abs(127-ADCH)>maxAmplituda)
  {
    maxAmplituda = abs(127-ADCH);
  }
  if (ampTajmera==1000)
  {
    ampTajmera = 0;
    provjeriMaxAmp = maxAmplituda;
    maxAmplituda = 0;
  }

}

void reset(){
  index = 0;
  Match = 0;
  maxNagib = 0;
}




void drugeLediceUgasi(int LED5, int LED4,int LED3,int LED2,int LED1){

  digitalWrite(LED5,0);
```

```
  digitalWrite(LED4,0);
  digitalWrite(LED3,0);
  digitalWrite(LED2,0);
  digitalWrite(LED1,0);
}


void provjeraNiza(){
  if(frekvencija>328&frekvencija<335){
    drugeLediceUgasi(2,3,4,5,6);                    // 1. žica e
    digitalWrite(3,1);
    ispravnaFreq = 331;//432
  }


 if(frekvencija>245&frekvencija<253){
    drugeLediceUgasi(2,3,5,6,7);                    // 2. žica h
    digitalWrite(2,1);
    ispravnaFreq = 249;
  }

  if(frekvencija>205&frekvencija<213){
    drugeLediceUgasi(2,4,5,6,7);
    digitalWrite(5,1);                              // 3. žica gis
    ispravnaFreq = 209;
  }

  if(frekvencija>161&frekvencija<169){
    drugeLediceUgasi(2,3,5,6,7);                    //4. žica  e
    digitalWrite(6,1);
    ispravnaFreq = 165; //185
  }
}


void provjeraFrekvencije(){
  if(frekvencija>ispravnaFreq+1){
    analogWrite(A3,255);//prva desno

  }
  if(frekvencija>ispravnaFreq+2){
    analogWrite(A2,255);//druga desno

  }
  if(frekvencija>ispravnaFreq+3){
    analogWrite(A1,255);//treci desno
  }
  if(frekvencija>ispravnaFreq-1){
    analogWrite(A5,255);//prva lijevo
  }
  if(frekvencija<ispravnaFreq-2){
    digitalWrite(9,1);//druga lijevo
  }
  if(frekvencija<ispravnaFreq-3){
    digitalWrite(8,1);//treca lijevo
     //analogWrite(A5,0);//prva lijevo
    // digitalWrite(9,0);//druga lijevo
  }
  if(frekvencija>ispravnaFreq-
1&frekvencija<ispravnaFreq+1)//if(frekvencija==ispravnaFreq)//ako  je  dobra
frekvencija upali zelenu
```

```
    analogWrite(A4,255); //glavna tocna
    delay(100);

   //if(frequency==0)//ako je dobra frekvencija upali zelenu

  }



void ugasiSveLedice(){
  digitalWrite(2,0);
  digitalWrite(3,0);
  digitalWrite(5,0);
  digitalWrite(6,0);
  digitalWrite(8,0);
  digitalWrite(9,0);
  analogWrite(A1,0);
  analogWrite(A2,0);
  analogWrite(A3,0);
  analogWrite(A4,0);
  analogWrite(A5,0);
}

void loop(){

  ugasiSveLedice();

  if (provjeriMaxAmp>ampPrag){
    frekvencija = 38462/float(period);  //float(period);
  }
    if(provjeriMaxAmp>ampPrag)

{
int filtriranaFreq;
if (frekvencija>160&frekvencija<335)
{
filtriranaFreq = frekvencija;
}
/*} else {
filteredFreq = 0;//
}*/


  //provjeriFrekvenciju();

  provjeraNiza();
  provjeraFrekvencije();
  Serial.println(filtriranaFreq);
}

  delay(400);

}
```