

Implementacija robota za RoboCup Junior Soccer Natjecanje

Kolarić, Ivan

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:959548>

Rights / Prava: [Attribution 3.0 Unported/Imenovanje 3.0](#)

Download date / Datum preuzimanja: **2025-02-05**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Ivan Kolarić

**IMPLEMENTACIJA ROBOTA ZA
ROBOCUP JUNIOR SOCCER
NATJECANJE**

DIPLOMSKI RAD

Varaždin, 2024.
SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Ivan Kolarić

Matični broj: 0035207443

Studij: Informatika u obrazovanju

IMPLEMENTACIJA ROBOTA ZA ROBOCUP JUNIOR SOCCER
NATJECANJE

DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Ivan Magdalenić

Varaždin, kolovoz 2024.

Ivan Kolarić

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj diplomski rad bavi se izradom robota u skladu s pravilima RoboCup Junior natjecanja, koje promovira robotiku i programiranje među mladima. Kroz osobno iskustvo kao član organizacijskog tima na svjetskoj razini od 2023. godine, cilj je bio razumjeti izazove s kojima se natjecatelji susreću te ponuditi praktična rješenja. Teorijska polazišta rada uključuju pregled osnovnih principa RoboCup natjecanja, istraživanje elektroničkih komponente te programiranje mikrokontrolera. Metodološki, rad obuhvaća nabavu dijelova, elektroničko i mehaničko povezivanje, dizajn PCB pločice u EasyEDA programu te 3D modeliranje konstrukcije robota u SolidWorks-u. Nakon dizajna, slijedi 3D printanje dijelova i konačno sklapanje robota. Glavne teze rada su identifikacija i prevladavanje problema u izradi robota, stjecanje praktičnih vještina u dizajnu i implementaciji robotskih sustava te doprinos unapređenju RoboCup Junior natjecanja kroz osobno iskustvo i inovacije. Kroz ovaj proces, rad detaljno opisuje sve faze razvoja robota, od početnog dizajna do završnog sklapanja i programiranja samog robota. Zaključci rada naglašavaju važnost praktične primjene stečenog znanja te pružaju smjernice budućim natjecateljima i mentorima u RoboCup Junior natjecanju. Iskustva i saznanja stečena kroz ovaj projekt imaju potencijal unaprijediti organizaciju i pravila natjecanja, potičući mlade na daljnji razvoj u području robotike i programiranja. Također na kraju vjerujem kako će mi ovo biti dobra podloga da krenem u RoboCup Major lige, gdje ću se moći natjecati kao član jednog tima ili kreirati vlastiti tim.

Ključne riječi: Robotika, RoboCup junior, Programiranje, 3D modeliranje, Elektronika, Arduino, SolidWorks, EasyEDA.

Sadržaj

| | |
|---|-----|
| Sadržaj..... | iii |
| 1. Uvod..... | 1 |
| 2. RoboCup..... | 3 |
| 3. RoboCup Junior..... | 5 |
| 3.1. RoboCup Junior Rescue i Dance kategorije | 6 |
| 3.2. RoboCup Junior Soccer kategorija..... | 7 |
| 3.3. Pravila Soccer Lightweight lige | 9 |
| 4. Materijali potrebni za rad | 12 |
| 4.1. Glavne komponente robota | 12 |
| 4.1.1. Teensy 4.1..... | 12 |
| 4.1.2. OpenMV H7+..... | 14 |
| 4.1.3. Baterija..... | 15 |
| 4.1.4. Step-Down Regulatori | 17 |
| 4.1.5. Motori..... | 19 |
| 4.1.6. Kontroleri motora | 20 |
| 4.2. Senzori..... | 21 |
| 4.2.1. IMU Kompas..... | 21 |
| 4.2.2. Senzor za loptu..... | 23 |
| 4.2.3. Senzori udaljenosti | 24 |
| 4.2.4. Barijera | 26 |
| 4.2.5. Mehanizam za udaranje lopte | 26 |
| 4.3. Ostali potrebni materijali | 28 |
| 5. Izrada elektroničke pločice (PCB-a)..... | 29 |
| 5.1. Izrada pločica u programu EasyEDA | 30 |
| 5.1.1. Izrada Glavne PCB pločice | 30 |
| 5.1.2. Izrada Linijske PCB pločice..... | 35 |

