Mehanizam za zakretanje robotske glave koji reagira na zvučne podražaje

Koren, Leon

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:136707

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-20**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Leon Koren

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

doc. dr. sc. Tomislav Stipančić, dipl. ing.

Student:

Leon Koren

Zagreb, 2020.

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, doc. dr. sc. Tomislavu Stipančiću, na stručnim savjetima i pruženoj pomoći pri izradi ovoga rada.

Leon Koren

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602 - 04 / 20 - 6 / 3	
Ur. broj: 15 - 1703 - 20 -	

DIPLOMSKI ZADATAK

Student: **LEON KOREN** Mat. br.: 0035196867

Naslov rada na hrvatskom jeziku: **Mehanizam za zakretanje robotske glave koji reagira na zvučne podražaje**

Naslov rada na engleskom jeziku: **Robotic head rotating mechanism that responds to auditory stimuli**

Opis zadatka:

U radu je potrebno izraditi cjelovito mehaničko i softversko rješenje vratnog mehanizma za okretanje robotske glave s jednim stupnjem slobode gibanja koje omogućuje zakretanje u lijevu ili desnu stranu. Mehanizam treba biti dizajniran tako da zakreće glavu u smjeru iz kojeg dolaze zvučni podražaji iz okoline. S time u vezi potrebno je:
1. Dizajnirati i ostvariti konstrukcijsko rješenje vrata robota. Prilikom akvizicije zvučnih podražaja iz okoline koristiti mikrofone smještene u vratni mehanizam.
2. Razviti te potom ostvariti softversko rješenje tako da su kretnje glave robota kontekstualne tako da reagiraju na smjer zvuka te njegovu jačinu. Što je zvuk jačeg intenziteta kretnje robotske glave trebaju biti intenzivnije.
Razvijeno rješenje je potrebno eksperimentalno evaluirati uključivši ljudske subjekte. U radu je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:
5. ožujka 2020.

Rok predaje rada:
7. svibnja 2020.

Predviđeni datum obrane:
11. svibnja do 15. svibnja 2020.

Zadatak zadao:

doc. dr. sc. Tomislav Stipančić

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ . . . . . **I**

POPIS SLIKA . . . . . **IV**

POPIS TEHNIČKE DOKUMENTACIJE . . . . . **V**

SAŽETAK **1**

SUMMARY **2**

1. UVOD **3**

2. Zvuk **4**
   2.1 Percepcija zvuka kod ljudi 4
      2.1.1 Visina zvuka (ton) 5
      2.1.2 Trajanje zvuka 6
      2.1.3 Jačina ili intezitet zvuka 6
      2.1.4 Boja zvuka 7
      2.1.5 Tekstura zvuka 7
      2.1.6 Prostorna lokacija 7
   2.2 Percepcija zvuka kod robotskih sustava 8
      2.2.1 Metoda eholokacije 9
      2.2.2 Metode prepoznavanja ljudskog govora 10

3. Razrada ideje **11**

4. Izrada mehanike **12**
   4.1 Gornji poklopac 13
      4.1.1 Postupak izrade 13
   4.2 Velika remenica 16
      4.2.1 Postupak izrade 16
   4.3 Nosač motora 19
      4.3.1 Postupak izrade 19
   4.4 Rotacijski stol 21
      4.4.1 Postupak izrade 21
   4.5 Kućište 23
      4.5.1 Postupak izrade 23
   4.6 Nosač predpojačala 26
      4.6.1 Postupak izrade 26
   4.7 Nosač projektora 27
      4.7.1 Postupak izrade 28

5. Ispis na 3D pisaču **29**

5.1 Ispis gornjeg poklopca . . . 32
5.2 Ispis velike remenice . . . 34
5.3 Ispis nosača motora . . . 36
5.4 Ispis rotacijskog stola . . . 38
5.5 Ispis kućišta . . . 40
5.6 Ispis manjih dijelova . . . 43

6. Izrada elektronike . . . **44**
6.1 Predpojačalo mikrofona . . . 45
6.2 Kontrolna pločica . . . 48
6.2.1 Mikrokontroler . . . 49
6.3 Popratna elektronika . . . 51

7. Izrada koda . . . **53**
7.1 Inicijalizacija i poziv biblioteka . . . 53
7.2 Inicijalizacija i konfiguracija I2S i ADC modula . . . 56
7.3 Izračun kuta zakreta . . . 59
7.4 Proračun vremenskog odmaka . . . 61
7.4.1 Akvizicija i parsiranje podataka . . . 61
7.4.2 Izračun vremenskog pomaka . . . 63
7.4.3 Upravljanje motorom . . . 66
7.5 Funkcija za početno postavljanje *setup()* . . . 67
7.6 Funkcija beskonačnog izvođenja *loop()* . . . 68

8. ZAKLJUČAK . . . **69**

9. PRILOG . . . **71**
9.1 Arduino kod . . . 71
9.2 Sheme sklopa . . . 83
9.2.1 Kontrolna pločica . . . 83
9.2.2 Predpojačalo . . . 84

POPIS SLIKA

Slika 1. Primjer visine zvuka 5
Slika 2. Primjer jačine zvuka 6
Slika 3. Primjer eholokacije 9
Slika 4. Google Home i Amazon Echo 10
Slika 5. Osnova sklopa (Motor i ležajevi) 13
Slika 6. Gornji poklopac 14
Slika 7. Sklop s gornjim poklopcem 15
Slika 8. Velika remenica 16
Slika 9. Velika remenica prikaz zubi 17
Slika 10. Velika remenica prikaz izreza za spajanje stola 17
Slika 11. Sklop s remenicom 18
Slika 12. Nosač motora 19
Slika 13. Sklop s nosačem motora 20
Slika 14. Prilagodba stola na vrat mehanizma 21
Slika 15. Izgled donje strane stola 22
Slika 16. Kućište s elektroničkim komponentama 23
Slika 17. Nosač gornjeg poklopca 24
Slika 18. Prihvat mikrofona 25
Slika 19. Nosač tiskane pločice 26
Slika 20. Princip vezanja donjeg nosača 27
Slika 21. Nosač projektora 28
Slika 22. Diskretizacija kružnih oblika 29
Slika 23. Prikaz potpornja 30
Slika 24. Pravilno postavljanje na ispisni stol 31
Slika 25. Prikaz izrezanog gornjeg poklopca 32
Slika 26. Početak ispisa gornjeg poklopca 33
Slika 27. Izrađen gornji poklopac 33

Slika 28. Prikaz izrezane velike remenice 34
Slika 29. Izrađena velika remenica 35
Slika 30. Prikaz izrezanog nosača motora 36
Slika 31. Izrađen nosač motora 37
Slika 32. Prikaz izrezanog rotacijskog stola 38
Slika 33. Izrađen rotacijski stol 39
Slika 34. Prikaz izrezanog kućišta 40
Slika 35. Prikaz gornjeg izrezanog kućišta 41
Slika 36. Izrađeno kućište 42
Slika 37. Nosač predpojačala 43
Slika 38. Nosač projektora 43
Slika 39. Shema mikrofona 45
Slika 40. Operacijsko pojačalo u spoju pojačala 46
Slika 41. Predpojačalo u EAGLE-u 47
Slika 42. Kontrolna pločica u EAGLE-u 48
Slika 43. ESP32 DOIT DevKit V1 49
Slika 44. Sklop spuštač napona 51
Slika 45. Sklop pojačala za zvučnik 52
Slika 46. Metodologija izračuna kuta 59
Slika 47. Snimljeni signal 63

POPIS TEHNIČKE DOKUMENTACIJE

05-20-000	Okretni stol
05-20-001	Donje kućište
05-20-002	Gornje kućište
05-20-003	Velika remenica
05-20-004	Stol
05-20-005	Nosač predpojačala
05-20-006	Nosač projektora donji
05-20-007	Nosač projektora gornji
05-20-008	Nosač motora

SAŽETAK

Tema ovog rada je izrada vratnog mehanizma koji služi za zakretanje afektivne robotske glave. Prvotno će biti dizajniran mehanički sustav nalik rotacijskom stolu kako bi se omogućilo normalno zakretanje cijele postave. Dodatna svrha postava je za postavljanje mikrofona koji akviziraju zvučne signale u okolini. Nakon toga dizajnirana je elektronika koja pojačava signal s mikrofona. Konačno je dizajniran program u Ardiuno okruženju koji omogućuje praćenje izvora zvuka na temelju primljenog signala sa sva tri mikrofona.

Ključne riječi: rotacijski stol, mikrofon, Arduino, zvuk, 3D ispis

SUMMARY

The theme of this paper is to make neck mechanism which serves for rotating affective robotic head. Firstly it will be designed mechanical system, similar to rotating table, to enable normal rotation of whole setup. Additional purpose of this system is to hold microphones that acquire sound signals from enviroment. After that it will be designed electrical system to amplify signals from microphones. Finally the software is designed in Arduino enviroment to enable folowing sound source on basis of acquired signal from the three microphones.

Keywords: rotational table, microphone, Arduino, sound, 3D print

1. UVOD

Prva robotika je bila zamišljena i implementirana kao relativno jednostavan sustav za pomoć ljudima u obavljanju repetativnih, teških ili opasnih radnji. Pošto smo u većoj mjeri uspjeli postići željenu namjeru, robotika je krenula prema sustavima koji mogu prirodnije komunicirati s ljudima. Na ovakvom putu javljaju se mnogi problemi od kojih je možda najveći postići kvalitetnu interakciju između čovjeka i robota.

Ljudi su evolucijski naučeni komunicirati i interaktirati s vrlo složenim biološkim sustavima, bile to druge jedinke ili druge vrste. Pod tom predpostavkom kvalitetan sustav treba biti u stanju barem približno oponašati visoko razvijene biološke sustave. Jedna od metoda koja omogućuje lakšu interakciju je opisana u ovome radu.

Većina inteligentnih živih bića, a posebice ljudi kod komunikacije gledaju jedan u drugog, što osobi koja prenosi informaciju daje osjećaj pozornosti te poticaj na nastavak prijenosa informacije. Ako se želi postići takav poticaj kada čovjek interaktira s robotom potrebno je robotu omogućiti sličan način rada. Drugim riječima robot mora reagirati na zvučni podražaj te se okrenuti licem prema njemu.

Ovaj rad počiva na temeljima fizike zvuka te mehaničkog ustroja rotacijskog stola. Fizika zvuka omogućuje definiranje matematičkog okvira za određivanje smjera izvora zvuka dok mehanika rotacijskog stola omogućuje sam zakret vrata mehanizma.

Principijelno najsličniji sustav ovome je Furhat robot [2] koji također može zakretati glavu te projicirati lice na 3D površinu kao [1]. Bitna razlika između makete u nastavku i ovog robota je u cijeni izrade.

Kroz ostatak rada opisane su metode izrade sustava od samog nastanka ideje preko izrade mehanike do izrade elektroničkih komponenti i programa. Svaki dio je objašnjen i potkrijepljen fotografijama ili prikazima iz nekih programskih alata.

2. ZVUK

Prema fizikalnoj definiciji zvuk je vibracija koja propagira kroz neki medij kao akustični val [4]. Zvuk se može širiti kao longitudinalni ili transverzalni val ovisno o mediju kojim se širi. Izvor zvuka je uvijek mehaničke prirode te se svodi na titranje nekog elementa(membrana zvučnika, glasnice, zvono itd.). Brzina širenja valova definirana je prema mediju kroz koji se isti kreću.

Prema biološkoj definiciji zvuk je vibracija frekvencije unutar slušnog raspona ljudi, koji je od 20Hz do 20kHz [5]. U ovom slučaju pojam zvuk se svodi na percepciju raspona frekvencija unutar mozga.

2.1. Percepcija zvuka kod ljudi

Područje koje se bavi percepcijom zvuka naziva se psihoakustika. Kao i kod većine životinja tako i kod ljudi zvuk služi za snalaženje u prostoru, detektiranje opasnosti te komunikaciju. Unutar tog konteksta suprotno zvuku može se definirati šum.

Šum je većinom neugodan zvuk kao takav, ako se pojavljuje u prijenosu signala ili automatici potrebno ga je poništiti da ne narušava korisni signal. S druge strane šum je važan za prepoznavanje izvora zvuka ta definiranje kvalitete istog.

Puna percepcija zvuka kod ljudi može se definirati sa šest parametara [6]:

- Visina zvuka (ton)
- Trajanje
- Jačina
- Boja
- Tekstura
- Prostorna lokacija

U novije vrijeme dodatno se mogu definirati temporalna envelopa te temporalna struktura koje kod ljudi uvjetuju percepciju promjene zvuka u vremenu.

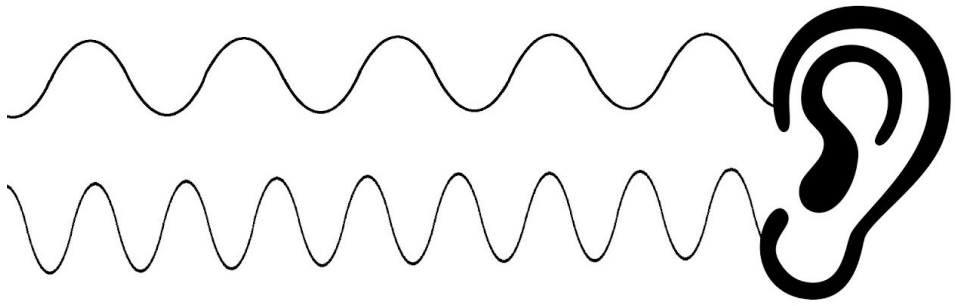
2.1.1. Visina zvuka (ton)

Visina zvuka ili ton je jedan od parametara kod percepcije zvuka.

Definirana je frekvencijom najsporije vibracije u zvuku (tzv. fundamentalni harmonik). Sama percepcija visine zvuka može varirati od osobe do osobe sukladno njihovim prijašnjim iskustvima.

Proučavanje visine zvuka izvodi se pomoću dijagrama frekvencijske karakteristike na kojem su vidljive amplitude u ovisnosti o frekvenciji samog zvuka, dok kod ljudi percepcija proizlazi iz nesvjesne analize.

Kod pojma visine zvuka važno je prikazati bijeli te ružičasti šum. Kod bijelog šuma amplituda je jednaka na svim frekvencijama tj. amplitudno frekvencijska karakteristika je horizontalan pravac. Ružičasti šum je definiran pomoću energije signala za određenu frekvenciju, pa se može reći da je kod tog šuma energija obrnuto proporcionalna frekvenciji signala.



Slika 1. Primjer visine zvuka

Na slici gornji signal je percipiran kao niži ton pošto se radi o tonu s manjim brojem titraja u jedinici vremena. Shodno tome, donji signal je višeg tona.

2.1.2. Trajanje zvuka

Teoretski trajanje zvuka se definira kao vrijeme od kad je određeni zvuk primijećen do trenutka kada se isti promijeni ili prestane. Realno trajanje zvuka ne mora biti čvrsto vezano uz prvotnu definiciju.

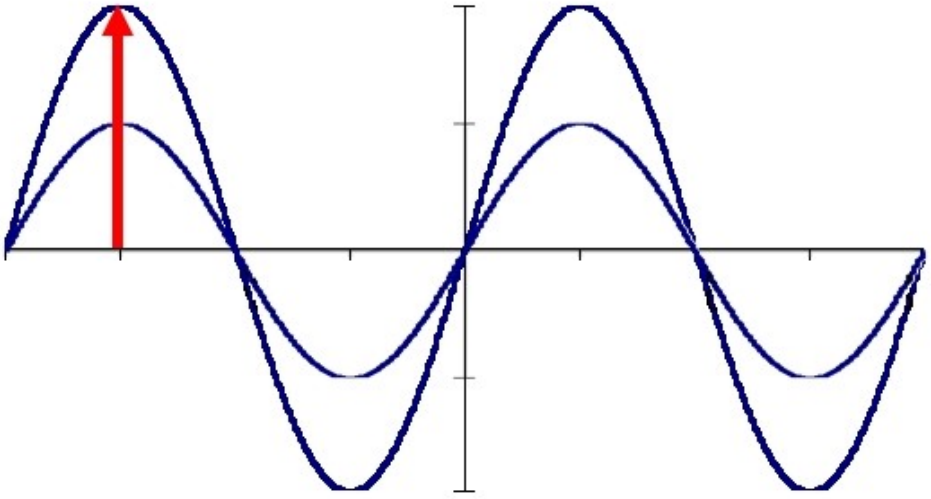
U slučaju vrlo zašumljenog okruženja percepcija zvuka može biti kontinuirana bez obzira što se stvarno radi o isprekidanom zvuku. Takav efekt se događa iz razloga što je šum sličnog inteziteta i frekvencije kao i prenošeni ton.

2.1.3. Jačina ili intezitet zvuka

Pojam jačine zvuka nije direktno vezan s fizikalnom definicijom visine amplitude tog zvuka. Primarno se radi o sumaciji količine energije određenog zvuka unutar recepcijskog sustava tokom nekog vremena.

Jednostavnije rečeno određeni zvukovi mogu imati više amplitude ali zvučati slabije nego drugi. Razlog tome je što kompliciraniji zvukovi imaju veći raspon frekvencija te okidaju više neuronskih receptora.

Dodatno vrlo kratni zvukovi (manje od 200ms) iako veće amplitude, percipiraju se kao manjeg inteziteta, iz razloga navedenog odlomak ranije.



Slika 2. Primjer jačine zvuka

2.1.4. Boja zvuka

Boja zvuka je percipirana kao kvaliteta određenog zvuka te omogućuje prepoznavanje različitih zvukova prije svjesne obrade istih.

Kako bi to bilo moguće u zvuku se traže frekvencijski tranzijenti, količina šuma, visina i dr.

Pomoću boje zvuka moguće je raspoznati različite zvukove u prirodi ili umjetne zvukove u okruženju.

2.1.5. Tekstura zvuka

Tekstura zvuka predstavlja pojam koji govori od koliko različitih izvora se sastoji određeni zvuk te kako se ti izvori isprepleću.

Pojam je vrlo često korišten u smislu glazbe kao opis uloge instrumenata u skladbi. Prema tome je i definiran pojam jedinstva, polifonije i homofonije.

S druge strane ljudi mnogo puta primjenjuju pojam kakofonija za vrlo mnogo izvora zvuka npr. prepunjen kafić ili postaja javnog prijevoza.

2.1.6. Prostorna lokacija

Prostorna lokacija podrazumijeva prepoznavanje prostornog položaja izvora određenog zvuka. Ljudi zbog vrlo složene strukture recepcijskog sustava mogu samo s dva uha prepoznati položaj izvora u potpunom trodimenzionalnom prostoru.

Dodatno je moguće pomoću kombiniranja prije navedenih parametara zvuka, prepoznati točno određen izvor zvuka tj. izdvojiti ga iz mase izvora.

2.2. Percepcija zvuka kod robotskih sustava

Kako bi se robotima omogućila percepcija zvuka potrebno je definirati različite elektromehaničke sustave. Prvi dio sustava je elektromehanički receptor ili mikrofon. Nakon toga je potreban sustav obrade i prepoznavanja primljenih signala koji se većinom sastoji od mikroračunala ili računala te specijaliziranog čipa za akviziciju signala u zadanom rasponu frekvencija (A/D konverteri sa dovoljno visokim brojem uzoraka po sekundi).

Mikrofoni mogu biti u mnogo različitih izvedba, no ipak je moguće izdvojiti neke od više korištenih:

- Piezoelektrični [8]
- Electret kondenzatorski [7]
- MEMS (Mikro električko mehanički sustav)
- Laserski
- Mikrofon od optičkog vlakna

Ovisno o primjeni primljenog zvuka sustavi za obradu mogu biti jednostavniji ili složeniji. Primjer jednostavnijeg sustava je detekcija smjera dolaska zvučnih valova dok se kao kompliciranije mogu navesti osobni asistenti koji prepoznavaju same uzorke zvuka(ljudski glas) te se ponašaju prema tome.

Metoda detekcije smjera dolaska zvučnih valova bit će primijenjena i podrobno objašnjena unutar ovog rada, dok će naprednije metode biti ukratko objašnjene u nastavku.

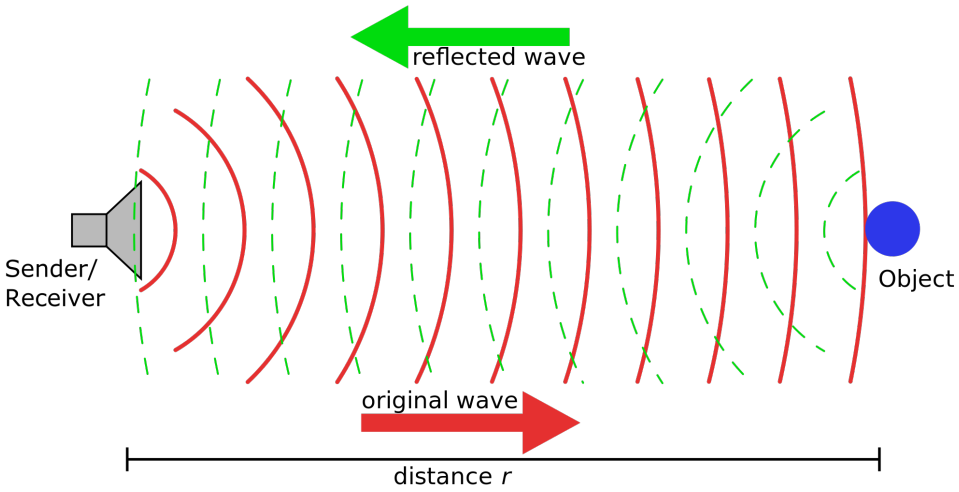
Posebni slučaj detekcije zvuka kod robota korijene vuče iz prirode, te ne spada direktno u kategoriju prepoznavanja zvuka već u kategoriju prepoznavanja prostora. Metoda je poznata pod imenom eholokacija.

Eholokacija je metoda koja se sastoji od slanja visokofrekventnih audio valova u okolinu te mjerenja vremena do kada se ti valovi vrate. Važno je napomenuti da ova metoda iziskuje poznavanje čim točnije brzine zvuka u određenom mediju. Uređaj koji se koristi u robotici za eholokaciju naziva se sonar.

2.2.1. Metoda eholokacije

Eholokacija je metoda prepoznavanja prostora pomoću zvuka. Najjednostavniji tip eholokacije je sonar za mjerenje udaljenosti, dok su kompliciraniji primjeri sonari kod podmornica, brodova i zrakoplova [9]. U prirodi eholokaciju koriste mnoge životinje od kojih je najpoznatiji šišmiš. Važna razlika između sonara i sustava koje koriste šišmiši je u brzini i preciznosti prepoznavanja prostora oko sebe. Dok sonar ima problema s prepoznavanjem malih predmeta šišmiš je u stanju prepoznati žicu debljine manje od milimetra.

Sonar radi na principu odbijanja visoko frekventnog zvuka od prepreka te mjerenjem proteklog vremena od slanja do primanja zvučnog signala. Kao što je bilo ranije napomenuto, točnost mjerenja se zasniva na točno određenoj brzini zvuka za medij u kojem se radi sa sonarom. Dodatan važan zaključak koji se može izvesti iz svega prije navedenog je da sonar radi samo unutar nekog medija dok se u zrakopraznom prostoru mogu primjenjivati metode zasnovane na drugim sustavima [9].



Slika 3. Primjer eholokacije

2.2.2. Metode prepoznavanja ljudskog govora

Ljudski govor je jedan od najsloženijih sustava prijenosa informacija stoga je prepoznavanje pomoću računala vrlo kompliciranо.

Metode koje se danas primijenjuju većinom ukjučuju neku vrstu umjetne inteligencije [11] te baze govorenih riječi za određeni jezik ili metode poznate kao skriveni Markov model(eng. hidden Markov model).

Primjena ovih metoda je danas vrlo raširena, najpoznatija je u osobnim asistentima (Google Home, Siri, Amazon Echo itd.). Dodatna primjena je u automobilima kao glasovne naredbe, kao zamjena za agente u pozivnim centrima, za učenje drugog jezika kao učitelj itd.

Slika 4. Google Home i Amazon Echo

3. RAZRADA IDEJE

Cilj ovog rada je izraditi okretni stol sposoban detektirati prostornu lokaciju zvuka te prema tome rotirati predmet koji se nalazi na istome. Sekundarna funkcija je mogućnost prihvata nano projektora na stol te provođenje kabela kroz stol. Kabeli moraju biti u središtu rotacije kako ne bi došlo do savijanja istih ili pucanja. Pod tom pretpostavkom potrebno je dizajnirati sustav koji u centru rotacije ima prolaznu rupu. Kako se većina rotacijskih stolova uležišćuje u središtu rotacije izrada ovog sustava iziskuje posebnu metodologiju. Jedno od mogućih rješenja je korištenje dva ležaja većeg promjera kako bi se postigao zahtjev za rotacijom te dovoljno prostora za kabele. Slijedeći problem je pogon samog stola. Zahtjev na pogon je postići točnost rotacije relativno dobre brzine te tihi rad. Jedini način za zadovoljiti sve parametre je koristiti remenski prijenos. Problem remenskog prijenosa je potreba za remenicom točno određenih dimenzija, no pošto je promjer definiran unutarnjim promjerima ležajeva potrebnih za rotaciju stola, vrlo teško će biti naći komercijalno dostupnu remenicu tog promjera.

Kada su razrađeni mehanički problemi potrebno je razraditi koncept prostorne lokacije zvuka. Pošto je zahtjev na sustav da se rotira u smjeru izvora zvuka potrebno je kreirati prostorni raspored mikrofona takav, da se omogući optimalno prikupljanje zvuka iz svih smjerova. Najbolji način za postići ovo je postaviti mikrofone pod 120° u ravnini okomitoj na os rotacije.

4. IZRADA MEHANIKE

Unutar ovog dijela bit će objašnjena izrada mehanike potrebne za funkcionalan okretni stol. Svrha okretnog stola je zakretanje glave prema osobi koja trenutno vrši verbalnu interakciju s agentom. Samo zakretanje glave potaknuto je detekcijom smjera iz kojeg dolazi zvuk.

Pokretanje stola omogućeno je pomoću step motora kako bi se osigurala točnost pozicioniranja uz relativno nisku cijenu izrade. Unutar samog sustava dodan je prekidač krajnjeg položaja kako bi se pri paljenju mogao odrediti referentni položaj stola.

Sustav se sastoji od okretnog stola, kućišta, upravljačke i energetske elektronike te mehanizma za smanjenje momenta rotacije stola(ležajevi s popratnim ulažištenjima).

U nastavku bit će prikazana izrada svakog pojedinačnog dijela i podrobnije objašnjena funkcija istog. Dodatno, u prilogu se nalaze tehnički crteži svih dijelova.

Dijelovi su projektirani s predpostavkom izrade na 3D pisaču, a kasnije su isti i izrađeni. Korišteni 3D pisač je Creality Ender 3.

Projektiranje sustava provedeno je unutar CAD alata Solidworks na način da su postavljeni unaprijed definirani dijelovi te je oko njih građen ostatak sustava. Stoga će u slijedećem dijelu, dijelovi biti prikazani u poretku kojem su izrađeni.

Kod ovakvog sustava važno je unaprijed definirati sve potrebne standardne dijelove kako bi se unutar samog sklopa nestandardni dijelovi mogli konstruirati prema njima. Stoga je prije izrade bilo potrebno naći ili nacrtati 3D modele svih standardnih dijelova.

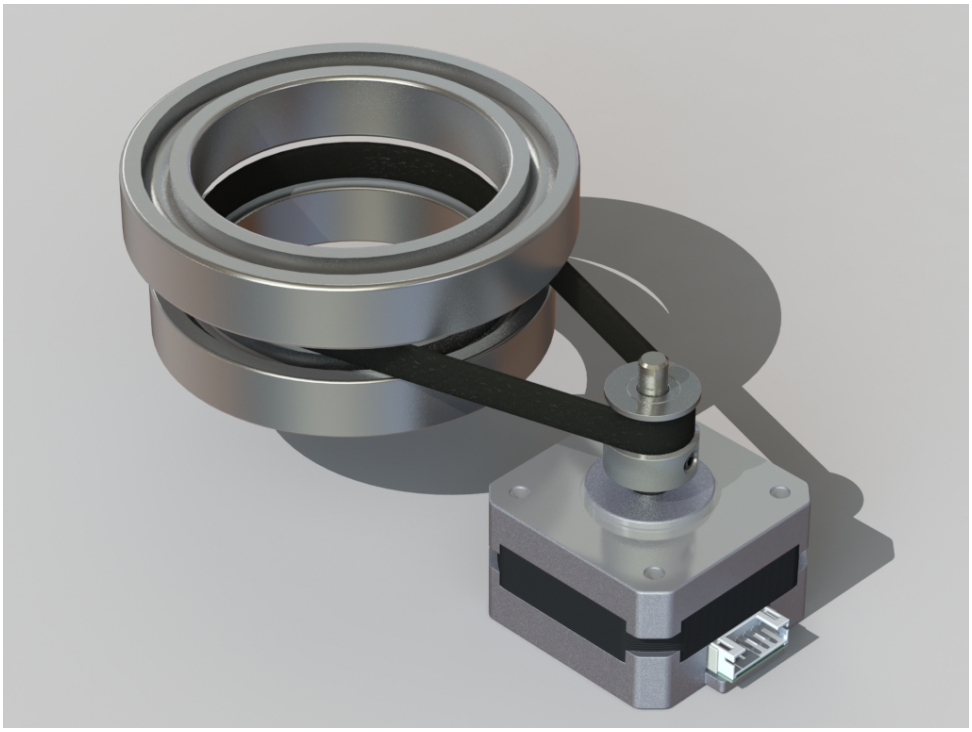
4.1. Gornji poklopac

Gornji poklopac je prvi dio koji je projektiran. Svrha istog je uležištenje stola i kao nosač motora. Dodatna svrha je zaštititi elektroniku od mogućih oštećenja. U središte stola ugrađuju se dva ležaja kako bi se uklonila mogućnost zakretanja stola. Između dva ležaja proveden je remen tipa GT2 za koji je definiran prorez na poklopcu. Nadalje na donjoj strani poklopca su izrađene staze na koje se dalje veže nosač motora.

