

Električni bubenjevi upravljeni mikroupravljačem

Petar Klinac, Mate Pušić

7. lipnja 2022.



Sadržaj

1	Uvod	1
2	Arduino Due	2
2.1	Arduino Due specifikacije	2
3	Korišteni računalni programi	3
3.1	PlatformIO	3
3.2	Addictive Drums 2	4
4	Akustični bubenjevi	4
5	MIDI standard	6
5.1	"Channel Voice" poruke	6
6	Izrađeni električni bubenjevi	7
6.1	Snare bubenj	7
6.2	Tom-Tom bubenj	9
6.3	Rim	10
6.4	Bass pedala	11
6.5	Hi-Hat pedala	12
6.5.1	Arduino Nano na Hi-Hat pedali	13
6.6	Ride činela	14
6.7	Ostale činele	15
7	Ožičenje	16
7.1	Spoj za ispravljanje signala	17
7.2	Analogni multiplexer	18
8	Konstrukcija	19
9	Tijek izrade	20
9.1	Izrada bubenjeva	20
9.2	Izrada konstrukcije	21
9.3	Izrada pedale	22
9.4	Izrada činele	23
9.5	Izrada tiskane pločice	24
10	Glavni program	25
10.1	Očitani napon sa piezo senzora na Analognom ulazu	25
10.2	Obrađivanje očitanog napona: Klasa <code>hitDetect()</code>	26
10.2.1	Podesivi parametri klase <code>hitDetect()</code>	26
10.2.2	Varijable i dodatni parametri	26
10.2.3	<code>readNewHitSensor()</code>	27
10.2.4	<code>process()</code>	28
10.3	Obrađivanje ulaza sa Hi-Hat pedale: Klasa <code>pedalHH()</code>	30
10.3.1	Varijable	30
10.3.2	Podesivi parametri	31
10.3.3	<code>processSpeed()</code>	31

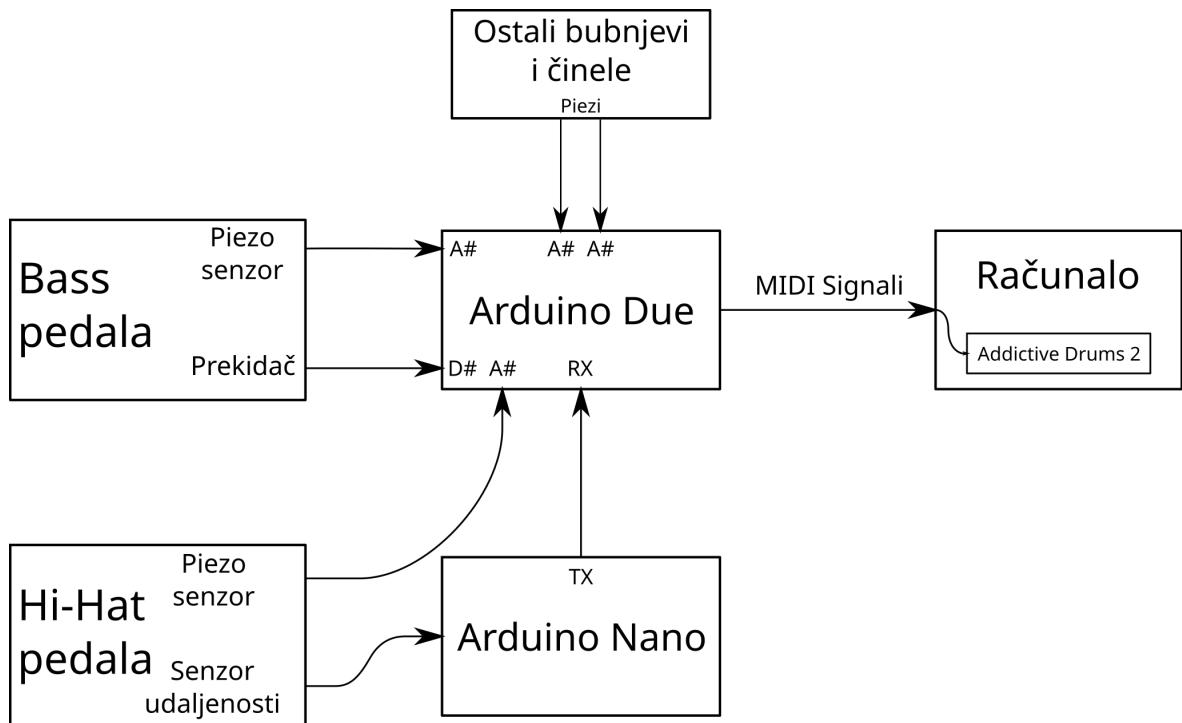
10.3.4	processPedalCC()	32
10.3.5	readNewOpenSensor()	32
10.3.6	process()	32
10.4	Obradivanje ulaza sa Tom-Tom bubenja: Klasa twoZone()	33
10.4.1	Podesivi parametri	34
10.4.2	applySettings()	34
10.4.3	process()	34
10.5	Glavna datoteka programa: main.cpp	35
11	Analiza ulaza sa Snare bubenja.	36
12	Zaključak	38

1 Uvod

Bubnjevi su jedan od glavnih instrumenata u većini glazbenih djela. Glavna svrha im je da daju pjesmi ritam i da pomognu drugim glazbenicima ostati u ritmu. Naime, bubenjevi su glasni i veliki, što onemogućuje sviranje u malim prostorima ili u prostorima u kojima ima puno ljudi (npr. stambena zgrada).

Električni bubenjevi riješavaju ta ograničenja. Oni su tiki i uglavnom manji u veličini od akustičnih bubenjeva. Na njima razni senzori očitavaju udarce na pojedinim bubenjevima i činelama, zatim te informacije šalju računalu koje pušta odgovarajući zvuk za pojedini bubenj i činelu.

Pojednostavljena blok shema ovoga projekta se vidi na slici 1.



Slika 1: Blok shema projekta

2 Arduino Due

Arduino Due [2] je mikroupravljačka pločica sa integriranim Atmel SAM3X8E 32-bitnim ARM procesorom. Ima 54 ugrađenih digitalnih ulaza/izlaza, od kojih 12 mogu stvarati PWM signal, 12 analognih ulaza, procesor koji operira na frekvenciji od 84MHz i ugrađene DEBUG pinove preko OpenOCD (Open On-Chip Debugger).

Arduino Due bio je odabran za izradu ovoga rada zbog velikog broja ugrađenih analognih ulaza, 12-bitnog analogno-digitalnog pretvornika, visoke frekvencije procesora, velike količine radne memorije, kompatibilosti sa već postojećim bibliotekama i mogućnosti da se prikaže na računalu na koji je spojen kao MIDI uređaj.



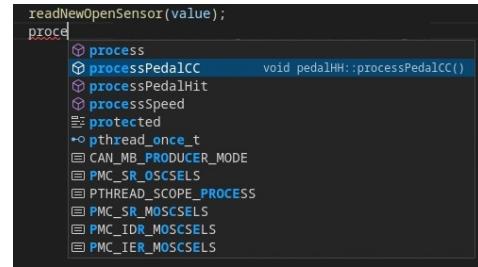
Slika 2: Arduino Due

2.1 Specifikacije mikroupravljača Arduino Due

· Mikroupravljač:	AT91SAM3X8E
· Radni napon:	3,3 V
· Preporučeni ulazni napon:	7-12 V
· Granice ulaznog napona:	6-16 V
· Digitalni I/O pinovi:	54
· Analogni I/O pinovi:	12
· FLASH memorija:	512 kB
· SRAM:	94 kB
· Frekvencija procesora:	84 MHz
· Širina:	101,52 mm
· Duljina	53,3 mm
· Masa:	36 g

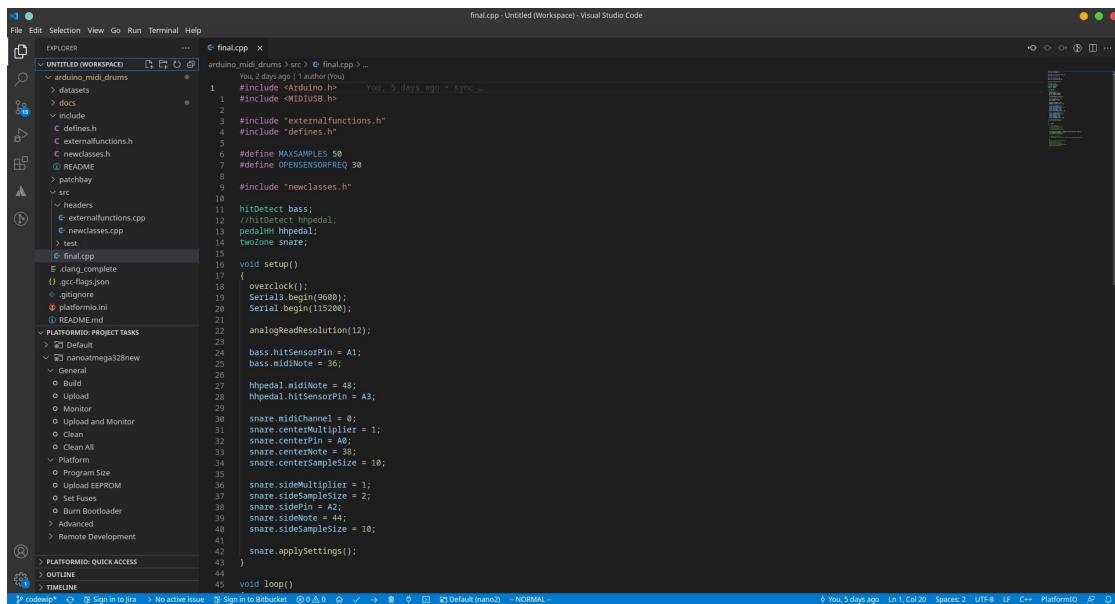
3 Korišteni računalni programi

Za stvaranje programa za Arduino Due bio je odabran program Visual Studio Code s dodatkom PlatformIO za kompajliranje i prenasanje programa na Arduino Due. Visual Studio Code je bio odabran zbog naprednih funkcija koje pruža programeru (npr. IntelliSense, što omogućuje pametno automatsko dovršavanje koda - vidljivo u slici 3), interaktivno debuganje softvera unutar programa i velike količine dodataka koji daju programu dodatne mogućnosti.



```
readNewOpenSensor(value);
process
processPedalCC void pedalH::processPedalCC()
processPedalHit
processSpeed
protected
pthread_once_t
CAN_MB_PRODUCER_MODE
PMC_SR_OSCSELS
PTHREAD_SCOPE_PROCESS
PMC_SR_MOSCSELS
PMC_IDR_MOSCSELS
PMC_IER_MOSCSELS
```

Slika 3: IntelliSense dovršavanje koda

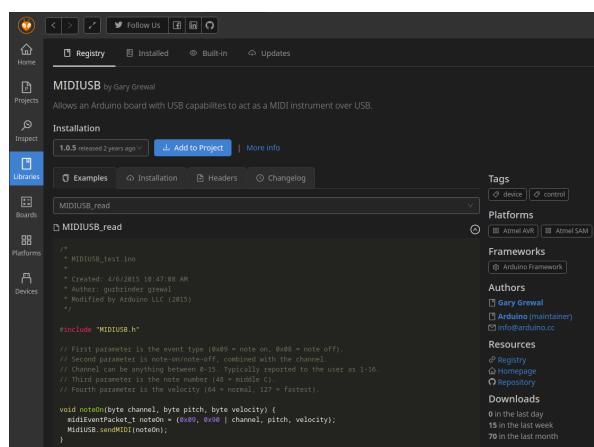


Slika 4: Sučelje programa Visual Studio Code

3.1 PlatformIO

PlatformIO se sastoji od PlatformIO IDE i PlatformIO Core. PlatformIO Core je skup alata napisani u Pythonu koji pruža platformu za izradu programa za integrirane uređaje, ugrađeno skidanje i organiziranje vanjskih biblioteka i monitor za serial port.[3]

PlatformIO IDE ugrađuje alate od PlatformIO Core unutar Visual Studio Codea zbog lakšeg korištenja. Uz to i sruđuje sa IntelliSense, pruža mogućnost debuganja uređaja unutar Visual Studio Code, automatsko kreira projekte za pojedini uređaj i ima ugrađen preglednik za biblioteke (Slika 5). [4]



Slika 5: Ugrađen preglednik za biblioteke

3.2 Addictive Drums 2

Zbog nedovoljne memorije i nemogućnosti Arduina Due da reproducira visoko kvalitetan zvuk, potrebno je koristiti dodatan računalni softver koji prihvaca MIDI podatke Arduina Due i pušta odgovarajuće uzorke (kratke zvučne datoteke) bubnjeva ili činela. Za tu potrebu je bio odabran program Addictive Drums 2 od XLN Audio.

Program je odabran zbog velikog broja ugrađenih, visoko kvalitetnih uzoraka bubnjeva, činela i drugih sličnih instrumenta, zbog kompatibilnosti sa MIDI uređajima i zbog mogućnosti korištenja ASIO zvučnih drivera za izlazni zvuk. ASIO driveri omogućuju vrlo kratak razmak između programa od kada primi MIDI signal do puštanja zvuka na zvučnicima računala. Uz navedene prednosti, program ima ugrađenu miksetu za pojedine bubnjeve, funkciju mijenjanja na koje MIDI signale reagira pojedini bubanj (tzv. "Map window") i vizualizaciju jačine udaraca.



Slika 6: Sučelje programa Addictive Drums 2 [5]

4 Akustični bubnjevi

U set akustičnih bubnjeva se podrazumijevaju i bubnjevi i činele. Standardni set bubnjeva (Slika 7) sastoji se od jednog para "Hi-Hat" činela, jednog "Snare" bubnja, jednog "Bass" bubnja, više "Tom-Tom" bubnjeva i više različitih činela. Svi bubnjevi i činele se razlikuju po promjeru i materijalu od kojih su izrađeni, i bubnjevi ili činele pod istim imenom mogu imati različite promjere i materijale od kojih su izrađeni.



Slika 7: Standardni set bubnjeva [6]



Slika 8: Snare bubanj [7]

Snare bubanj jedan je od glavnih bubenjeva u setu bubenjeva, i često se koristi samostalno izvan seta. Može proizvoditi četiri različita zvuka ovisne o tome kako se svira. Može se udariti izravno po sredini, praveći puni zvuk (tzv. "Open Hit"), može se udariti bliže rubu bubenja, praveći plići zvuk ("Shallow Hit"), može se udariti rub bubenja bez udaranja glave bubenja ("RimClick") i može se istovremeno lupiti i glava bubenja i rub bubenja ("RimShot").



Slika 9: Bass bubanj sa pedalom [8]



Slika 10: Hi-Hat na stalku sa pedalom.

Hi-Hat zadnji je glavni dio seta bubenjeva. Sastoji se od gornje i donje Hi-Hat činele koje se dodiruju i stalka na kojem stoje. Stalak ima ugrađenu pedalu na dnu kojom se može tijekom sviranja postaviti razmak između činela i pritisak kojim gornja činela pritišće u donju. Veći pritisak gornje činele skratiti će vrijeme koje treba čineli da prestane vibrirati i stvarati zvuk nakon udarca, a povećanje razmaka će produljiti to vrijeme.

Hi-Hat može proizvoditi pet različita zvuka ovisno gdje se udari. Može se udariti po rubu činele stranom palice ("Shaft"), može se udariti vrhom palice blizu ruba činele ("Tip"), može se činela udariti blizu vrha, gdje je učvršćena u stalak ("Bell"). Također može proizvesti zvuk od naglog zatvaranja, tako da gornja činela lapi u donju. Ako Hi-Hat ostane zatvoren nakon tog udarca, zvuk će trajati kratko ("Foot Close"), a ako se brzo nakon Hi-Hat otvoriti, zvuku će trebati vrijeme da prestane ("Foot Splash").