| | |
|--|----|
| 6. Izrada robota | 38 |
| 6.1. Dizajn 3D dijelova | 38 |
| 6.1.1. Donja šasija | 38 |
| 6.1.2. Gornja šasija | 40 |
| 6.1.3. Držać pucaljke (kickera) | 41 |
| 6.1.4. Udarač | 42 |
| 6.1.5. Držać senzora za loptu | 43 |
| 6.1.6. Držać kamere | 44 |
| 6.1.7. Držać ultrazvučnih senzora | 45 |
| 6.1.8. Zrcalo i ručka | 46 |
| 6.2. 3D Printanje dijelova | 47 |
| 6.2.1. Filamenti | 49 |
| 6.2.2. Prusa slicer | 50 |
| 6.3. Sklapanje robota | 52 |
| 7. Programiranje i testiranje robota | 56 |
| 7.1. Arduino okolina | 56 |
| 7.2. OpenMV okolina | 57 |
| 7.3. Program | 57 |
| 8. Zaključak | 71 |
| Popis literature | 73 |
| Popis slika | 75 |

1. Uvod

Tema ovog diplomskog rada je izrada robota od početka do kraja, uključujući implementaciju koda koji će omogućiti njegovu primarnu funkciju. Robot je dizajniran u skladu s pravilima RoboCup Junior natjecanja, u čijem sam organizacijskom timu na svjetskoj razini od 2023. godine. Upravo kroz sudjelovanje u organizaciji ovog natjecanja, dobio sam ideju za završni rad, te sam kroz proteklu godinu aktivno sudjelovao u oblikovanju pravila natjecanja. RoboCup Junior je međunarodno natjecanje koje potiče mlade da se bave robotikom i programiranjem kroz praktične projekte. Kao član organizacijskog tima, imao sam priliku iz prve ruke vidjeti izazove i probleme s kojima se natjecatelji suočavaju. Cilj ovog rada je stavljanje u ulogu učenika i studenata koji se natječu u RoboCup Junioru, te identificiranje problema s kojima se suočavaju tijekom izrade robota. Smatram da će mi ovo iskustvo značajno pomoći u budućnosti, kako u vođenju timova kao mentor na RoboCup natjecanjima, tako i u kreiranju novih pravila.

Tijekom izrade rada, prošao sam kroz sve faze izrade robota, počevši od nabave dijelova, njihove elektroničke i mehaničke integracije, pa sve do programiranja robota. Većinu dijelova nabavio sam online od raznih dobavljača, njihove web stranice sam naveo i uključio cijenu pojedinih komponenata. Osim toga, istražio sam različite izvore i dobavljače kako bih osigurao najkvalitetnije i najpouzdanije dijelove za moj projekt. Nakon toga, povezo sam komponente u programu EasyEDA gdje sam dizajnirao PCB pločicu. EasyEDA je online alat za dizajn elektroničkih shema i PCB-ova, koji omogućava jednostavno i intuitivno povezivanje komponenata. Po dovršetku dizajna i narudžbe PCB pločice, kreirao sam konstrukciju robota u programu za 3D modeliranje, SolidWorks. SolidWorks je moćan alat za 3D modeliranje koji omogućava precizno kreiranje i simulaciju mehaničkih dijelova robota. Slijedi 3D printanje dijelova, što je omogućilo izradu prilagođenih komponenti koje savršeno odgovaraju dizajnu robota. Korištenjem 3D printera, brzo i učinkovito sam izradio dijelove, testirao ih i po potrebi modificirao. Nakon što su svi dijelovi isprintani i spremni, slijedilo je konačno sklapanje robota u cjelinu. Ovaj korak uključuje pažljivo povezivanje svih elektroničkih i mehaničkih dijelova, testiranje funkcionalnosti i otklanjanje eventualnih problema. Na samom kraju isprogramirao sam robota u programskom jeziku C++, u Arduino okolini. Sa predsjednikom udruge „Robofreak“ dogovorio sam kako će mi ta udruga pribaviti sve potrebne materijale za izradu robota.

Uz tehničke aspekte izrade robota, ovaj rad se osvrnuo i na metodologiju rada, planiranje projekta, upravljanje vremenom i resursima, te dokumentiranje svih koraka procesa. Također, u sklopu rada uključena je i analiza izazova s kojima sam se susreo tijekom izrade robota i kako sam ih prevladao, što će biti vrijedno iskustvo za buduće projekte. Ovaj rad

detaljno prikazuje sve korake izrade robota, te će poslužiti kao vodič budućim natjecateljima i mentorima u RoboCup Junior natjecanju. Nadam se da će moje iskustvo i znanje stečeno kroz ovaj projekt doprinijeti unapređenju RoboCup Junior natjecanja i poticanju mladih na bavljenje robotikom i programiranjem, kao i potaknuti mene da nastavim u tom smjeru razvijati robota za Small size League u RoboCup natjecanju.