Dimenzije samog stola prilagođene su dimenzijama stola na printeru i zahtjevima drugih dijelova.

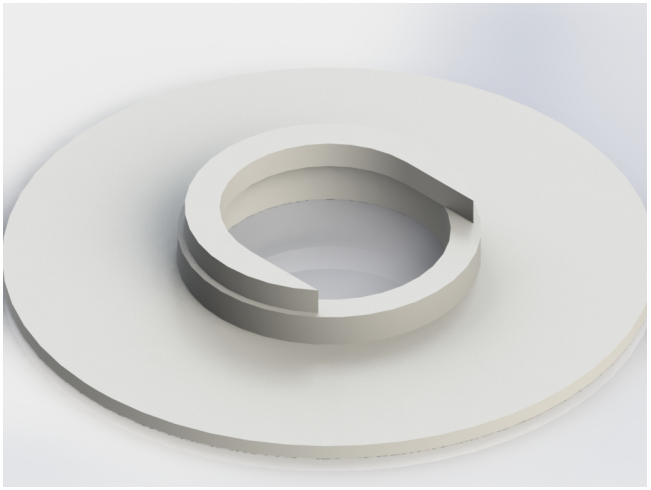
4.1.1. Postupak izrade

Prvotno su postavljeni ležajevi u međuodnos te je dodan motor sa remenicom. Nadalje, udaljenost motora je definirana preko automatskog alata u CAD za dizajniranje remenskih prijenosa.



Slika 5. Osnova sklopa (Motor i ležajevi)

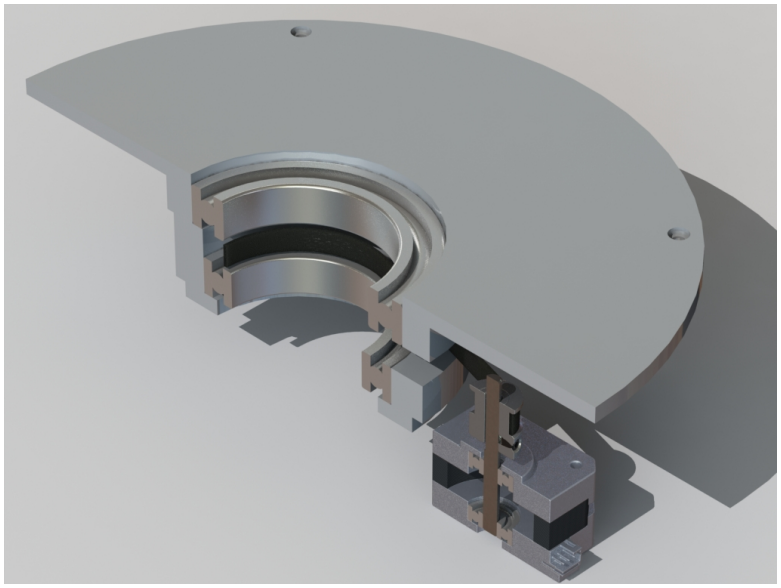
Kada su svi dijelovi stavljeni u međuodnos, te su potpuno definirane sve relacije, moguće je pomoću alata za mjerenje dimenzija definirati veličine dijela. Nakon toga može se pristupiti projektiranju dijela.



Slika 6. Gornji poklopac

Vanjski promjer dijela definiran je veličinom vrata glave iz rada Afektivna robotska glava koja imitira izražaje lica osobe u interakciji[1]. Dodatno, visina je definirana razmakom i debljinom ležajeva koji su stavljeni.

Nakon svih definiranih dimenzija i izrađenog dijela dodaje se u već prije definiran sklop radi provjere točne izrade.



Slika 7. Sklop s gornjim poklopcem

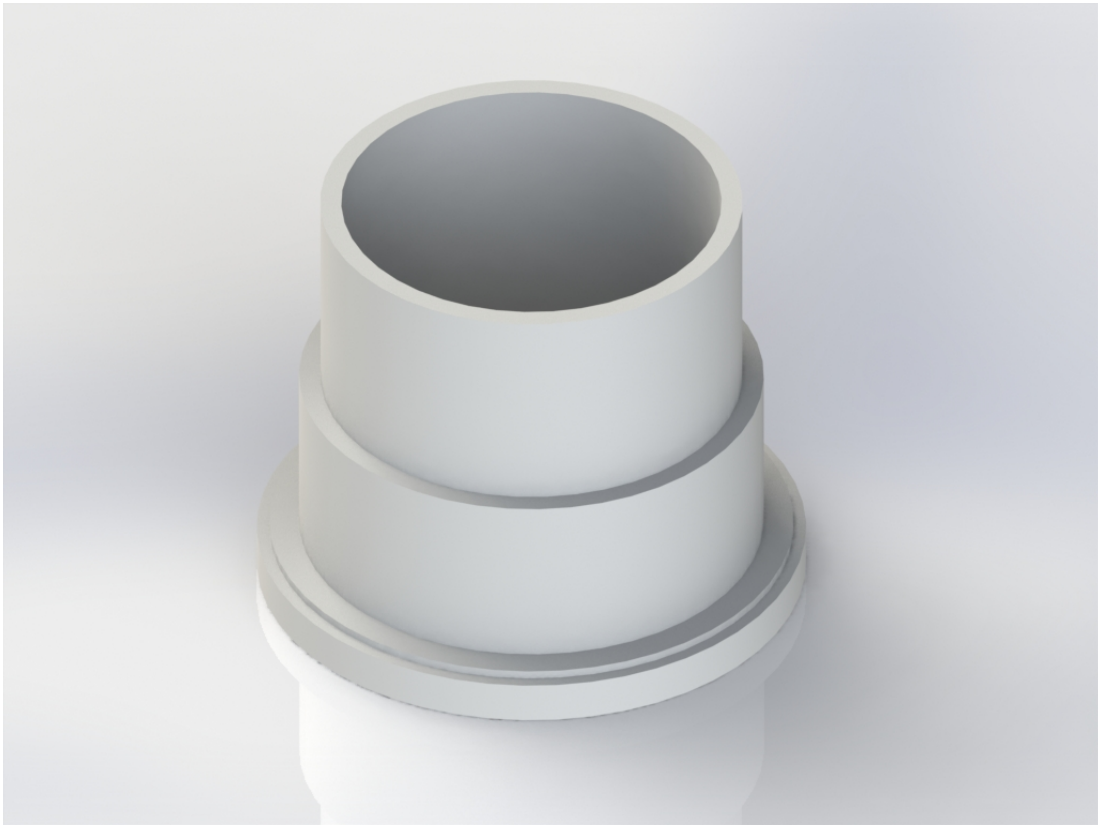
4.2. Velika remenica

Svrha velike remenice je rotacija stola te unutarnje uležištenje. Velika remenica je definirana unutarnjim promjerima ležajeva i s dodatnim zahtjevima da kroz središte mogu prolaziti kablovi.

Najkompliciraniji dio je generiranje remenice koja se poklapa sa zubima remena. Kako bi bilo moguće izraditi ovakvu remenicu, potrebno je uzeti profil zuba s tehničkog opisa te ga izraditi na već prije modeliranom dijelu, što će biti detaljnije prikazano u nastavku.

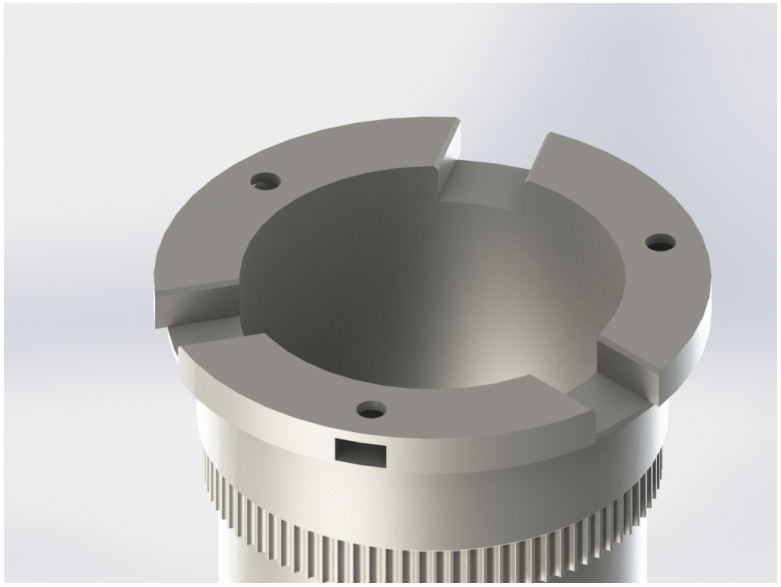
4.2.1. Postupak izrade

Dvije cirkularne površine koje nasjedaju na ležaj definirane su dimenzijama koje se uzimaju iz kataloga. Visina remenice je definirana mjerenjem razmaka ležajeva i dodavanjem visine potrebne za prihvat stolne ploče te izdanka za prekidač nultog položaja.



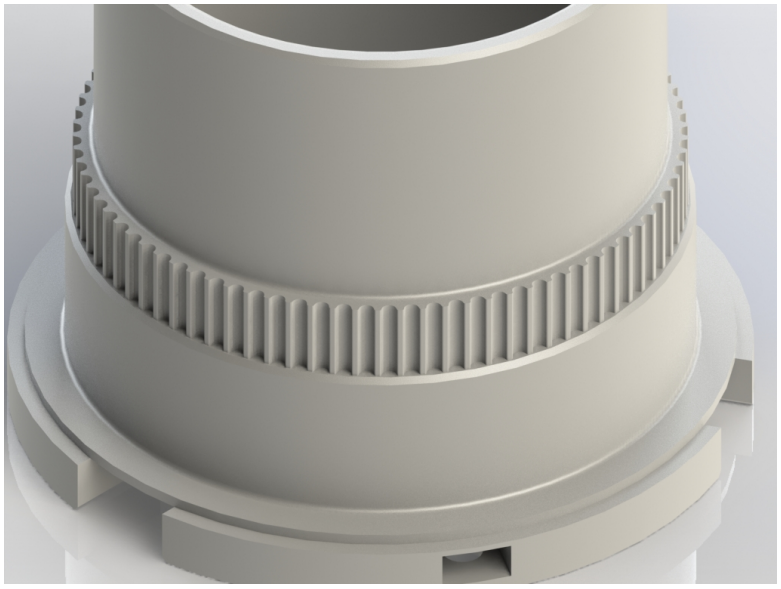
Slika 8. Velika remenica

Kako bi bilo moguće povezati stol s remenicom potrebno je dizajnirati tri utora pod 120 stupnjeva što omogućuje centriranje dijela. Dodatno se projektiraju rupe te utori za matice kako bi bilo moguće spojiti stol s remenicom.



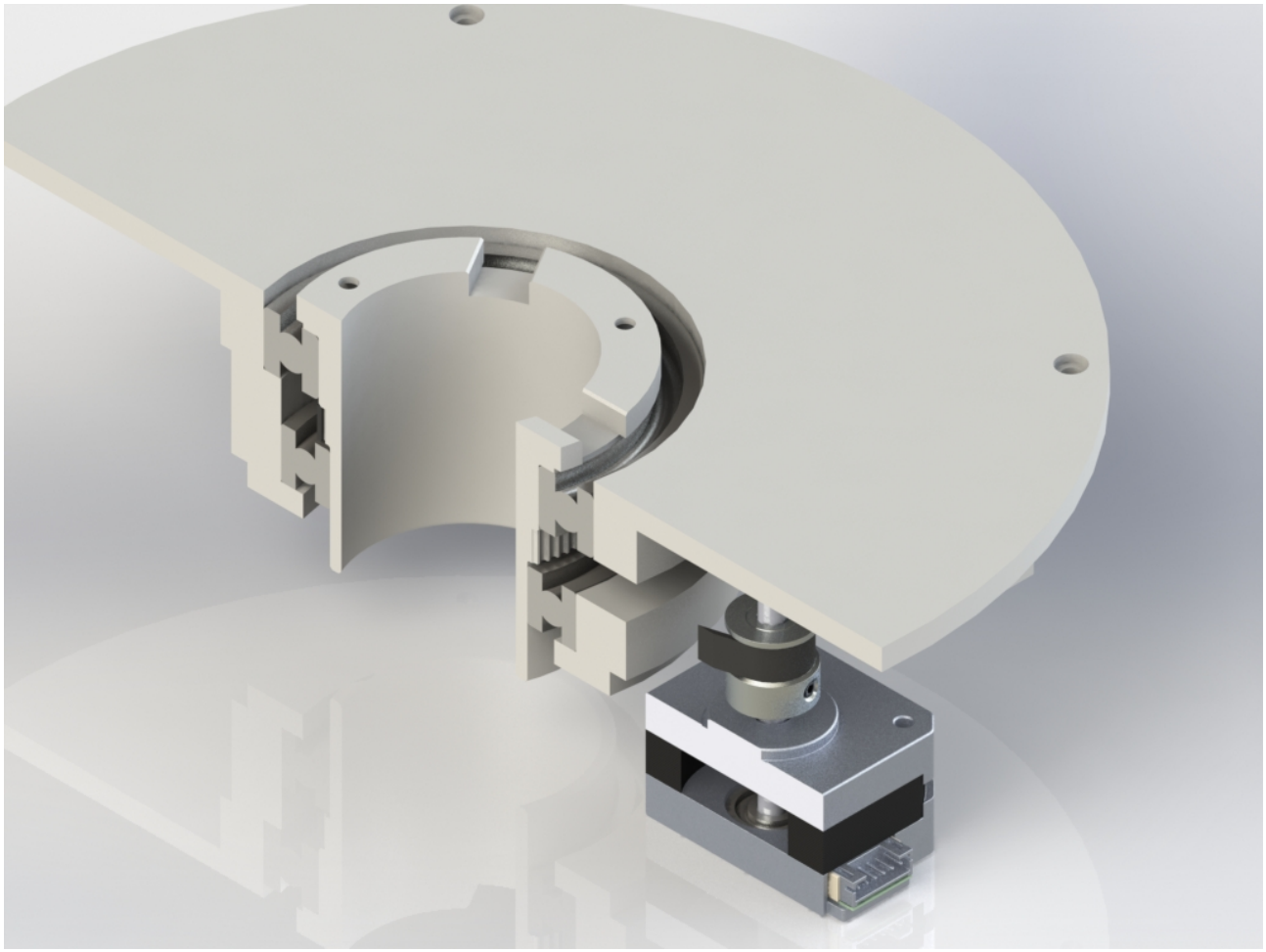
Slika 9. Velika remenica prikaz zubi

Kako je bilo ranije objašnjeno potrebno je definirati remenicu. Profil je preuzet s referentnog priručnika. Nakon toga je profil prenesen na površinu te je pomoću *Extruded-cut* izvučen konačan profil.



Slika 10. Velika remenica prikaz - izreza za spajanje stola

Kada je sve izrađeno postupak je isti kao i kod prvog dijela, provjera se radi dodavanjem u sklop.



Slika 11. Sklop s remenicom

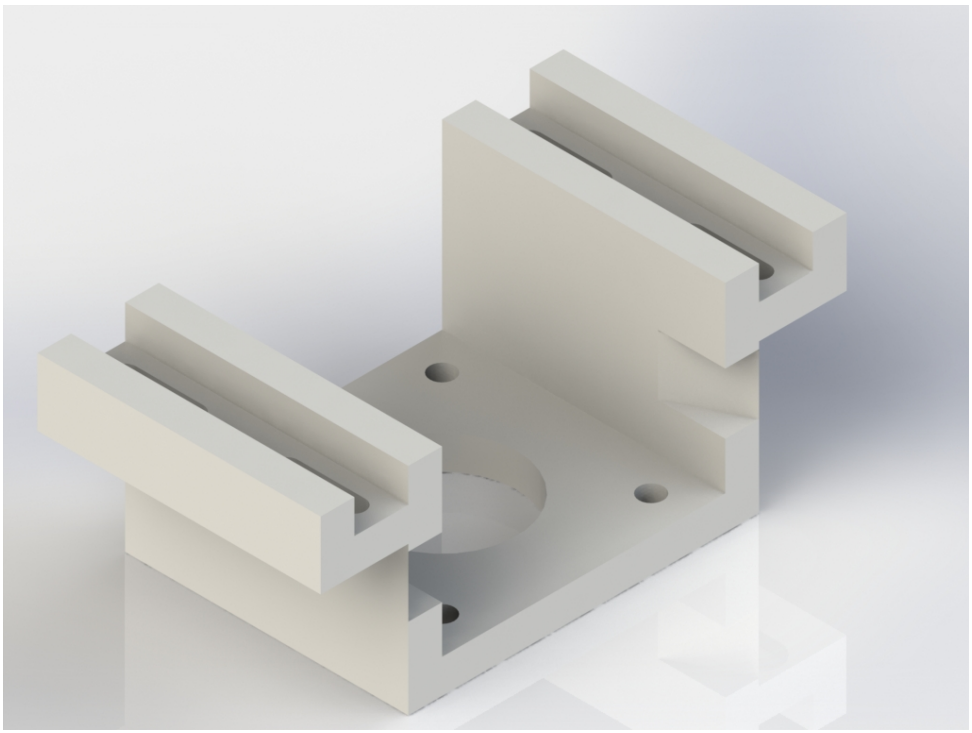
Na slici je vidljivo da sve komponente sjedaju te je moguće zaključiti da su spremne za izradu. No pošto se radi o postupku 3D ispisa komponente treba dodatno obraditi prije same izrade što je opisano u sljedećem poglavlju.

4.3. Nosač motora

Jedan od jednostavnijih dijelova cijelog sklopa je nosač Nema17 step motora. Motor je u cijelom sklopu ovješen na nosač te shodno tome i na gornji poklopac. Kod izrade je bilo važno voditi računa o mehanizmu zatezanja remena. Unutar ovog rada zatezanje je izvedeno pomoću kliznih staza na koje se stavlja nosač. S druge strane nosač je dizajniran s utorima za vijke kako bi bilo moguće pomicanje cijelog nosača.

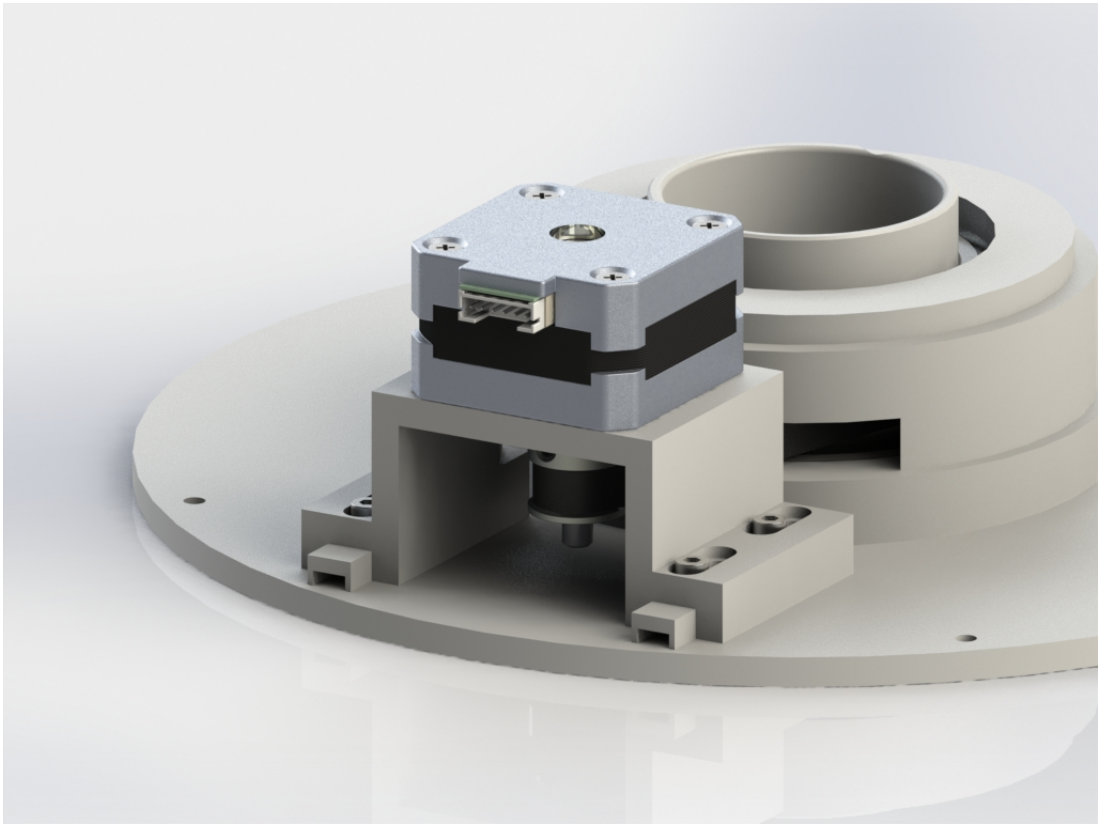
4.3.1. Postupak izrade

Prvotno su uzeti CAD modeli motora koji je uporabljen u ovom radu te tehnička dokumentacija prema kojoj je definiran prihvatna ploča na nosaču motora. Nakon toga preko sklopnih mjera je definirana visina nosača, uzimajući u obzir duljinu vratila na motoru. Ostale mjere samog nosača su definirane prihvatom na gornjem poklopcu te vanjskim dimenzijama motora.



Slika 12. Nosač motora

Nakon dodavanja nosača u sklop rade se prilagodbe kliznih utora prema unaprijed dizajniranom remenskom prijenosu. Cilj prilagodbe je osigurati mogućnost kvalitetnog zatezanja remena.



Slika 13. Sklop s nosačem motora

Unutar programa Solidworks su odabrani realni parametri remena što omogućuje dobivanje točnog odmaka od velike remenice za maksimalno napnut remen. Kako bi omogućili realno zatezanje, cilj je idealni slučaj staviti u položaj koji omogućuje maksimalni pomak u jednu i drugu stranu. Pomak bliže središtu omogućuje određeni faktor sigurnosti kod sastavljanja, dok odmak od središta omogućuje uklanjanje elastičnosti remena.

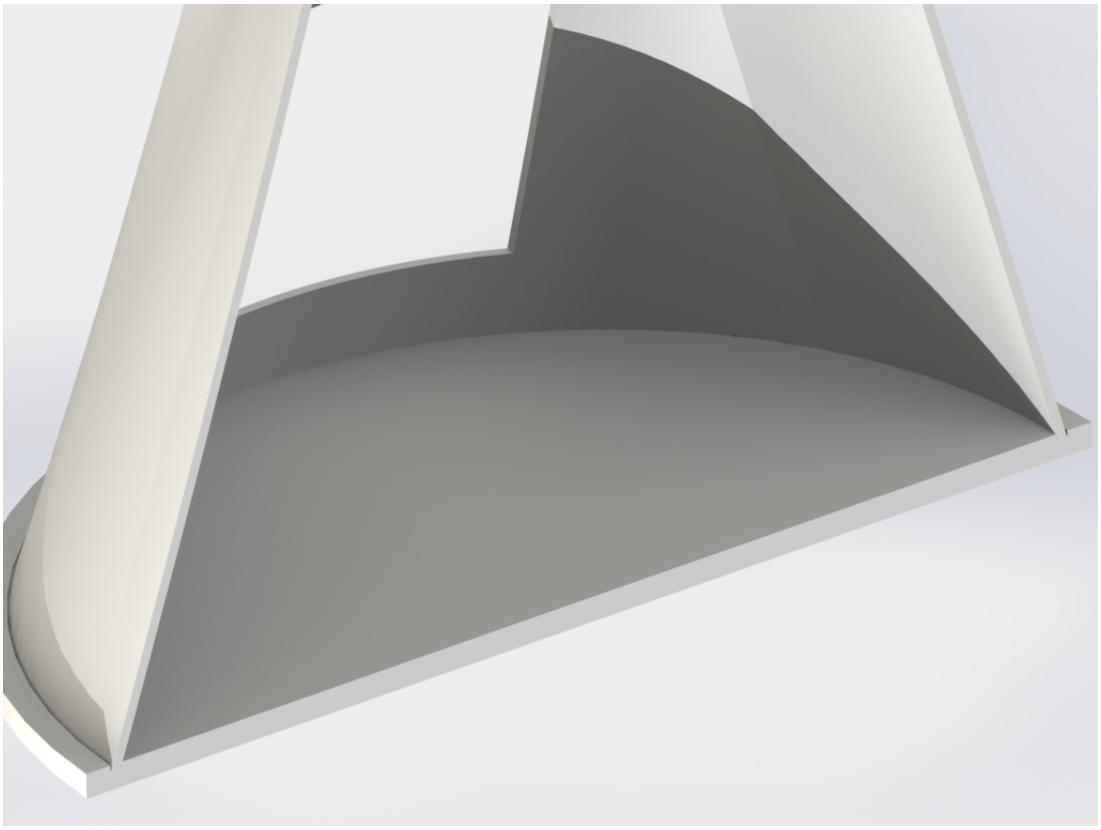
4.4. Rotacijski stol

Svrha rotacijskog stola je omogućiti stabilnu platformu na kojoj stoji komponenta koja se želi rotirati, u slučaju ovog rada vrat, glava i projektor. Dimenzije stola određene su prema tim komponentama i naravno prema maksimalnoj veličini stola za 3D ispis.

Stol je dizajniran s rubom koji skriva gornju pričvrsnu ploču i vijke. Dodatno je bilo potrebno konstruirati suprotne utore koji se spajaju s velikom remenicom te upuštene rupe za vijke s kojima se sve to stegne.

4.4.1. Postupak izrade

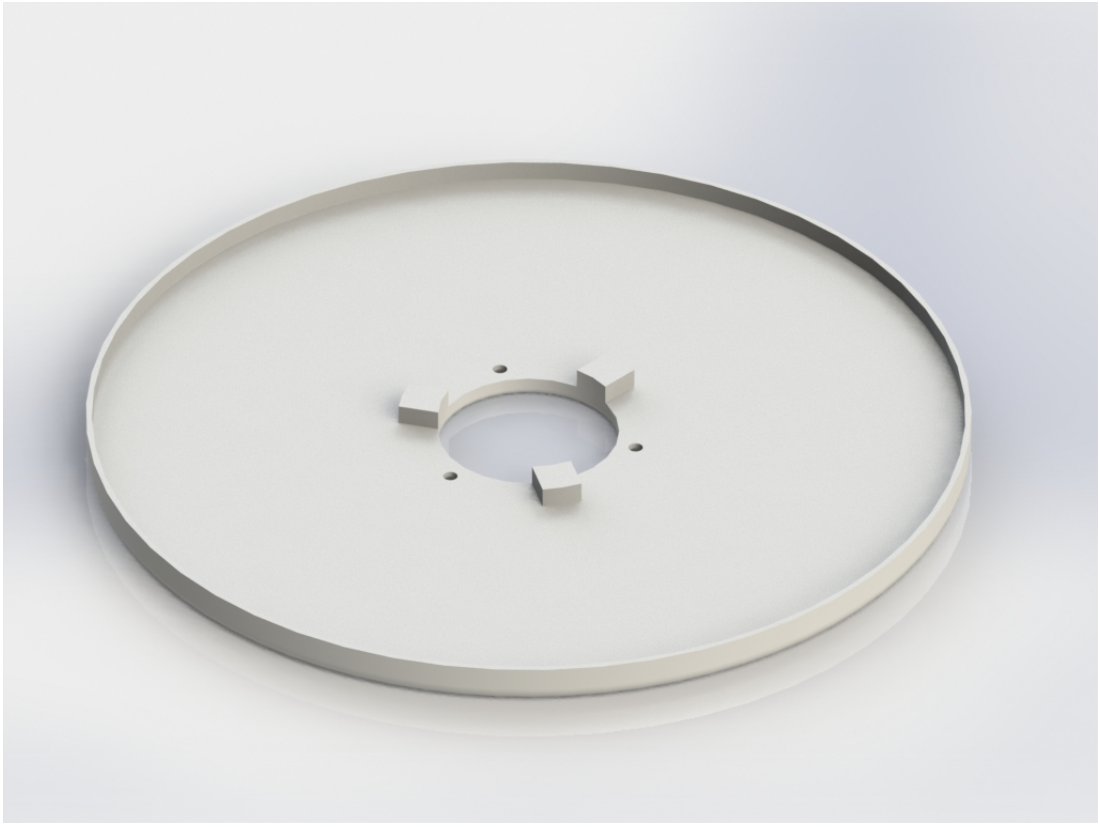
Kao predložak za stol je uzet vrat robota iz rada [1] te je krenuto od toga kao osnovne dimenzije. Na stolu je kreiran utor kako bi se omogućilo kvalitetno nasjedanje to istog vrata.



Slika 14. Prilagodba stola na vrat mehanizma

Nakon izrade osnovne ploče opisane prethodno dodani su izdanci koji sjedaju u utore velike remenice. Kod konstruiranja izdanaka važno je uzeti u obzir tolerancije mjera kako bi se omogućilo dobro nasjedanje elemenata.