Slika 11: Zatvoreni Hi-Hat



Slika 12: Djelomično otvoreni Hi-Hat



Slika 13: Otvoreni Hi-Hat

5 MIDI standard

MIDI (Musical Instrument Digital Interface) je tehnički standard koji opisuje komunikacijski protokol, digitalno sučelje i elektroničke utore koji povezuju veliku količinu elektroničkih glazbenih instrumenata. [9] MIDI koristi MIDI poruke kao upute koje kontroliraju dio instrumenta. MIDI poruka se sastoji od bajta stanja, koji ukazuje vrstu poruke, i dodatna dva bajta podataka. MIDI poruke se mogu dijeliti na kanalne poruke i sistemske poruke. U kanalnim porukama se kanal uključuje u bajt statusa, dok sistemske poruke nemaju uključen kanal u bajt statusa. [10]

MIDI kanalne poruke se dodatno dijele na "Channel Voice" poruke i "Mode" poruke. "Mode" poruke mijenjaju kako instrument reagira na MIDI signale (npr. mogu se resetirati sve upravljačke vrijednosti ili podesiti instrument da ignorira lokalni upravljač). "Channel Voice" poruke se koriste u ovom projektu i šalju informacije o glazbenoj izvedbi. Vrste "Channel Voice" poruka u ovome projektu su "Note On", "Note Off", "Control Change" i "Aftertouch". [10]

5.1 "Channel Voice" poruke

"Note On" i "Note Off" poruke sastoje se od bajta statusa i dva dodatna bajta koji govore koja tipka je bila pritisнута i sa kojom jačinom. U ovom projektu se koriste za signaliziranje koji bubanj ili činela je bila udarena i koliko jako. Te dvije vrijednosti imaju maksimalnu vrijednost od 127. [10]

"Control Change" poruke se koriste za upravljanje širokog izbora funkcija unutar instrumenta. Te poruke se sastoje od status bajta i dodatna dva bajta koji govore koju upravljačku vrijednost mijenjaju i na koju vrijednost. [10] Upravljačke vrijednosti su definirane [11], a vrijednost na koju se mijenja može biti između 0 i 127.

"Aftertouch" poruke se koriste za slanje količine pritiska na tipku kada je već stisnuta. Sastoje se od bajta statusa i podatka koliko je jako pritisnuto. [10] U ovom projektu je planirano da se koristi za jačinu pritiska na zatvorenu Hi-Hat pedalu.

6 Izrađeni električni bubenjevi

6.1 Snare bubanj

Najkompleksniji po izvedbi od svih bubenjeva ovog projekta je upravo Snare bubanj. Često nazivan i doboš, na akustičnom je setu najčešće udaran bubenj s vrlo visokom razinom podesivosti. Većina elektroničkih setova bubenjeva prema izvedbi ne razlikuje snare bubanj od ostalih zbog uštede materijala pri proizvodnji. Ta puno manja dimenzija, s obzirom na akustični element koji pokušava zamijeniti, predstavlja prepreku pri navikavanju te je iz tog razloga odlučeno izraditi Snare bubanj u dimenzijama bliskim akustičnom Snare bubenju.

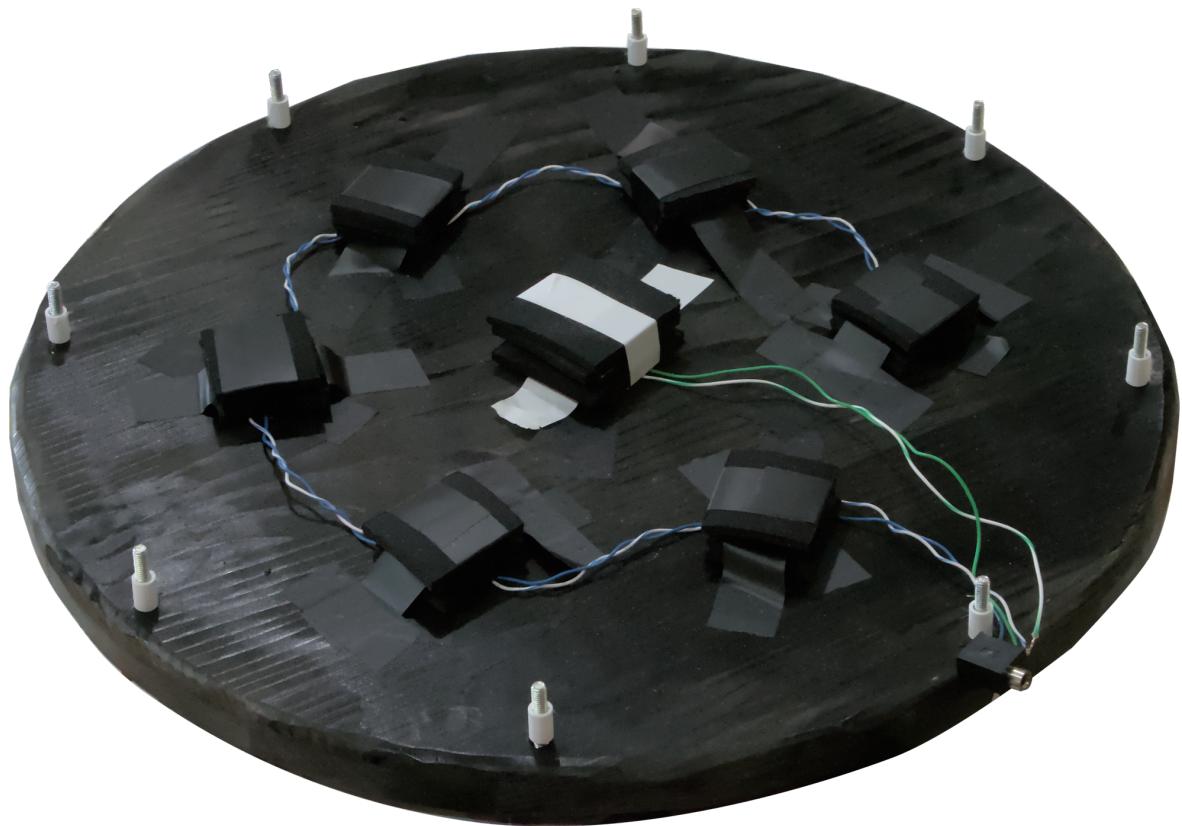


Slika 14: Izrađeni električni Snare bubanj

Fizička je građa elektroničkog Snare bubenja ovog projekta značajno drugačija od građe akustičnog. Zbog čvrstoće samog bubenja te elemenata koji povezuju bubenj s kosturom, za temelj je izabrana ploča od lijepljenoga drva. Njena je debљina 18mm, a promjer je 350mm. Na nju je učvršćeno 7 piezo senzora te dva 3,5mm priključka za ožičenje. Jedan je piezo senzor samostalno povezan na jedan kontakt 3,5mm priključka dok su ostalih 6 pieza paralelno spojeni i dijele drugi kontakt na tom priključku. Takav je dizajn ožičenja odabran s ciljem postizanja što autentičnijeg zvuka te osjećaja sviranja sličnom onom na akustičnom bubenju. Zasebni piezo senzor određen je za središnju zonu bubenja, a ostali su paralelno spojeni kako bi funkcionali poput piezo senzora u obliku vijenca te oni određuju zonu bližu rubu bubenja. Na drugi 3,5 mm priključak spojeni su jedino piezo senzori iz ruba bubenja (tzv. *Rim*, detaljiziran na stranici 10). Zajedno te dvije zone omogućuju razlikovanje "Open Hit" i "Shallow Hit" udaraca (Str. 5).

Između površine koja je namijenjena za udaranje palicama i senzora nalazi se gumeno-spužvasti materijal dimenzija 30x30x12mm. Njegova je funkcija prenošenje sile udarca na senzor dok ga istovremeno zaštićuje zbog elastičnosti.

Površinu za udaranje palicama čine plastika Hobby Color i gumeno-spužvasti materijal, obje promjera 350mm i svaka debljine 3mm. Njihova kombinacija se pokazala kao najpogodnija zamjena opne akustičnog Snare bubnja. Plastika je dovoljno tanka i elastična da bi ukupna veličina i masa bubnja ostale prihvatljive jer bi upotreba materijala poput ranije spomenute drvene ploče bila neekonomična. Gumeno-spužvasti upija potencijalno veliku silu udaraca te zbog svoje elastičnosti dopušta bubnjarskoj palici da se relativno realistično odbije nakon udarca. Plastika podupire gumeno-spužvasti materijal te površini daje ravninu i čvrstoću.

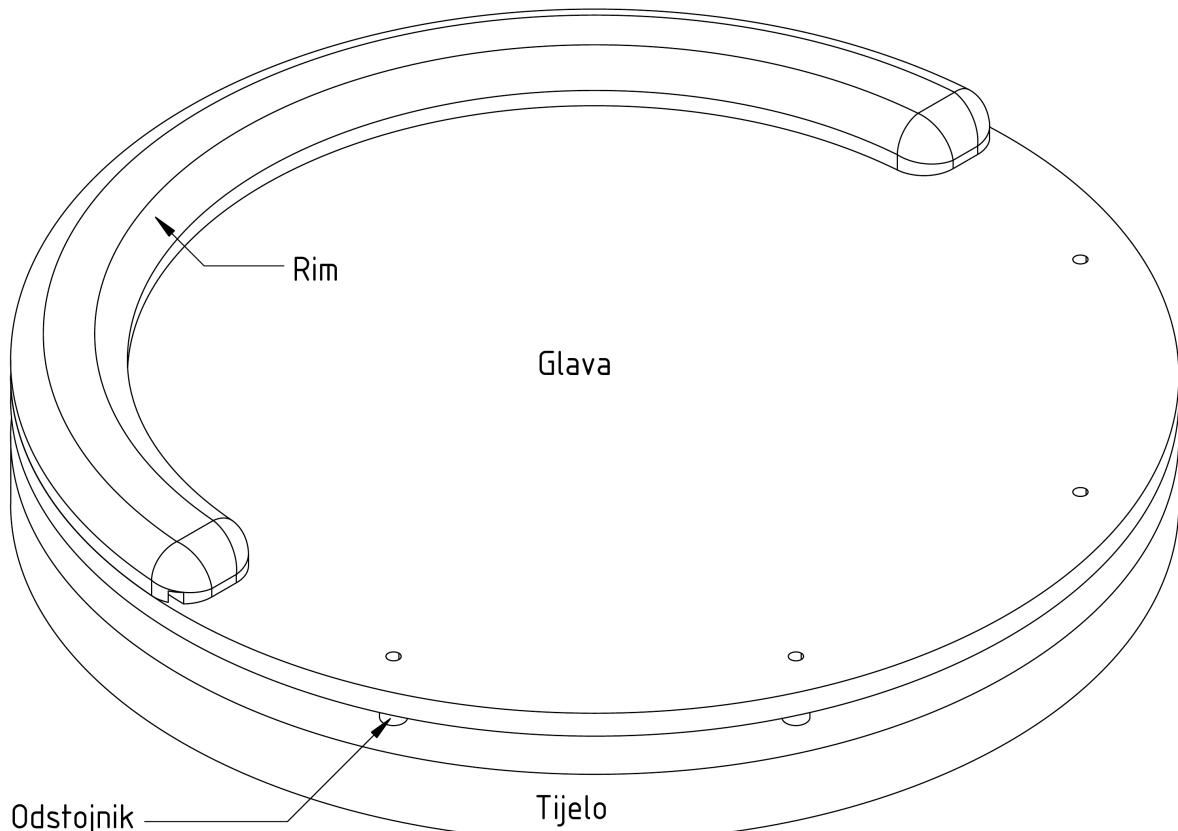


Slika 15: Raspored piezo senzora na bubnju

Drvena ploča, plastika te gumeno-spužvasti materijal su povezani s 8 M4 vijaka, matica te podloški, a zbog ravnomjernosti napetosti podloge po kojoj se udara (plastika, podloga za miš) dizajnirane su posebni odstojnici visine 10mm koje se nalaze između drvene ploče i plastike. One su izrađene pomoću 3D printera.

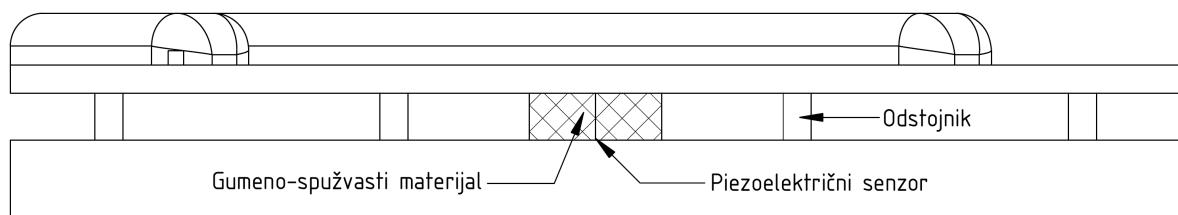
6.2 Tom-Tom bubanj

Tom-Tom bubanj drugi je od dvije vrste bubenja koji su bili izrađeni. Jednostavniji je od Snare bubenja zbog toga što Tom-Tom bubenjevi mogu proizvoditi samo dva različita zvuka. Izrađeno je dvije vrste Tom-Tom bubenja koji se razlikuju u dimenzijama. Oba bubenja su visoka 40 mm, a promjeri su 250 mm (Mid Tom, Slika 16) i 300 mm (Floor Tom). Konstrukcija bubenja identična je konstrukciji Snare bubenja (Stranica 7). Razlika je u količini senzora i 3,5 mm priključaka. Zbog toga što na Tom-Tom bubenju okružnja zona nije potrebna, on ima samo piezo senzor u sredini i u Rimu. Zbog istog razloga, Tom-Tom bubanj ima samo jedan 3,5 mm priključak.



Slika 16: Izgled Tom-Tom bubenja

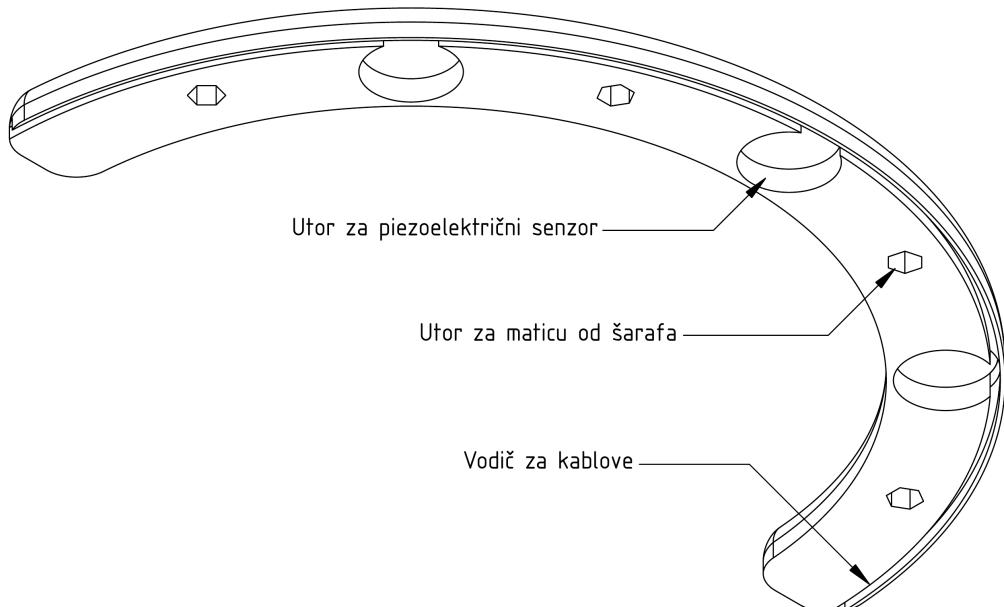
Na sredini unutrašnjosti tijela bubenja učvršćen je okrugli piezoelektrični senzor promjera 20 mm. Postavljen na senzor je gumeni-spužvasti materijal dimenzija 30 mm x 30 mm sa visinom od 12 mm (Slika 17). Taj materijal ostvaruje kontakt između piezoelektričnog senzora i glave bubenja i dodatno smanjuje fleksibilnost plastične glave stoga glava manje upija udarac.