2. RoboCup

RoboCup je inicijativa Svjetskog kupa za robote, osmišljena kako bi potaknula istraživanje u području umjetne inteligencije (AI) i inteligentne robotike pružanjem standardiziranog problema u kojem se mogu integrirati i ispitivati razne tehnologije. Inicijativu RoboCup započeo je Hiroaki Kitano 1996. godine, s ciljem stvaranja platforme za istraživanje i razvoj u AI i robotici. Prvo RoboCup natjecanje održano je na IJCAI-97 u Nagoyi. Kako bi robotski tim uspješno sudjelovao u nogometnoj utakmici, potrebno je uključiti različite tehnologije, uključujući dizajnerske principe autonomnih agenata, suradnju više agenata, stjecanje strategije, razmišljanje u stvarnom vremenu, robotiku i fuziju senzora. RoboCup se predlaže kao novi standardni problem za istraživanje u AI i robotici. Korištenje nogometne igre kao platforme za istraživanje širokog spektra AI i robotskih tehnologija omogućuje integraciju i ispitivanje različitih tehnologija u dinamičnom okruženju. Stvarni cilj RoboCupa je do 2050. godine razviti tim autonomnih robota koji će pobijediti ljudsku nogometnu momčad prvaka svijeta u skladu s pravilima FIFA-e.[1]

Za uspješan rad robotskog tima na nogometnoj utakmici potrebno je inkorporirati dizajnerske principe autonomnih agenata, suradnju više agenata, stjecanje strategije, razmišljanje u stvarnom vremenu, robotiku i fuziju senzora. Suprotno AAAI robotskom natjecanju, koje je prilagođeno za jednog teškog robota, RoboCup je zadatak za tim brzih robota u dinamičnom okruženju. Postojeći robotski igrači dizajnirani su za izvođenje pojedinačnih akcija poput guranja, driblanja, rolanja ili udaranja. RoboCup igrač trebao bi izvoditi više podzadatka poput pucanja, driblanja, dodavanja i bacanja lopte, što zahtijeva dizajn koji može obavljati više zadataka. Vizualne informacije pružaju podatke za percepciju vanjskog svijeta, ali igrač može trebati dodatne senzore poput sonara i dodira za diskriminaciju situacija. Pojačano učenje je obećavajuća metoda jer omogućava reaktivna i prilagodljiva ponašanja. RoboCup je platforma za skaliranje ovih metoda na složenije zadatke. Pravila za natjecanje sa stvarnim robotima uključuju standarde za veličinu igrališta, robota, timove, ciljeve, boje terena i duljinu igre. U simulacijskom dijelu koristi se Soccer Server, simulator koji pruža virtualno igralište za nogometne utakmice s robotima. RoboCup pruža istraživačke probleme za AI i robotiku, pozivajući na sudjelovanje u inicijativi za definiranje pravila, razvoj istraživačkog okruženja i organizaciju natjecanja i radionica.[1]

RoboCup je odlična inicijativa sa odličnim ciljem kojeg je veoma teško sprovesti, te se sve nade polažu u budućnost kada bi ekipe studenata riješile jedne od najvećih problema današnjice vezanih uz robotiku, te ih inkorporirali u svoje robotičke zamisli. Mišljenja sam kako je ovaj pristup generalno dobar i kako potiče na razvoj ne samo robotike već i cijele industrije od kada je i sam RoboCup krenuo.

3. RoboCup Junior

RoboCup Junior je edukativna inicijativa usmjerena na mlade učenike, pružajući im priliku da sudjeluju u lokalnim, regionalnim i međunarodnim događanjima vezanim za robotiku. Program je namijenjen učenicima osnovnih i srednjih škola, kao i studentima koji nemaju pristup resursima potrebnim za sudjelovanje u naprednijim RoboCup ligama. Fokus RCJ-a je prvenstveno na obrazovanju, s ciljem uvođenja mladih u svijet robotike.[2]