Nakon toga definira se provrt u središtu promjera istog, kao unutarnji promjer velike remenice da bi se omogućilo prolaz kabela. Konačni dio je konstruirati rub koji sakriva gornji poklopac tako da se definira rubna kružnica te izvuče materijal.



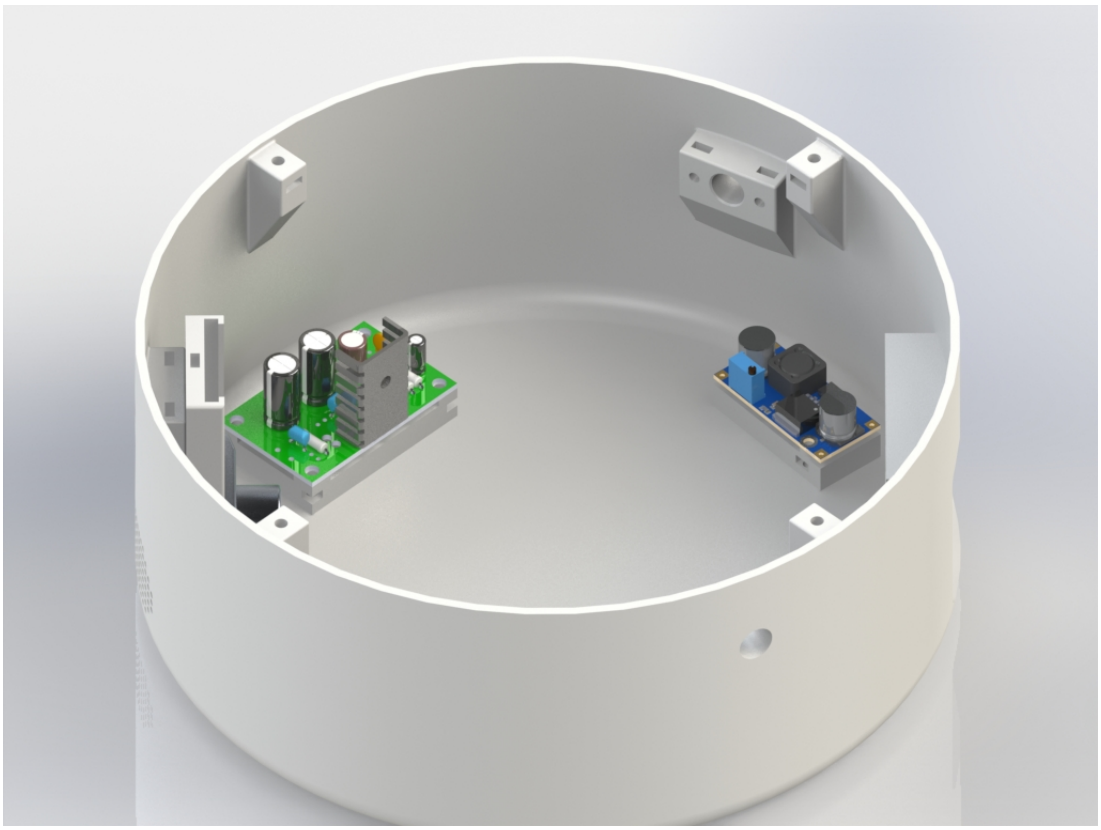
Slika 15. Izgled donje strane stola

4.5. Kućište

Kućište je jedan od primarnih i definitivno najvažnijih dijelova cijelog sklopa pošto je temelj na koji se pričvršćuju svi dijelovi. Isto tako kućište pruža potporu za svu potrebnu elektroniku. Unutar kućišta se nalazi glavna kontrolna ploča, napajanje, pojačalo zvuka te tri predpojačala za mikrofone. Dodatno na kućištu se nalazi konekcijska ploča koja je dodatno pričvršćena na konstrukciju radi jednostavnije zamjene konektora u slučaju promjene funkcionalnosti.

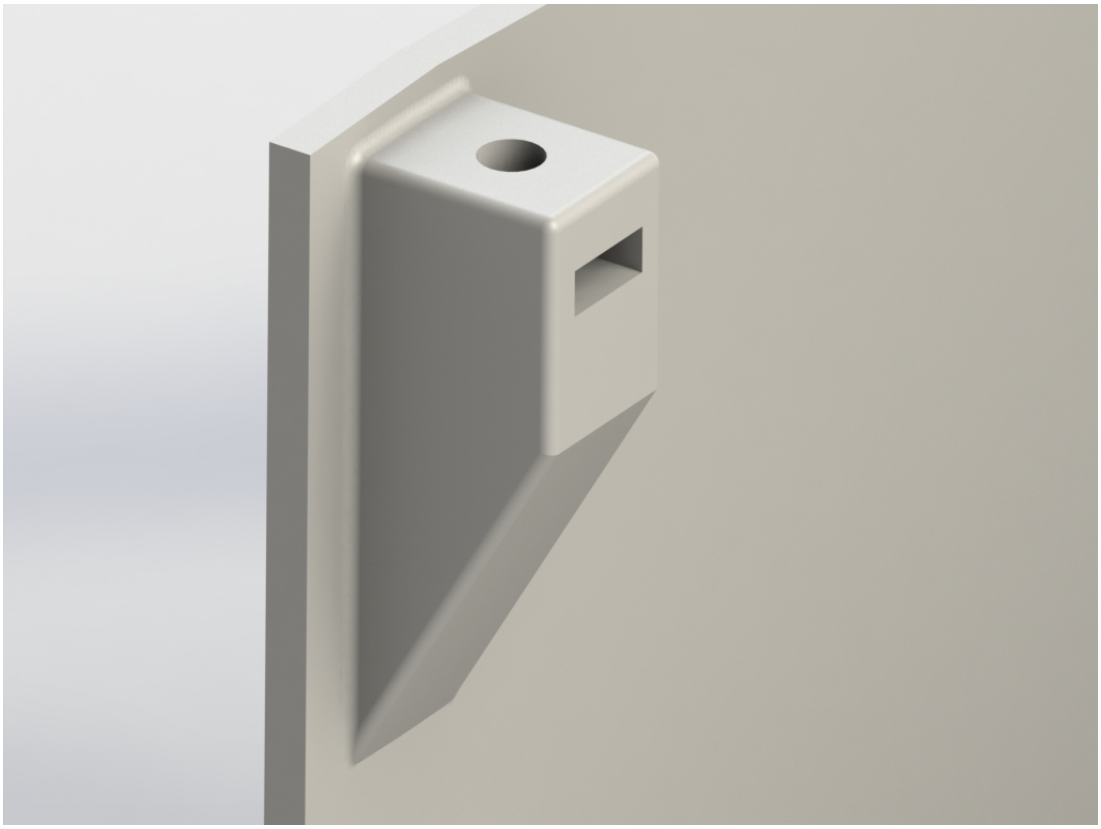
4.5.1. Postupak izrade

Kako bi bilo moguće izraditi kućište sa svim nosačima za elektroniku bilo je potrebno prvotno izraditi elektroničke pločice. Pločice su izrađene u programu Eagle PCB kompanije Autodesk. Prednost ovakve izrade je mogućnost izvoza modela pločice u STL formatu koje je moguće uvesti u SolidWorks. Na ovakav način je moguće vrlo brzo konstruirati nosače za svaku konstruiranu tiskanu pločicu.



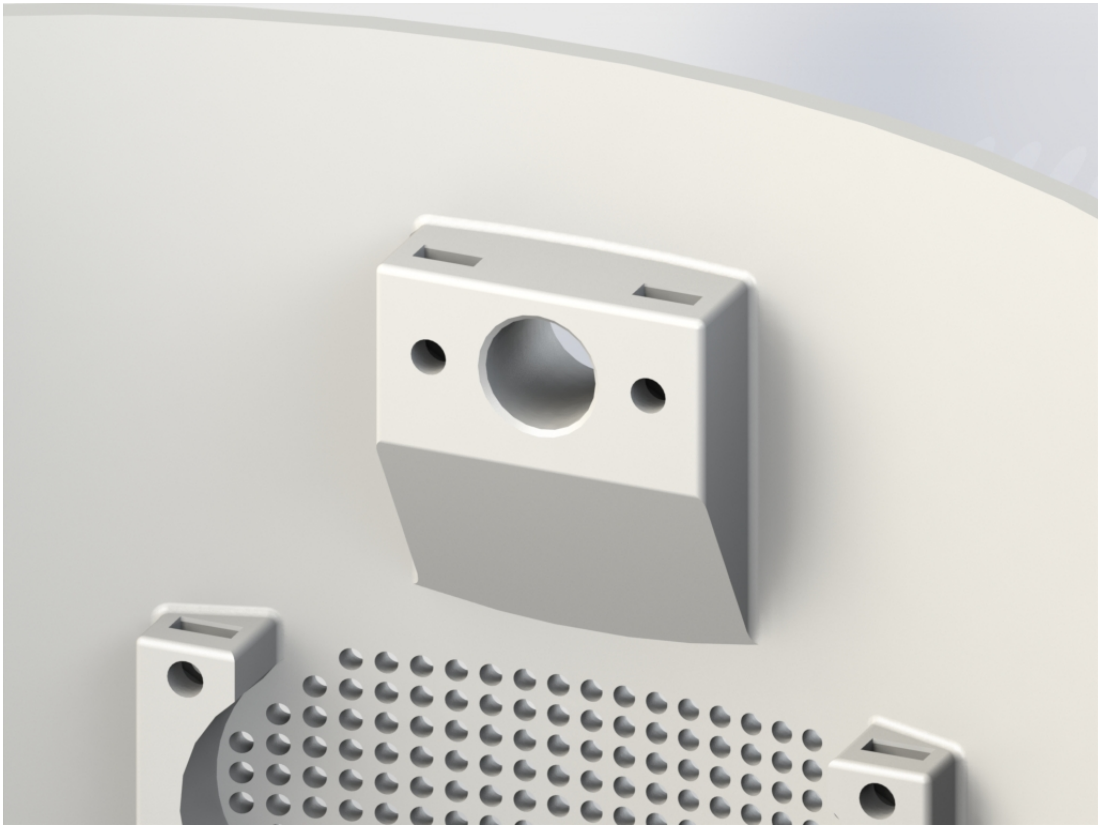
Slika 16. Kućište s elektroničkim komponentama

Slijedeći dio koji se izrađuje na kućištu je nosač gornjeg poklopca. Pošto se radi o izdancima s unutarnje strane kućišta, zbog uštede materijala izdanci su samo na vrhu kućišta. No kako se radi postupkom FDM ispisa potrebno je s donje strane izraditi skošenja pod kutem koji se određuje iskustveno.



Slika 17. Nosač gornjeg poklopca

Nakon toga izrađena su tri provrta za mikrofone oko kojih je s unutarnje strane dodan izdanak na koji se pričvršćuje tiskana pločica predpojačala. Dodatno je izrađen nosač za pločicu predpojačala te je osmišljen mehanizam za pričvršćivanje. Izdanci sa svake strane provrta za mikrofon imaju provrte za vijke na koje se pričvršćuje pločica s utorima. U ovakvu pločicu se umeće tiskana pločica koja se s gornje strane pričvrsti s dodatnim vijkom.



Slika 18. Prihvat mikrofona

4.6. Nosač predpojačala

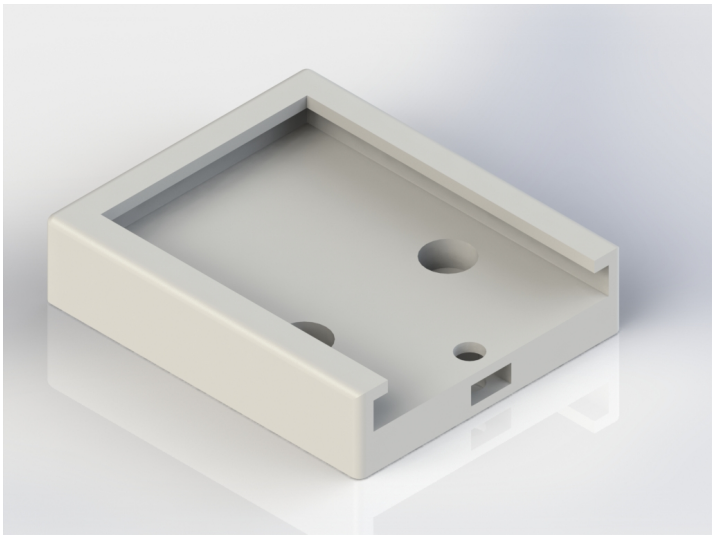
Nosač predpojačala ima dvojaku namjenu:

- nosi tiskanu pločicu predpojačala te osigurava da ne dođe do kratkog spoja s ostatkom elektronike
- osigurava mikrofon unutar ležišta od aksijalnog pomaka

Nosač je dizajniran tako da se mora poštivati točan redoslijed sklapanja kako bi bilo moguće pričvrstiti cijeli sklop. Prvi korak je umetanje mikrofona u ležište, nakon toga se pričvršćuje sam nosač pomoću dva vijka, te se konačno umetne tiskana pločica. Tiskana pločica je zaštićena od ispadanja dodatnim vijkom.

4.6.1. Postupak izrade

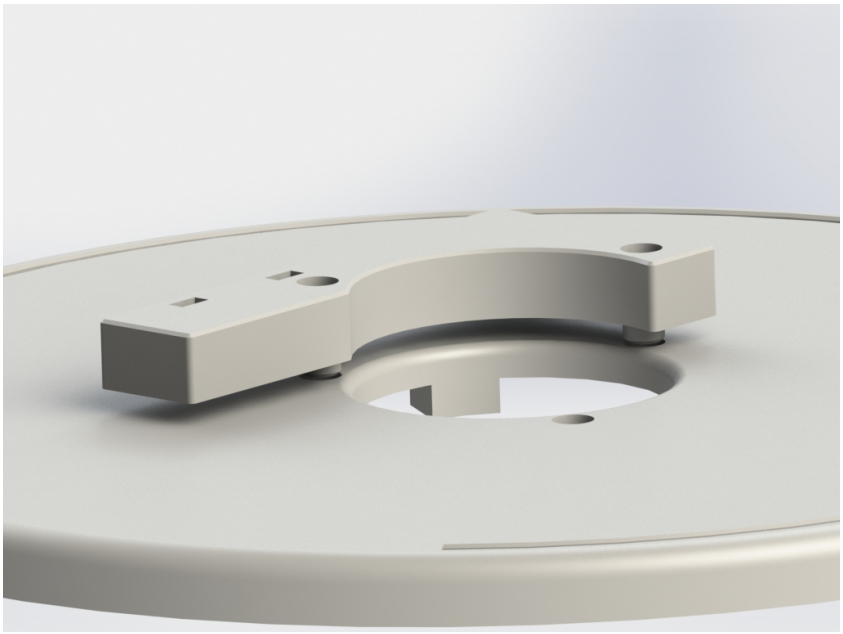
Kako bi se mogao definirati nosač bilo je potrebno točno odrediti mjere tiskanih pločica, uz uvjet da su sve tri tiskane pločice istih dimenzija. Prema njima dizajniran je utor u koji sjedne pločica s ciljem da je dobro pričvršćena, a opet da je moguća relativno laka ugradnja. Dodatno je dizajniran prorez sa stražnje strane nosača kako bi se omogućilo provlačenje žica s mikrofona prema pločici. Kako su vijci koji nose nosač točno ispod tiskane pločice, bilo je potrebno dodatno upustiti iste da ne bi došlo do kratkog spoja.



Slika 19. Nosač tiskane pločice

4.7. Nosač projektora

Nosač projektora dizajniran je iz dva zasebna dijela kako bi se omogućilo što lakše sastavljanje.Dizajniran je na način da se može dodati na stol bez preinaka samog stola. Drugim riječima ako se želi koristiti stol za neku drugu namjenu nosač se makne te se dobiva potpuno ravan stol na koji je moguće smjestiti na primjer obradak za 3D skeniranje. Pošto su provrti za vijke na rotacijskom stolu upušteni, donji nosač projektora dizajniran je s izdancima koja se stavljanju u ta upuštenja dok se gornji dio nosača pričvršćuje s boka te omogućuje dio linearnog pomaka za dodatno fino namještanje po jednoj osi.

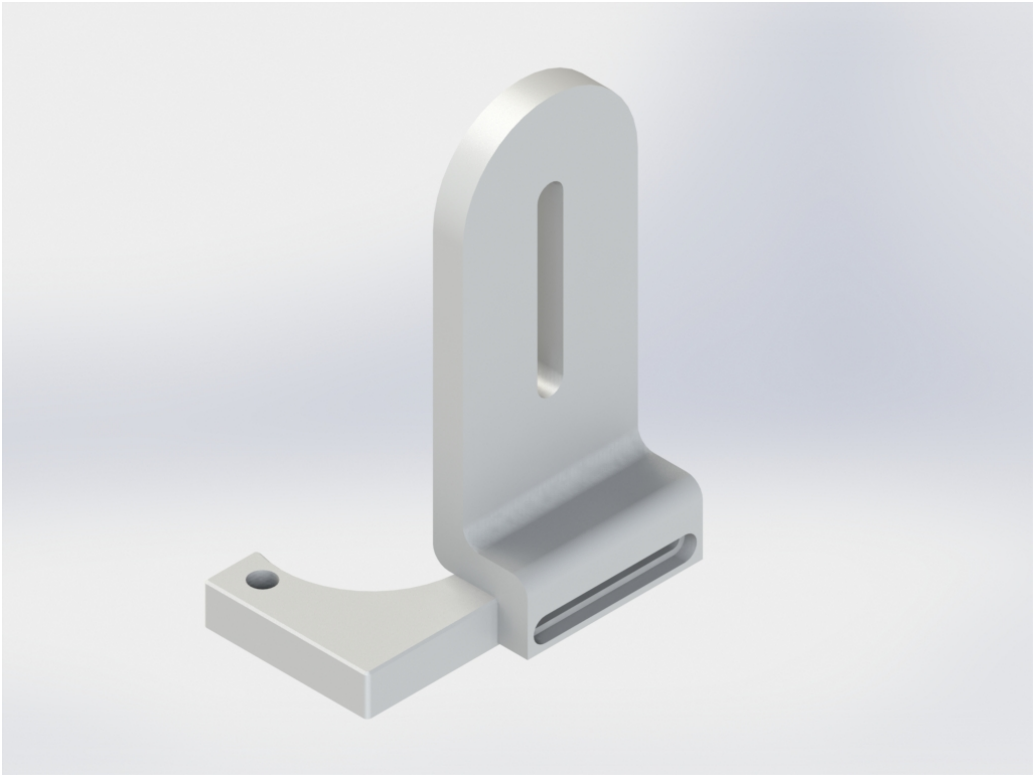


Slika 20. Princip vezanja donjeg nosača

Gornji dio nosača kako je napisano ranije veže se na bok donjeg nosača te služi pričvršćivanju projektora. Dizajniran je tako da omogući pomak projektora u vertikalnoj osi te rotaciju oko mjesta nošenja.

4.7.1. Postupak izrade

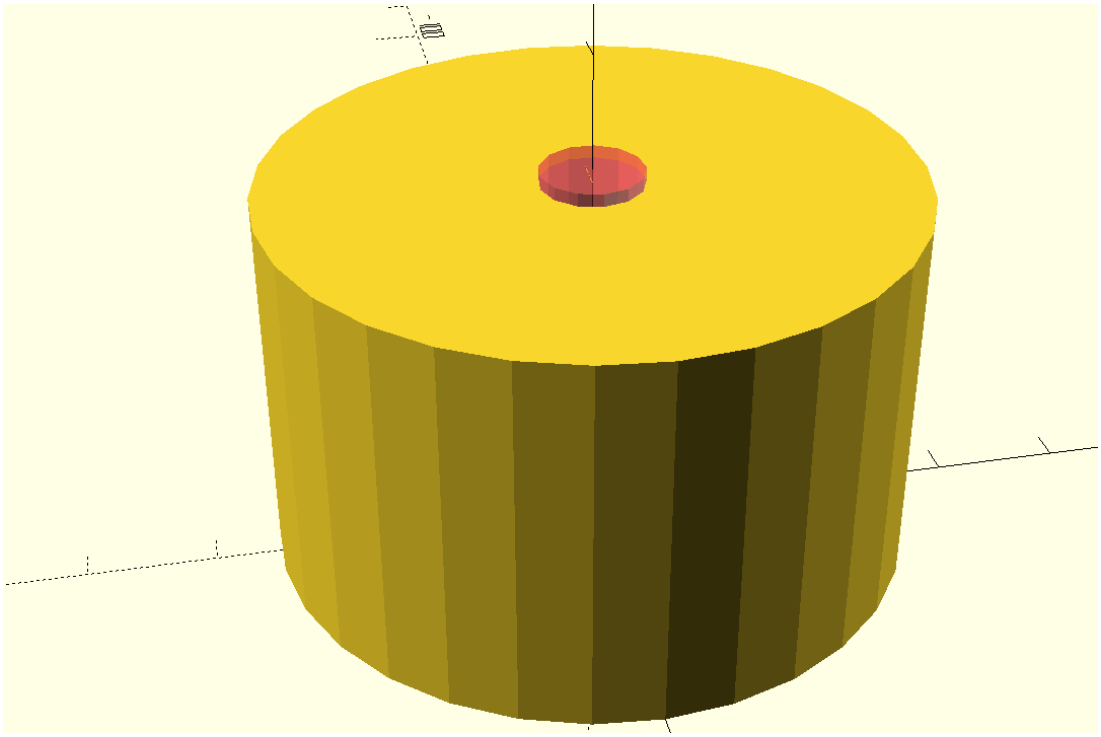
Donji dio nosača izrađen je prema profilu rotacijskog stola kako bi bilo omogućeno dobro nasjedanje. Dodatno je dizajniran s poluprovrtom radijusa jednakog provrtu na stolu, kako svojim sastavljanjem ne bi spriječio prolaz kabela prema projektoru. S druge strane gornji dio nosača je prilagođen dimenzijama projektora i položaja istoga u odnosu na vratni dio i sam stol. Samom izradom pokušao se zadržati pricip jednostavnosti sklapanja dok je jednostavnost izrade sekundarna.



Slika 21. Nosač projektora

5. ISPIS NA 3D PISAČU

Kako je bilo ranije navedeno cijeli sustav je ispisan na Creality Ender 3 3D pisaču i sa svijetlo sivim Fiberology EasyPLA materijalom. Kada se radi postupkom FDM ispisa neke napomene su važne kako bi isto bilo uspješno izvedeno. Bez ulaženja previše duboko u tehnologiju 3D ispisa vrlo teško je objasniti neke odabrane parametre pa stoga se može reći da su vrijednosti spomenutih parametara dobivene iskustveno. Unutarnji provrti suosni s osi ispisne glave uvijek su promjera manjeg od nazivnog, većinom za promjer ispisne brizgaljke, dok su vanjski kružni oblici većinom relativno točnih dimenzija. Dodatno, promjeri ovise o veličini diskretizacijskog elementa korištenih kod spremanja STL datoteke. Primjer ovakve diskretizacije kružnih oblika prikazan je na slici niže. Iste napomene vrijede i za ravne oblike, no utjecaj je manji zbog nepostojanja diskretizacije na tim dijelovima.



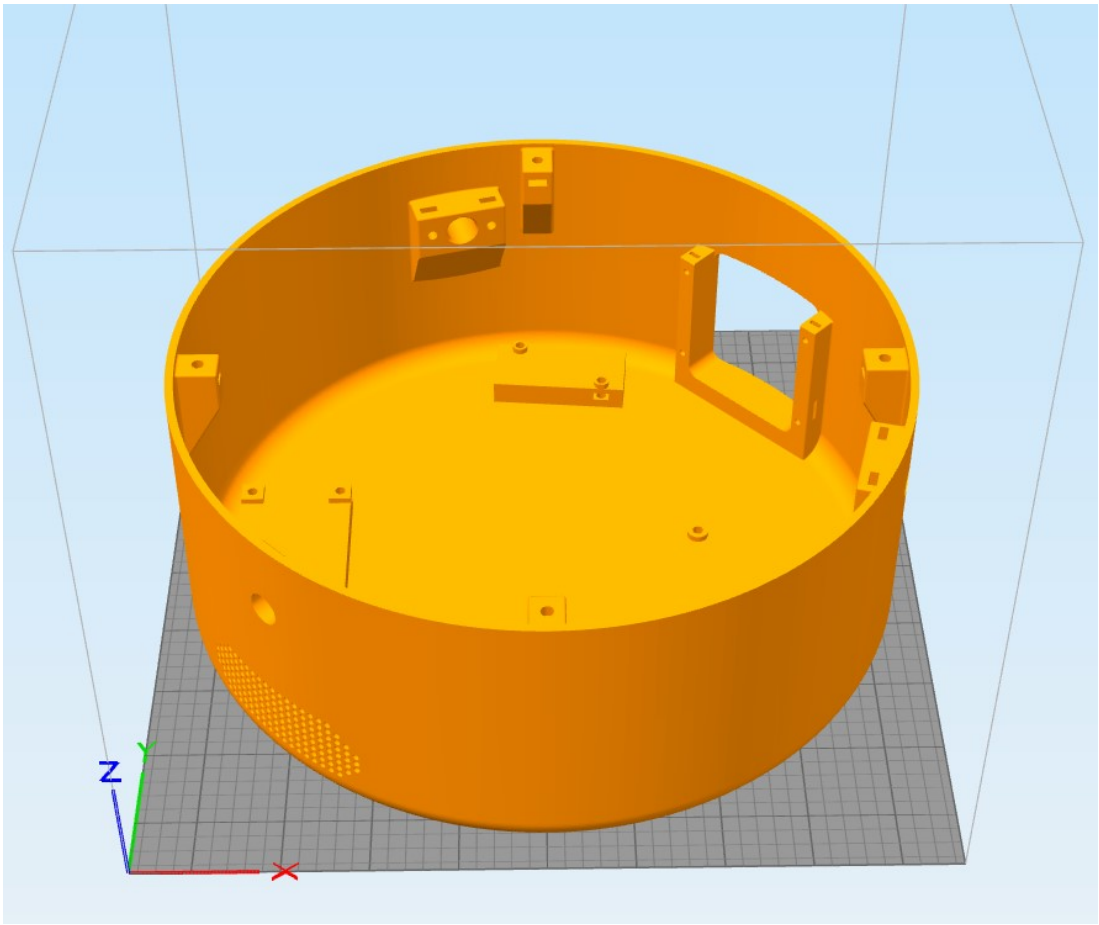
Slika 22. Diskretizacija kružnih oblika

Minimalna debljina stijenke je jednaka promjeru ispisne brizgaljke. Iz ovog zaključka proizlazi i zaključak da bi debljina stijenke trebala u većini sučajeva biti višekratnik promjera brizgaljke. Moguće je koristiti i ostale debljine, no dolazi do preklapanja ili se koristi ispuna, što kod relativno tankih stijenki usporava ispis.

Dijelovi čija površina raste proporcionalno promjeni visine te uz uvjet da je rast relativno velik tj. ako su bočnice pod kutem većim od maksimalnog, za kombinaciju postavljenih pisača te same sposobnosti pisača, iziskuju dodatni potporanj. Potpornji troše materijal te produžuju vrijeme ispisa pa ih je uputno izbjegavati gdje je god to moguće. Uz to, ponekad može doći do vrlo dobrog stapanja potpornja i dijela, što može uzrokovati pucanje dijela kod skidanja istog.

Slika 23. Prikaz potpornja

Krajnji važan parametar je postavljanje virtualnog dijela na virtualni stol unutar programa za kreiranje putanja. Postavljanje se mora provesti imajući u vidu sve prije navedene napomene te dodatnu napomenu, kako je uputno, najveću ravnu površinu staviti na stol.



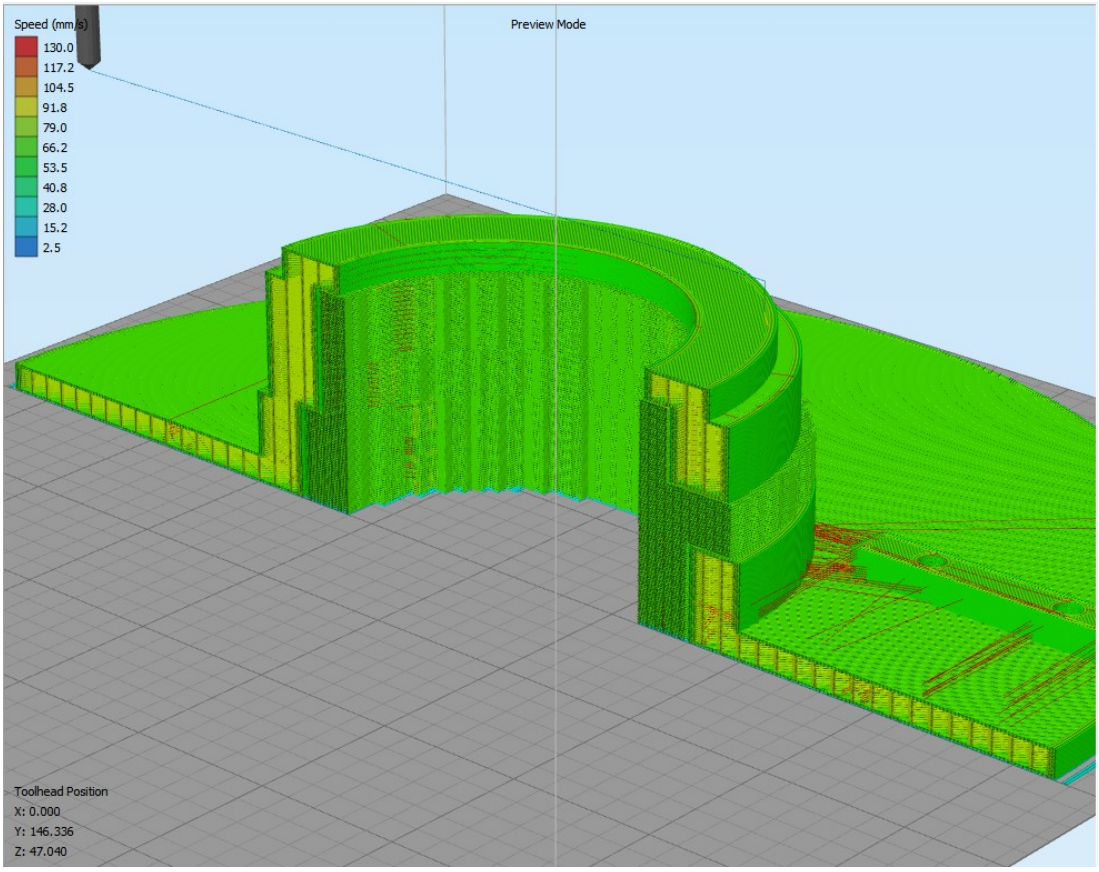
Slika 24. Pravilno postavljanje na ispisni stol

Program koji se koristi za parametriziranje i izradu putanji brizgaljke kolokvijalno se naziva *slicer* što bi na hrvatskom bilo rezač pošto mu je svrha razrezati model u mnogo tankih slojeva debljine odabrane u postavkama. Uputno je debljinu sloja odabrati maksimalno pola promjera brizgaljke.