Slika 17: Položaj senzora unutar Tom-Tom bubenja

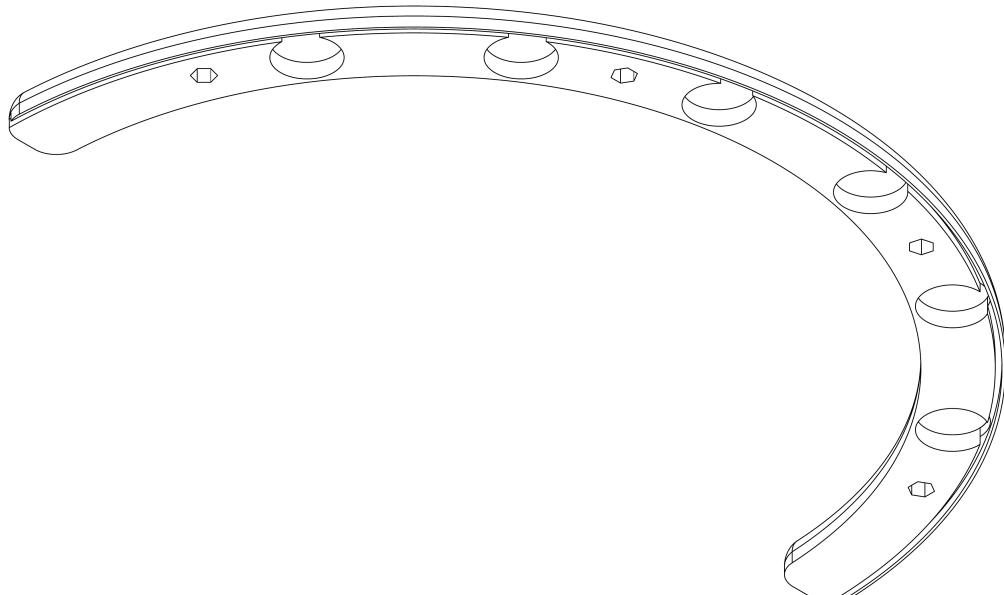
6.3 Rim

Rim je područje ruba glave. Lupanje tog dijela bубnja proizvodi drugačiji zvuk od lapanja sredine glave, stoga je bilo potrebno dizajnirati poseban dio koji se učvršćuje na vrh bубnja i u kojem su postavljeni piezoelektrični senzori (Slika 18). Ovalnog je oblika zbog postizanja ugodnijeg načina sviranja te nam tako sa strane proizvođača olakšava smještanje piezo senzora.



Slika 18: Izgled dna rima

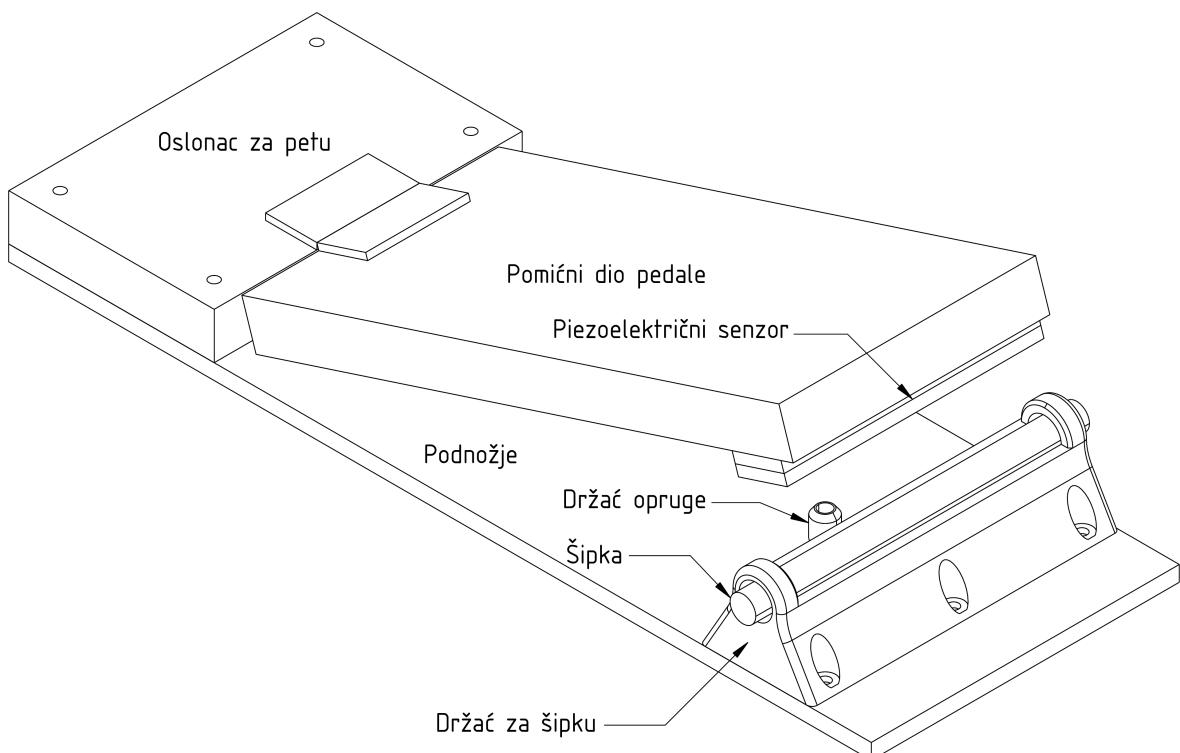
Zbog veličine Snare bубnja, dizajniran je dodatan Rim sa više piezoelektričnim senzorima (Slika 19).



Slika 19: Izgled dna rima za Snare bубnaj

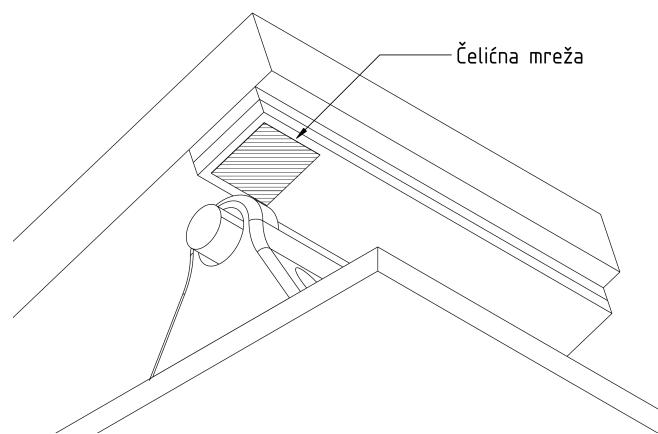
6.4 Bass pedala

Bass pedala dizajnirana je slično pedali akustičnih bubnjeva. Podnožje pedale izrađeno je od Hobi plastike i od ispod učvršćen je komad metala koji očvršćuje pedalu i spriječava da se savija. Na jednom kraju podnožja učvršćen je oslonac za petu od stopala izrađen od rezanog lijepjenog drva. Na oslonac je pomoću panta učvršćen pomični dio pedale koji se stiče. Odozdo, na suprotnom dijelu pomičnog dijela pedale, učvršćen između dva sloja gumenog-spužvastog materijala, je piezoelektrični senzor koji osijeti udarce Bass pedale. Taj senzor udara u metalnu šipku koja je učvršćena na podnožje pedale pomoću posebno 3D isprintanog plastičnog dijela. Pomični dio pedale, nakon pritiska, na otvoreno stanje vrača opruga.



Slika 20: Izgled Bass pedale

Zbog mogućnosti piezoelektričnog senzora da generira napon i kada je pedala zatvorena dodatno stiščući pedalu, bilo je potrebno dodati poseban električni prekidač koji se zatvara kada pedala dotiče šipku. Taj prekidač sastoji se od same šipke i čelične mreže koja se nalazi iznad piezoelektričnog senzora, na vrhu gumenog-spužvastog materijala. Svrha tog prekidača je utvrđivanje da li je pedala bila otvorena nakon posljednjeg udarca.

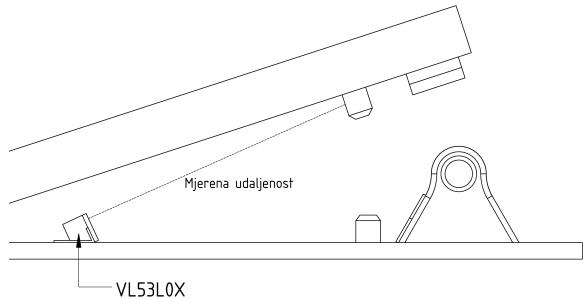


Slika 21: Položaj mreže na pedali

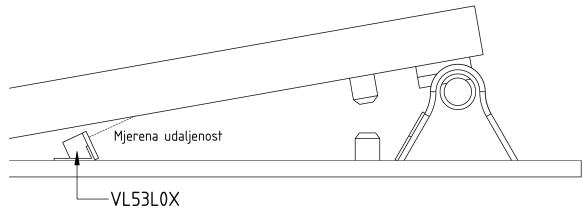
6.5 Hi-Hat pedala

Hi-Hat pedala izrađena je vrlo slično Bass pedali sa dodanim senzorom koji mjeri koliko je pedala otvorena. Za tu svrhu odabran je senzor VL53L0X infracrveni senzor udaljenosti (Slika 22).

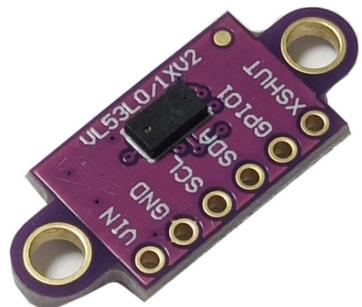
VL53L0X radi na principu mjerjenja upadnog kuta IR svjetlosti koju je odaslao. Sa senzorom se komunicira preko I2C načina komunikacije i može očitati udaljenost svakih 20 ms. Uz prednost velike frekvencije očitanja udaljenosti, velika prednost toga senzora je pomoću čega mjeri udaljenost. Za razliku od ultrazvučnih senzora udaljenosti, na senzore udaljenosti infracrvenom svjetlošću ne utječe kut pod kojim zraka upadne. Ta prednost vrlo je bitna zbog položaja senzora unutar pedale. Senzor je postavljen ispod pomičnog dijela pedale, blizu oslonca. Okrenut je prema drugom kraju pedale i nagibljen tako da mjeri vrh pomičnog dijela pedale kada je pedala otvorena. Takav način mjerjenja poveća raspon udaljenosti koju senzor mjeri iako se sami pomični dio pedale pomiče samo nekoliko centimetara (Slika 23 i 24).



Slika 23: Mjerena udaljenost otvorene pedale



Slika 24: Mjerena udaljenost zatvorene pedale



Slika 22: VL53L0X Infracrveni senzor udaljenosti



Slika 25: Izrađena Hi-Hat pedala

Iako Hi-Hat pedala pravi zvuk ako se naglo zatvori, poseban prekidač koji osjeti je li pedala otvorena kao na Bass pedali nije potreban jer se to može utvrditi preko senzora udaljenosti.

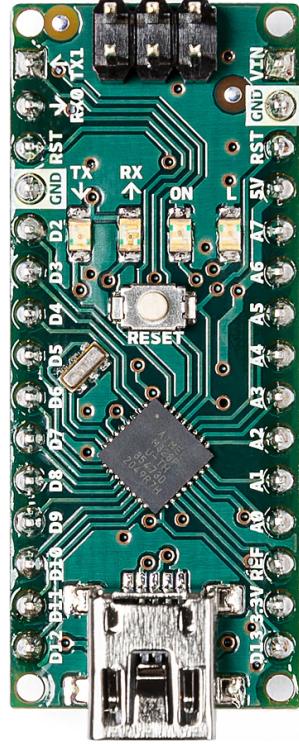
6.5.1 Arduino Nano na Hi-Hat pedali

Zbog toga što VL53L0X koristi I2C komunikaciju, duljina kabela je ograničena na manje od jedan metar. To je nedovoljno za svrhe ovoga projekta zbog toga što bi se pedala nalazila na podu, a Arduino Due u razini prsa svirača. Kao riješenje toj tehničkoj poteškoći bio je uveden Arduino Nano koji je učvršćen za podnožje pedale i koji komunicira sa glavim Arduino Due preko Serial komunikacije.

Arduino Nano komunicira sa senzorom udaljenosti preko I2C komunikacije i zatim šalje očitanu udaljenost preko svog Tx izlaza na RS485 modul koji omogućuje Serial komunikaciju na većim udaljenostima.

Arduino Nano bio je odabran za ovu svrhu zbog svoje male veličine i male potrošnje energije. Zbog male veličine stane ispod pomičnog dijela pedale, a zbog male potrošnje energije se može napajati preko 5V pina na Arduino Due.

```
1 #include <Arduino.h>
2
3 // Biblioteke potrebne za VL53L0X senzor
4 // → udaljenosti
5 #include <Wire.h>
6 #include <VL53L0X.h>
7
8
9 VL53L0X sensor;
10
11 void setup()
12 {
13     Serial.begin(9600);
14     Wire.begin();
15
16     sensor.setTimeout(500);
17     if (!sensor.init())
18     {
19         while (1) {
20             // Blica ugrađenu LED diodu ako senzor ne radi.
21             digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
22             delay(200);
23         }
24
25         // Počne mjerjenje svakih 50 ms
26         sensor.setMeasurementTimingBudget(50000);
27         sensor.startContinuous();
28     }
29
30     void loop()
31     {
32         // Neprestano šalje podatke na glavni Arduino Due
33         Serial.write(sensor.readRangeContinuousMillimeters());
34     }
35 }
```

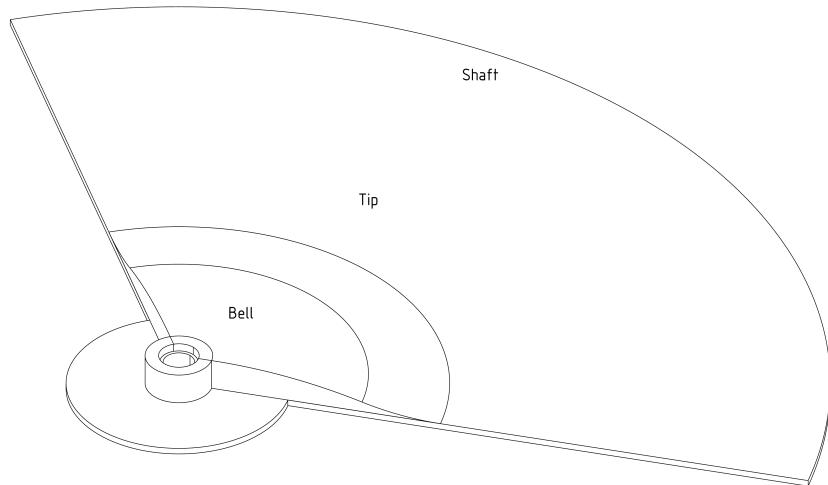


Slika 26: Arduino Nano [12]

Program 1: Program unutar Arduino Nano

6.6 Ride činela

Ride činela jedna je od najvećih činela u setu bubnjeva i, kao Hi-Hat činela, može proizvoditi tri različita zvuka lupanjem palicom. Izrađena je u obliku kružnog isječka polumjera 30 cm. Shaft i Tip zvukovi postižu se dijeljenjem površine činele u područja koristeći više piezoelektričnih senzora, a Bell se postiže printanjem zasebnog dijela koji se lijepi na činelu i u kojemu su učvršćeni piezoelektrični senzori.