Povijest i razvoj RoboCupJunior-a seže više od dvadeset godina unatrag, kada su osobna računala započela novu eru u tehnologiji i obrazovnim aplikacijama. Inspiriran teorijama konstruktivizma Jean Piageta i Seymour Papertovim konceptom konstrukcionizma, RCJ promovira učenje kroz aktivno sudjelovanje učenika u stvaranju nečega što im je osobno važno. Od 1998. godine, RCJ se razvio u globalni pokret, uključujući tisuće učenika iz cijelog svijeta.[3]

Za razliku od drugih programa za mlade, RCJ se ističe nekoliko ključnih aspekata. Prvo, naglasak je na obrazovanju, a ne na natjecanju. Izazovi u RCJ-u ostaju isti iz godine u godinu, omogućujući učenicima da postupno razvijaju složenija rješenja kako stječu više znanja. RCJ lige - Nogomet, Spašavanje i OnStage - su jednostavne za razumijevanje, čak i za gledatelje bez tehničkog predznanja. Također, RCJ potiče dublje proučavanje računalnih znanosti i programiranja, naglašavajući autonomne robote. RCJ predstavlja početnu razinu međunarodne RoboCup inicijative, koja je posvećena obrazovanju i uključivanju mladih u tehnologiju. [2]

Izazovi u RCJ-u su osmišljeni kako bi poticali suradnju, rješavanje problema i postizanje zadataka u timu. U nogometnoj ligi, timovi od dva autonomna robota natječu se na dinamičnom terenu, dok u ligi Spašavanje roboti identificiraju "žrtve" u simuliranim scenarijima katastrofe. Liga OnStage potiče kreativnost, gdje roboti nastupaju uz glazbu, obučeni u kostime, usklađujući pokrete s ritmom. [3]

Natjecanja za učenike i studente organiziraju se na regionalnoj razini, omogućujući im da razvijaju svoje vještine u manje formalnom okruženju. Jedno takvo regionalno natjecanje se događa i svake godine u Hrvatskoj. Na RoboCup Junior Croatia Open natjecanjima sudjelujem od samih začetaka 2011. godine, kao učenik, te kao organizator u posljednjih nekoliko godina. Iz primijećenog tvrdim kako RoboCupJunior ne samo da pruža uvod u robotiku, već i motivira

mlade da razvijaju tehničke vještine, uče o timskom radu, te dijele svoja znanja i tehnologije s vršnjacima. Ovaj program predstavlja jedinstvenu priliku za mlade različitih interesa i sposobnosti da zajedno rade na zajedničkom cilju, stvarajući pritom temelje za buduće tehnološke inovacije.

3.1. RoboCup Junior Rescue i Dance kategorije

Među popularnijim kategorijama unutar RoboCupJunior-a su Rescue i Dance, koje nude jedinstvene izazove i mogućnosti za učenike različitih uzrasta.

Rescue kategorija osmišljena je da simulira stvarne situacije spašavanja u kojima roboti moraju identificirati i pomoći "žrtvama" u simuliranim scenarijima katastrofa. Ova kategorija potiče učenike da razviju robote sposobne za autonomno kretanje, prepoznavanje prepreka i žrtava, te donošenje odluka u stvarnom vremenu. Glavni cilj Rescue kategorije je razviti robota koji može što brže i preciznije proći kroz prepreke, slijediti crte na terenu i prepoznati žrtve u simuliranom okruženju. Ovo zahtijeva od učenika da se fokusiraju na napredne aspekte robotike, poput programiranja senzora, mapiranja terena, te optimizacije algoritama za prepoznavanje i reagiranje na različite situacije.

Jedan od ključnih izazova u ovoj kategoriji je nepredvidivost terena. Učenici moraju dizajnirati robote koji se mogu prilagoditi promjenjivim uvjetima, poput neravnog terena, nagiba i prepreka. Ovo potiče inovativno razmišljanje i vještine rješavanja problema, jer timovi moraju osigurati da njihovi roboti mogu uspješno navigirati i izvršiti zadatak unatoč svim preprekama. Rescue kategorija potiče suradnju među timovima, budući da učenici često moraju dijeliti ideje i strategije kako bi postigli što bolje rezultate. Osim tehničkih vještina, učenici razvijaju i sposobnosti timskog rada, komunikacije i upravljanja projektima, što su ključne kompetencije u današnjem tehnološkom svijetu.