U sljedećem dijelu bit će opisan način pripreme svakog dijela te će biti prikazane fotografije za vrijeme i nakon ispisa.

5.1. Ispis gornjeg poklopca

Gornji poklopac treba biti prilagođen za ispis prema napomenama opisanim ranije. Prvotno je bilo potrebno povećati promjer uležištenja za ležajeve što je bilo izvršeno već unutar CAD alata. Nakon toga izvozi se dio u STL format te se otvara u *slicer-u* i postavlja se na virtualni stol pisača. U slučaju ovog dijela isti je okrenut naopako, pošto je to jedina potpuno ravna površina, dok je bilo potrebno dodati potporanj za dio ležišnog mjesta. U nastavku je prikaz iz programa izrezanog dijela gdje boje označavaju brzinu pomaka ispisne glave.

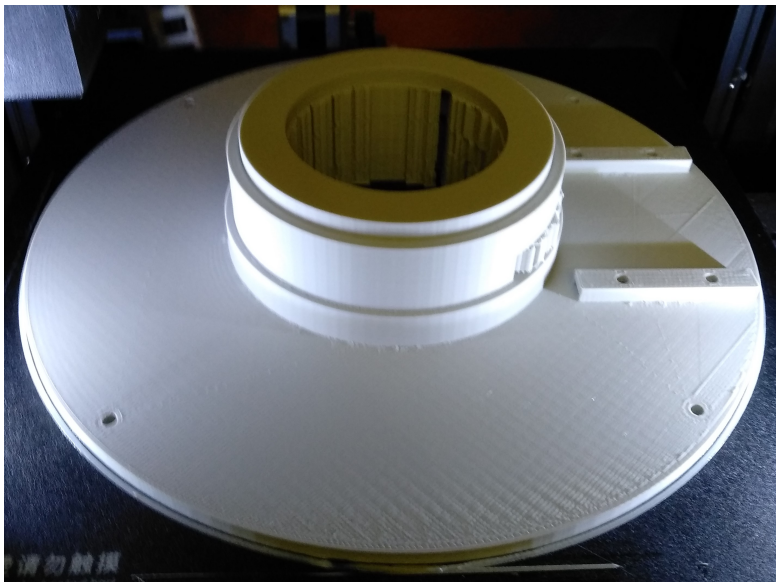


Slika 25. Prikaz izrezanog gornjeg poklopca

Kada su podešeni svi parametri i kada su iskustveno zadovoljene brzine ispisa na određenim površinama, kod se može prebaciti na 3D pisač te se može započeti s ispisom.

Slika 26. Početak ispisa gornjeg poklopca

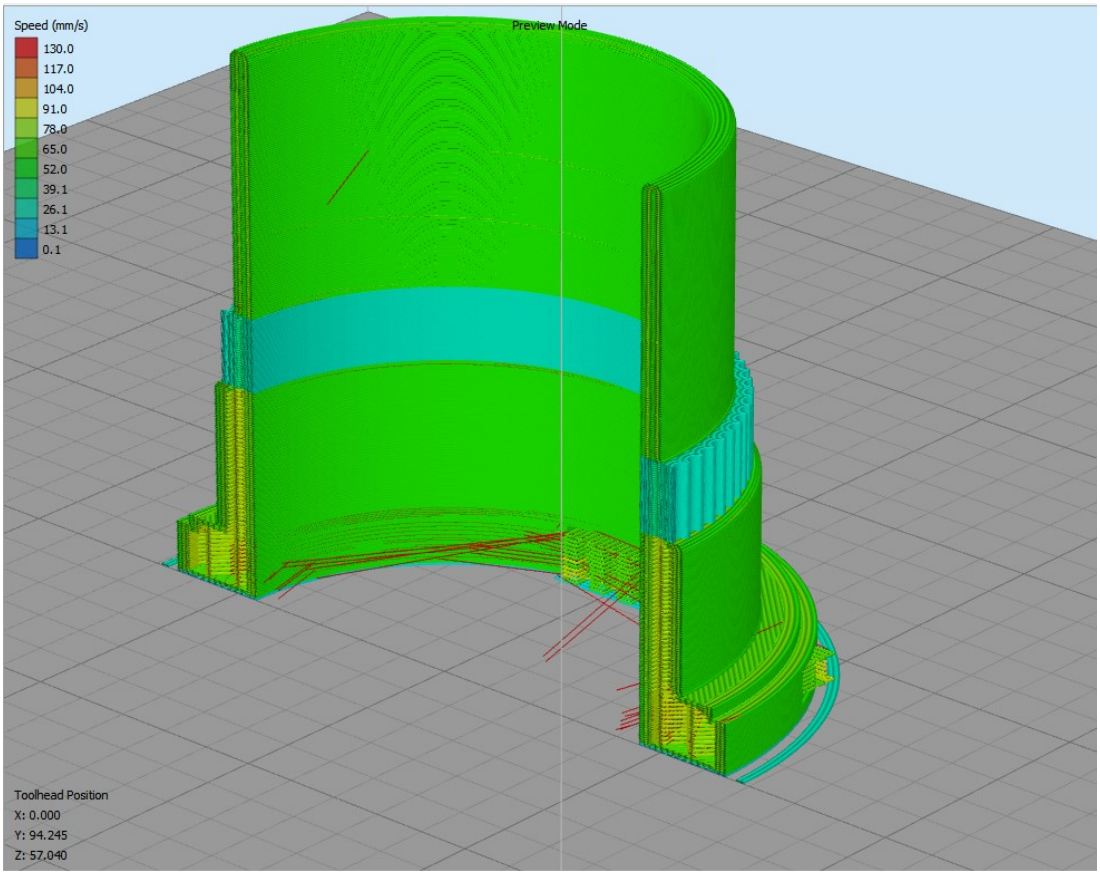
Konačno, nakon određenog vremena, ako nije došlo do prekida ili greške, dobivamo potpuno izrađen model. Ako su korišteni potpornji potrebno ih je skinuti te se takav dio može koristiti u sastavljanju.



Slika 27. Izrađen gornji poklopac

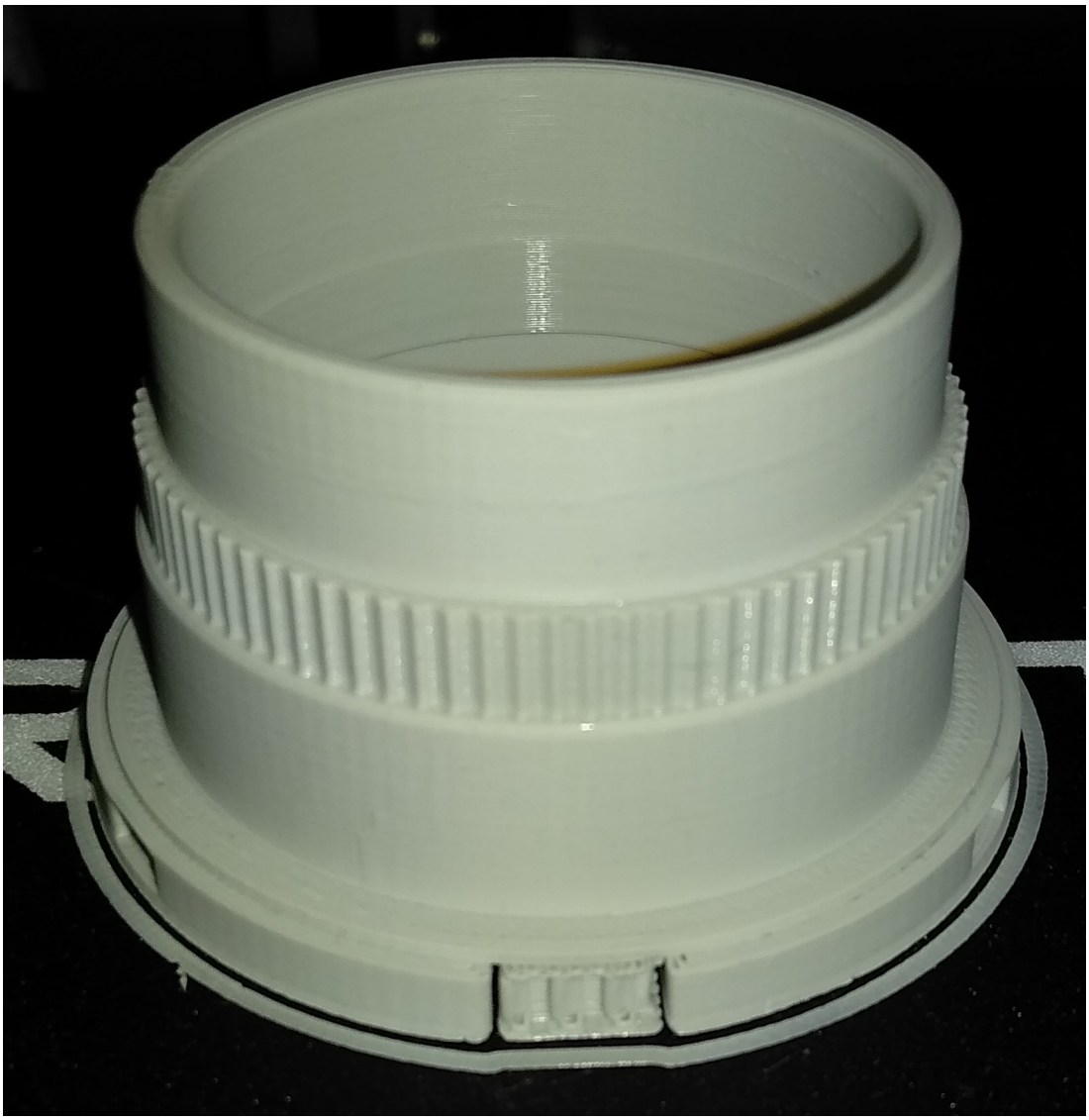
5.2. Ispis velike remenice

Ispis velike remenice zahtijeva posebno podešenje pisača kako bi ispis bio uspješan i dimenzijski točan. Naime, uz podešavanje parametara moguće je podesiti više profila tj. grupa parametara, koji se naknadno mogu dodijeliti određenim sekcijama dijela koji se ispisuje. Shodno tome za ovaj dio korištena su tri različita profila: prvi je bio za ispis do same remenice, drugi ispis remenice dok je treći bio nakon remenice do kraja dijela. Iako su prvi i treći dio ispisivani s istim parametrima, unutar korištenog *slicer-a* nije moguće na više segmenata dodijeliti isti profil. Promjene učinjene na drugom segmentu, segmentu remenice, bile su smanjenje visine sloja i smanjenje brzine ispisa. Razlog smanjenju brzine ispisa je jasan; dobivanje veće kvalitete istog dok je visina sloja smanjena s razlogom veće točnosti i kvalitete ispisa tog kritičnog dijela. Naravno, prije samog ispisa kompletne remenice prvo su ispisani samo prstenovi sa zubima kako bi se moglo provjeriti kvalitetno nasjedanje remena.



Slika 28. Prikaz izrezane velike remenice

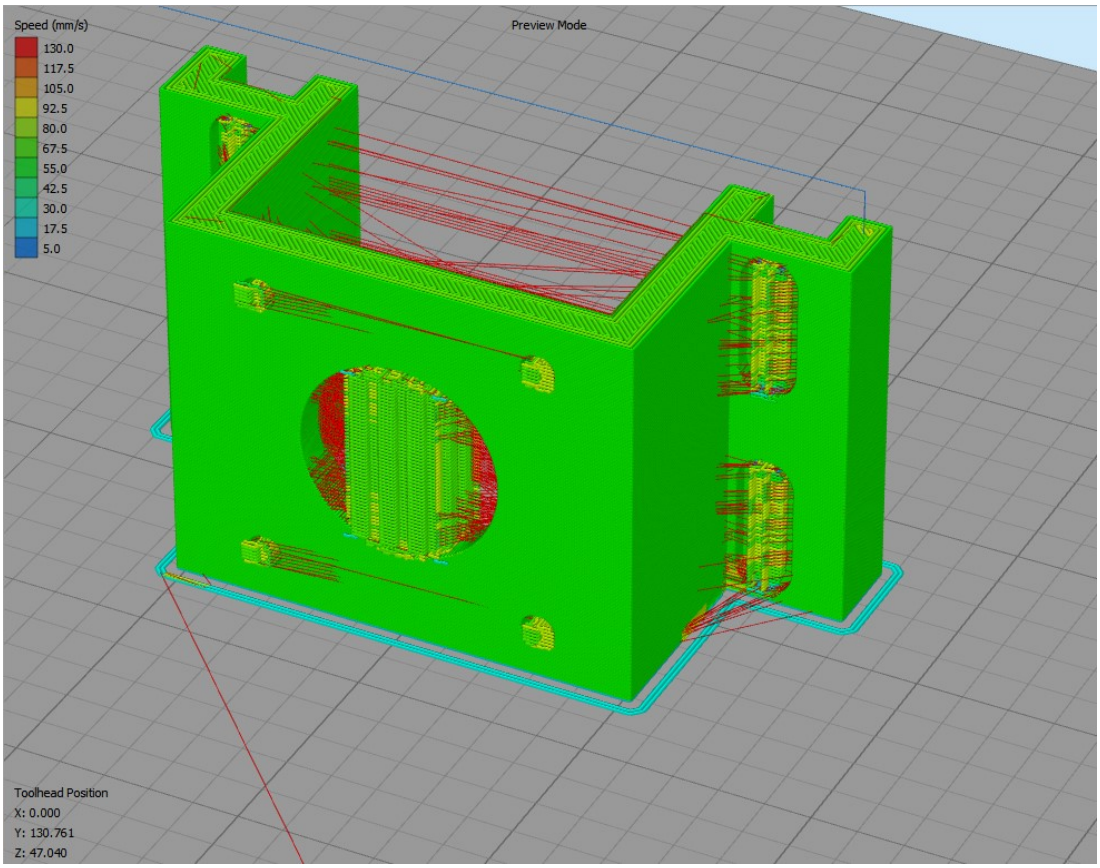
Na gornjoj slici moguće je vidjeti slojeve koji su ispisivani s manjom brzinom pošto je opet pripremni model prikazan bojama u skladu s brzinom ispisa. Na sljedećoj slici se vidi taj isti model, ispisan. Kod njega je također moguće s unutarnje strane primijetiti pojas na kojem je promijenjen parametar brzine i visine sloja.



Slika 29. Izrađena velika remenica

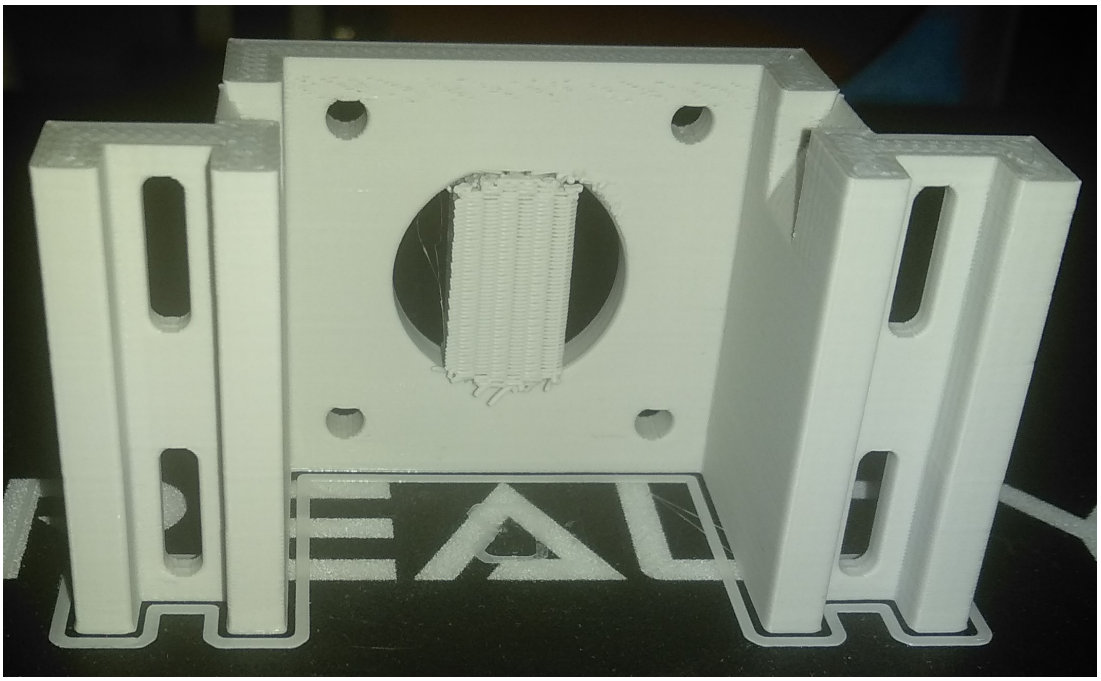
5.3. Ispis nosača motora

Kako je nosač motora jedan od jednostavnijih dijelova gledano sa strane ispisa, nije potreban duži opis. Postavljanje na ispisni stol bilo je uvjetovano potrebom za minimalnom količinom potpornja što je i razlog smještaja prikazanog na slici u nastavku.



Slika 30. Prikaz izrezanog nosača motora

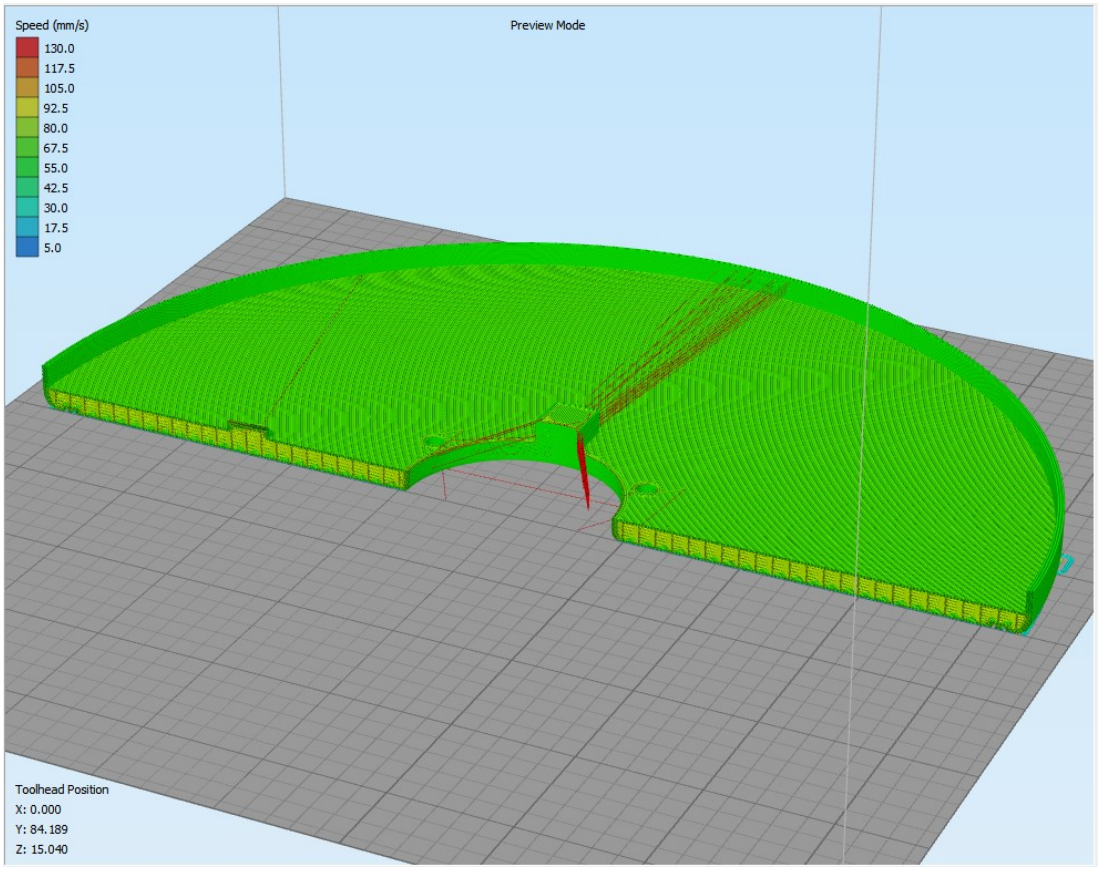
Kako je vidljivo na slici brzina ispisa je relativno uniformna te se limitira samo mehanikom pisača, što je opet neka iskustvena vrijednost. Nakon ispisa dobiva se dio kao na slici iz kojeg je potrebno ukloniti postavljene potpornje te je spreman za ugradnju.



Slika 31. Izrađen nosač motora

5.4. Ispis rotacijskog stola

Kako se radi o relativno velikom, ravnom i tankom dijelu ovo će biti prvo susretanje, u sklopu ovog rada, sa termalnim iskrivljenjem dijela. Kako se radi o polimernom materijalu nehomogene strukture vrlo lako može doći do iskrivljenja kod krivog izbora punjenja ili loše konstrukcijske odluke. Najjednostavnija metoda za izbjegavanje većih iskrivljenja je koristiti simentrično punjenje, te čim bolje simetrični dio. U slučaju ovog dijela iskorišteno je kvadratno punjenje te jednak broj gornjih i donjih punih slojeva. Ostatak parametara je jednak kao i kod ostalih dijelova.



Slika 32. Prikaz izrezanog rotacijskog stola

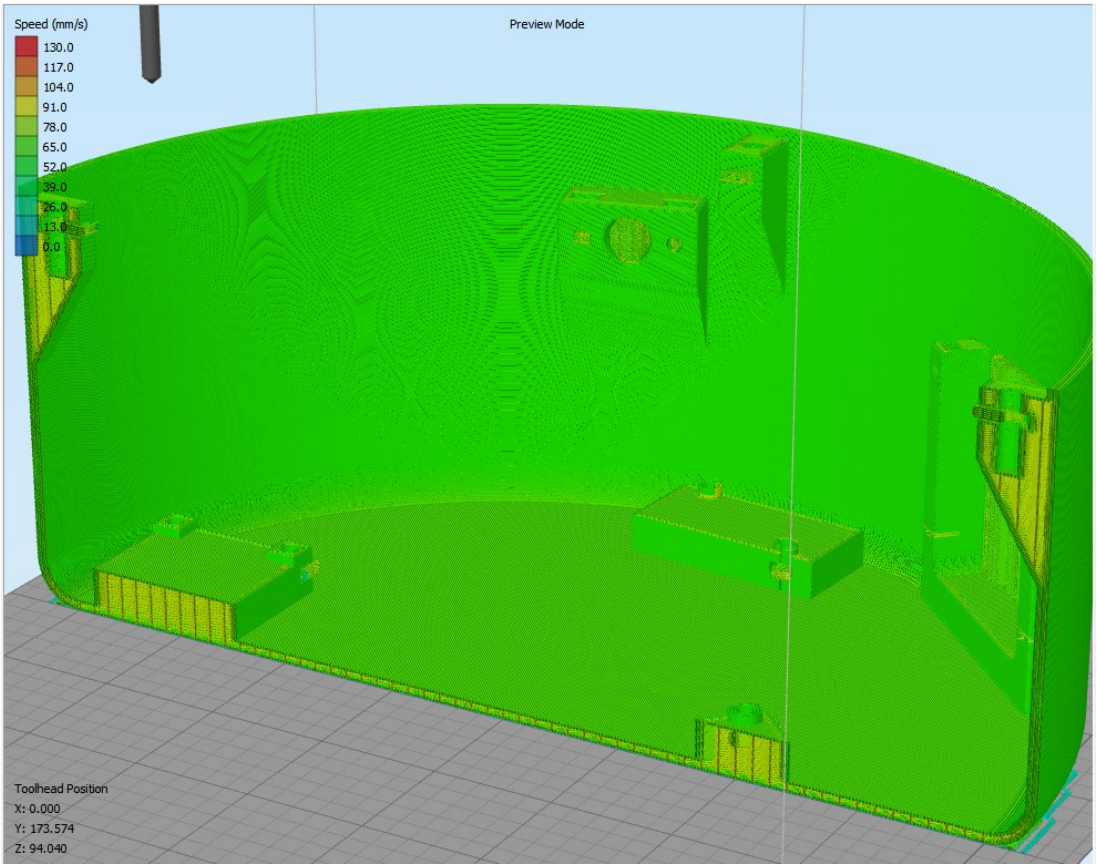
Jedan od parametara koji je promijenjen je brzina ispisa, pošto se radi o relativno kružnom obliku, moguće je bilo koristiti veće brzine ispisa. Razlog tome je smanjenje akceleracije uslijed nepostojanja oštrih kuteva u ispisu. Nakon ispisa dobiva se dio prikazan na slici ispod.

Slika 33. Izrađen rotacijski stol

5.5. Ispis kućišta

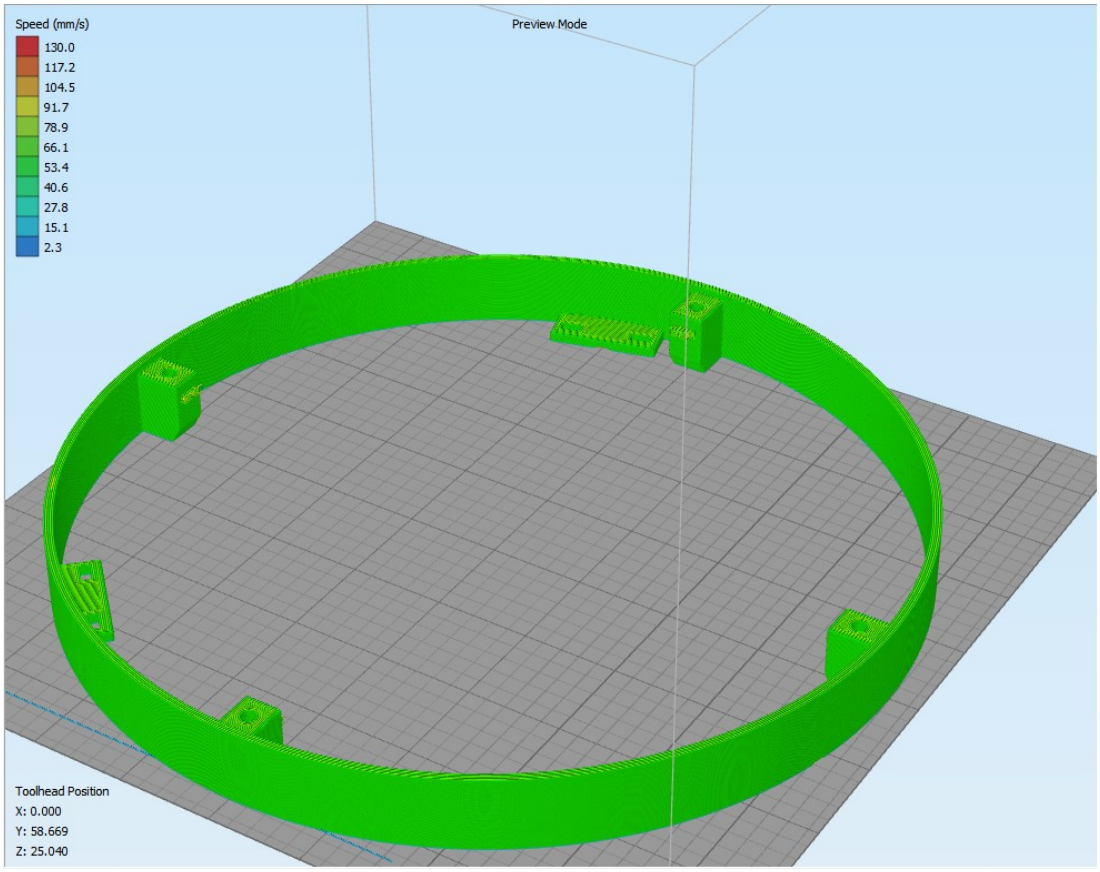
Kako i kod same konstrukcije ovog dijela isto tako i kod ispisa ovo je definitivno najkompliciraniji dio u cijelom radu. Stoga je bilo potrebno vrlo dobro podesiti parametre ispisa kako bi se optimizirala kvaliteta ispisa u odnosu na vrijeme. Dodatno, jedan od većih problema je kao i kod stola temperaturno izvijanje donjih slojeva. Razlika u odnosu na stol je u tome što unutar kućišta postoje nosači za elektroničke komponente što doprinosi povećanju izvijanja.