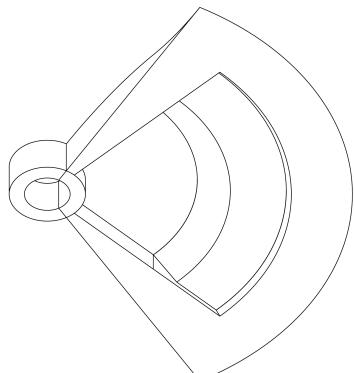
Slika 27: Izgled Ride činele

Na unutrašnjosti dijela za Bell ostavljeno je prostora za više piezoelektričnih senzora kako bi se povećala osjetljivost kroz tvrdnu plastiku samog dijela. Senzori su učvršćeni za printani dio umjesto za plastiku na kojem stoji kako udarac na druga dva područja činele minimalno utječe na senzore unutar Bella (Slika 29).

Činelu je moguće primiti rukom nakon udarca kako bi se rano zaustavio zvuk ("Choke"). To je postignuto printanim dijelom koji je nalijepljen ispod činele koji služi kao gumb kada se pritisne (Slika 31).



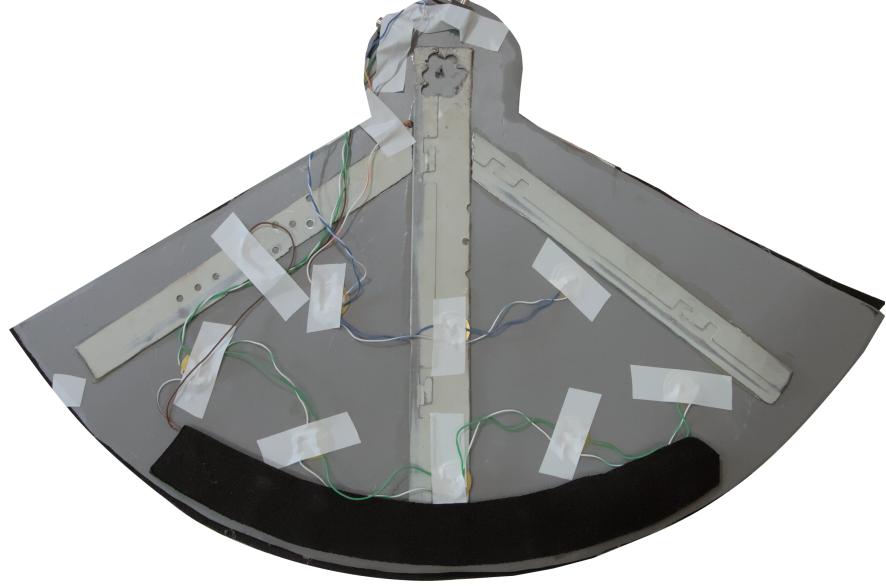
Slika 28: Izgled unutrašnjosti Bella



Slika 29: Izgled unutrašnjosti Bella

Temeljni materijal ove činele je upravo Hobby Collor plastika koja je korištena i u bubnjevima ovog rada. Ona se pokazala kao vrlo izdržljiv materijal koji je vrlo jednostavan za oblikovanje. Na nju je nalijepljen 3D isprintan predmet koji pridonosi autentičnom izgledu činele. Uz to, taj se dio ističe kao lako uočljiva zona činele koja na akustičnom setu proizvodi značajno drugačiji zvuk od udarca po nekom drugom dijelu činele.

Odlučeno je ride činelu izraditi s rasporedom senzora na tri područja: Bell, područje 1 te područje 2. Naziv "Bell" se odnosi na maloprije spomenut 3D isprintan dio, a područja su dodane kako bi se dobilo još više mogućnosti sviranja te činele pa time i autentičnosti izvedbe (Slika 30).



Slika 30: Raspored senzora na Ride čineli

Ride činela se rijetko prigušuje, ali zbog izrade sustava prigušivanja na drugim činelama, odlučeno je skalirati isti sustav na dimenzije ove činele. On se sastoji od gumenospužvastog materijala kao površina ugodna za dodir prstima, dvije plohe čelične mreže koje funkcioniraju kao tipkalo te odstojnika izrađenog pomoću 3D printera (Slika 31).

Na ride se čineli nalazi 9 piezo senzora raspoređenih u zone, jedan sistem prigušivanja te dva 3,5 mm priključka za ožičenje. Dotični se priključci nalaze u blizini rupe za šipku jer je tamo činela najčvršća te je sila udarca u većini slučajeva najmanja.



Slika 31: Pozicija dijela za Choke

6.7 Ostale činele

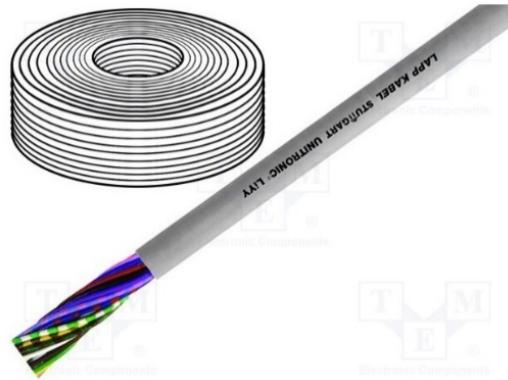
Po uzoru na ride činelu, lijeva, desna i hi-hat činela izrađene su u dimenzijama proporcionalno manjim od ridea. Bell dio je također skaliran na manje dimenzije te je zbog suvišnosti odlučeno izraditi činele sa samo dvije zone. Sistem prigušivanja, limeni pravokutnici te predmet oko rupe za šipku su također umanjeni kako bi odgovarali manjim dimenzijama. Jedina razlika među te tri dimenzijama identične činele je što hi-hat činela nema sustav prigušivanja jer se ta funkcija ostvaruje preko njene pedale o kojoj će kasnije biti govora.

Postupak izrade je identičan izradi ride činele, ali je korištena manja količina piezo senzora. Isti izgled i dimenzije mogu imati i dodatne činele za koje je dizajnirano sklopovlje zbog mogućnosti dodatka više činela u budućnosti.

7 Ožičenje

Kada su bile određene pozicije elemenata u programu bilo je moguće odrediti duljine pojedinih kabela. Odlučeno je koristiti vodove debljine 0.14mm^2 raznih duljina. Koriste se dva tipa kabela – trožilni i dvožilni. Trožilni u većini slučajeva nosi dva signala i masu dok dvožični nosi jedan signal i masu ili napon i masu.

Na krajevima kabela zalemljeni su muški 3.5mm stereo konektori s tri površine – tip (vrh), ring (prsten) te sleeve (rukav).

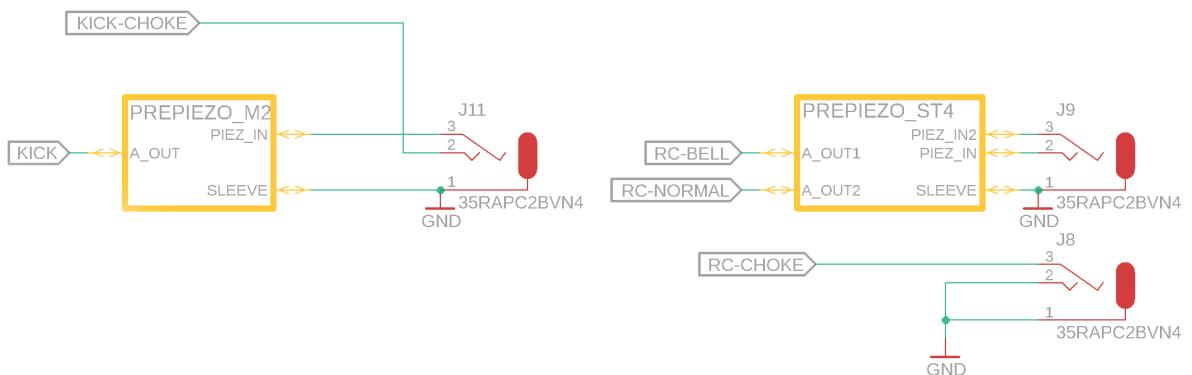


Slika 32: Izgled kabela [1]



Slika 33: 3,5 mm muški konektor za lemljenje

Na svim elementima po kojima se svira te na razvodnoj kutiji nalaze se ženski 3.5mm stereo priključci. Na njihove su kontakte spojeni signali te napajanja prema sljedećoj shemi.

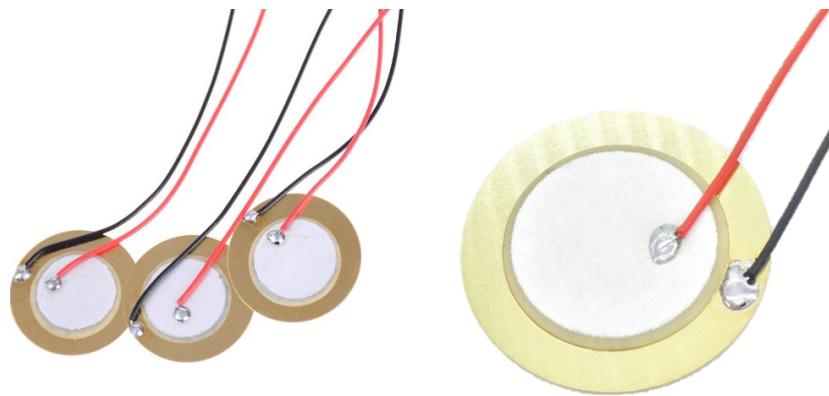


Slika 34: Shema ožičenja ženskog priključka

7.1 Spoj za ispravljanje signala

Piezo elementi su generatori napona uslijed njihove elastične deformacije pod djelovanjem vanjske sile. Nakon deformacije, stvaraju napon obrnutog polariteta dok se vraćaju u prvobitni oblik.

Generirani napon ovisi o strukturi i dimenzijama piezo elementa, a njihov sami otpor je dovoljno velik da jedan drugoga u paralelnom spoju ne mogu pobuditi. Odabrani su senzori okruglog oblika promjera 20 mm te 50 mm (Sl. 35) čiji maksimalni napon pri deformaciji doseže 5V.

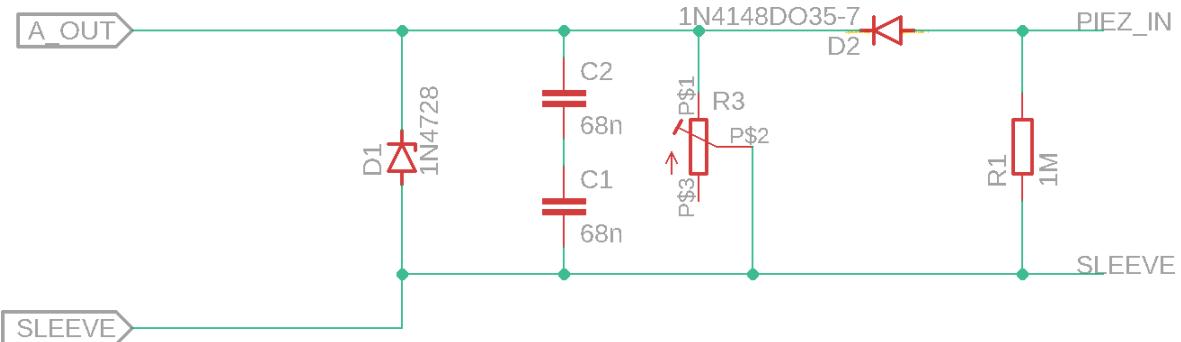


Slika 35: Odabrani piezo elementi (20 mm i 50 mm)

Zbog veličine površina koje se nadziru ovim elementima u ulozi senzora, bilo je neophodno povezati ih paralelno. Tako je postignut efekt dobivanja proizvoljne veličine te oblika mjerene zone pa tako i figurativnog piezo elementa. Nedostatak ovog pristupa je postojanje manje osjetljivih područja koja nastaju na dijelovima zona gdje nema pieza.

Paralelno povezivanje pieza znači da se generirani naponi zbrajaju, a AD pretvornici na analognim ulazima Arduino Due primaju napon do 5V. U slučaju jakog udarca na mjestu između dva paralelno povezana pieza, ukupni napon bi lako mogao prijeći 5V. Uz to, napon obrnutog polariteta koji se stvara nakon deformacije te smetnje koje dođu na nezaštićene vodove stvorili su potrebu za dizajniranjem posebnog sklopa.

Kako bi se odsjekao obrnuto polarizirani napon, postavljena je dioda opće namjene 1N4148 u seriju sa vodom za pozitivni pol pieza. Kako bi se spriječio dovod smetnji na priključak Arduina dok piezo element ne stvara napon, paralelno je postavljen jedan stalni otpornik od $1M\Omega$.



Slika 36: Shema projektiranoga spoja za obradu signala (prepiezo)

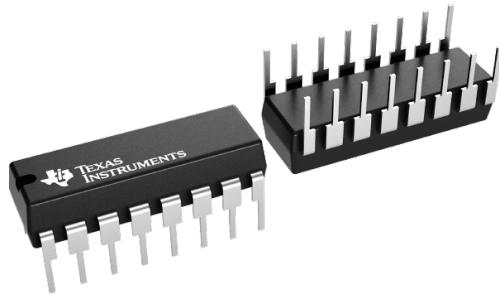
Svaki je piezo senzor drugačije osjetljivosti koja se može namjestiti prije dovoda signala na AD pretvornik projektiranjem nekog otpora između vodova signala s pieza. Kako bi se olakšalo umjeravanje, odlučeno je postaviti trimer otpornik te ručno namjestiti osjetljivost svakog signala svake površine po kojoj se svira.

Stalne smetnje koje su bile očitavane od strane mikroupravljača dolazile su kroz nezaštićeno ožičenje. Značajno su utjecale na sustav zbog male jačine korisnog signala te su aktivirale tihe udarce. Kako bi se smanjio njihov utjecaj te izravnao izgled signala s piezo elementa nakon udarca, projektirana su dva međusobno serijski spojena keramička kondenzatora iznosa 68 nF u paralelnom spoju s vodovima piezo senzora.

Zener dioda je postavljena kako bi ograničila napon na 5V prije dolaska signala na mikrokontroler. Nju koristimo najčešće kao obrnuto polariziranu jer se može obnoviti nakon prijelaza stope napona probijanja.

7.2 Analogni multiplexer

Zbog nedovoljne količine analognih ulaza na mikrokontroleru, dodani su analogni multiplexeri modela CD4051BE (Slika 37). Glavna uloga svakog multiplexera je proširenje količine ulaza/izlaza. Konkretan model korišten u ovom projektu zahtjeva konekciju na tri digitalna pina za odabir pina čiji će signal biti proslijeden na pin spojen na mikrokontrolerov analogni pin. Tako se zauzimanjem jednog analognog pina dobito proširenje na njih 8, jedini nedostatak je što multiplexeru treba određena količina vremena da promijeni koji pin spaja na pin spojen na mikrokontroler. Ta je količina vremena zanemariva za ovaj projekt jer je frekvencija multiplexera vrlo visoka.



Slika 37: Izgled CD4051BE analognog multiplexera

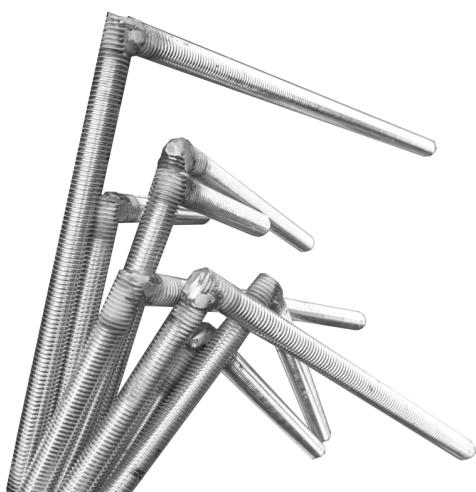
8 Konstrukcija

Glavna komponenta zaslužna za povezivanje svih elemenata je konstrukcija tj. kostur. Izrađena je od cijevi namijenjenih za noge stola, a u ovom projektu služe sličnu svrhu. Neke je bilo potrebno skratiti, neke produljiti varenjem jednih na druge, a na nekima je samo odrezano kvadratno dno. Sve su noge obojene ranije spomenutom bojom u spreju. Međusobno su povezane 3D isprintanim spojnicima te se sličnim učvršćuju šipke koje će biti opisane u nastavku.