OnStage kategorija, s druge strane, nudi potpuno drugačiji set izazova i naglašava kreativnost i umjetnički izraz. U ovoj kategoriji, učenici razvijaju robote koji izvode plesne točke, sinkronizirane s glazbom, često u kombinaciji s vlastitim kostimima i scenografijom. Glavni cilj OnStage kategorije je stvoriti robota koji može autonomno izvesti koreografiju u skladu s glazbom, pokazujući pritom preciznost i koordinaciju pokreta. Ova kategorija potiče učenike da istraže kreativne aspekte robotike, poput dizajna, umjetničkog izraza i prezentacije. Jedan od izazova u OnStage kategoriji je postizanje fluidnih i koordiniranih pokreta robota koji prate ritam glazbe. Učenici moraju programirati robote tako da prepoznaju ritam i dinamiku glazbe te da se kreću u skladu s njom. Ovo zahtijeva precizno programiranje, ali i dobru suradnju

među članovima tima kako bi se osigurala usklađenost između tehnoloških i umjetničkih aspekata izvedbe.

Kao i Rescue, OnStage kategorija također potiče interdisciplinarni pristup učenju, kombinirajući tehnologiju s umjetnošću i dizajnom. Učenici često koriste elemente scenskog dizajna, kostimografije i koreografije kako bi stvorili vizualno i tehnički impresivne izvedbe. Ova kategorija posebno je popularna među učenicama, jer nudi priliku za izražavanje kreativnosti i umjetničkog talenta kroz tehnologiju. Time se potiče veći interes djevojaka za STEM područja, što se također ističe kao jedan od ciljeva RoboCupJunior. [3]

3.2. RoboCup Junior Soccer kategorija

RoboCupJunior Soccer kategorija jedan je od najdinamičnijih i najuzbudljivijih dijelova RoboCupJunior natjecanja, koji privlači mlade entuzijaste robotike iz cijelog svijeta. Ova kategorija kombinira tehničke vještine, timski rad, strategiju i uzbuđenje nogometne igre, omogućujući razvijanje i demonstraciju znanja u području robotike, programiranja i mehatronike kroz natjecateljski duh.

U RoboCupJunior Soccer kategoriji, dva tima autonomnih robota suočavaju se na posebno dizajniranom terenu, u pokušaju postizanja više golova od protivničkog tima unutar zadanog vremena. Timovi obično imaju dva robota, a svaki robot mora biti potpuno autonoman, što znači da nije dopuštena nikakva vanjska kontrola tijekom igre. Roboti moraju samostalno prepoznati loptu, navigirati terenom i pokušati postići gol, dok istovremeno brane svoj vlastiti gol.

Soccer kategorija unutar RoboCupJunior-a podijeljena je u dvije glavne podkategorije: Soccer Lightweight i Soccer Open. U Soccer Lightweight kategoriji koristi se infracrvena (IR) lopta, koja emitira IR svjetlo i koju roboti koriste za prepoznavanje i praćenje lopte na terenu. Ova lopta omogućuje robotima da precizno detektiraju položaj lopte i prilagode svoje kretanje kako bi bili što uspješniji u igri. S druge strane, Soccer Open kategorija koristi narančastu loptu, koja se oslanja na vizualnu prepoznatljivost. Ova lopta omogućuje robotima da koriste različite senzore i kamere kako bi pratili kretanje lopte i usmjeravali se prema голу.

Ono što ovu kategoriju čini izazovnom i uzbudljivom jest činjenica da su roboti suočeni s vrlo dinamičnim okruženjem. U Soccer Lightweight kategoriji, gdje se koristi IR lopta, lopta i roboti stalno mijenjaju pozicije, pa roboti moraju brzo analizirati situaciju, donositi odluke i prilagoditi svoje kretanje. U Soccer Open kategoriji, gdje se koristi narančasta lopta, izazov je prilagoditi

algoritme tako da roboti mogu precizno prepoznati loptu unatoč mogućim smetnjama u okolini i promjenama u osvjetljenju. Kategorija koju sam izabrao za razvoj i implementaciju robota je Soccer Lightweight budući da je sama liga dinamičnija i gledljivija promatraču.

Jedan od najvećih tehničkih izazova u obje potkategorije Soccer natjecanja je razvoj učinkovitih algoritama za prepoznavanje i praćenje lopte, kao i za navigaciju terenom. Roboti moraju biti opremljeni sensorima koji omogućuju prepoznavanje različitih elemenata na terenu, uključujući loptu, golove i granice terena. Ovi senzori, često kombinacija infracrvenih senzora, kamera i drugih detektora, pružaju podatke potrebne za preciznu kontrolu robota.