Na ovaj dio primijenjeni su isti parametri kao i za ostale te su i brzine ostale iste, pošto se kao i kod stola radi o relativno kružnom obliku. Brzine ispisa po dijelu prikazane su kao i kod svih prijašnjih dijelova u vidu toplinske mape.



Slika 34. Prikaz izrezanog kućišta

Kod ispisa ovoliko velikog dijela uvijek postoji šansa za prekidom ispisa zbog greške na pisaču ili u programskom paketu za rezanje komada. Realan primjer je prikazan na ovom dijelu jer se i u stvarnosti to dogodio. Kada je ispis bio na visini od 64mm što je približno 60% zbog artefakta u G-kodu dijela pisač je stao te se mikrokontroler istog resitirao. Nakon toga jedini način za uspješan ispis je pokušati definirati novu nul točku koju će printer shvaćati kao površinu stola te nakon toga generirati novi kod koji obuhvaća samo neispisani dio komada, kao u nastavku.



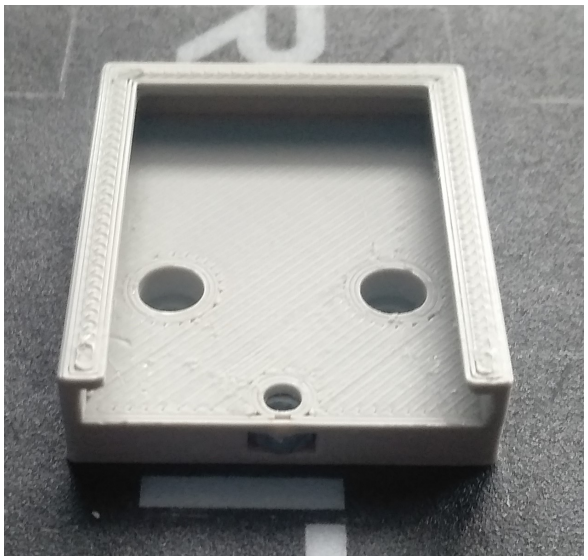
Slika 35. Prikaz gornjeg izrezanog kućišta

Nakon toga bilo je potrebno ponovno pokrenuti pisač, no s vrlo niskom brzinom prvog sloja kako ne bi došlo do udaranja u već prije ispisan dio. Kada je postignuto da se slojevi slažu, pisač je bilo moguće dalje upogoniti standardnom brzinom te je sam ispis bio uspješan.

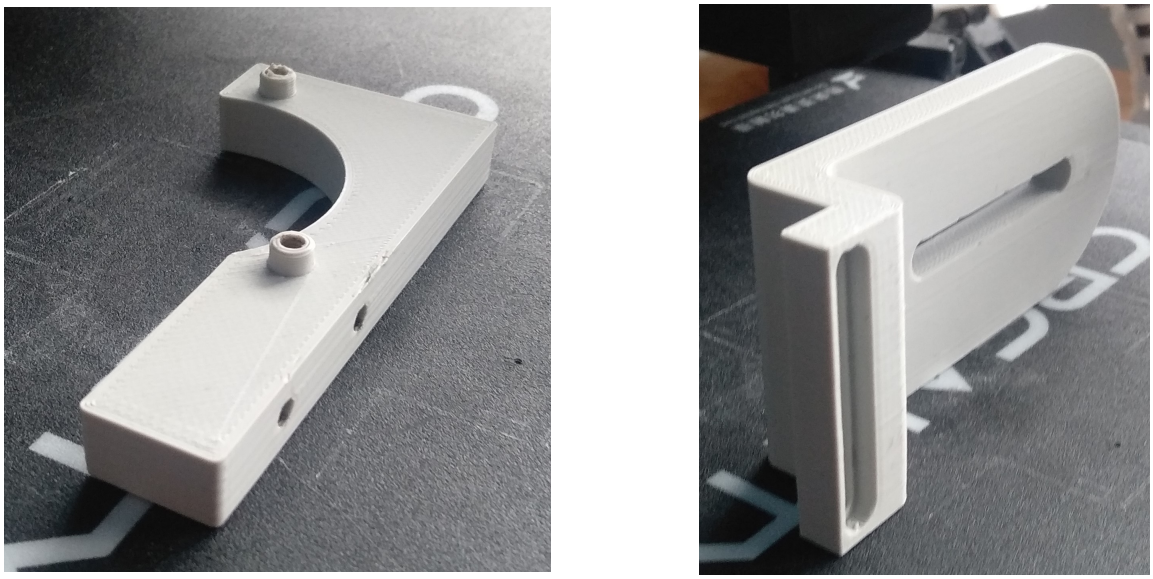
Slika 36. Izrađeno kućište

5.6. Ispis manjih dijelova

Ovim podpoglavljem je obuhvaćen ispis nosača predpojačala te nosača projektora. Pošto se radi o vrlo jednstavnim dijelovima oba tj. sva tri dijela ispisivana su sa zadanim postavkama. Jedino kod nosača predpojačala bilo je potrebno raditi dodatne prilagodbe nakon ispisa te ispisivati više puta isti dio, pošto su same tiskane pločice izrađene na probnim pločicama te dimenzije nisu potpuno točne.



Slika 37. Nosač predpojačala



Slika 38. Nosač projektora

6. IZRADA ELEKTRONIKE

Izrađen mehanički dio nije dovoljan za postići potpunu funkcionalnost sklopa. Kako bi to bilo dostignuto potrebno je izraditi elektroničke sklopove zadužene za akviziciju, obradu signala i u konačnici upravljanje zakretom.

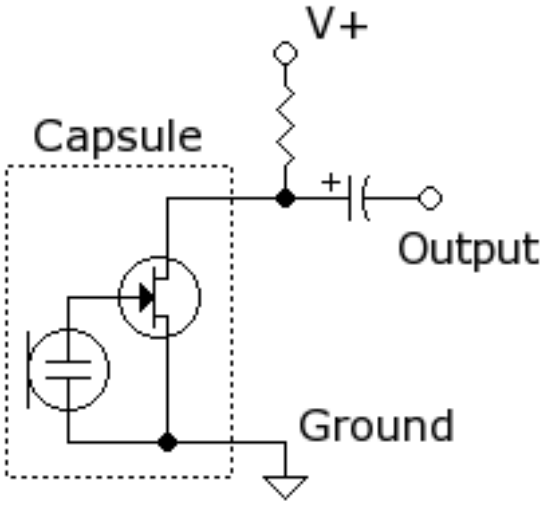
Elektronički sklop se sastoji od tri mikrofona postavljena na kućište u pravilnim razmacima od 120 stupnjeva. Kako bi bilo moguće koristiti signal s mikrofona unutar kućišta su dodane tiskane pločice sa sklopom predpojačala. Konačno, predpojačala su spojena na glavni kontrolni sklop koji se sastoji od mikrokontrolera sa pripadajućom periferijom. Dodatno, na kontrolnoj pločici se nalazi upravljač koračnog motora.

U nastavku će biti prikazani prije navedeni sklopovi s pripadajućim objašnjenjima određenih podsklopova, dok će cijele sheme biti prikazane u prilogu.

6.1. Predpojačalo mikrofona

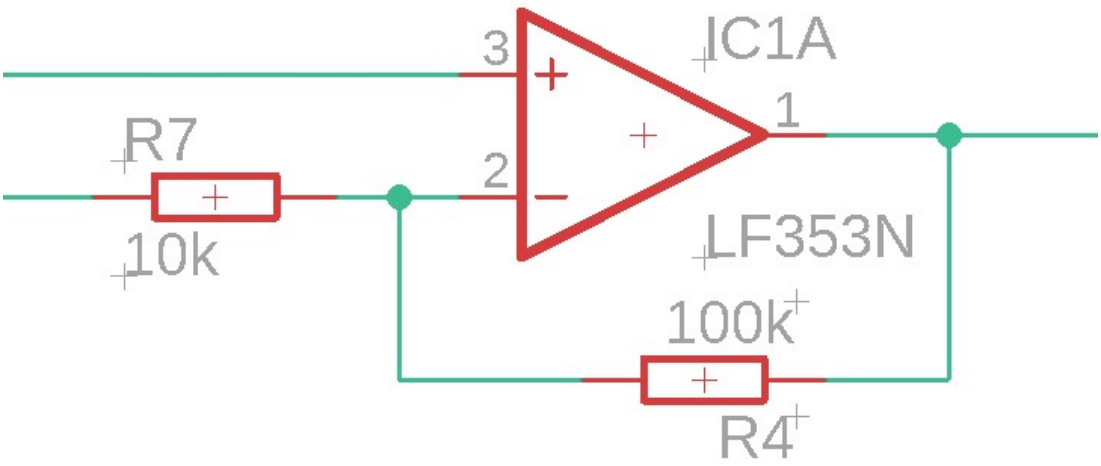
Prvi sklop koji je korišten unutar ovog rada je predpojačalo mikrofona. Kako bi bila jasna primjena ovog skopa potrebno je prvo objasniti rad samog mikrofona korištenog u ovom radu.

Za ovaj rad je korišten electret kondenzatorski mikrofon. Ovaj mikrofon sadrži pomičnu membranu te vrlo malen FET (eng. field-effect transistor). Membrana je od dipolnog materijala te se njezinim pomicanjem stvara vrlo malene varijacije napona koje mijenjaju provodljivost FET-a. Tranzistor je spojen u konfiguraciju zajedničkog izvora (kao što je prikazano na shemi niže) te se na taj način promjenom struje koja prolazi kroz njega dobiva napon na V_m. Iako pojačan, ovaj napon je i dalje vrlo male vrijednosti, svega par milivolta te je potrebno signal dodatno pojačati.



Slika 39. Shema mikrofona

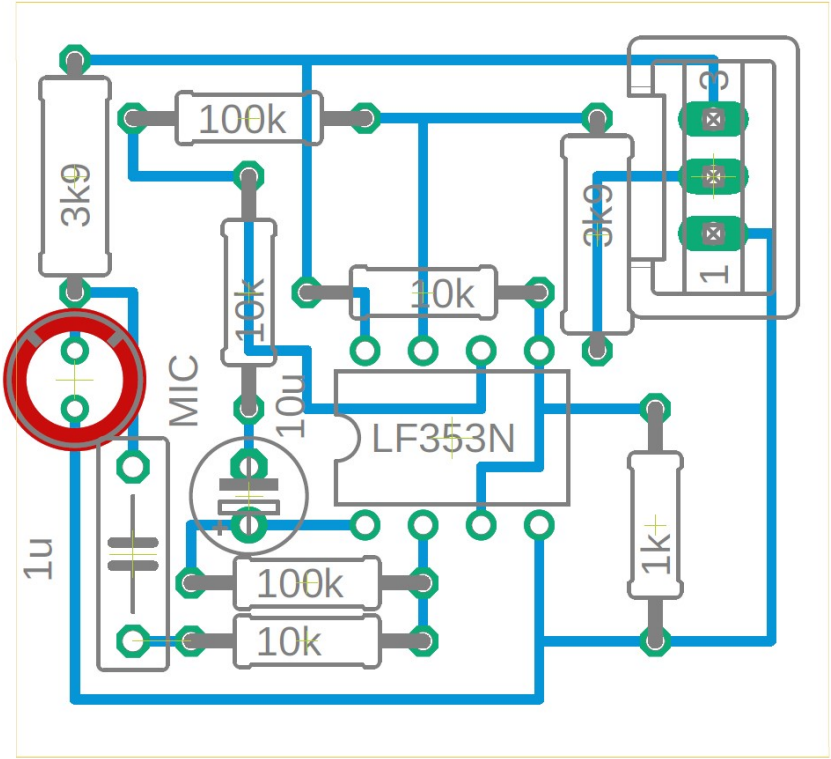
Unutar ovog rada pojačalo je dizajnirano kao dvostupanjsko pojačalo ukupnog pojačanja A=100. Stupnjevi su simetrični tj. s pojačanjima $A_1 = A_2 = 10$. Kako bi se izbjeglo korištenje simetričnog napajanja, ulaz u pojačala je pomaknut u pozitivnom smjeru (tzv. biased ulaz). Ovakvom konfiguracijom dobiven je signal u rasponu od 0-3V, kod normalnog rada, što je u potpunom skladu s ulazom AD konvertora na mikrokontroleru. Primjer spoja sklopa pojačala za jedan stadij prikaza je u nastavku.



Slika 40. Operacijsko pojačalo u spoju pojačala

Na pozitivni ulaz pojačala doveden je napon s naponskog djelila kako bi se postiglo prije spomenuto pomicanje izlaznog signala od nule. Slijedeće je vidljivo da se kao otpornik R_4 koristi vrijednost od $100k\Omega$ dok se za otpornik R_7 koristi vrijednost od $10k\Omega$ što daje pojačanje od 10 puta.

Nakon izrađene sheme potrebno je bilo izraditi tiskanu pločicu. Kako bi se lakše radile promjene na samim pločicama (ako bi to bilo potrebno), kao i zbog jednostavnosti izrade korištene su testne lemne pločice. Prednost ovakvih pločica je jednostavnost izrade zalemljenih sklopova dok je jedan velik nedostatak nemogućnost vrlo detaljnog prostornog rasporeda. Kako bi se omogućila lakša izrada pločica prvotno su iste bile dizajnirane u programskom paketu EAGLE tako što je mreža fiksirana na veličinu jednaku mreži stvarne pločice. Ovakvom metodom olakšano je planiranje rasporeda komponenti prije nego se krene u izradu.

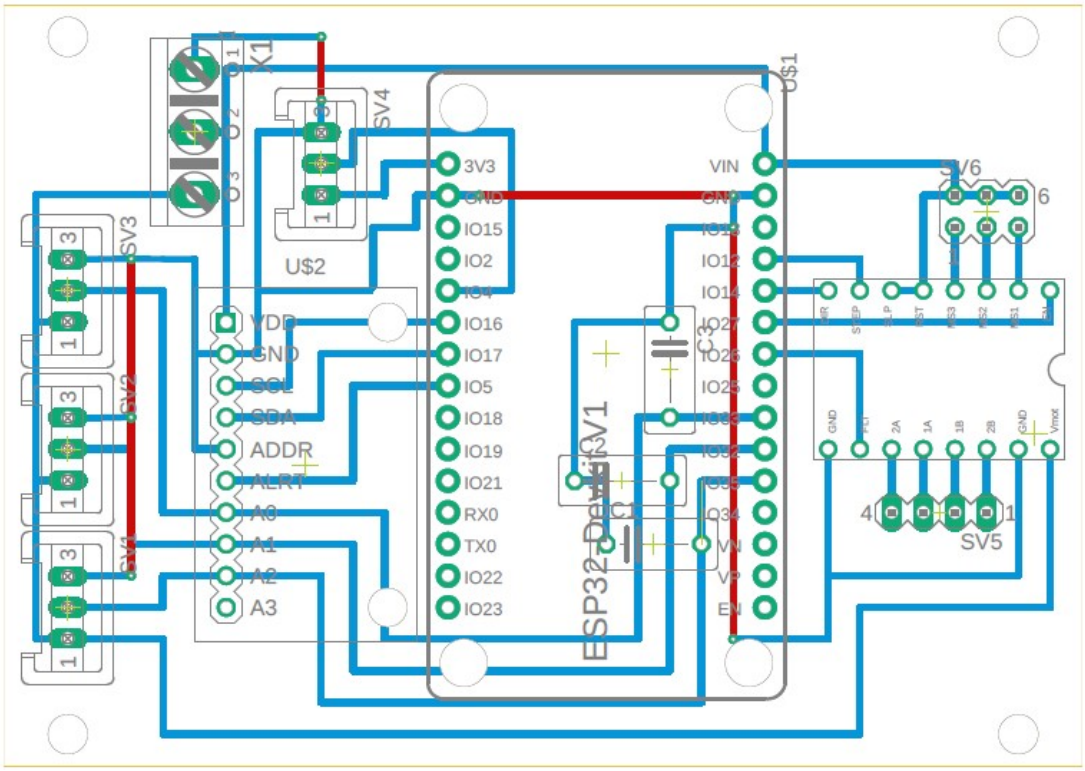


Slika 41. Predpojačalo u EAGLE-u

6.2. Kontrolna pločica

Primarna svrha kontrolne pločice je povezivanje svih komponenti sklopa u jednu cjelinu. Na istoj se nalazi mikrokontroler koji upravlja cijelim sklopom, konektori za spajanje mikrofonskih sklopova i prekidača krajnjeg položaja. S druge strane na nju je spojen kontroler za motor sa pripadajućim konektorom i vrlo malo popratne elektronike. Pod popratnu elektroniku definitivno spadaju kondenzatori velikog kapaciteta (1mF i 2,2mF) kao stabilizatori napajanja, te kondenzatori malog kapaciteta(0.1uF) kao stabilizatori analognih ulaza. Kako su jedine komponente na ovoj pločici kondenzatori moglo bi se reći da je to razvodna pločica s kondicioniranjem signala i napajanja.

Metoda izrade ove pločice ista je kao i izrada predpojačala s iznimkom da se ovdje radi o relativno većoj pločici s relativno više konekcija pa je i složenost izrade nešto veća. U nastavku je prikazana pločica u programskom paketu EAGLE kao i izrađena pločica.

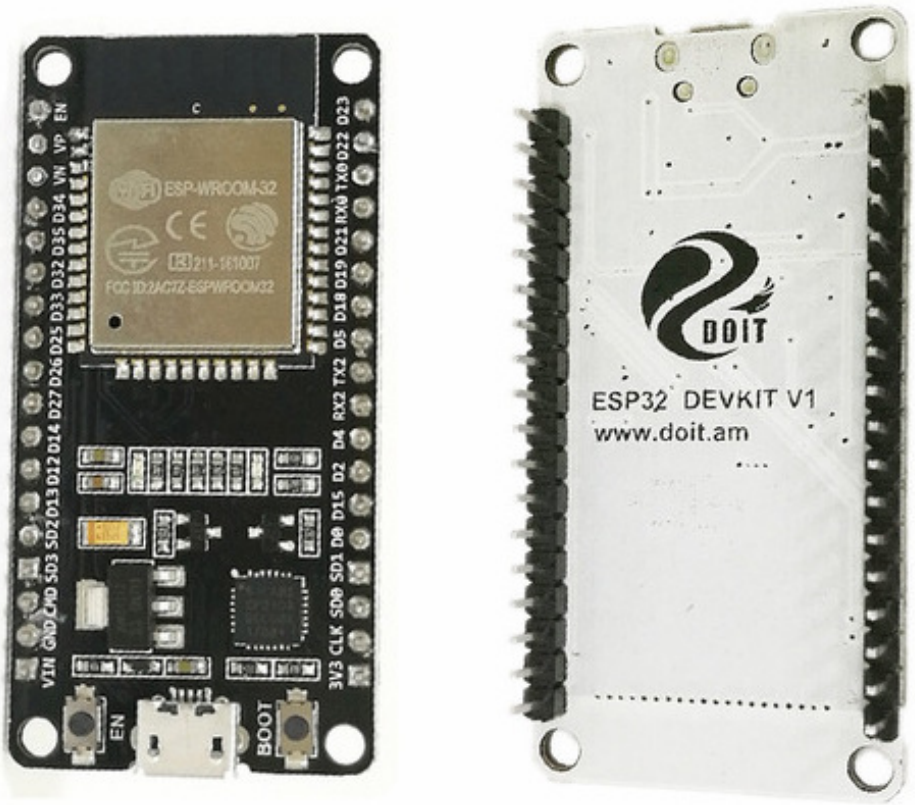


Slika 42. Kontrolna pločica u EAGLE-u

6.2.1. Mikrokontroler

Kao najvažniji dio cijelog sklopa, ujedino i ove pločice je mikrokontroler. Pošto njegova funkcija postaje jasna tek kada unutar njega postoji program koji definira njegovu funkciju obajšnjenja istog, bit će dana u slijedećem poglavlju, dok će u nastavku biti definirane za ovaj rad važne karakteristike.

Korišten mikrokotroler je ESP32 kineskog proizvođača Espressif. Pošto je sam čip izveden u SMD tehnologiji bez mnogo prateće elektronike mnogo je jednostavnije pribaviti gotovu pločicu koja posjeduje čip za komunikaciju, memoriju i elektroniku za napajanje. U slučaju ovog rada to je bila razvojna pločica ESP32 DOIT DevBoard V1.



Slika 43. ESP32 DOIT DevKit V1

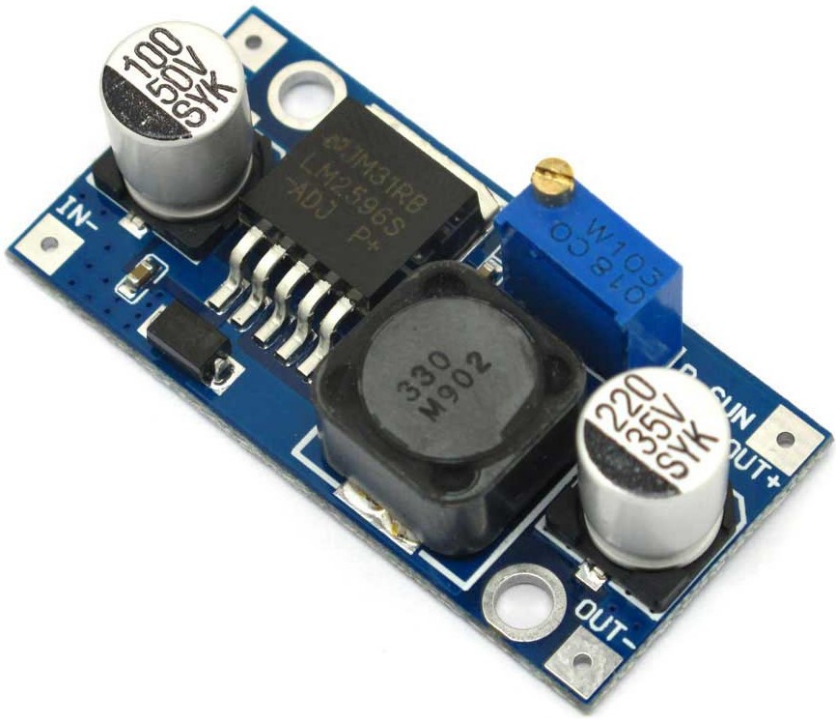
Kako je unutar ovog rada jedan od najvećih zahtjeva visoka brzina uzorkovanja signala ovaj mikrokontroler se pokazao kao odličan odabir. Razlog tome je njegova velika brzina rada (240MHz) te posjedovanje dvije jezgre. Dodatno ovaj mikrokotroler posjeduje analogno digitalni konverter tipa SAR koji je izrazito brz. Kombinirano to s vrlo pristupačnom cijenom te lakoćom programiranja istog potvrđuje se prije navedena teza.

Glavne karakteristike mikrokontrolera [13]:

- Frekvencija: do 240MHz
- Broj jezgri: 2
- WiFi modul
- Bluetooth 4.2 modul
- AD konverter: 2 x SAR ADC 12bit 8 kanala
- Brzina uzorkovanja: do 1Msps razdijeljeno na broj kanala
- I^2S modul
- Serijska komunikacija
- I^2C komunikacija
- SPI komunikacija
- "*Touch*" senzor
- Temperaturni senzor
- Hall senzor
- 8-bitni D/A pretvornik
- Brojač realnog vremena (RTC)
- Mogućnost programiranja u Arduinu

6.3. Popratna elektronika

Pozornom čitatelju je vrlo lako primijetiti da se unutar prije navedenih sklopova nigdje ne spominje izvor napajanja iako je izrazito potreban, pogotovo jer sam sklop radi na nižem naponu (5V) dok je napajanje motora na višem (12-24V). Kako bi se omogućilo napajanje iz samo jednog izvora potrebna je dodatna elektronika za konverziju napona. U slučaju ovog rada to je izvedeno s gotovim sklopom spuštača napona (eng. step-down) čija je glavna prednost visoka efikasnost i mogućnost korištenja bilo kojeg ulaznog napona u rasponu $V_u + 1$ do 32V. Postoje i sklopovi s većim rasponom, no ovakav je potpuno dovoljan za ovu primjenu.



Slika 44. Sklop spuštač napona

Slijedeći dio gotove elektronike je pojačalo zvuka s pripadajućim zvučnikom. Iako nije direktno vezan uz funkcionalnost ovog rada svrha toga je mogućnost reproduciranja zvuka iz vanjskog uređaja. Sklop pojačala je bilo najlakše izraditi iz gotovih dijelova koji dolaze kao kitovi za sastavljanje nekih renomiranih proizvođača dok je zvučnik standardni dio. Maksimalna snaga zvučnika i pripadajućeg pojačala je 3W RMS što je potpuno dovoljno za većinu primjena.

Slika 45. Sklop pojačala za zvučnik

7. IZRADA KODA

Ranije je definirano da je kod izrađen u razvojnom okruženju Arduino, no važno je napomenuti da je ovo okruženje izabrano primarno iz razloga jednostavnosti programiranja samog čipa. Sam kod je u mnogo čemu različit od standardnog Arduino koda te ima mnoge dodatke koji su specifični za odabrani mikrokontroler, primarno u dijelu akvizicije zvuka.

Detaljno objašnjenje koda bit će dano u nastavku, dok će zbog jednostavnosti raščlana po podpoglavljima biti jednaka raščlani samog koda. Važna napomena je kako su neke funkcije u kodu sastavljene od više modula radi kompaktnosti koda.

7.1. Inicijalizacija i poziv biblioteka

Prvi korak gotovo svakog programa je inicijalizacija ili poziv biblioteka ili oboje ovisno o kompleksnosti samog programa. Pošto ovaj program nije izuzetak prvo će biti prokomentirane korištene biblioteke te inicijalizirane globalne varijable. Dodatno će biti dana objašnjenja za određene korake načinjene u inicijalizaciji.

```
11 #include <Arduino.h>
12 #include <driver/i2s.h>
13 #include <driver/adc.h>
14 #include "SpeedyStepper.h"
15 #include <SimpleKalmanFilter.h>
```

Prvi korak je svakako poziv biblioteka za osnovnu funkcionalnost čemu služi prvih pet biblioteka redom:

- Arduino.h biblioteka omogućuje interakciju Arduino specifičnog koda s mikrokontrolerom koji nije direktno podržan od Arduino proizvođača
- i2s.h biblioteka specifična za ovaj mikrokontroler te omogućuje iterakciju s I2S modulom
- adc.h također specifična za ovaj mikrokontroler te omogućuje interakciju s ADC periferijom

- SpeedyStepper.h biblioteka za jednostavnu kontrolu koračnog motora
- SimpleKalmanFilter.h implementacija jednostavnog kalmanovog filtra za brzu filtraciju akviziranog signala

U nastavku se nalaze dvije biblioteke koje olakšavaju definiranje registara za postavljanje I2S i ADC modula tako što adrese i ulazne podatke zamijenjuju s unaprijed definiranim imenima. Primjer korištenja ovih biblioteka će biti dan u dijelu inicijalizacije prije navedenih modula.

```
18 extern "C" {
19   #include "soc/syscon_reg.h"
20   #include "soc/syscon_struct.h"
21 }
```

Nakon poziva svih biblioteka potrebno je definirati vrijednost frekvencije uzorkovanja koja se kasnije koristi kod postavljanja AD konvertera te vrijednost vremena između dva uzorka. Vrijeme između dva uzorka nije jednostavno inverz frekvencije uzorkovanja jer konverter posjeduje samo jedan modul dok se kanali multipleksiraju što u ovo slučaju daje tri puta manje vrijeme uzorkovanja po kanalu.

```
23 #define Fs   450000              // Hz
24 #define T    1000.0/(Fs/3.0);    // ms
```

Slijedeći dio koji je potreban je stvaranje objekata za kalmanov filtar, potrebna su tri pošto se akviziraju tri kanala, te dodatno stvaranje objekta za upravljanje koračnim motorom. Kod stvaranja kalmanovog filtra potrebno je definirati tri argumenta: nepouzdanost mjerenja, nepouzdanost estimacije i procesnu varijancu. Pošto u slučaju ovog sustava nije naglasak na preciznosti i točnosti mjerenja nego samo na vremenskim odmacima između dva mjerenja parametri kalmanovog filtra su određeni eksperimentalno za dobro smanjenje šuma uz zadržavanje relativno dobre karakteristike signala.

```
32 SimpleKalmanFilter filter_adc0(3, 3, 0.015);
33 SimpleKalmanFilter filter_adc1(3, 3, 0.015);
34 SimpleKalmanFilter filter_adc2(3, 3, 0.015);
```

Konačno, potrebno je definirati neke globalne varijable. Svrha ovih varijabli je da se mogu mijenjati iz svih funkcija te bez obzira koji dio programa se izvršava vrijednosti su postojane. Iako nije najbolja metoda za prijenos podataka između funkcija vrlo je jednostavan za implementaciju te kao takav i vrlo korišten. U ovom dijelu definirani su spremnici za očitanje vrijednosti AD konvertera te definirane vrijednosti korištene za upravljanje koračnim motorom.

```
40 float adc0f[3000],adc1f[3000],adc2f[3000];
41 uint16_t i2s_read_buff[9000];
42 size_t bytes_read;
43
44 float belt_ratio=4.25;
45 int motor_steps=400;
46 int u_step=8;
47 int full_rotate=int(u_step*motor_steps*belt_ratio);
48 bool homeState;
49 int homePin=4;
50 int hommingSpeed=600;
```

7.2. Inicijalizacija i konfiguracija I2S i ADC modula

Ranije su bile navedene sve relevantne specifikacije ovih modula pa će u nastavku biti objašnjenji načini konfiguriranja modula kako bi se postigle performanse potrebne za ovaj sustav. Prije nego se krene objašnjavati funkcija potrebno je definirati nekoliko pravila koja možda i nisu programski logična, no definirana su samom arhitekturom čipa. Prvo iako je vrlo logično prvo konfigurirati sve postavke te nakon toga pokretati module unutar ovog sustava to nije tako. Poredak radnji je čvrsto definiran te ne slijedi prije navedeni zaključak. Slijedeće kako bi bila konfiguracija uspješna potrebno je popuniti neke registre, jedan od njih je i registar za definiciju kanala koji se skeniraju. Ovaj registar dozvoljava upis četiri kanala, no ako ih je potrebno manje također treba definirati sva četiri te u drugom registru definirati koliko ih se zaista koristi. Metode definiranja su prikazane u nastavku.

```
54 void configure_i2s()
55 {
56   i2s_config_t i2s_config =
57   {
58     // Mod je ADC s I2S modulom
59     .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX |
       ↪  I2S_MODE_ADC_BUILT_IN),
60     // Brzina uzorkovanja
61     .sample_rate = Fs,
62     // 16bit I2S
63     .bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT,
64     // Odabran samo lijevi kanal(izbjegavanje greške krivog uzorkovanja)
65     .channel_format = I2S_CHANNEL_FMT_ALL_LEFT,
66     // I2S format MSB
67     .communication_format = (i2s_comm_format_t)(I2S_COMM_FORMAT_I2S |
       ↪  I2S_COMM_FORMAT_I2S_MSB),
68     // Alokacija broja interrupt-a
```

```
69      .intr_alloc_flags = 1,
70      // Broj DMA(brza memorija) spremnika
71      .dma_buf_count = 9,
72      // Broj uzoraka po spremniku
73      .dma_buf_len = 1000,
74      // Bez Audio PLL-a
75      .use_apll = false,
76    };
```

Iz koda je vidljivo da se prvo popunjava struktura s konfiguracijskim parametrima I2S modula, no lako je vidljivo da se unutar te konfiguracije može mijenjati relativno malen broj postavki. Razlog ovakvom načinu je slabo korištenje modula od krajnjih korisnika, što uvjetuje nedovršenost ovakvih biblioteka.