Slika 38: 3D render seta bubnjeva

U komponentu konstrukcije ubrajaju se i pune aluminijske šipke koje su razreznane u komade koji su zavareni pod kutom od 90° . Njihova je zadaća povezati bubnjeve i činele sa ranije spomenutim dijelom konstrukcije.



Slika 39: Šipke zavarene pod kutem od 90°

9 Tijek izrade

9.1 Izrada bubenjeva

S obzirom na dostupnost materijala te alata, bilo je potrebno stvoriti novi dizajn svih elemenata. Testirani su mnogi materijali za glavu bubenja te je odlučeno koristiti Hobby Color plastiku te gumenu podlogu za miš. Materijal osnove bubenja je ploča od lijepljjenog drva smreke. Njena čvrstoća je čini idealnom za temelj bubenja, a laka obradivost do sad spomenutih materijala presudna je zbog korištenja ručnih alata.

Drvena je ploča izrezana u n-terokut ubodnom pilom pa je brusilicom oblikovana u krug. Podloga za miš je izrezana uredskim škarama u krug dok je plastika izrezana lemilicom te su neravni rubovi izglađeni brusilicom. Nakon oblikovanja su drvena ploča i plastika oslikani crnom bojom u spreju marke Dupli Fly Color.

Sljedeći je korak bio testiranje pozicije piezo elementa. Njegova krhkost ograničenje je koje zahtjeva da ga se postavi na neelastičnu plohu. S obzirom da je elastičnost površine po kojoj se udara poželjna zbog autentičnog osjećaja sviranja bubenjeva, odlučeno je piezo postaviti na drvenu ploču.



Slika 41: Rezanje podloge za miš

od jakih udaraca. Primjena spomenutih dijelova ostala je ista i u završnom dizajnu.

Nakon tog je koraka sastavljen bubanj bez ožičenja i senzora te je uočen problem oblika plastike. Ona se naime savijala i bila previše slobodna oko ruba. Privremeno je rješenje bilo postaviti kupljene matice na dio vijka između drvene ploče i plastike. Njihove dimenzije nisu odgovarale visini stupca gumene podloge koja je postavljena na senzor pa je odlučeno izmodelirati odstojnike tj. matice odgovarajućih dimenzija te ih isprintati na 3D printeru. To se rješenje pokazalo vrlo dobrim te je ostala u finalnom dizajnu.



Slika 40: Radno okruženje

Povezivanje svih dijelova ostvareno je bušenjem 8 rupa na jednakoj udaljenosti jedne od druge te provođenjem M4 vijaka koji su pričvršćeni zatezanjem matice. Vibracije koje su se prenosile preko samih vijaka do piezo senzora na drvu nisu bile dovoljno jake da bi se koristile za detekciju. Taj se izazov riješio dodavanjem stupca od 5 kvadratična gumene podloge za miš na senzor (Slika 41). Podloga za miš bolje prenosi vibracije s glave bubenja na senzor od dosadašnjeg rješenja, ali i zbog svoje elastičnosti dobro štiti senzor

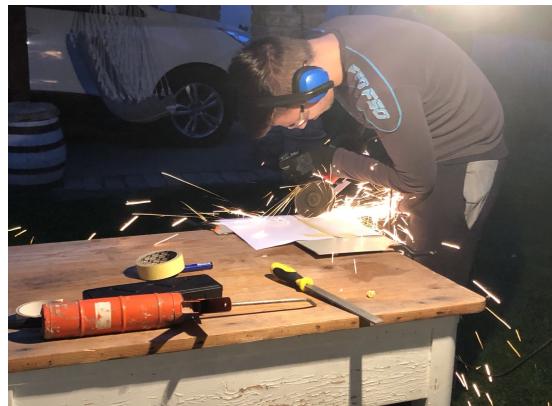
S obzirom da mnogi setovi bubnjeva imaju i dodatke poput cowbella, bongosa, tam-burina i sl. postavilo se pitanje kako uvesti i tu funkciju. Proučavanjem komercijalnih električnih bubnjeva, zaključeno je da se površine postavljene po rubu glave bubnja najčešće koriste za te zvukove. Ti se rubovi na akustičnim bubnjevima rijetko koriste pa je zamjena autentičnog zvuka udaranja o rub nekim drugim zvukom prihvatljiva. S obzirom na dostupnost alata, bio je problem kreirati bilo što sa zahtjevima tog ruba. On treba sadržavati rupe za maticice kako bi se mogao učvrstiti na ostatak bubnja, uz to treba imati rupe promjera 20mm kako bi se mogli postaviti senzori te usku kanalicu za vođenje žica. Iz tog razloga odlučeno je kreirati dizajn u programu za 3D modeliranje te testirati integritet plastike za 3D printer. Njen je oblik sličan obliku ruba komercijalnih električnih bubnjeva no opseg je prepolovljen zbog ekonomičnosti. Ta se opcija pokazala pouzdanom i jednostavnom te se koristi u finalnom dizajnu. Svi su bubnjevi izrađeni istim postupkom.

9.2 Izrada konstrukcije

Velik je izazov cijelog projekta odabir odgovarajućih komponenata i materijala. Iz opisa namjene odabranih nogu za stol zaključeno je da mogu podnijeti težinu cijelog seta bubnjeva, konkretan materijal od kojih su izrađene je vrlo lagan te su šuplje poput cijevi što je poželjno. Četiri su noge po dizajnu postavljene okomito dok su tri vodoravno. Srednje dvije noge su bile prekratke pa su zavarene svaka s još jednom kako bi ukupna duljina bila veća.



Slika 42: Brušenje



Slika 43: Rezanje



Slika 44: 3D isprintani spojnik

Isto je rješenje odabранo za jednu horizontalnu nogu. Kod nje i ostalih vodoravno položenih nogu bilo je potrebno ukloniti kvadratni dio na jednom završetku noge, on bi služio za učvršćivanje drvene plohe stola. Svi su rezani rubovi te vareni spojevi pobrušeni kako bi se uklonili oštari rubovi i neravnine nastale varenjem zbog sigurnosti krajnjeg korisnika i estetskog izgleda.

Za povezivanje cijelog kostura zaduženi su 3D isprintani dijelovi (Slika 44). Oni su s unutarnjih strana premazani tekućom gumom koja nakon 24h očvrsne te poboljša silu trenja koja doprinosi čvrstoći spoja. Ključno je da se niti jedan element ne pomiče nakon što su M4 vijci stegnuti odgovarajućim maticama.

9.3 Izrada pedale

Pedale su oblikom najjednostavnije za obraditi. Oblikom su jednostavne, sastoje se od plastike, koja je i dosad korištena, kao temeljnog dijela koji povezuje sve, dva komada drva te 3D isprintane komponente. Spojeve čine M4 vijci te matice.

Isti postupak korišten je za izradu obje pedale, prvotno je izrezana plastika u oblik pravokutnika, zatim su izrezani također pravokutni dijelovi drvenih ploča različitih dimenzija. Pant slobode kretanja 120° postavljen je na rubove oba komada drva, probušeno je 6 rupa na odgovarajućim mjestima te je sve učvršćeno zajedno.

U manjem komadu drva su izbušene još 4 rupe u svakom uglu zbog poboljšanja integriteta pedale. 3D isprintani dio također je odlučeno učvrstiti vijcima te su po predlošci izbušene rupe. Nakon toga je postavljena kraća navijena šipka u predviđeni utor te je protukliznim maticama učvršćena. Probušena je još po jedna rupa i u plastici i u većem drvenom dijelu kako bi se mogla učvrstiti opruga između njih. Opruga je odabrana prema osjećaju u lokalnoj željezariji.



Slika 45: Pedala u izradi

Kako bi se utišao zvuk koji pedala fizički proizvodi, dodana je podloga za miš na dijelu površine većeg drvenog dijela koja udara o šipku. Odlučeno je pozicionirati piezo element između dvije podloge za miš pa bi one imale i ulogu prenošenja sile na senzor. Potom je zaledljen 3,5mm priključak koji nosi signal s piezo senzora.

Od ovog koraka, postupak izrade poklapa se jedinu u završnom koraku koji je navijanje kukice za špagu te vezanje špage. Razlog razilaženja postupka izrade su elektronički zahtjevi pojedine pedale. Hi-Hat pedala u svom sastavu koristi i senzor za mjerjenje udaljenosti pa je potrebno više žica koje nose signal te shodno tome još jedan 3,5mm priključak. Kick (Bass) pedala koristi samo jedan priključak koji nosi masu i signal.

9.4 Izrada činele

S obzirom da je odabran nesvakidašnji oblik činele, odabran je jednako neobičan alat za oblikovanje. Stara lemilica s vrlo oksidiranim vrhom pokazala se kao idealni alat jer se s njom mora sporo rukovati za razliku od ubodne pile koja je tijekom testiranja značajno oštećivala plastiku. Nakon njenog izrezivanja, uočeno je da je vrlo savitljiva te da je treba nekako poduprijeti, a to je odlučeno napraviti komadima lima izrezanima od bočnih strana kućišta za računalo. Oni su zalijepljeni epoxy ljepilom na temeljnu plastiku.

Potom je dizajniran Bell dio za činelu jer je plastiku preteško permanentno formirati u zaobljeni oblik koji je poželjan za prirodnije sviranje. On je učvršćen epoxy ljepilom za plastiku te je cijela gornja ploha činele obložena podlogom za miš izrezanom tako da se lakše uočavaju zone. Zalemljeni su piezo senzori u paralelne spojeve, zalijepljeni na predviđena mjesta te su im vodovi zalemljeni na dva 3,5mm priključka.

Kako bi se ublažile sile koje djeluju na 3D isprintan dio činele, postavljen je filc između matice na šipki za činelu te same činele. Tako ona može biti dovoljno dobro učvršćena, ali i vrlo dobro može podnositi udarce.

Podloga za miš nalijepljena je vrućim ljepilom na plastiku kako bi utišala zvuk udarca štapića o činelu no i zaštitila je od sile udaraca (Slika 46). Ostale su činele izrađene istim postupkom, no s manjim dimenzijama te manjim brojem senzora.



Slika 46: Lijepljenje podloge za miš



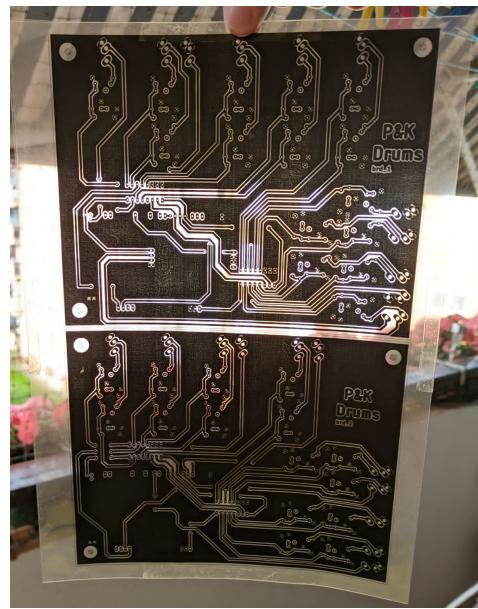
Slika 47: Izrađena Ride činela

9.5 Izrada tiskane pločice

S obzirom na jedinstvenost ovog rada, ne postoji mogućnost korištenja već izrađene kutije sa mikroupravljačem, priključcima, elektroničkim krugovima za obradu signala i sl. Stoga je dizajnirana tiskana pločica koju je odlučeno napraviti u Tehničkoj školi Bjelovar zbog male količine (nije serijska proizvodnja).

Kupljena je „PLOČICA VETRONIT FOTO POSITIV 200x300 JEDNOSTRANA“ jer oba dizajna pločica se mogu izraditi na ploči tih dimenzija pa kasnije odvojiti ubodnom pilom. Odlučeno je to odvajanje napraviti prije samog osvjetljavanja. Važno je biti brz pri rukovanju s pločicom s koje je uklonjena zaštita od osvjetljavanja kako bi postupak razvijanja bio uspješan. Zato je izbušena mala rupa u uglu pločice prije samog osvjetljavanja kako bi se mogla provući tanka nit koja će pomoći oko rukovanja s pločicom pri razvijanju u lužini i skidanju bakra u kiselini.

Potom je uklonjena zaštita od osvjetljavanja, predlošci za razvijanje učvršćeni su na pločice te su obje pločice osvjetljene na 5 minuta. Nakon toga su uklonjene predloške s pločica te su one jedna po jedna razvijane otprilike pola minute u lužini. Sljedeći korak je bio ispiranje vodom, zatim skidanje bakra u kiselini, ponovno ispiranje vodom pa osvjetljavanje bez predložaka. Ponovno osvjetljavanje je nužno kako bi se fotoosjetljivi spoj aktivirao te isprao lužinom da bi se lakše provodilo lemljenje.



Slika 48: Predložak za osvjetljavanje

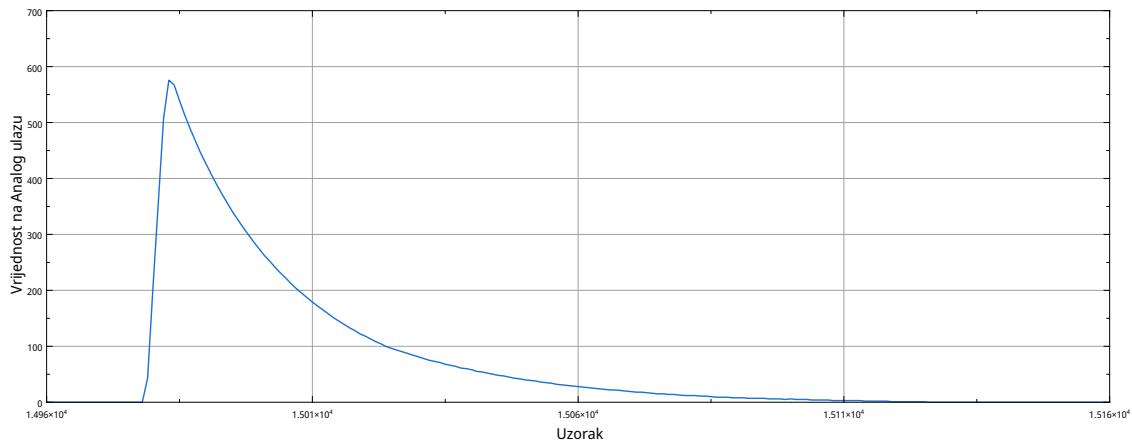
10 Glavni program

Glavni program nalazi se na Arduino Due i on je odgovoran za obrađivanje svih ulaza sa svih piezoelektričnih senzora, određivanje i detektiranje udarca i slanje MIDI signala na Addictive Drums 2.

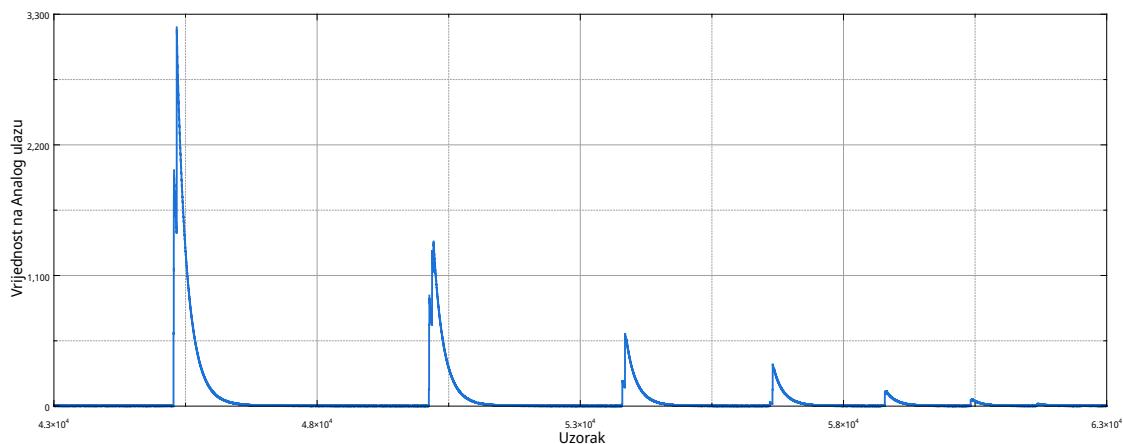
10.1 Očitani napon sa piezo senzora na Analognom ulazu

Piezoelektrični senzor proizvodi izmjenični napon na svojim izlazima koji može biti veći od 5V, maksimalnog napona na Analognom ulazu Arduina Due. Stoga je bilo potrebno ograničiti, ispraviti i filtrirati napon prije očitanja na Analognom ulazu. Za to je bio dizajniran spoj kondenzatora, dioda i otpornika.