Programiranje igra ključnu ulogu u uspjehu tima. Potrebno je razviti softver koji omogućava robotima da donose brze i točne odluke u stvarnom vremenu. To uključuje algoritme za prepoznavanje objekata, izbjegavanje prepreka, strategije napada i obrane, te timsku koordinaciju. Uspješan robot mora biti sposoban ne samo prepoznati loptu i kretati se prema голу, već i surađivati s drugim robotom iz svog tima, što zahtijeva sofisticirano programiranje i temeljito testiranje.

RoboCupJunior Soccer nije samo test tehničkih vještina; to je i test timskog rada i strategije. Potrebno je blisko surađivati kako bi se dizajnirali, izgradili i programirali roboti. Svaki član tima obično ima specifičnu ulogu, bilo da se radi o dizajnu hardvera, programiranju softvera ili analizi podataka sa senzora. Timska koordinacija i komunikacija su ključni, jer roboti moraju raditi zajedno kao jedinstvena cjelina tijekom igre.

Strategija je također ključni aspekt natjecanja. Potrebno je razviti plan igre, uzimajući u obzir snage i slabosti robota, kao i potencijalne taktike protivničkih timova. To može uključivati odluku o tome koji robot će biti zadužen za napad, a koji za obranu, ili kako će roboti reagirati u različitim situacijama na terenu.

Soccer kategorija nudi ogromne obrazovne koristi. Kroz sudjelovanje u ovom natjecanju, stječu se vrijedne vještine u programiranju, elektronici, mehanici i algoritamskom razmišljanju. Također se razvijaju vještine rješavanja problema, kritičkog mišljenja i timskog rada, koje su neprocjenjive u daljnjem obrazovanju i karijeri. Osim tehničkih vještina, Soccer kategorija potiče i sportski duh, gdje se natječe, ali i međusobno uči i surađuje. Natjecanja pružaju priliku za međunarodnu razmjenu ideja i kultura, što dodatno obogaćuje iskustvo sudionika. [3]

3.3. Pravila Soccer Lightweight lige

Budući da su pravila lige veoma opširna, odlučio sam izdvojiti nekoliko najbitnijih pravila kako bi ona stvorila sliku načina rada i igre samih automatiziranih robota. Ova kategorija koristi posebnu loptu koja emitira infracrveno (IR) svjetlo, što omogućuje robotima da detektiraju i prate loptu pomoću infracrvenih senzora. Pravila natjecanja su stroga, a njihova svrha je osigurati fer igru i poticati tehničku izvrsnost među sudionicima.



Slika 1. IR lopta korištena na RCJ natjecanjima

[Preuzeto sa: <https://www.robocupjunior.org.au/product/robocup-infrared-electronic-soccer-ball/>]

Roboti koji sudjeluju u Soccer Lightweight kategoriji ne smiju prelaziti dimenzije od 22 cm u širini i visini, dok maksimalna težina robota iznosi 1,4 kg. Ovo ograničenje osigurava da svi timovi imaju jednake uvjete te da se natjecanje fokusira na vještine programiranja i konstrukcije robota, a ne na fizičke prednosti. Jedno od ključnih pravila u ovoj kategoriji je zabrana IR interferencije. Komponente koje emitiraju infracrveno svjetlo, poput LiDAR-a ili IR LED-ova, nisu dopuštene, a materijali koji reflektiraju infracrveno svjetlo moraju biti minimalno vidljivi ili potpuno pokriveni. Ova mjera sprječava ometanje drugih robota tijekom igre i osigurava da natjecanje bude pošteno za sve timove. Također, maksimalni dozvoljeni napon na bilo kojem dijelu robota je 48V DC ili 25V AC RMS, što osigurava sigurnost robota i sprječava eventualne tehničke probleme tijekom utakmica. Roboti moraju biti pravilno zaštićeni, s čvrsto pričvršćenim dijelovima, a električne i pneumatske komponente moraju biti zaštićene od izravnog kontakta kako bi se spriječilo oštećenje tijekom igre.