Nakon definiranja postavki I2S modula potrebno je konfigurirati ADC modul za čitanje iz jednog kanala točno poretkom definiranim u predloženom kodu, prvo broj kanala i raspon mjerenja, naknadno rezoluciju mjerenja. Time je pokrenut ADC modul te se pristupa pokretanju I2S modula te spajanja istog s prije pokrenutim ADC modulom. Kada je izvršeno spajanje, pristupa se upisivanju stvarne konfiguracije u registre koji potpuno uklone sve unaprijed definiranje postavke, osim osnovne konfiguracije, te ponovno rekonfiguriraju module.

```
78    //Postavljanje jednog kanala za inicijalizaciju
79    //Mjerni raspon 0-3.6V
80    adc1_config_channel_atten(ADC1_CHANNEL_7, ADC_ATTEN_11db);
81    //Mjerna rezolucija 12bit (0-4095)
82    adc1_config_width(ADC_WIDTH_12Bit);
```

```
84  //Instalacija I2S driver-a
85  i2s_driver_install(I2S_NUM_0, &i2s_config, 0, NULL);
86  //Spajanje I2S modula s ADC modulom te pokretanje istog
87  i2s_set_adc_mode(ADC_UNIT_1, ADC1_CHANNEL_7);
88  i2s_adc_enable(I2S_NUM_0);
89
90  //Definiranje registara za upravljanje ADC čitanjem
91  //Treći argument u prvoj funkciji je (N-1) kanala za čitanje
92  SET_PERI_REG_BITS(SYSCON_SARADC_CTRL_REG, SYSCON_SARADC_SAR1_PATT_LEN,
    ↪  2, SYSCON_SARADC_SAR1_PATT_LEN_S);
```

Najvažnija naredba u ovom dijelu je popunjavanje registra ***SYSCON_SARADC_SAR1_PATT_TAB1_REG*** čija je svrha definirati koji kanali se čitaju. U ovaj registar se upisuje 32 bitni broj koji je moguće rastaviti na četiri dijela, koji odgovaraju željenom kanalu i postavkama na istome.

```
102 WRITE_PERI_REG(SYSCON_SARADC_SAR1_PATT_TAB1_REG, 0x5F4F7F0F);
```

Pretpostavimo opći zapis broja *0xXYXYXYXY* možemo reći:

- X - broj korištenog kanala u heksadecimalnom zapisu (0-F)
- Y - definicija parametara za pripadajući kanal preko bitnog raspisa broja *0bAABB*

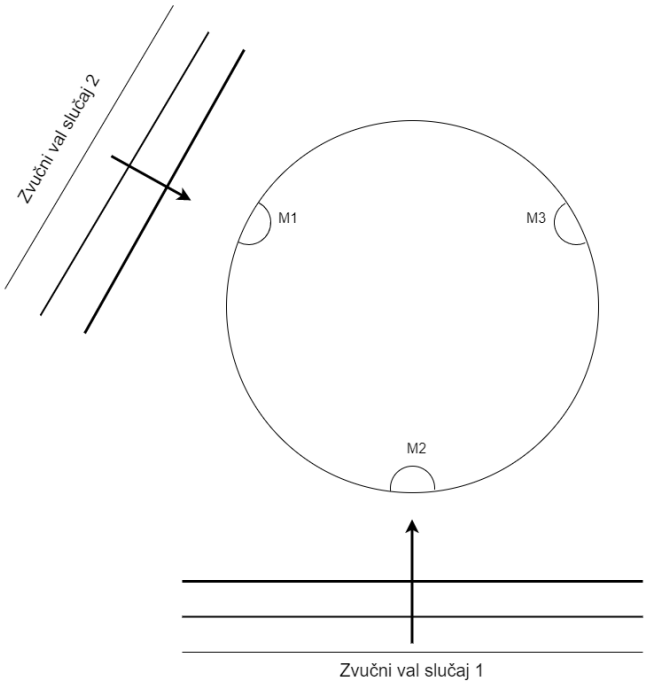
  AA - rezolucija kanala 9bit do 12bit

  BB - širina kanala 0-[1,1.34,2.3.6]V

Konačno se definira da spremljene podatke treba invertirati, pošto ADC podatke sprema kao inverz. Važna napomena prije slijedećeg modula je da nakon incijalizacije ovih modula treba staviti stanku u izvođenju minimalno 2 sekunde.

7.3. Izračun kuta zakreta

Važna napomena prije diskusije ovog dijela je kako ovaj proračun radi na principu aproksimacije između šest diskretnih elemenata. Prvo algoritam odredi iz kojeg smjera dolazi zvuk na temelju predznaka vremenskih pomaka. Prema principu prikazanom na slici u nastavku lako je zaključiti da ovisi o smjeru dolaska zvuka val će prvo dostići jedan od mikrofona. Shodno tome predznak udaljenosti između dva mikrofona definira koji je od njih bliže izvoru zvuka.



Slika 46. Metodologija izračuna kuta

Ako se razmotri slučaj 1 na gornjoj slici te se za primjer uzme kašnjenje između kanala M1 i M2 vidljivo je da će kašnjenje biti negativno jer zvuk prvo udara u mikrofon 2 tek onda u 1, dok kod slučaja 2 je situacija suprotna. Shodno tome kod slučaja 1 samo provjerom predznaka vremenskog pomaka možemo zaključiti da je izvor zvuka u smjeru Y osi, dok je kod slučaja 2 smjer zvuka 30 stupnjeva odmaknut od smjera X osi.

Prema izvedenom zaključku za dva slučaja dolazi se do svih šest mogućih slučaja ako postoje 3 mikrofona. Dodatna uzeta pretpostavka za preciznije određivanje kuta da je omjer najmanjeg vremenskog pomaka i ukupnog vremenskog pomaka jednak omjeru ukupnog diskretnog kuta (u ovom slučaju 60 stupnjeva) i kuta odmaka od početka zadanog segmenta. Pošto se sam algoritam ponavlja za svaki diskretni dio kružnice u nastavku je prikazan samo jedan dio.

```
112 float calc_angle(float d12,float d13, float d23)
113 {
114   float fdist,ratio,angle_rel,angle;
115   if((d12<0) and (d13<0) and (d23<0))
116   {
117     fdist=abs(d12)+abs(d23);
118     ratio=abs(d23)/fdist;
119     angle_rel=60*ratio;
120     angle=330+angle_rel;
121     if(angle>=360){angle=angle-360;}
122   }
```

Pošto jedan diskretni segment prelazi kroz nula stupnjeva potrebno je nakon izračuna relativnog kuta na tom segmentu, ako je vrijednost veća od punog kruga, oduzeti 360 stupnjeva. Primjer toga je linija 121 u kodu iznad.

7.4. Proračun vremenskog odmaka

Kako svaki program ima neki algoritam koji čini njegovu jezgru i ključan je za funkcionalnost istog tako ima i ovaj. Proračun vremenskog odmaka je ključan za točno računanje kuta izvora zvuka te se može podijeliti na tri podmodula. Prvi od njih je akvizicija signala te njezino parsiranje u tri zasebna kanala, drugi dio je obrada signala te izračun vremenskih odmaka, dok je treći izračun kuta pomoću ranije opisanog algoritma te pomak stola u smjeru izvora zvuka. Kada se govori o akviziciji i obradi podataka unutar ove funkcije važno je napomenuti da ovo nije kontinuirano aktivna funkcija nego se pokreće tek kad su zadovoljeni uvjeti u glavnoj petlji. Na početku se akvizira 1500 podataka po kanalu s kojima se radi proračun.

7.4.1. Akvizicija i parsiranje podataka

Prvi modul vrši akviziciju podataka u spremnik te nakon toga unutar *FOR* petlje raspodjeljuje podatke u tri zasebna spremnika. Vrlo je važno zamijetiti da petlja prolazi samo kroz parne brojeve tj. uzima samo svaki drugi podatak na parsiranje. Razlog tome je greška kod akvizicije podataka gdje se pri svakoj akviziciji spremi duplirani podatak što je unutarnji problem samog čipa i kao takvo nije predmet ove obrade. Shodno ovakvom spremanju podataka važno je napomenuti da iako kod svake akvizicije sustav spremi duplirani podatak, vrijeme uzorkovanja je nazivno vrijeme uzorkovanja pomnoženo s brojem kanala što govori da je uzorkovanje u redu i naravno daje točno vrijeme između dva uzorka.

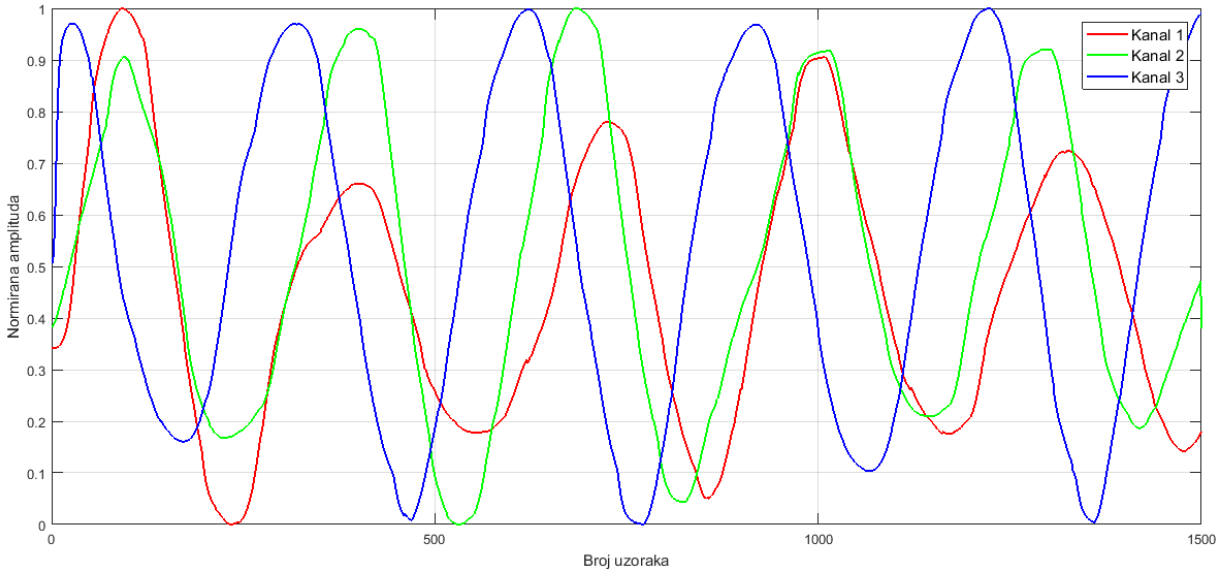
```
164 void data_processing()
165 {
166   // Čitanje podataka iz spremnika
167   i2s_read(I2S_NUM_0, (char*)i2s_read_buff,9000 *
      ↪  sizeof(uint16_t),&bytes_read, portMAX_DELAY);
```

```
176   int j0=0,j1=0,j2=0;
177   for(int i=2;i<9000;i=i+2){
178     uint16_t vals=i2s_read_buff[i]&0xfff;
179     uint16_t ch=i2s_read_buff[i]>>12;
180
181     if(ch==5){adc0f[j0]=filter_adc0.updateEstimate(vals);j0++;}
182     if(ch==4){adc1f[j1]=filter_adc1.updateEstimate(vals);j1++;}
183     if(ch==7){adc2f[j2]=filter_adc2.updateEstimate(vals);j2++;}
184   }
```

Svaki podatak spremljen u spremnik je dug 16 bitova je iako sam podatak dug 12 bitova. Prema referentnim uputstvima ostala četiri bita su rezervirana za broj kanala na kojem je podatak uzet. Takav način zapisa omogućuje razdjeljivanje podataka po kanalima i spremanje svakog podatka u svoj spremnik. Metoda primijenjena u ovom kodu je rotiranje kroz sve podatke i ovisno o vrijednosti prva četiri bita, ostatak bitova spremiti u određeni spremnik.

7.4.2. Izračun vremenskog pomaka

Vremenski pomak signala je definiran fizikom zvuka i brzinom njegovog kretanja kroz neki medij. Kako bi vjernije bila predstavljena metoda kojom se dobivaju odmaci vremena uputno je prvotno prikazati dijagram s očitanim podacima sa sva tri kanala. U nastavku je prikazan dijagram s 1500 uzoraka snimljen na pravom postavu u svrhu testiranja brzine uzorkovanja.



Slika 47. Snimljeni signal

Iz dijagrama je lako vidljivo kako je smjer zvuka na pravcu koji siječe središte te mikrofon tri dok je usmjeren tako da prvo upada na mikrofon tri, a tek onda na ostale. Kako je bilo veoma lako izvući te podatke iz dijagrama iste zaključke treba dizajnirati i programski, no veliko olakšanje je baš ta mogućnost potvrđivanja rezultata samo čitanjem dijagrama.

Izračun vremenskog pomaka dva signala moguće je raditi na mnogo različitih načina, no jedan od jednostavnijih je naći pomak za koji je razlika između dva signala najmanja. Temelj metode se svodi na umnoška dva normirana signala što će za najveće poklapanje davati najveći rezultat. Ovakvom metodom dobiveni rezultat se može ponovno normirati te gledati koliko je maksimalna vrijednost odmaknuta od nultog pomaka.

Kako bi bilo moguće raditi ovakvu usporedbu potrebno je dva signala potpuno odmaknuti jednog od drugog što automatski stvara polje dvostruke dužine od dužine signala. Nakon toga signali se pomiču jedan preko drugog. Pošto se taj proces može izvesti samo pomoću dvostruke petlje radi se o relativno vremenski zahtjevnoj metodi, no s druge strane računski vrlo jednostavnoj.

```
230   for (i=0;i<n;i++)
231   {
232     j = i + dly;
233     if (j < 0 || j >= n){continue;}
234     else
235     {
236       sc12 += (adc0f[i] - mch1) * (adc1f[j] - mch2);
237       sc13 += (adc0f[i] - mch1) * (adc2f[j] - mch3);
238       sc23 += (adc1f[i] - mch2) * (adc2f[j] - mch3);
239     }
240
241     // r predstavlja koeficijent pomaka te ujedino maksimum r je index
        ↪  pomaka
242     r12 = abs(sc12 / denom12);
243     if(r12>max12){max12=r12;idx12=dly;}
244     r13 = abs(sc13 / denom13);
245     if(r13>max13){max13=r13;idx13=dly;}
246     r23 = abs(sc23 / denom23);
247     if(r23>max23){max23=r23;idx23=dly;}
248   }
```

Druga metoda koja se mogla primijeniti je bila metoda kros korelacije, no ta metoda iziskuje računanje brze furierove transformacije za oba signala, množenje transformiranih signala, konjugaciju jednog od njih te ponovno inverz transformacije. Memorijski i procesorski gledano zahtjenost algoritma je slična dok je implementacija mnogo složenija.

Konačno kada se dobiju kašnjenja signala, ista se množe s vremenom uzorkovanja po signalu te brzinom zvuka, naravno uz vrlo veliku pozornost na jedinice. Dobivene vrijednosti su udaljenosti paralelne s pravcem koji prolazi kroz središte kružnice na kojoj se nalaze mikrofoni te točke koju se smatra izvorom zvuka. Slijedeće se uklanjaju rezultati koji ne pripadaju u absolutnu vrijednost manju od 250mm pošto udaljenost između dva mikrofona nikada neće, bez obzira na kut pravca, biti veća.

```
251  dist12=float(idx12)*T*330.0;
252  dist13=float(idx13)*T*330.0;
253  dist23=float(idx23)*T*330.0;
254  //Udaljenost ne može biti veća od 250mm (procijenjeni parametar)
255  if((abs(dist12)>250) or (abs(dist13)>250) or (abs(dist23)>250)){return;}
```

7.4.3. Upravljanje motorom

Ovaj modul je jedan od najjednostavnijih s gledišta implementacije. Razlog tome je korištenje gotove biblioteke za upravljanje radom motora. Metoda upravljanja se sastoji od paljenja kontrolera te nakon toga davanja naredbe za apsolutni pomak motora. Naredba apsolutnog pomaka je zadana u broju okretaja stola pošto je kod konfiguracije definiran prijenosni omjer i broj koraka po okretaju motora. Biblioteka dodatno definira ubrzanja i maksimalne brzine pomaka koje se isto definiraju kod konfiguracije. Vrijednosti brzine i ubrzanja dobiveni su eksperimentalno, na način da se minimizira vrijeme pomaka bez gubitka koraka motora.

```
268   digitalWrite(27,LOW);
269   delay(100);
270   stepper.moveToPositionInRevolutions(angle/360.0);
271   delay(100);
272   digitalWrite(27,HIGH);
```

7.5. Funkcija za početno postavljanje *setup()*

Razlog ove funkcije je rad u Arduinu, naime prema metodi programiranja svaki program mora imati ovu funkciju. Svrha funkcije je je početno postavljanje modula i ostalih korištenih parametara, dok se ona pokreće samo kod paljenja ili ponovnog pokretanja mikrokontrolera. Važna napomena je da se ova funkcija pokreće samo jedanput te nije dopušteno unutar nje koristiti beskonačne petlje.

Unutar ove funkcije pokreće se modul za serijsku komunikaciju, izvrši se funkcija za inicijalizaciju i konfiguraciju I2S i ADC modula te se vrši postavljanje biblioteke za rad s motorom. Kada su postavljeni parametri motora pokreće se funkcija za traženje nultog položaja, nakon čega se definira da je taj položaj zamaknut od definirane nule za negativnih 60 stupnjeva te se kao zadnji korak pomiče stol na 90 stupnjeva, što je položaj ravno ispred.

```
279 void setup(){
280   Serial.begin(115200);
281   configure_i2s();
282   delay(4000);
283   pinMode(27,OUTPUT);
284   digitalWrite(27,LOW);
285   stepper.connectToPins(12,14);
286   stepper.setSpeedInStepsPerSecond(2200);
287   stepper.setAccelerationInStepsPerSecondPerSecond(900);
288   stepper.setStepsPerRevolution(full_rotate);
289   stepper.moveToHomeInSteps(-1,hommingSpeed,full_rotate+400,homePin);
290   delay(500);
291   stepper.setCurrentPositionInRevolutions(-60/360.0);
292   delay(500);
293   stepper.moveToPositionInRevolutions(90/360.0);
294   digitalWrite(27,HIGH);
```

7.6. Funkcija beskonačnog izvođenja *loop()*

Unutar ove funkcije nalazi se kod koji se pokreće kontinurano u beskonačnoj petlji. U slučaju ovog koda pokreće se funkcija za povlačenje zadnjeg spremnika akviziranih podataka te se podaci provlače kroz komparator koji odlučuje je li bilo koji podatak veći od prije definirane razine. Ako je podatak veći ulazi se u funkciju za proračun vremenskog odmaka te se pomiče stol.

Funkcija ne uzima podatke kontinuirano nego u razmacima od 100 milisekundi kako bi se usporila akvizicija i smanjila mogućnost krivih očitanja podataka. Ako je izvršen pomak program će čekati dodatnih pola sekunde, kako ne bi dolazilo do stalnih pomaka stola.

```
299 void loop()
300 {//Unutar ove funkcije kontinuirano se čita ADC te se čeka na prelazak
    ↪  limita na bilo kojem kanalu
301   i2s_read(I2S_NUM_0, (char*)i2s_read_buff,9000 *
      ↪  sizeof(uint16_t),&bytes_read, portMAX_DELAY);
302   for(int i=2;i<9000;i=i+2){
303     uint16_t vals=i2s_read_buff[i]&0xfff;
304     uint16_t ch=i2s_read_buff[i]>>12;
305     if(vals>2075)
306     {
307       // Limit dostignut pokrenu procesuiranje signala i zakret
308       data_processing();
309       delay(500);
310     }
311   }
312   delay(100);
313 }
```

8. ZAKLJUČAK

Izrada ovog rada mi je omogućila rad na pravom mehatroničkom sustavu i ne samo to već i pružila priliku za izraditi svaki dio sustava i uklopiti ga u cjelinu koja izvršava preddefiniranu funkciju. Uspjeh ovakvog projekta, što mi je sada mnogo jasnije, uvelike ovisi o dobroj povezanosti svih sustava u jedan. Samim time primjetio sam kako izrada nije linearna, od hardvera do programa, nego kontinuirano treba biti više cirkularna.

Ovakav sustav ne samo da mi je omogućio primjenu širokog znanja nego me prisilio da naučim potpuno novi dio fizike, barem za mene, za koji sam predpostavljao da je vrlo jednostavan. Naravno radi se o fizici zvuka kojom je uvelike bio pogonjen, a i limitiran ovaj rad.

Posebno mi je drago što sam imao priliku raditi na nečemu što spada na samu granicu današnje robotike te nadam se barem malo pridonjeti napretku iste. S druge strane ovo je prvi sustav koji sam mogao izvesti prvo na računalu i nakon toga kreirati taj isti sustav na 3D pisaču, što je kod dizajna stvorilo mnoge nove prilike ali i probleme.

Konačno nadam se da će ovaj rad potaknuti barem nekog da dublje uđe u područje koje se ovdje obrađuje te stvori nove ideje ili poboljšanja.

LITERATURA

[1] Mario M.: Afektivna robotska glava koja imitira izražaje lica osobe u interakciji, Diplomski rad, FSB Zagreb, 2019.

[2] Al Moubayed, Samer and Beskow, Jonas and Skantze, Gabriel and Granström, Björn: Furhat: A Back-Projected Human-Like Robot Head for Multiparty Human-Machine Interaction, KTH Royal Institute of Technology, 2012.

[3] Milhorat, Pierrick, Divesh Lala, Koji Inoue, Tianyu Zhao, Masanari Ishida, Katsuya Takanashi, Shizuka Nakamura, Tatsuya Kawahara: A conversational dialogue manager for the humanoid robot ERICA, Advanced Social Interaction with Agents, 2019.

[4] ANSI/ASA, S1.1, 2013.

[5] Olson, Harry F.: Music, physics and engineering, New York, 1967.

[6] Burton, Russell: The elements of music: What are they, and who cares?, Conferencijski članak, Sveučilište Adelaide, 2015.

[7] Sessler G. M., West J. E.: Self-Biased Condenser Microphone with High Capacitance, The Journal of the Acoustical Society of America, 1962.

[8] Woon Seob Lee, Seung S. Lee: Piezoelectric microphone built on circular diaphragm, Korea Advanced Institute of Science and Technology, 2008.

[9] Gareth Jones: Echolocation, Sveučilište Bristol, Članak iz Current Biology, 2005.

[10] Uređivači Encyclopaedia Britannica: Sonar, Encyclopaedia Britannica inc., 2019.

[11] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. J. Lang: Phoneme recognition using time-delay neural networks, IEEE Transactions on Acoustics, Speech, and Signal Processing, 1989.

[12] Fiberology, https://fiberlogy.com/en/, 5.5.2020.

[13] ESP32 Technical Reference Manual, https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf, 5.5.2020.