Nakon ispravljanja, očitani napon jednog udarca na Analognom ulazu može se vidjeti na slici 49, a od palice koja je bila ostavljena da oskače na senzoru se može vidjeti na slici 50.



Slika 49: Očitani napon na Analognom ulazu nakon jednog udarca



Slika 50: Očitani napon na Analognom ulazu uz odskakujuću palicu

10.2 Obradivanje očitanog napona: Klasa `hitDetect()`

Najosnovnija funkcija obrade Analognog ulaza je objekt `hitDetect()`. Taj objekt, nakon deklariranja i postavljanja parametra, svaki put kada se zove procesira jedan uzorak na postavljenom Analognom ulazu i vrati vrijednost udarca ako se jedan dogodio. `hitDetect()` koriste i druge klase unutar programa koje kombiniraju više `hitDetect()` objekta u jedan objekt bubenja.

10.2.1 Podesivi parametri klase `hitDetect()`

`hitDetect()` deklariran je u datoteci `drumSpecificFunctions.h`, i njene funkcije definirane su u datoteci `drumSpecificFunctions.cpp`. Klasa u cijelovitosti se sastoji od dvije funkcije i četiri podesivih parametara.

Parametar/funkcija()	Svrha
<code>hitSensorSampleSize</code>	Količina uzoraka za računanje srednje vrijednosti.
<code>hitSensorPin</code>	Koji Analogni ulaz očitava.
<code>hitThreshold</code>	Granična vrijednost za udarac.
<code>hitMul</code>	Množenje očitane Analognе vrijednosti.
<code>readNewHitSensor()</code>	Očita novu vrijednost sa Analognog ulaza.
<code>process()</code>	Procesira jedan uzorak očitane Analognе vrijednosti.

Tablica 1: Podesivi parametri i funkcije klase `hitDetect()`

Uz parametre unutar klase, postoji jedan parametar definiran za vrijeme kompilacije koji vrijedi za sve deklarirane `hitDetect()` objekte unutar programa. Taj parametar, nazvan `MAXSAMPLES`, označava najveći broj uzoraka koji može biti postavljen tijekom vremena izvođenja.

10.2.2 Varijable i dodatni parametri

Uz parametre koji se moraju podesiti postoje i parametri koji se ne moraju nužno podesiti, ali su bili dodani zbog proširenja fleksibilnosti klase `hitDetect()`.

Parametar	Svrha
<code>returnOnly</code>	Ako istinito, <code>hitDetect()</code> vrati jačinu udarca ako se jedan dogodio ili 0 ako se nije. Ako nije istinito, <code>hitDetect()</code> preuzme slanje MIDI signala ako detektira udarac.
<code>midiChannel</code>	Na kojem kanalu objekt šalje MIDI signale ako se dogodio udarac. Bitno ako je <code>returnOnly</code> lažan.
<code>↪ midiNote</code>	Koju MIDI notu šalje objekt ako se dogodio udarac. Bitno ako je <code>returnOnly</code> lažan.

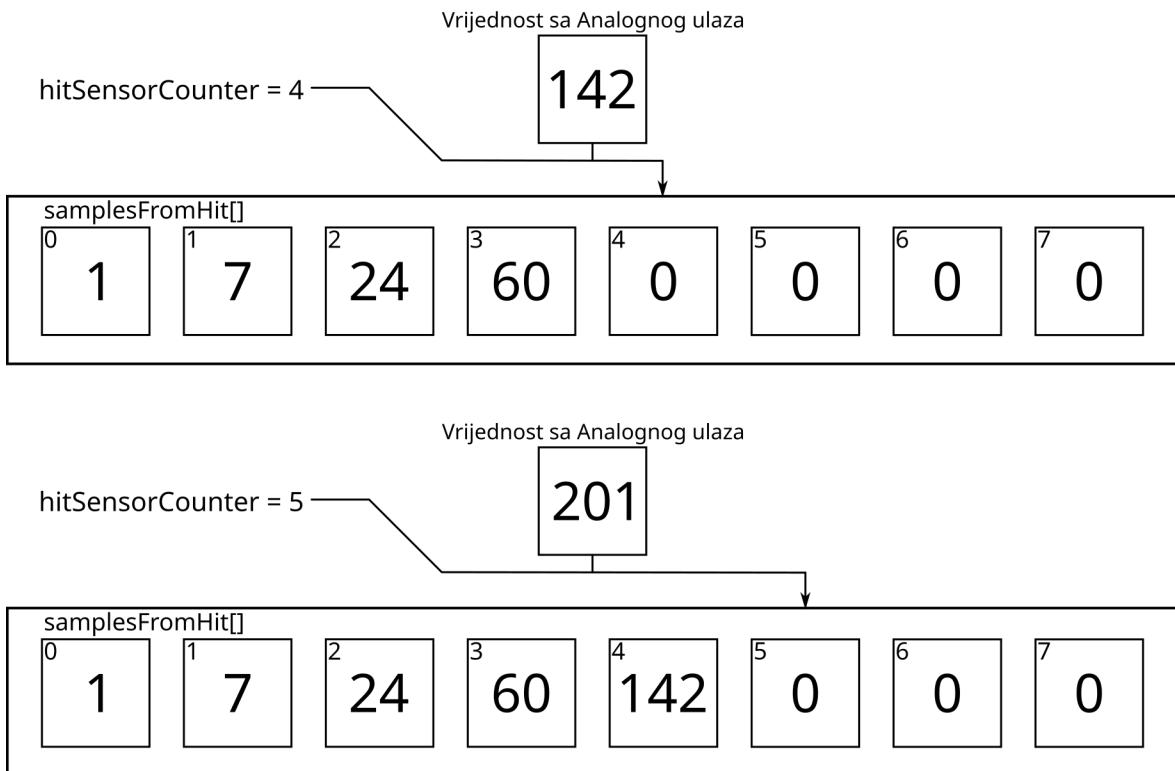
Tablica 2: Dodatni parametri klase `hitDetect()`

Varijabla	Svrha
<code>samplesFromHit[]</code>	Niz podataka koji sadrži očitane vrijednosti sa Analognog ulaza.
<code>hitSensorHit</code>	Označava ako se dogodio udarac.
<code>hitSensorMaximum</code>	Maksimalna vrijednost udarca.
<code>currentHitAvg</code>	Trenutna srednja vrijednost Analognog ulaza. Izračunata iz <code>samplesFromHit[]</code> .
<code>hitSensorCounter</code>	Varijabla koja označava u koju poziciju unutar niza <code>samplesFromHit</code> se spremi očitana vrijednost sa Analognog ulaza.
<code>hitSentNote</code>	Označava da li je objekt poslao MIDI notu. Nebitno ako je <code>returnOnly</code> istinit.

Tablica 3: Varijable klase `hitDetect()`

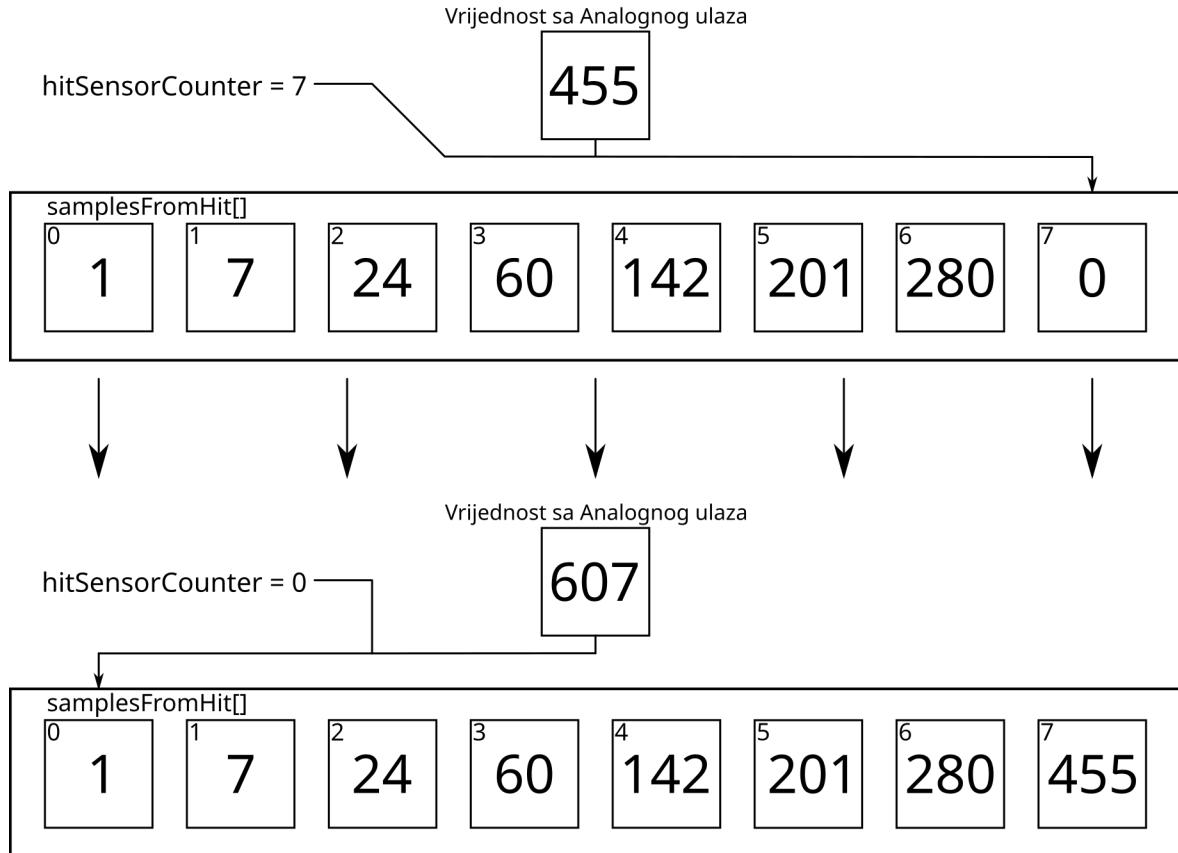
10.2.3 `readNewHitSensor()`

`readNewHitSensor()` je funkcija koja očitava jedan uzorak sa Analognog ulaza i priprema ga za obradu. Ona očitava Analogni ulaz, sprema očitanu vrijednost u niz `samplesFromHit` na poziciju koju označava `hitSensorCounter` (Slika 51). Zatim, po potrebi, vrati `hitSensorCounter` na 0.



Slika 51: Primjer upisivanja dvije očitane vrijednosti u niz uzoraka

Ako bi iduća pozicija bila izvan granica niza, `hitSensorCounter` se vrati na 0 (Slika 52).



Slika 52: Primjer upisivanja dvije očitane vrijednosti u niz uzoraka

Nakon svakog zvanja funkcija `readNewHitSensor()` se ponovno računa srednja vrijednost cijelog niza. Funkcija je deklarirana u datoteci `drumSpecificFunctions.h`, i definirana je u datoteci `drumSpecificFunctions.cpp` (Program 2)

```

1 void hitDetect::readNewHitSensor(){
2     samplesFromHit[hitSensorCounter] = analogRead(hitSensorPin)*
3         ↵ drumMul;
4     (hitSensorCounter >= hitSensorSampleSize) ? hitSensorCounter = 0 :
5         ↵ hitSensorCounter++;
}
```

Program 2: Funkcija `readNewHitSensor()`

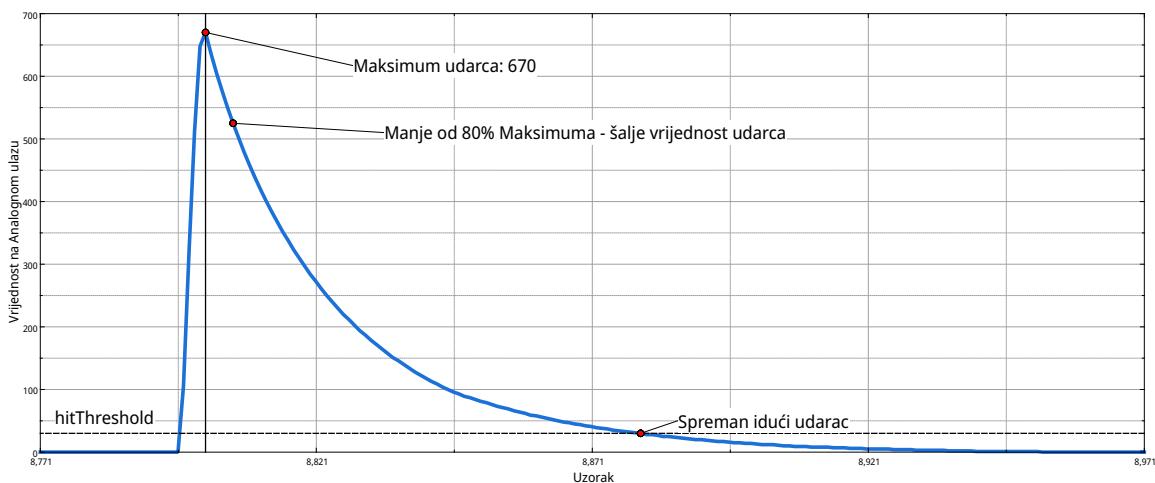
10.2.4 process()

`process()` je glavna funkcija unutar klase `hitDetect()` koja se zove svaki put kada je potrebno procesirati ulaz sa bubnja. Ona traži trenutak kada se udarac dogodio, traži maksimum tog udarca i vraća ga (ili šalje MIDI signal, ovisno o varijabli `returnOnly`) kao `return` vrijednost.

```

1  uint16_t hitDetect::process(){
2      readNewHitSensor();
3      currentHitAvg = calcMeanInt(samplesFromHit, hitSensorSampleSize);
4
5      // Traži udarac
6      (currentHitAvg > hitThreshold) ? hitSensorHit = true :
7          ↪ hitSensorHit = false;
8      (!hitSensorHit) ? hitSentNote = false : hitSentNote = hitSentNote;
9
10     // Ako se udarac pogodio i nije poslan MIDI signal.
11     if(hitSensorHit && !hitSentNote){
12         // Traži maksimum
13         if(currentHitAvg >= hitSensorMaximum) hitSensorMaximum =
14             ↪ currentHitAvg;
15
16         // Ako trenutna vrijednost padne ispod 80% maksimuma vraća
17         ↪ vrijednost udarca ili šalje MIDI signal.
18         else if(currentHitAvg < hitSensorMaximum * 0.8){
19             hitSentNote = true;
20             // Jačinu udarca potrebno je podijeliti sa 32 zbog toga što
21             ↪ to je vrijednost sa Analognog ulaza između 0-4095, a MIDI signal
22             ↪ šalje jačine između 0-127.
23             uint8_t noteVelocity = hitSensorMaximum / 32;
24             if(!returnOnly) noteOn(midiChannel, midiNote, noteVelocity
25             ↪ );
26             else return hitSensorMaximum;
27             hitSensorMaximum = 0;
28         }
29     }
30     if(returnOnly) return 0;
31 }
```

Program 3: Funkcija process()