9. PRILOG

9.1. Arduino kod

```
/*****************************************************************
** Program za prepoznavanje lokacije zvuka u ravnini te upravljanje
** rotacijom okretnog stola.
** Primjena u sklopu Diplomskog rada:
** Mehanizam za zakretanje robotske glave koji reagira na zvučne
   ↪ podražaje
**
** Leon Koren, 4.2020
*****************************************************************/

//Poziv svih relevantnih biblioteka
#include <Arduino.h>
#include <driver/i2s.h>
#include <driver/adc.h>
#include "SpeedyStepper.h"
#include <SimpleKalmanFilter.h>

//Pošto su mnoge biblioteke pisane u C jeziku za ESP-IDF potrebna je
   ↪ definicija
extern "C" {
  #include "soc/syscon_reg.h"
  #include "soc/syscon_struct.h"
}

```

```
#define Fs   450000              // Hz
#define T    1000.0/(Fs/3.0);    // ms

/* Inicijalizacija jednostavnog Kalmanovog filtra
** argumenti: (NM,NE,q)
** gdje je: NM - nepouzdanost mjerenja
**          NE - nepouzdanost estimacije(početna vrijednost)
**          q - procesna varijanca
*/
SimpleKalmanFilter filter_adc0(3, 3, 0.015);
SimpleKalmanFilter filter_adc1(3, 3, 0.015);
SimpleKalmanFilter filter_adc2(3, 3, 0.015);

//Inicijalizacija objekta za kontrolu motora
SpeedyStepper stepper;

//Inicijalizacija globalnih varijabli
float adc0f[3000],adc1f[3000],adc2f[3000];
uint16_t i2s_read_buff[9000];
size_t bytes_read;

float belt_ratio=4.25;
int motor_steps=400;
int u_step=8;
int full_rotate=int(u_step*motor_steps*belt_ratio);
bool homeState;
int homePin=4;
int hommingSpeed=600;
```

```
51
52
53  //Funkcija za postavljanje I2S i ADC modula
54  void configure_i2s()
55  {
56    i2s_config_t i2s_config =
57    {
58      // Mod je ADC s I2S modulom
59      .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX |
        ↪  I2S_MODE_ADC_BUILT_IN),
60      // Brzina uzorkovanja
61      .sample_rate = Fs,
62      // 16bit I2S
63      .bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT,
64      // Odabran samo lijevi kanal(izbjegavanje greške krivog uzorkovanja)
65      .channel_format = I2S_CHANNEL_FMT_ALL_LEFT,
66      // I2S format MSB
67      .communication_format = (i2s_comm_format_t)(I2S_COMM_FORMAT_I2S |
        ↪  I2S_COMM_FORMAT_I2S_MSB),
68      // Alokacija broja interrupt-a
69      .intr_alloc_flags = 1,
70      // Broj DMA(brza memorija) spremnika
71      .dma_buf_count = 9,
72      // Broj uzoraka po spremniku
73      .dma_buf_len = 1000,
74      // Bez Audio PLL-a
75      .use_apll = false,
76    };
```

```
77
78      //Postavljanje jednog kanala za inicijalizaciju
79      //Mjerni raspon 0-3.6V
80      adc1_config_channel_atten(ADC1_CHANNEL_7, ADC_ATTEN_11db);
81      //Mjerna rezolucija 12bit (0-4095)
82      adc1_config_width(ADC_WIDTH_12Bit);
83
84      //Instalacija I2S driver-a
85      i2s_driver_install(I2S_NUM_0, &i2s_config, 0, NULL);
86      //Spajanje I2S modula s ADC modulom te pokretanje istog
87      i2s_set_adc_mode(ADC_UNIT_1, ADC1_CHANNEL_7);
88      i2s_adc_enable(I2S_NUM_0);
89
90      //Definiranje registara za upravljanje ADC čitanjem
91      //Treći argument u prvoj funkciji je (N-1) kanala za čitanje
92      SET_PERI_REG_BITS(SYSCON_SARADC_CTRL_REG, SYSCON_SARADC_SAR1_PATT_LEN,
        ↪  2, SYSCON_SARADC_SAR1_PATT_LEN_S);
93
94      /* Postavljanje kanala za skeniranje  prema pravilu:
95      ** HEX broj razdijeljen na 4 dijela:
96      ** 5F, 4F, 7F, 0F gdje je:
97      ** prvi broj broj kanala (5,4,7,0)
98      ** drugi broj postavka kanala F - 0bxxyy
99      ** xx -> 00:9bit 01:10bit 10:11bit 11:12bit
100     ** yy -> 00:0dB(0-1V) 01:2.5dB(0-1.34V) 10:6dB(0-2V) 11:11dB(0-3.6V)
101     */
102     WRITE_PERI_REG(SYSCON_SARADC_SAR1_PATT_TAB1_REG, 0x5F4F7F0F);
103
```

```
104   //Podaci u spremniku su invertirani SYSCON_SARADC_SAR1_INV ih ponovo
      ↪  invertira
105   SET_PERI_REG_MASK(SYSCON_SARADC_CTRL2_REG, SYSCON_SARADC_SAR1_INV);
106 }
107
108 /* Izračun kuta zakreta u ovisnosti o udaljenostima između mikrofona
109 ** Motoda izračuna je temeljena na predpostavci ako x predhodi y
110 ** dxy>0 u suprotnom dxy<0
111 */
112 float calc_angle(float d12,float d13, float d23)
113 {
114   float fdist,ratio,angle_rel,angle;
115   if((d12<0) and (d13<0) and (d23<0))
116   {
117     fdist=abs(d12)+abs(d23);
118     ratio=abs(d23)/fdist;
119     angle_rel=60*ratio;
120     angle=330+angle_rel;
121     if(angle>=360){angle=angle-360;}
122   }
123   else if((d12<0) and (d13<0) and (d23>0))
124   {
125     fdist=abs(d13)+abs(d23);
126     ratio=abs(d23)/fdist;
127     angle_rel=60*ratio;
128     angle=30+angle_rel;
129   }
130   else if((d12<0) and (d13>0) and (d23>0))
```

```
131     {
132       fdist=abs(d13)+abs(d12);
133       ratio=abs(d12)/fdist;
134       angle_rel=60*(1-ratio);
135       angle=90+angle_rel;
136     }
137     else if((d12>0) and (d13>0) and (d23>0))
138     {
139       fdist=abs(d12)+abs(d23);
140       ratio=abs(d12)/fdist;
141       angle_rel=60*(1-ratio);
142       angle=150+angle_rel;
143     }
144     else if((d12>0) and (d13>0) and (d23<0))
145     {
146       fdist=abs(d13)+abs(d23);
147       ratio=abs(d13)/fdist;
148       angle_rel=60*(1-ratio);
149       angle=210+angle_rel;
150     }
151     else if((d12>0) and (d13<0) and (d23<0))
152     {
153       fdist=abs(d13)+abs(d12);
154       ratio=abs(d13)/fdist;
155       angle_rel=60*ratio;
156       angle=270+angle_rel;
157     }
158     return angle;
```

```
159 }
160
161 /* Funkcija koja se aktivira kada detektirani zvuk pređe zadanu
    ↪  vrijednost(treshold)
162 ** Služi za akviziciju jednog slijeda podataka sa sva tri kanala te
    ↪  obradu vremenski odmaka
163 */
164 void data_processing()
165 {
166   // Čitanje podataka iz spremnika
167   i2s_read(I2S_NUM_0, (char*)i2s_read_buff,9000 *
      ↪  sizeof(uint16_t),&bytes_read, portMAX_DELAY);
168
169   /* FOR petlja za parsiranje podataka prema metodi:
170   ** učitana vrijednost je 16bit podatak (0xABBB) gdje je:
171   ** A - hexadecimalni broj kanala(0-15)
172   ** BBB - 12bit podatak
173   ** kako bi izvadili podatak radi se 0xABBB & 0xFFF
174   ** dok kanal se dobiva pomakom broja za 12 bitova ulijevo
175   */
176   int j0=0,j1=0,j2=0;
177   for(int i=2;i<9000;i=i+2){
178     uint16_t vals=i2s_read_buff[i]&0xfff;
179     uint16_t ch=i2s_read_buff[i]>>12;
180
181     if(ch==5){adc0f[j0]=filter_adc0.updateEstimate(vals);j0++;}
182     if(ch==4){adc1f[j1]=filter_adc1.updateEstimate(vals);j1++;}
183     if(ch==7){adc2f[j2]=filter_adc2.updateEstimate(vals);j2++;}
```

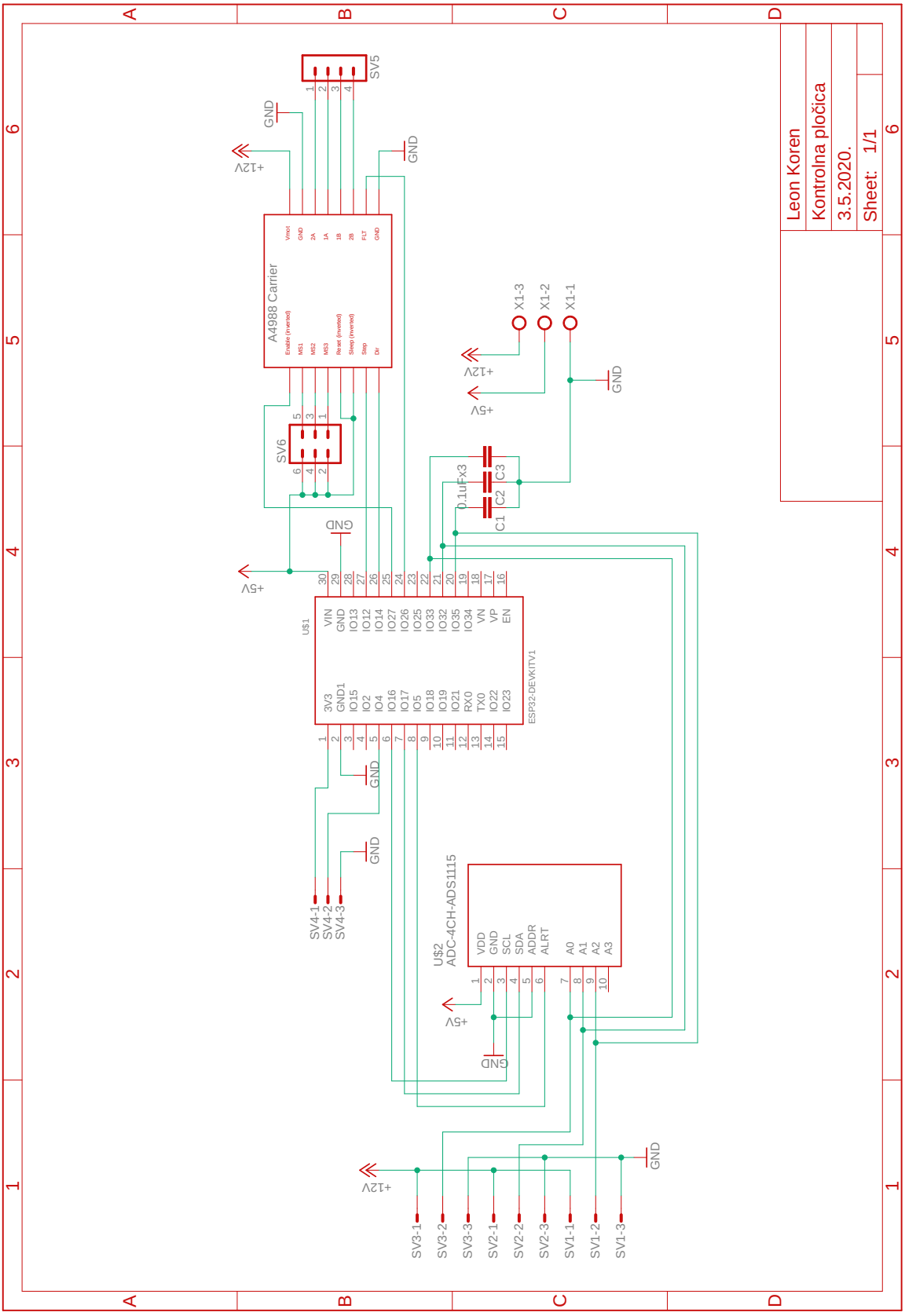
```
184   }
185
186   int i,j,n=1500,dly,maxdelay=n,idx12,idx13,idx23;
187   float mch1,mch2,mch3,sch1,sch2,sch3;
188   float sc12,denom12,r12,lag12,dist12,max12=0;
189   float sc13,denom13,r13,lag13,dist13,max13=0;
190   float sc23,denom23,r23,lag23,dist23,max23=0;
191
192   /* Metoda dobivanja algoritma se svodi na tzv. Goodness of fit
193   ** Algoritam pronalazi za koji pomak krivulje najviše nasjedaju
194   ** jedna na drugu čime se automatski dobiva zamak signala u broju
      ↪  uzoraka
195   */
196
197   //Proračun djelitelja za skaliranje konačnog rezultata
198   mch1 = 0;
199   mch2 = 0;
200   mch3 = 0;
201   for (i=0;i<n;i++)
202   {
203     mch1 += adc0f[i];
204     mch2 += adc1f[i];
205     mch3 += adc2f[i];
206   }
207   mch1 /= n;
208   mch2 /= n;
209   mch3 /= n;
210
```

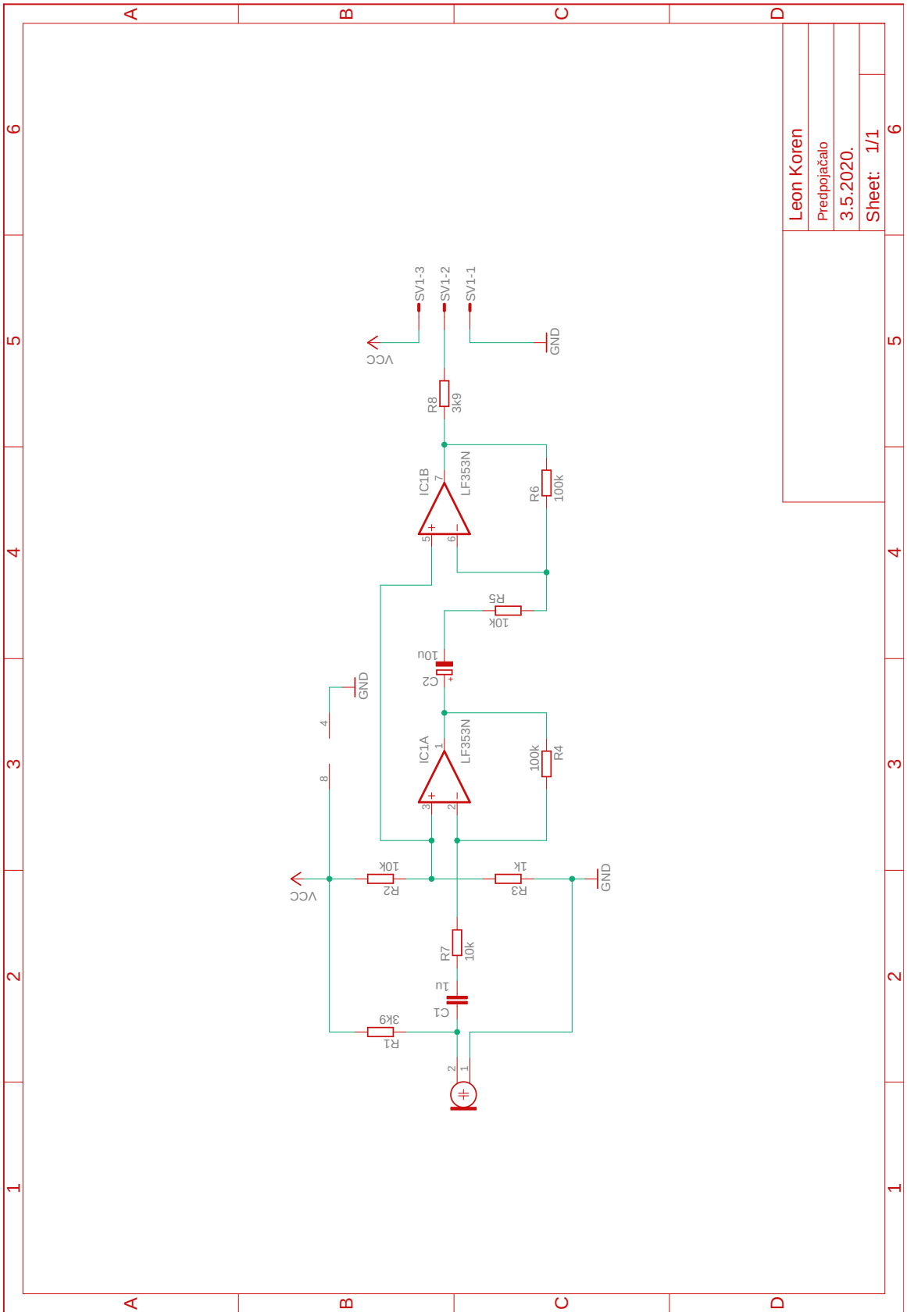
```
211     sch1 = 0;
212     sch2 = 0;
213     sch3 = 0;
214     for (i=0;i<n;i++)
215     {
216       sch1 += (adc0f[i] - mch1) * (adc0f[i] - mch1);
217       sch2 += (adc1f[i] - mch2) * (adc1f[i] - mch2);
218       sch3 += (adc2f[i] - mch3) * (adc2f[i] - mch3);
219     }
220     denom12 = sqrt(sch1*sch2);
221     denom13 = sqrt(sch1*sch3);
222     denom23 = sqrt(sch2*sch3);
223
224     //Glavni dio koji računa poklapanje krivulja pomoću RMS metode
225     for (dly=-maxdelay;dly<maxdelay;dly++)
226     {
227       sc12 = 0;
228       sc13 = 0;
229       sc23 = 0;
230     for (i=0;i<n;i++)
231     {
232       j = i + dly;
233       if (j < 0 || j >= n){continue;}
234       else
235       {
236         sc12 += (adc0f[i] - mch1) * (adc1f[j] - mch2);
237         sc13 += (adc0f[i] - mch1) * (adc2f[j] - mch3);
238         sc23 += (adc1f[i] - mch2) * (adc2f[j] - mch3);
```

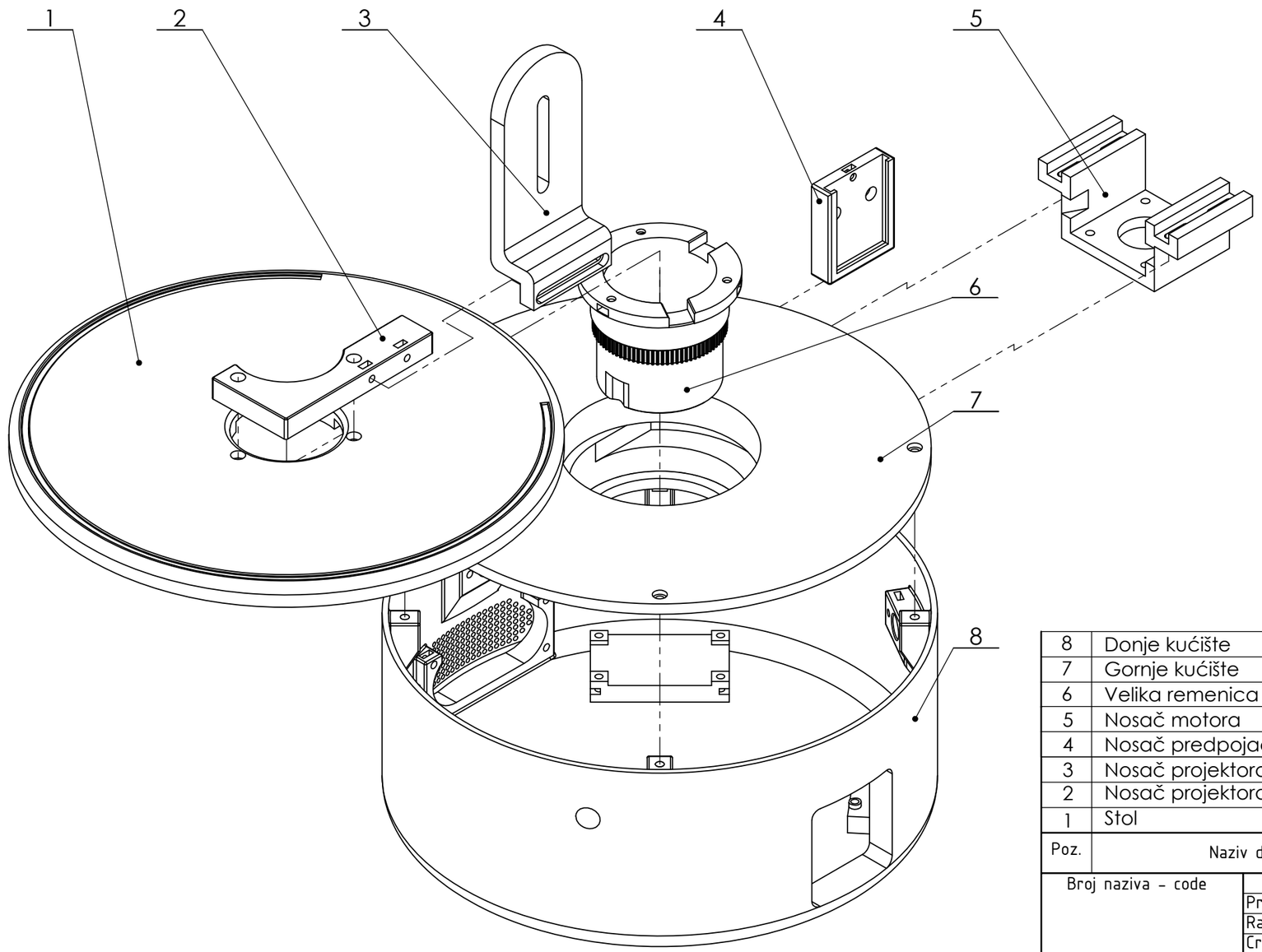
```
239       }
240
241       // r predstavlja koeficijent pomaka te ujedino maksimum r je index
          ↪  pomaka
242       r12 = abs(sc12 / denom12);
243       if(r12>max12){max12=r12;idx12=dly;}
244       r13 = abs(sc13 / denom13);
245       if(r13>max13){max13=r13;idx13=dly;}
246       r23 = abs(sc23 / denom23);
247       if(r23>max23){max23=r23;idx23=dly;}
248     }
249
250     //Izračun udaljenosti iz vremenskih pomaka (brzina zvuka je u mm/ms =
          ↪  m/s)
251     dist12=float(idx12)*T*330.0;
252     dist13=float(idx13)*T*330.0;
253     dist23=float(idx23)*T*330.0;
254     //Udaljenost ne može biti veća od 250mm (procijenjeni parametar)
255     if((abs(dist12)>250) or (abs(dist13)>250) or (abs(dist23)>250)){return;}
256     float angle=calc_angle(dist12,dist13,dist23);
257
258     //kut ne može biti manji od nule
259     if(angle<=0){return;}
260
261     //Prvo ispiši vrijednost kuta na serijski port
262     Serial.print("ANG: ");
263     Serial.println(angle);
264     Serial.println();
```

```
265
266     //Uključi driver za motor pričekaj neko vrijeme te zamakni motor za
        ↪  zadani kut
267     //Uključivanje drivera se radi kako nebi "rezanje" struje utjecalo na
        ↪  ADC
268     digitalWrite(27,LOW);
269     delay(100);
270     stepper.moveToPositionInRevolutions(angle/360.0);
271     delay(100);
272     digitalWrite(27,HIGH);
273   }
274   /* Funkcija koja se izvršava samo jednom na početku programa
275   ** unutar nje je izvršeno pokretanje svih potrebnih modula te
276   ** zadane naredbe za nuliranje položaja motora kako bi bili u mogućnosti
277   ** pomicati motor prema kutu
278   */
279   void setup(){
280     Serial.begin(115200);
281     configure_i2s();
282     delay(4000);
283     pinMode(27,OUTPUT);
284     digitalWrite(27,LOW);
285     stepper.connectToPins(12,14);
286     stepper.setSpeedInStepsPerSecond(2200);
287     stepper.setAccelerationInStepsPerSecondPerSecond(900);
288     stepper.setStepsPerRevolution(full_rotate);
289     stepper.moveToHomeInSteps(-1,hommingSpeed,full_rotate+400,homePin);
290     delay(500);
```

```
291   stepper.setCurrentPositionInRevolutions(-60/360.0);
292   delay(500);
293   stepper.moveToPositionInRevolutions(90/360.0);
294   digitalWrite(27,HIGH);
295
296   Serial.println("Initalization finished");
297 }
298 // Funkcija beskonačne petlje
299 void loop()
300 {//Unutar ove funkcije kontinuirano se čita ADC te se čeka na prelazak
    ↪  limita na bilo kojem kanalu
301   i2s_read(I2S_NUM_0, (char*)i2s_read_buff,9000 *
      ↪  sizeof(uint16_t),&bytes_read, portMAX_DELAY);
302   for(int i=2;i<9000;i=i+2){
303     uint16_t vals=i2s_read_buff[i]&0xfff;
304     uint16_t ch=i2s_read_buff[i]>>12;
305     if(vals>2075)
306     {
307       // Limit dostignut pokrenu procesuiranje signala i zakret
308       data_processing();
309       delay(500);
310     }
311   }
312   delay(100);
313 }
```


Leon Koren
Kontrolna pločica
3.5.2020.
Sheet: 1/1
A4988 Carrier
ESP32-DEVKITV1
ADC-4CH-ADS1115
SV5
SV6
X1-3
X1-2
X1-1
C1
C2
C3
0.1uFx3
GND
+12V
+5V
SV4-1
SV4-2
SV4-3
SV3-1
SV3-2
SV3-3
SV2-1
SV2-2
SV2-3
SV1-1
SV1-2
SV1-3


A
B
C
D
1
2
3
4
5
6
Leon Koren
Predpojačalo
3.5.2020.
Sheet: 1/1
SV1-3
SV1-2
SV1-1
VCC
GND
R8
3k9
IC1B
LF353N
R6
100k
R5
10K
10u
C2
IC1A
LF353N
R4
100k
R2
10K
R3
1k
R7
10k
1u
C1
R1
3k9



Poz.	Naziv dijela	Kom.	Crtež broj Norma	Materijal	Sirove dimenzije Proizvođač	Masa
8	Donje kućište	1	05-20-001	PLA		
7	Gornje kućište	1	05-20-002	PLA		
6	Velika remenica	1	05-20-003	PLA		
5	Nosač motora	1	05-20-008	PLA		
4	Nosač predpojačala	3	05-20-005	PLA		
3	Nosač projektora gornji	1	05-20-007	PLA		
2	Nosač projektora donji	1	05-20-006	PLA		
1	Stol	1	05-20-004	PLA		

Broj naziva - code

	Datum	Ime i prezime	Potpis
Projektirao	20.3.2020	Leon Koren	
Razradio			
Crtao	4.5.2020	Leon Koren	
Pregledao			
Mentor			

FSB Zagreb

ISO - tolerancije

Objekt:

Objekt broj:

R. N. broj:

Napomena:

Kopija

Materijal: PLA

Masa:

Naziv: Rotacijski stol

Pozicija:

Format: A3

Mjerilo originala M1:2

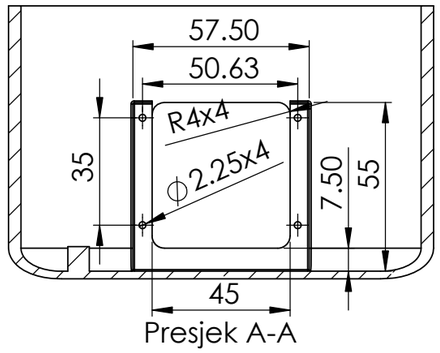
Listova:

Crtež broj: 05-20-000

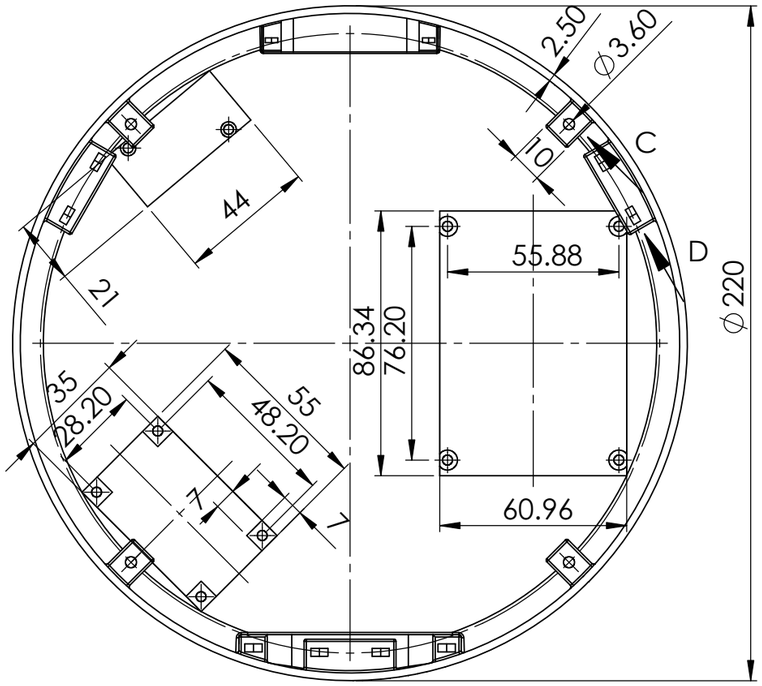
List:

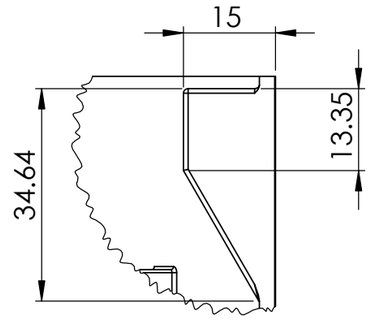
0 10 20 30 40 50 60 70 80 90 100

Design by CADLab

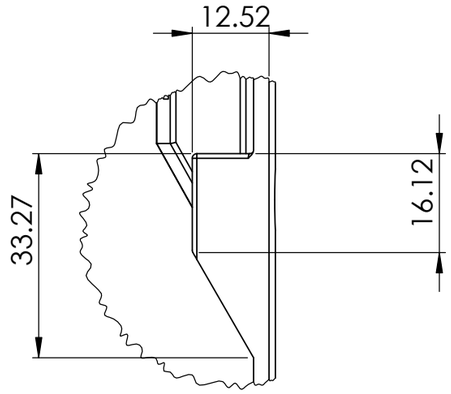


Presjek A-A

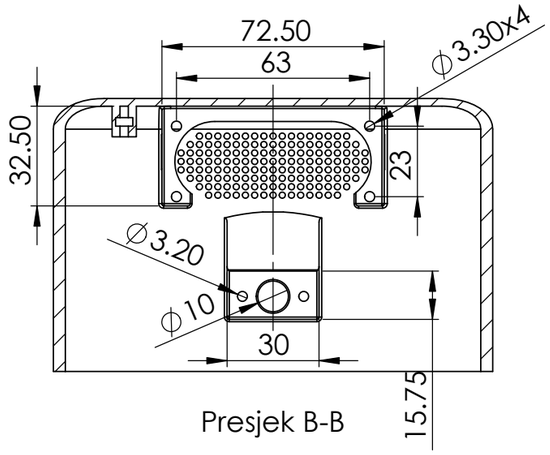




Pogled C



Pogled D

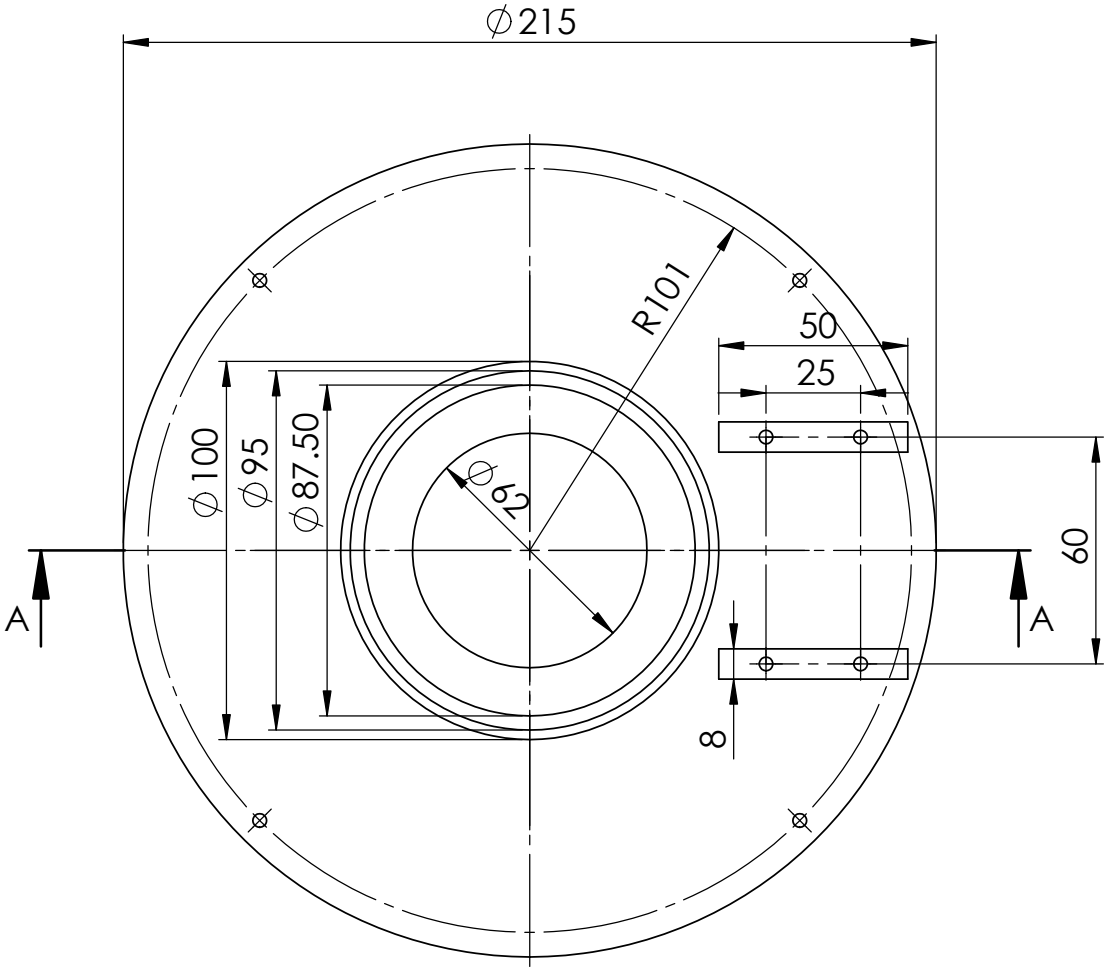


Presjek B-B

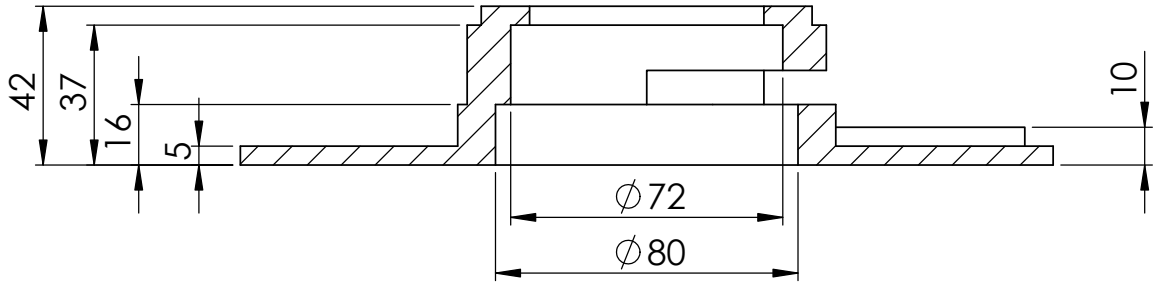
	Datum	Ime i prezime	Potpis
Projektirao	20.3.2020	Leon Koren	
Razradio			
Crtao	4.5.2020	Leon Koren	
Pregledao			
Mentor			