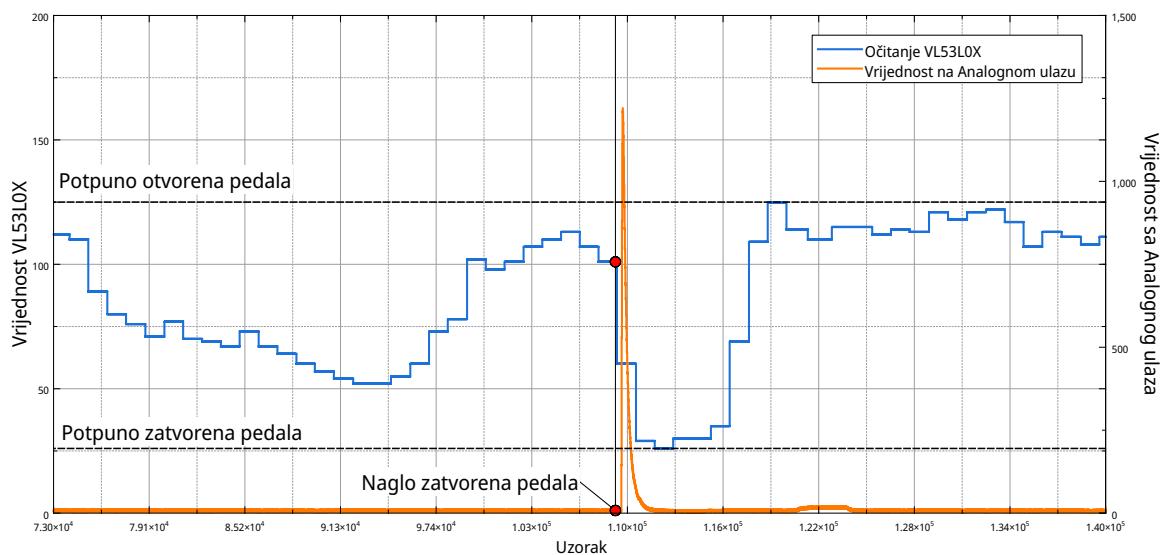


Slika 53: Vizualizacija procesiranja jednog udarca

10.3 Obradivanje ulaza sa Hi-Hat pedale: Klasa `pedalHH()`

Hi-Hat pedala, kao već obrađeno na stranici 12, ima uz piezoelektrični senzor i senzor udaljenosti koji mjeri koliko je pedala otvorena. Ta se vrijednost šalje programu Addictive Drums 2 u obliku Control Change (CC) MIDI signala. Tijekom normalnog rada, vrijednost senzora udaljenosti je između 25 mm i 130 mm (Slika 54). Zbog toga što je vrijednost CC signala između 0 i 127, a Addictive Drums 2 koristi cijeli raspon CC vrijednosti, potrebno je mapirati vrijednost senzora na vrijednosti CC signala.

Kao već spomenuto na stranici 5, Hi-Hat proizvodi poseban zvuk kada se naglo zatvori. Addictive Drums 2 ima funkciju kojom preko CC signala prepozna naglo zatvaranje pedale i pusti odgovarajući zvuk. Naime, kao što se vidi na slici 54, vrijednost senzora udaljenosti kasni kada se usporedi sa stvarnovremenskim signalom sa piezoelektričnog senzora. To kašnjenje je uzrokovano nedovoljno velikom brzinom očitanja senzora i bilo je dovoljno veliko da se nije moglo osloniti na Addictive Drums 2 za prepoznavanje zatvaranja pedale u stvarnom vremenu. Kao rješenje, uveden je spomenuti piezoelektrični senzor koji detektira kada se pedala potpuno zatvori.



Slika 54: Vizualiziranje vrijednosti senzora uz jedno zatvaranje pedale

Klasa `pedalHH()` odgovorna je za obradivanje ulaza. Ona kombinira funkcije klase `hitDetect()` za obradivanje piezoelektričnog senzora i vlastite funkcije za obradivanje ulaza sa senzora.

10.3.1 Varijable

Varijabla	Svrha
<code>samplesFromOpen[]</code>	Niz podataka koji sadrži očitane vrijednosti sa senzora udaljenosti.
<code>currentOpenAvg</code>	Trenutna srednja vrijednost senzora udaljenosti.
<code>sendCC</code>	Vrijednost CC MIDI signala koju funkcija šalje.
<code>previousCC</code>	Prošla vrijednost CC MIDI signala.

Tablica 4: Varijable klase `pedalHH()`

10.3.2 Podesivi parametri

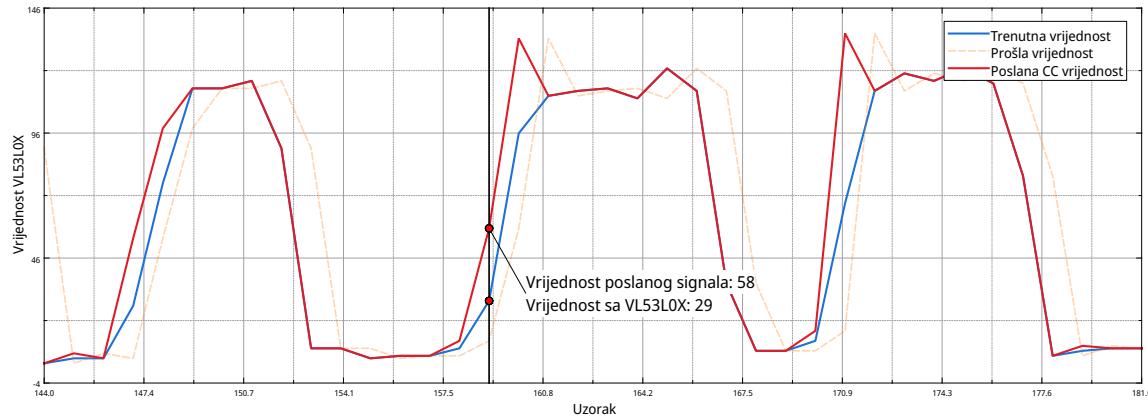
Podesivi parametri su gotovo isti parametrima klase `hitDetect()`, uz dodanih nekoliko vezanih za procesiranje senzora udaljenosti.

Parametar/funkcija()	Svrha
<code>midiCC</code>	Upravljačka vrijednost CC signala.
<code>openSensorSampleSize</code>	Količina uzoraka senzora udaljenosti za računanje srednje vrijednosti.
<code>openSensorMin</code>	Vrijednost senzora udaljenosti uz potpuno zatvorenu pedalu (Slika 54).
<code>openSensorMax</code>	Vrijednost senzora udaljenosti uz potpuno otvorenu pedalu (Slika 54).
<code>quantizeCC</code>	Uključuje ili isključuje kvantiziranje CC vrijednosti.
<code>quantizeRanges [5]</code>	Označava područja za kvantizaciju.
<code>processPedalCC()</code>	Procesira ulaz sa senzora udaljenosti.
<code>processSpeed()</code>	Procesira brzinu otvaranja Hi-Hat pedale.
<code>process()</code>	Procesira cijelu Hi-Hat pedalu.

Tablica 5: Podesivi parametri i funkcije klase `pedalHH()`

10.3.3 `processSpeed()`

`processSpeed()` bio je uveden kao način prividnog smanjenja kašnjenja očitanja senzora udaljenost. Ta funkcija prepoznaje naglo otvaranje Hi-Hat pedale i poveća poslani signal kao pokušaj u predviđanju do koje će se razine pedala otvoriti.



Slika 55: Učinak funkcije `processSpeed()` na poslanu vrijednost

Funkcija pogleda razliku između prošlog poslanog CC signala i trenutnog očitanog CC signala. Ako je trenutna vrijednost dovoljno veća od prošle, funkcija će dodati vrijednost prošle na trenutnu.

```

1 void pedalHH::processSpeed(){
2     if(sendCC > previousCC*1.2){      // Provjeri za rast i poveća
3         sendCC+=sendCC-previousCC;    // poslanu vrijednost.
4     }
5 }
```

Program 4: Funkcija `processSpeed()`

10.3.4 processPedalCC()

processPedalCC() je glavna funkcija unutar klase pedalHH() koja se zove svaki put kada je potrebno i moguće procesirati koliko je pedala otvorena. Ona mapira i graniči vrijednost senzora udaljenosti na vrijednosti CC signala (0-127), procesira brzinu i šalje CC MIDI signal računalu.

```
1 void pedalHH::processPedalCC(){
2     // Mapira i ogranic udaljenost od senzora na vrijednosti CC
3     ↵ signala
4     currentOpenAvg = calcMeanInt(samplesFromOpen, openSensorSampleSize
5     ↵ );
6     sendCC = map(currentOpenAvg, openSensorMin, openSensorMax, 0, 127)
7     ↵ ;
8     processSpeed();
9
10    if (sendCC < 0) sendCC = 0;
11    if (sendCC > 127) sendCC = 127;
12    controlChange(midiChannel, midiCC, sendCC);
13 }
```

Program 5: Funkcija processPedalCC()

10.3.5 readNewOpenSensor()

readNewOpenSensor() uzima kao argument vrijednost senzora udaljenosti i stavlja ju u niz podataka samplesFromOpen. Funkcija radi identično funkciji readNewHitSensor() (Stranica 27).

10.3.6 process()

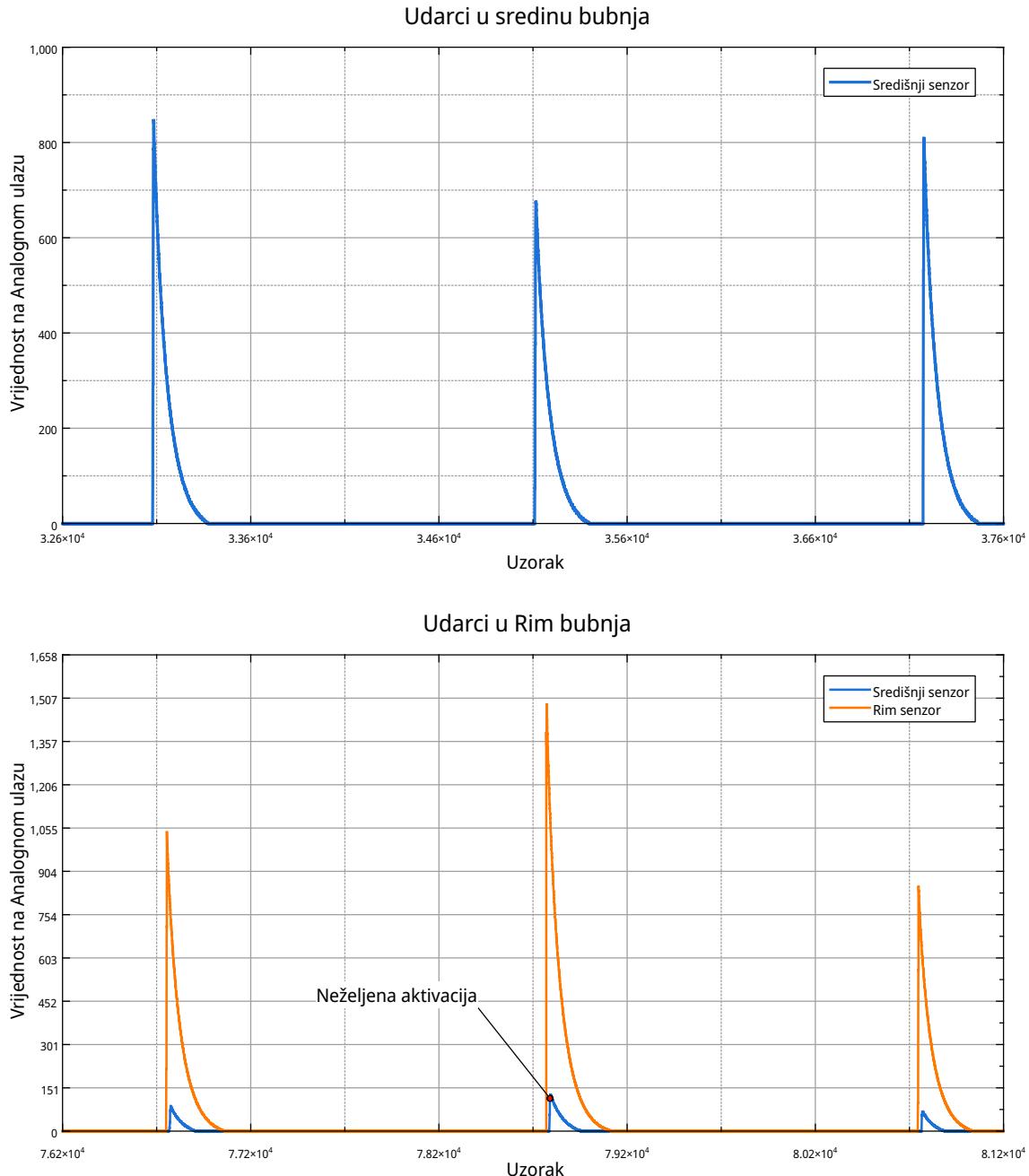
process() je funkcija unutar klase pedalHH() koja se zove svaki put kada je potrebno procesirati Hi-Hat pedalu u cjelovitosti. Ona prvo zove funkciju za procesiranje senzora udaljenosti (processCC()), zatim zove funkciju za procesiranje piezoelektričnog senzora. hitDetect::process(). processCC() zove jedino ako je dostupna nova vrijednost sa senzora udaljenosti, tj. Arduina Nano.

```
1 void pedalHH::process(int value){
2     // Procesira koliko je pedala otvorena ako je dostupna nova
3     ↵ vrijednost sa senzora udaljenosti
4     if(value != -1){
5         readNewOpenSensor(value);
6         processPedalCC();
7     }
8
9     // Procesira piezoelektrični senzor na pedali
10    close.process();
11 }
```

Program 6: Funkcija process()

10.4 Obrađivanje ulaza sa Tom-Tom bubnja: Klasa twoZone()

Tom-Tom bubanj sadrži dva piezoelektrična senzora za dva različita zvuka (Stranica 9). Jedan se nalazi na sredini glave, a drugi na rubu bubnja, unutar Rima (Stranica 10). Iako se oba senzora nalaze na istom bubnju, kada se udara sredina bubnja se ne aktivira senzor unutar Rima zbog toga što je Rim 3D isprintan od tvrde plastike. Naime, kada se udara Rim moguće je da se ponekada aktivira i senzor na sredini glave (Slika 56). Klasa koja obrađuje Tom-Tom bubanj, uz procesiranje oba ulaza, dodatno kompenzira za tu neželjenu aktivaciju središnjeg senzora.



Slika 56: Vizualizacija očitane vrijednosti sa udarcima na sredinu i na Rim bubnja

10.4.1 Podesivi parametri

Klasa se sastoji od samo dva `hitDetect()` objekta i sitan dio koda tako da su podešljivi parametri samo parametri od ta dvija `hitDetect()` objekta.

Parametar/funkcija()	Svrha
<code>midiChannel</code>	Na kojem kanalu objekt šalje MIDI signale ako se dogodio udarac.
<code>applySettings()</code>	Pošalje sve podesive parametre u <code>hitDetect()</code> objekte za koje su namijenjeni.
<code>process()</code>	Procesira cijeli Tom-Tom bubanj.
<code>centerNote</code>	Koju MIDI notu šalje objekt ako se dogodio udarac na sredini glave.
<code>centerSampleSize</code>	Količina uzoraka za računanje srednje vrijednosti središnjeg senzora.
<code>centerPin</code>	Koji Analogni ulaz očitava.
<code>centerMul</code>	Množenje očitane Analognog vrijednosti.
<code>centerThreshold</code>	Granična vrijednost za udarac.
<code>sideNote</code>	Koju MIDI notu šalje objekt ako se dogodio udarac na Rimu.
<code>sideSampleSize</code>	Količina uzoraka za računanje srednje vrijednosti Rim senzora.
<code>sidePin</code>	Koji Analogni ulaz očitava.
<code>sideMul</code>	Množenje očitane Analognog vrijednosti.
<code>sideThreshold</code>	Granična vrijednost za udarac.

Tablica 6: Podesivi parametri i funkcije klase `twoZone()`

10.4.2 `applySettings()`

Unutar klase `twoZone()` koriste se dva `hitDetect()` objekta: `hitDetect center` i `hitDetect side`. Zbog toga što je ova dva objekta potrebno podesiti da očitavaju točne Analognog ulaze, a njihovim podesivim parametrima nije moguće pristupiti izvan klase `twoZone()`, bilo je potrebno deklarirati unutar ove klase posebnu varijablu za svaki podešivi parametar od svakog `hitDetect()` objekta.

Da bi parametrima unutar `hitDetect()` objekata bili dodijeljene vrijednosti unutar varijabla, uvedena je funkcija `applySettings()` koja im dodijeli navedene vrijednosti.

10.4.3 `process()`

Funkcija `process()` se zove svaki put kada je potrebno obraditi Tom-Tom bubanj. Iako funkcija nije kompleksna (uglavnom se sastoji od zvanja druga dva `hitDetect()` objekta), obavlja dio logike koji sprječava slanje MIDI signala zbog neželjene aktivacije središnjeg senzora.

```

1 void twoZone::process(){
2     side.process();
3     if(side.hitSensorHit) center.hitSentNote = true;
4     center.process();
5 }
```

Program 7: Funkcija `process()`

10.5 Glavna datoteka programa: main.cpp

Unutar glavne datoteke programa se deklariraju svi bubenjevi, činele i pedale te se postave parametri. Uz deklariranje, dodatno se i poveća frekvencija procesora Arduina Due sa 80MHz na 114MHz, pokrene se Serial komunikacija na RX3/TX3 paru pinova i poveća se rezolucija ADC pretvornika sa 10 bitova na 12 bitova.

```

1 #include <Arduino.h>
2 #include <MIDIUSB.h> // 
    ↪ Biblioteka potrebna za
    ↪ slanje MIDI podataka preko
    ↪ USB porta Arduina Due.
3
4 #include "externalFunctions.h"
5 #include "defines.h"
6
7 #define MAXSAMPLES 50
8
9 #include "drumSpecificFunctions.h"
    ↪ "
10
11 hitDetect hihatCymbal;
12 pedalHH hhpPedal;
13 hitDetect tom;
14
15 void setup()
16 {
17     overclock(); // Poveća
        ↪ frekvenciju procesora sa 84
        ↪ MHz na 114MHz.
18     Serial3.begin(9600); // Serial
        ↪ komunikacija potrebna za Hi
        ↪ -Hat pedalu.
19     SerialUSB.begin(115200);
20
21     pinMode(5, OUTPUT);
22     pinMode(6, OUTPUT);
23     pinMode(7, OUTPUT);
24
25     analogReadResolution(12);
26
27     hihatCymbal.midiNote = 8;
28     hihatCymbal.hitSensorPin = A10;
29     hihatCymbal.hitSensorSampleSize
        ↪ = 15;
30
31
32     hhpPedal.midiNote = 48;
33     hhpPedal.hitSensorPin = A10;
34     hhpPedal.openSensorSampleSize =
        ↪ 2;
35     hhpPedal.hitThreshold = 70;
36     hhpPedal.applySettings();
37
38
39     tom.midiNote = 71;
40     tom.hitSensorPin = A10;
41     tom.hitSensorSampleSize = 10;
42
43 }
44
45 void loop()
46 {
47     //bass.process();
48     //delayMicroseconds(50);
49     //tom.process();
50     //delayMicroseconds(50);
51     digitalWrite(5, LOW);
52     digitalWrite(6, !LOW);
53     digitalWrite(7, LOW);
54     delayMicroseconds(50);
55
56     if(Serial3.available()) hhpPedal
        ↪ .process(Serial3.read());
57     else hhpPedal.process(-1);
58     delayMicroseconds(100);
59
60     digitalWrite(5, !LOW);
61     digitalWrite(6, LOW);
62     digitalWrite(7, LOW);
63     delayMicroseconds(50);
64
65     hihatCymbal.process();
66     delayMicroseconds(50);
67
68 }
69

```

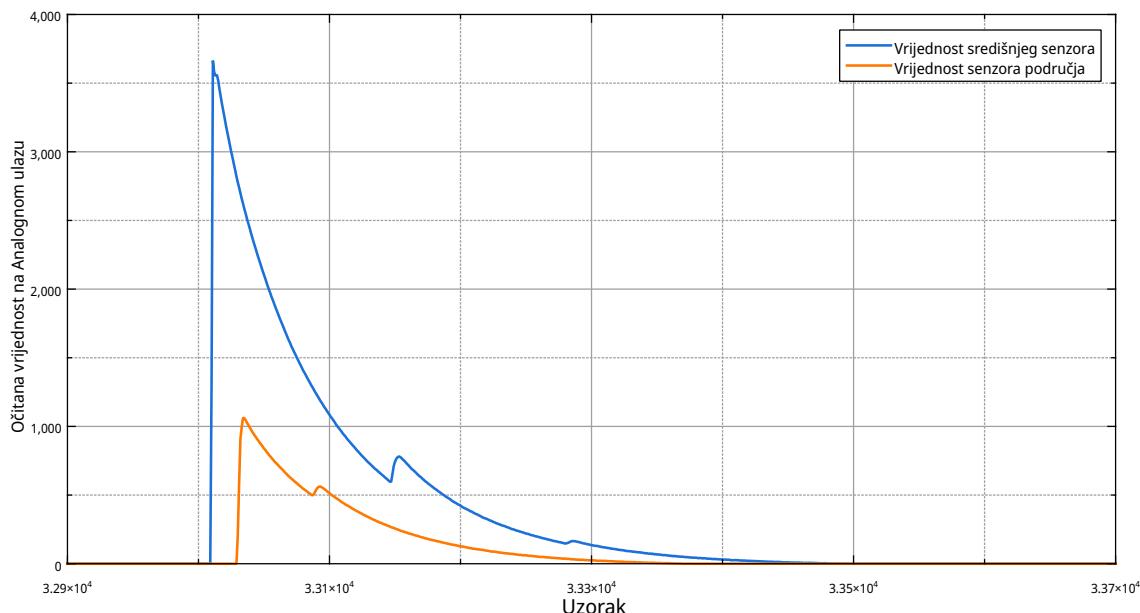
Program 8: Glavna datoteka programa.

Program iznad nije finaliziran i često se mijenja zbog eksperimentacije, debugganja i programiranja.

11 Analiza ulaza sa Snare bubenja.

Snare bubanj, kao već opisano na strani 7, je podijeljen na područja pomoću dodatnih piezoelektričnih senzora. Zbog toga što su i središnji senzor i svi okružnji senzori (kolektivno nazavni senzor područja) postavljeni na glavu bubnja, bilo je potrebno proučiti, analizirati i utvrditi način kako razlikovati udarce na sredinu bubnja i udarac bliže rubu bubnja.

Nakon proučavanja očitane vrijednosti uz različite udarce, bilo je utvrđeno kako razlikovati udarac na sredinu bubnja i udarac bliže rubu bubnja. Udarac u sredinu bubnja daje karakterističan i rijetko promjenjiv izgled očitanih vrijednosti na grafu (Slika 57).

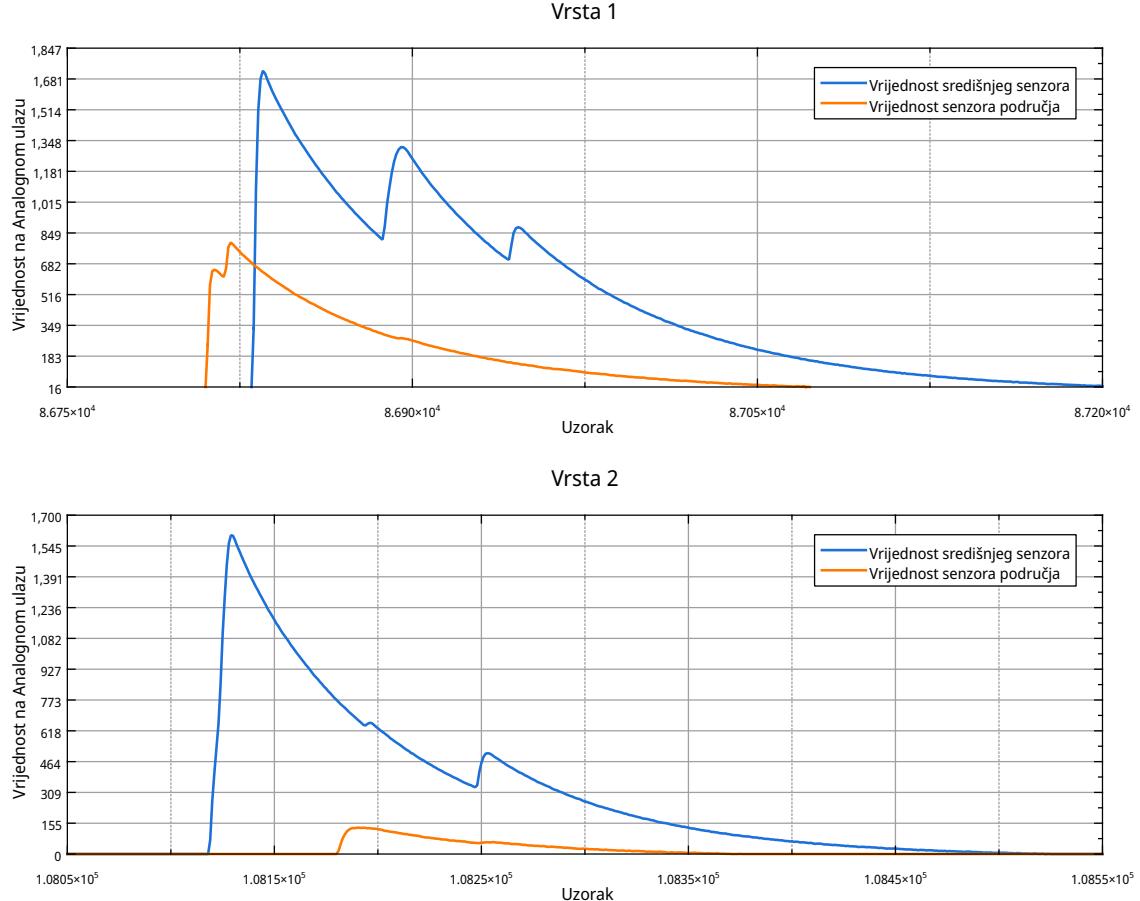


Slika 57: Vizualizacija vrijednosti senzora uz udarac u sredinu bubnja

Na grafu iznad bitno je uočiti dvije glavne karakteristike: omjer očitanih vrijednosti i međusobni vremenski razmak. Zbog toga što se udara izravno po središnjem senzoru, njegova očitana vrijednost znatno je veća od vrijednosti senzora područja. Također, zbog istog razloga, mehanički valovi od udarca prvo dosegnu središnji senzor, zatim se prošire i dostignu senzor područja. To vrijeme širenja uzrokuje mjerljivi vremenski razmak između pojave udarca na vrijednostima senzora.

Taj omjer vrijednosti i njihov međusobni vremenski razmak pojavljuje se konstantno kod udarca u sredinu bubnja. Pomoću te dvije karakteristike se može odrediti da li se udarac dogodio na sredini bubnja.

Dalnjim proučavanjem udaraca dalje od sredine pronađen je način kako ih razlikovati od udaraca na sredini bubnja. Razlikuju se dvije vrste reagiranja senzora na udarac udaljen od sredine bubnja (Slika 58).



Slika 58: Vrste reagiranja senzora na udarac udaljen od sredine bubnja.

U vrsti 1, karakteristična veličina je vremenski razmak između pojava udarca na senzorima. Bitno je uočiti da se, za razliku od udarca na sredini bubnja (Slika 57), udarac se pojavi na senzoru područja *prije* nego što se pojavi na središnjem senzoru. Pomoću te činjenice lako je utvrditi da se udarac dogodio dalje od sredine bubnja.

Naime, nije uvijek slučaj da se udarac pojavi prvo na senzoru područja. Ponekada se udarac pojavi prvo na središnjem senzoru iako se nije udarilo po sredini bubnja. Takvi udarci spadaju pod vrstu 2. Na toj vrsti, bitno je uočiti omjer maksimuma udarca. Na udarcu po sredini bubnja, maksimumi očitanih vrijednosti stalno su u omjerima između 2:1 ili 3:1, tj. da je očitani maksimum središnjeg bubnja stalno 2-3 puta veći od očitanog maksimuma na senzoru područja. Ovo se razlikuje od omjera maksimuma na vrsti 2, gdje je očitani maksimum središnjeg senzora znatno veći od maksimuma na senzoru područja.

Pomoću ovih uočavanja i svrstavanja grafova na vrste, moguće je konzistentno razlikovati udarac u sredini bubnja od udarca bliže rubu bubnja.

12 Zaključak

Električni bubenjevi nisu iznimno kompleksni uređaji ako se ograniči na jednostavno prepoznavanje jednog udarca na bubenju. Naime, ako se želi proširiti funkcionalnost i izraditi električne bubenjeve da su sličniji akustičnim bubenjevima, potrebno je isplanirati, dizajnirati i sastaviti kompleksnije dijelove što oduljuje, zakomplicira i poskupljuje cijelu građu.

Na bubenjevima, nakon odabiranja od kojih materijala izraditi bubenj, najviše planiranja bilo je uloženo u Rim i pozicije senzora po bubenju.

Na činelama, najviše planiranja bilo je uloženo u rad Chokea i područje Bella, oboje od kojih je bilo potrebno 3D modelirati i printati.

Na Bass pedali, najviše je vremena bilo uloženo u dizajniranje izgleda i sastava pedale. Od kojih materijala će biti pojedini dio napravljen, kako će se međusobno spojiti, kako osigurati da se pedala vrati u otvoreni položaj nakon pritisaka i osiguranja da pedala može izdržati teške udarce.

Na Hi-Hat pedali, uz svo uloženo vrijeme sa Bass pedale, dodatno je bilo potrebo odlučiti kako će se očitavati koliko je otvorena pedala. Nakon odlučivanja za infracrveni senzor udaljenosti, bilo je potrebno naći najbolji položaj tog senzora na pedali. Posljednje za Hi-Hat pedalu je bilo potrebno i odlučiti kako prijeći ograničenje od jednog metra na I2C komunikaciji.

Literatura

- [1] TME.eu
- [2] Arduino Due - <https://store.arduino.cc/products/arduino-due>
- [3] PlatformIO Core (CLI) - <https://docs.platformio.org/en/latest/core/index.html>
- [4] PlatformIO IDE - <https://docs.platformio.org/en/latest/integration/ide/pioide.html>
- [5] Addictive Drums 2 - https://www.xlnaudio.com/products/addictive_drums_2
- [6] Thomann, "Gretsch Drums Energy Standard Red" - https://www.thomann.de/be/gretsch_energy_standard_wine_red.htm
- [7] Thomann, "British Drum Company 14"x6,5" Merlin Snare Drum" - https://www.thomann.de/be/british_drum_company_14_x65_merlin_snare_drum.htm
- [8] The New Drummer, "How to Have Effective Bass Drum Technique" - <https://www.thenewdrummer.com/effective-bass-drum-technique/>
- [9] Swift, Andrew, "A Brief Introduction to MIDI"
- [10] midi.org, "About MIDI-Part 3:MIDI Messages" - <https://www.midi.org/midi-articles/about-midi-part-3-midi-messages>
- [11] midi.org, "MIDI 1.0 Control Change Messages (Data Bytes)" - <https://www.midi.org/specifications-old/item/table-3-control-change-messages-data-bytes-2>
- [12] Arduino Nano - <https://store.arduino.cc/products/arduino-nano>