Objekt:	Objekt broj:
	R. N. broj:
Napomena:	Kopija
Materijal: PLA	Masa:
Mjerilo originala M1:2	Naziv: Donje kućište
Pozicija: 1	Format: A3
Listova: 1	List: 1
Crtež broj: 05-20-001	

0 10 20 30 40 50 60 70 80 90 100



Presjek A-A

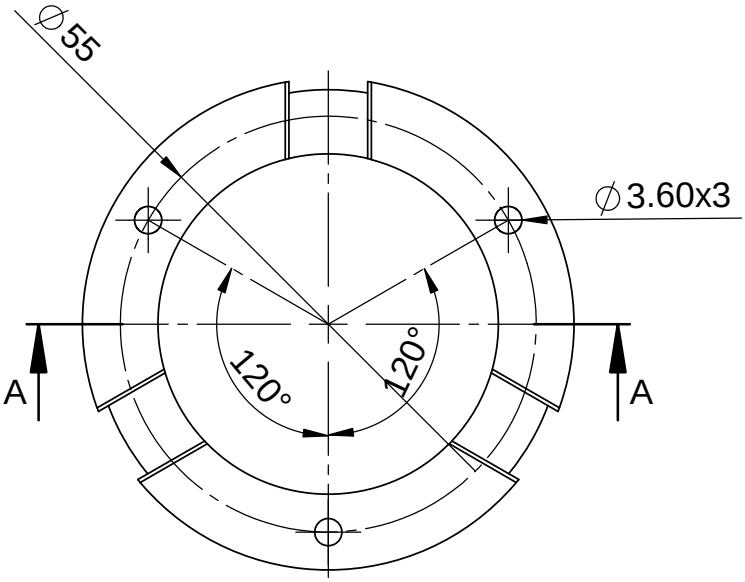


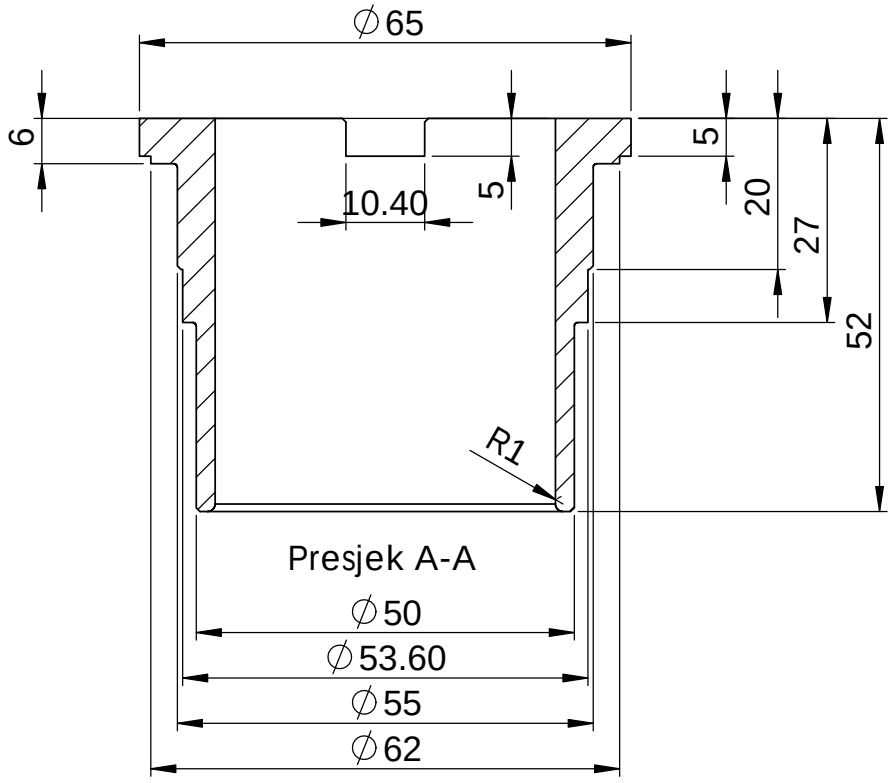
	Datum	Ime i prezime	Potpis
Projektirao	20.3.2020	Leon Koren	
Razradio			
Crtao	4.5.2020	Leon Koren	
Pregledao			

FSB Zagreb

Objekt:	Objekt broj:
	R. N. broj:
Napomena:	Kopija
Materijal: PLA / Masa:	
Mjerilo originala: M1:2	Naziv: Gornje kućište
Pozicija: 1	Format: A4
Listova: 1	List: 1
Crtež broj: 05-20-002	

Design by CADLab





Presjek A-A

∅50

∅53.60

∅55

∅62

	Datum	Ime i prezime	Potpis
Projektirao	20.3.2020	Leon Koren	
Razradio			
Crtao	4.5.2020	Leon Koren	
Pregledao			

FSB Zagreb

Objekt:

Objekt broj:

R. N. broj:

Napomena:

Kopija

Materijal: PLA

Masa:

Mjerilo originala

M1:1

Naziv:

Velika remenica

Pozicija: 1

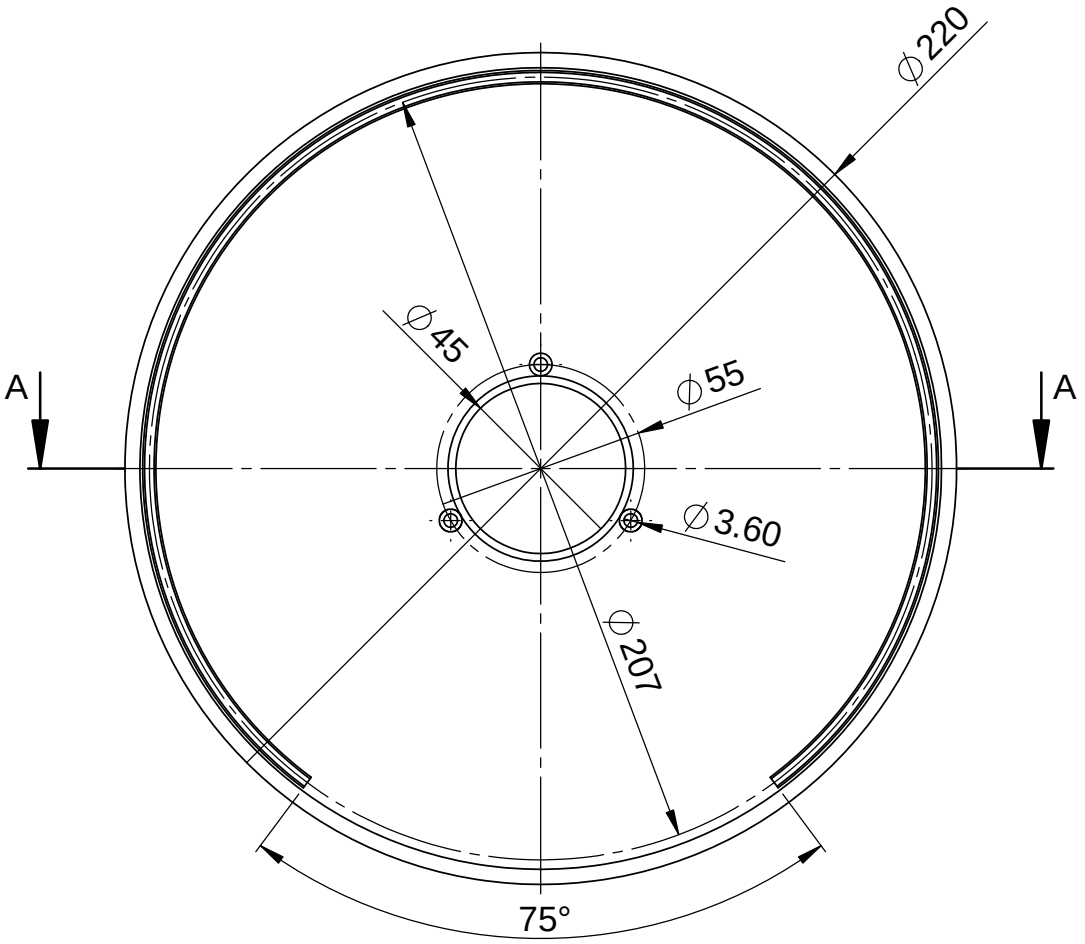
Format: A4

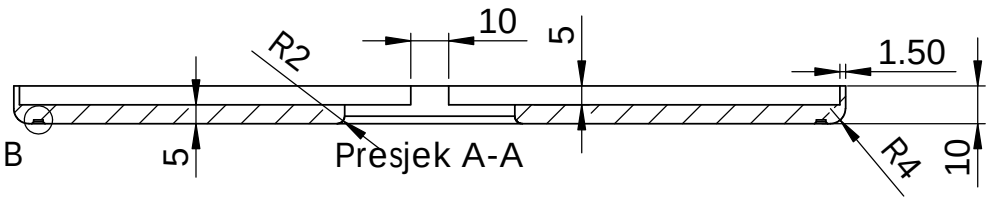
Listova: 1

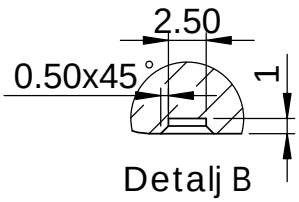
Crtež broj: 05-20-003

List: 1

Design by CADLab







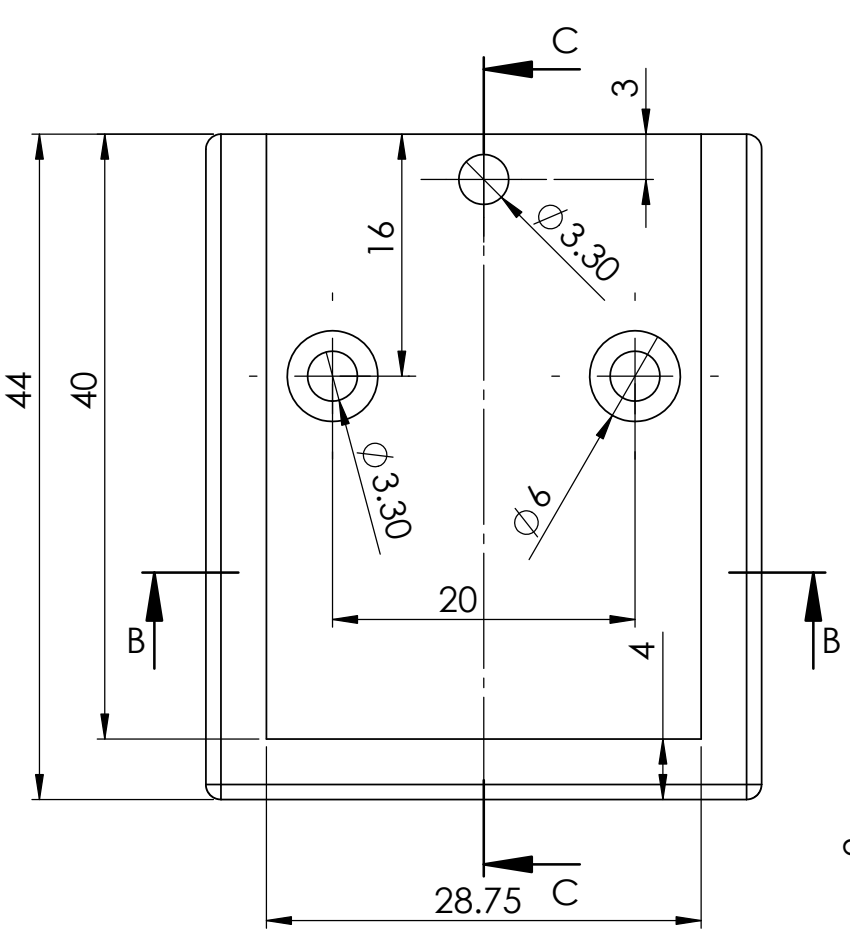
Detalj B

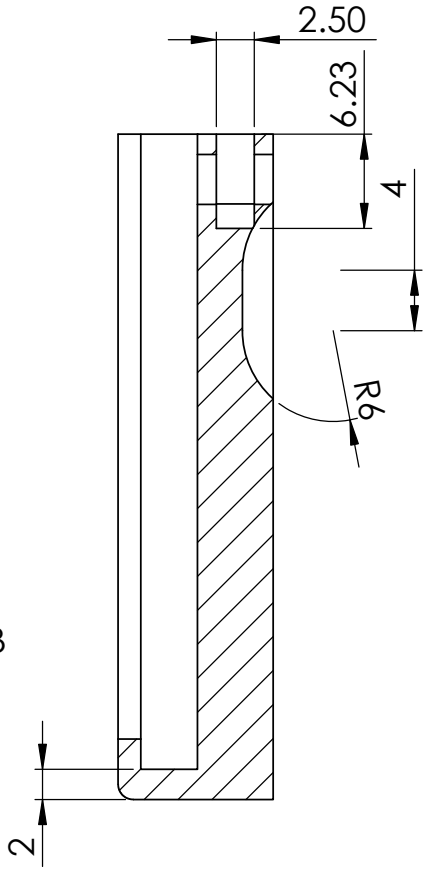
Napomena: Provrti ∅3.6 su jednako razmaknuti 120°
Izdanke širine 10mm s donje strane stola prilagoditi utorima na velikoj remenici

	Datum	Ime i prezime	Potpis	FSB Zagreb
Projektirao	20.3.2020	Leon Koren		
Razradio				
Crtao	4.5.2020	Leon Koren		
Pregledao				

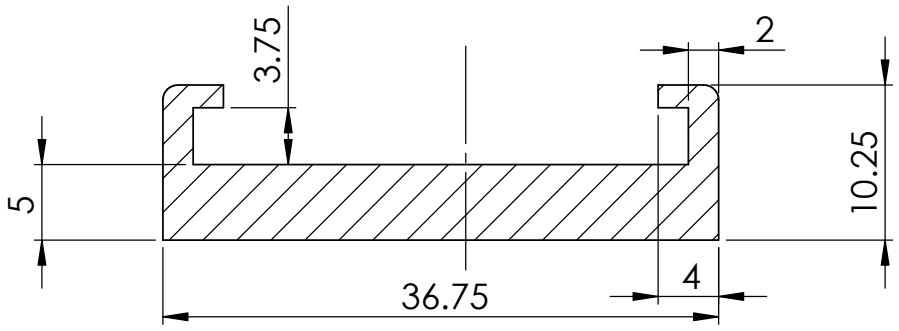
Objekt:	Objekt broj:	
	R. N. broj:	
Napomena:		Kopija
Materijal: PLA	Masa:	
Mjerilo originala M1:2	Naziv: Stol	Pozicija: 1
	Crtež broj: 05-20-004	Format: A4
		Listova: 1
		List: 1

Design by CADLab





Presjek C-C



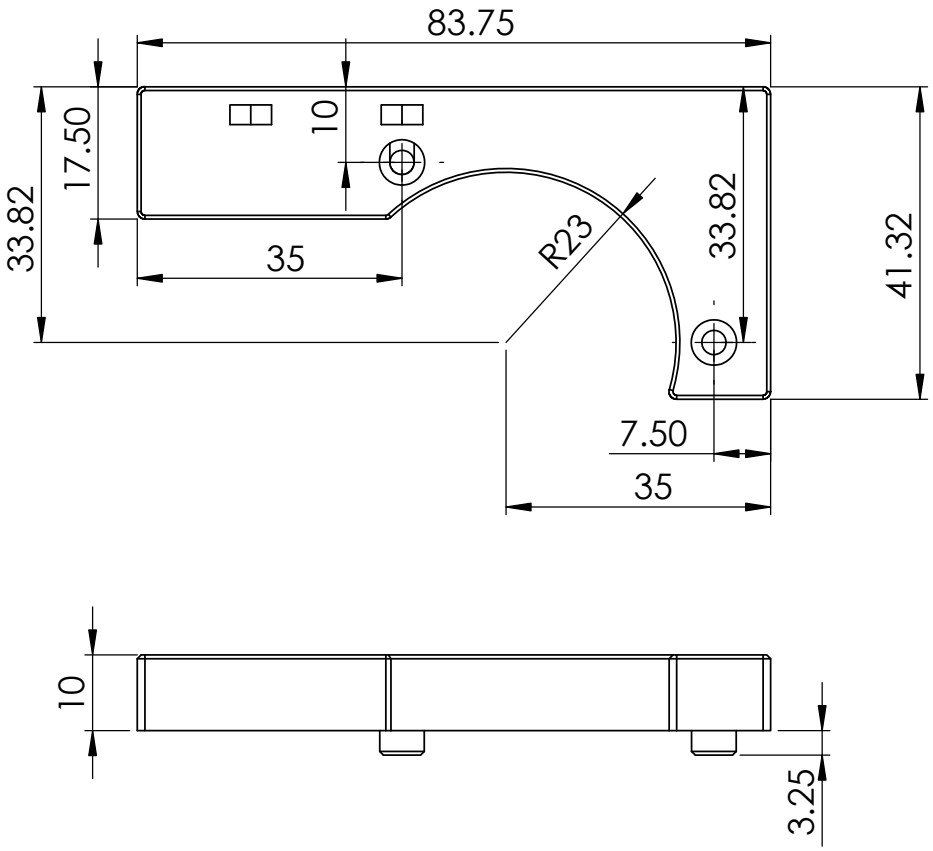
Presjek B-B

	Datum	Ime i prezime	Potpis	FSB Zagreb
Projektirao	20.3.2020	Leon Koren		
Razradio				
Crtao	4.5.2020	Leon Koren		
Pregledao				

Objekt:	Objekt broj:	
	R. N. broj:	
Napomena:		Kopija
Materijal: PLA	Masa:	

Mjerilo originala M2:1	Naziv: Nosač pojačala	Pozicija: 1	Format: A4
			Listova: 1
	Crtež broj: 05-20-005		List: 1

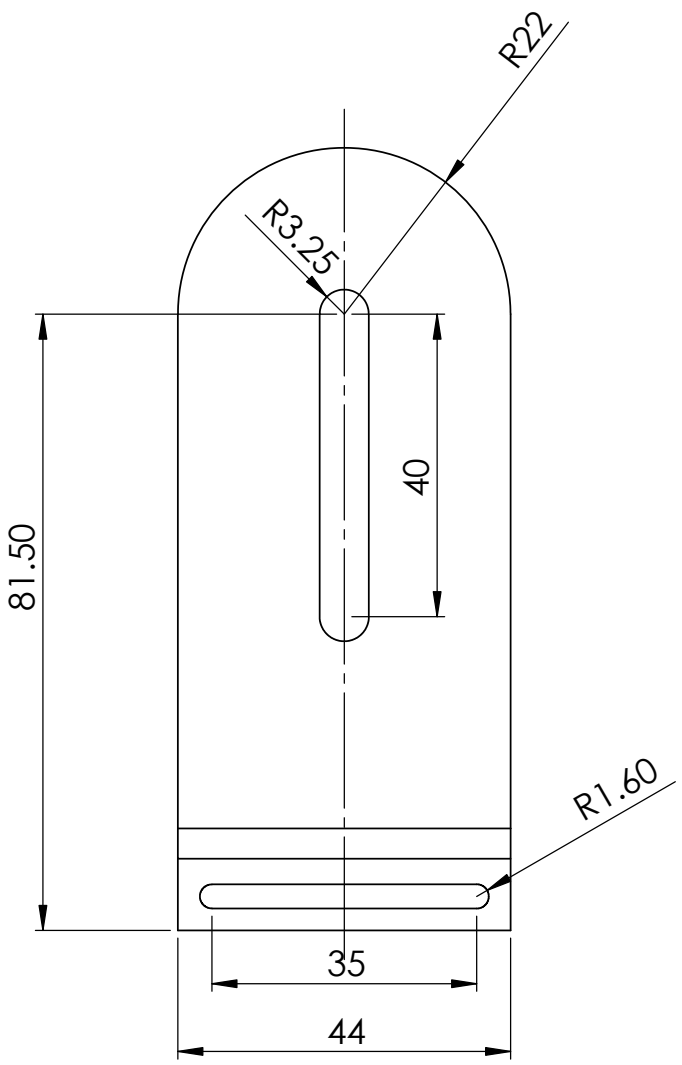
Design by CADLab

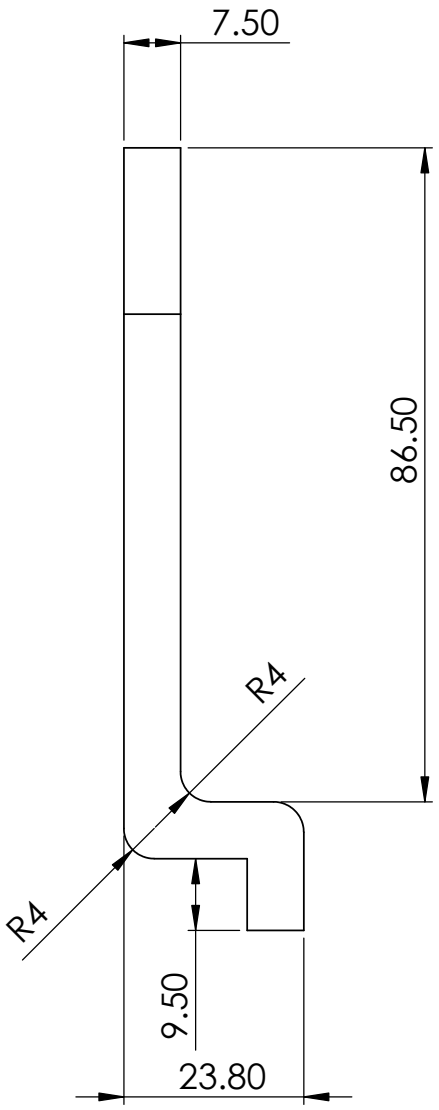


	Datum	Ime i prezime	Potpis	FSB Zagreb
Projektirao	20.3.2020	Leon Koren		
Razradio				
Crtao	4.5.2020	Leon Koren		
Pregledao				

Objekt:	Objekt broj:
	R. N. broj:
Napomena:	Kopija
Materijal: PLA	Masa:
Mjerilo originala M1:1	Naziv: Nosač projektora donji
Pozicija: 1	Format: A4
Listova: 1	List: 1
Crtež broj: 05-20-006	

Design by CADLab

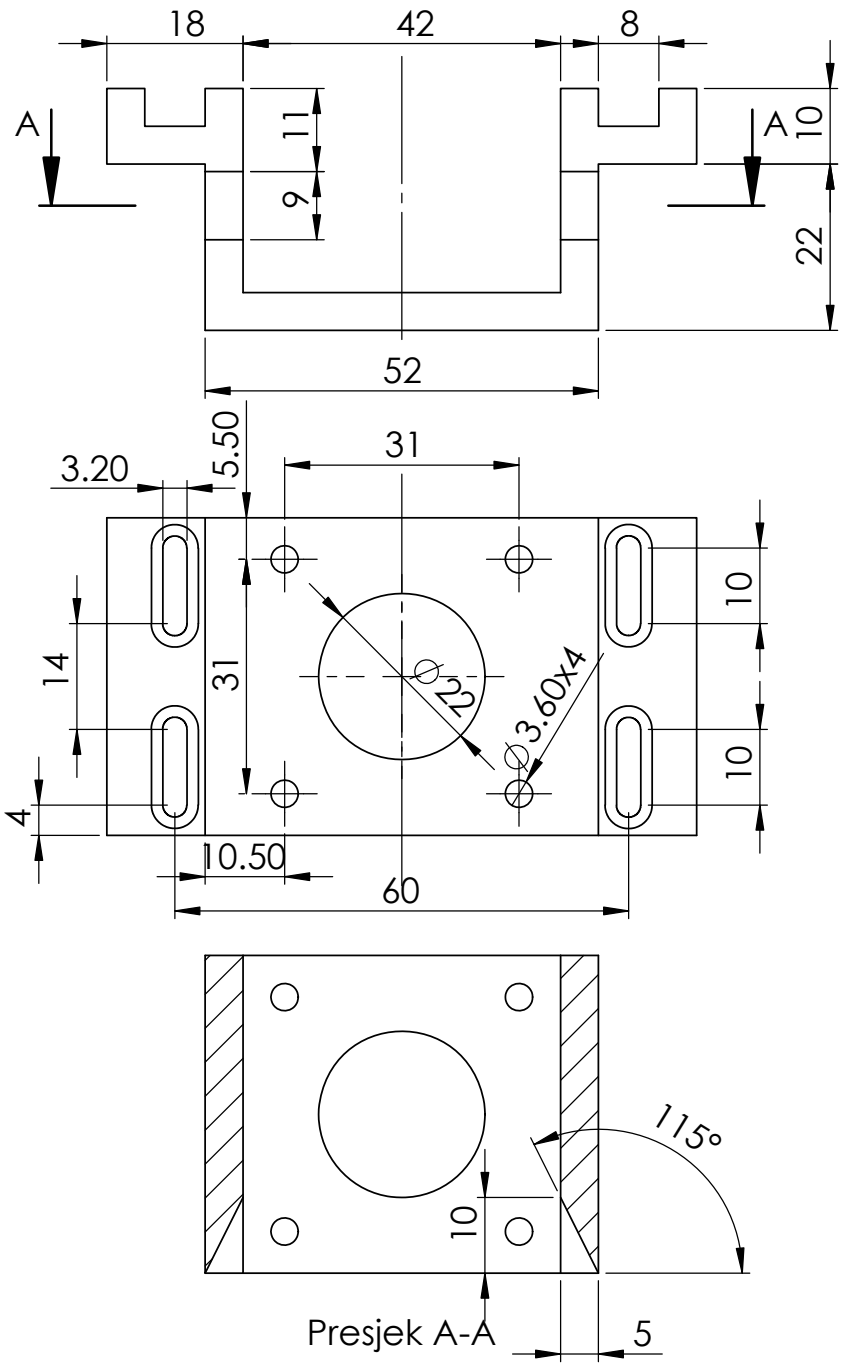




	Datum	Ime i prezime	Potpis	FSB Zagreb
Projektirao	20.3.2020	Leon Koren		
Razradio				
Crtao	4.5.2020	Leon Koren		
Pregledao				

Objekt:	Objekt broj:	
	R. N. broj:	
Napomena:		Kopija
Materijal: PLA	Masa:	
Mjerilo originala M1:1	Naziv: Nosač projektora gornji	Pozicija: 1
	Crtež broj: 05-20-007	Format: A4
		Listova: 1
		List: 1

Design by CADLab



	Datum	Ime i prezime	Potpis
Projektirao	20.3.2020	Leon Koren	
Razradio			
Crtao	4.5.2020	Leon Koren	
Pregledao			

FSB Zagreb

Objekt:	Objekt broj:
	R. N. broj:
Napomena:	Kopija
Materijal: PLA	Masa:
Mjerilo originala M1:1	Naziv: Nosač motora
Pozicija: 1	Format: A4
Listova: 1	List: 1
Crtež broj: 05-20-008	

Design by CADLab