Jedno od važnih pravila u vezi rukovanja loptom je da robot ne smije držati loptu. Držanje lopte definira se kao potpuno kontroliranje lopte bez mogućnosti pristupa drugih robota. Jedini izuzetak je upotreba rotirajućeg bubnja ("dribbler") koji može zadržati loptu na površini robota, ali bez potpunog ograničenja njezine slobode kretanja. Na svakoj utakmici prisutan je sudac koji nadzire tijek igre, donosi odluke o mogućim prekršajima i osigurava da se svi igrači pridržavaju pravila. Natjecanja se odvijaju u formatu 2 na 2, gdje dva tima, svaki s po dva robota, pokušavaju postići više golova od suparničkog tima unutar zadanog vremena. Utakmice su intenzivne, a strategija i timska koordinacija ključni su za postizanje uspjeha. Same utakmice odvijaju se unutar određenog terena na kojem se nalazi zeleni tepih, te je okružen crnim zidovima podignutim 22cm, zidovi imaju svrhu sprječavanja izlijetanja robota van terena, kao i sprječavanje vizualne interferencije kao što je to na primjer plava ili žuta majica, budući da su golovi žuti odnosno plavi. Također na zelenom terenu su iscrtane bijele linije koje označavaju aut liniju. Uz njih su nacrtani i golmanska područja odnosno tzv. šesnaesterci. U robotičkom nogometu robot ne smije izaći van tih linija, te se to smatra faulom i taj robot koji je izašao dobiva opomenu na način da ga se izbacuje iz igre na jednu minutu ili do sljedećeg gola. Utakmice se igraju dva puta po 10 minuta sa poluvremenom od 5 minuta.



Slika 2. Teren na kojem se igra Robotski nogomet

[Preuzeto sa: <https://grabcad.com/library/robocup-junior-soccer-field-2021-1>]

Zbog financijskih razloga, u svom radu koristio sam samo jednog robota, što znači da neću biti u mogućnosti održati pravu utakmicu, već je prikazan način igre robota usmjerenog prema jednom od golova. Ovaj robot detektira loptu, prati njezino kretanje i pokušava postići gol,

pružajući uvid u tehničke mogućnosti i algoritme. Robot kojeg sam napravio u potpunosti je osposobljen za sudjelovanje u takvoj ligi.

4. Materijali potrebni za rad

U ovom poglavlju prošao sam kroz proces razmišljanja nabavke svih materijala koji su mi potrebni za izradu samog robota. Svi materijali prikupljeni su u razgovoru sa predsjednikom udruge „Robofreak“ u namjeri kakobi kreirao što boljeg robota. Ovaj korak je veoma bitan kako bi se prikupili kompatibilni materijali, te kako se novac ne bi uzalud potrošio nekoliko puta. Također veoma je bitan i izbor samih materijala da budu kvalitetni, čvrsti, te na kraju krajeva da elektroničke komponente mogu izdržati pritisak igranja robotičkog nogometa 10 minuta po poluvrijemenu. Materijale koje sam nabavljao podijelio sam na nekoliko skupina a to su:

- Glavne komponente robota
- Senzori
- Ostali materijali potrebni za rad

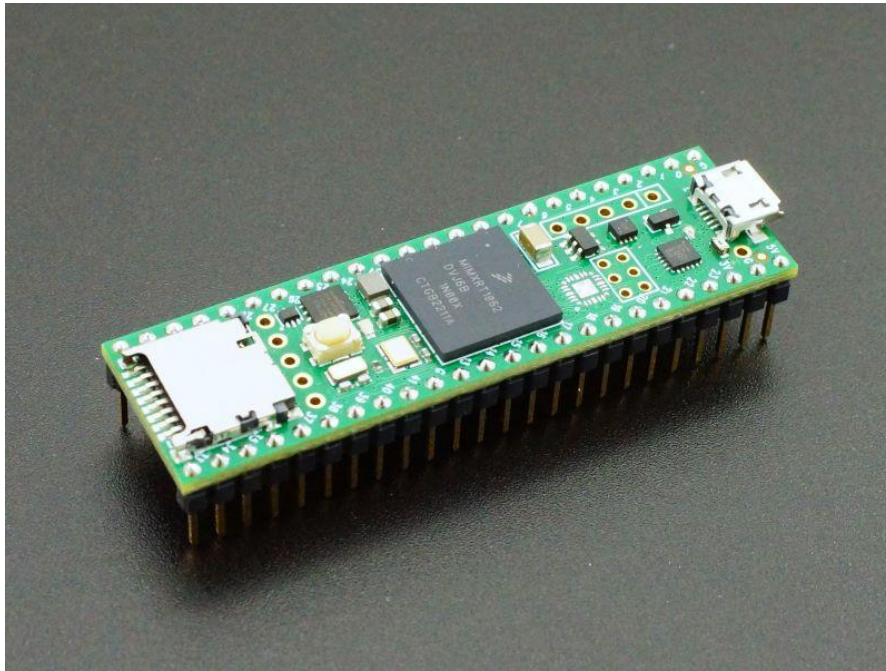
4.1. Glavne komponente robota

4.1.1. Teensy 4.1.

U svom radu koristio sam Teensy 4.1 mikrokontroler za izradu robota i mogu reći da je ovo pravi izbor za moj projekt. Teensy 4.1, kojeg je razvila kompanija PJRC, izuzetno je moćan zbog svoje visoke brzine i velikog kapaciteta memorije. Posebno je pogodan za složene projekte u robotici i automatizaciji, gdje je brza obrada podataka ključna.

Prije sam koristio Teensy 3.2, koji je također odličan, ali ima slabiji procesor i manje pinova u usporedbi s Teensy 4.1. Ovaj stariji model nije mogao podržati sve zahtjeve mog trenutnog projekta, posebno u pogledu brzine obrade i broja dostupnih ulazno/izlaznih pinova. Razmatrao sam i korištenje ESP32, koji je popularan zbog svoje integrirane Wi-Fi i Bluetooth funkcionalnosti. No, budući da sam se već upoznao s Teensy 4.1 u slobodno vrijeme i bio zadovoljan njegovim performansama, odlučio sam nastaviti s njim.

Teensy 4.1 ima 32-bitni ARM Cortex-M7 procesor koji radi na 600 MHz, 1 MB RAM-a i 8 MB flash memorije, što omogućava obradu složenih algoritama u stvarnom vremenu. Za moj projekt robota, to znači da mogu koristiti napredne algoritme za obradu podataka s različitim senzora, uključujući OpenMV H7+ kameru.



Slika 3. Teensy 4.1 Mikrokontroler

[Preuzeto sa: <https://protosupplies.com/product/teensy-4-1/>]

Teensy 4.1 nudi bogatstvo ulazno/izlaznih mogućnosti, kao što su USB, UART, I2C, SPI, i CAN sučelja, te 14 analognih i 26 digitalnih pinova. Na ploči se nalaze i PWM pinovi, putem kojih mogu kontrolirati motore, što je ključno za pokretanje robota. USB sučelje olakšava programiranje i debugging, dok su UART, I2C i SPI sučelja savršeni za komunikaciju s raznim sensorima i modulima. CAN sučelje je posebno korisno za sustave koji zahtijevaju otpornost na smetnje. U mom radu, UART konekcija se koristi za komunikaciju s OpenMV H7+ kamerom, omogućavajući prijenos video signala i podatke za obradu slike.[4]

Tokom godina prilikom rada u centru izvrsnosti Teensy 4.1 se pokazao kao izuzetno pouzdan i učinkovit, s visokom brzinom obrade i niskom latencijom. To je od presudne važnosti za aplikacije koje zahtijevaju brzu i preciznu kontrolu. Osim toga, cijena od oko 30 EUR čini ga pristupačnom opcijom za mnoge studente i hobiste. U mom slučaju, Teensy 4.1 se pokazao kao ključna komponenta koja je omogućila preciznu obradu podataka i učinkovitu komunikaciju između senzora, značajno poboljšavši performanse mog robotskog sustava.

4.1.2. OpenMV H7+

OpenMV H7+ kamera je napredni mikrokontrolerski modul s integriranom kamerom, dizajniran za zadatke računalnog vida. Ovaj uređaj je posebno koristan u projektima poput robotike, automatske inspekcije i prepoznavanja objekata. Ključne specifikacije uključuju ARM Cortex M7 procesor koji radi na 480 MHz, 32 MB RAM-a, 1 MB RAM-a i 2 MB flash memorije. Kamera koristi OV5640 senzor rezolucije 5 MP i podržava različita sučelja poput USB, I2C, SPI, UART i GPIO. Napajanje kamere varira između 3.3V i 5V, što je čini fleksibilnom za različite aplikacije.

OpenMV H7+ kamera pruža niz naprednih funkcionalnosti. Koristi ugrađene algoritme za prepoznavanje lica, QR kodova, linija, oblika i boja, te omogućuje detekciju i praćenje pokreta u stvarnom vremenu, kao i praćenje brzine i smjera kretanja objekata. Kamera se programira koristeći MicroPython, što omogućava brzo prototipiranje i razvoj aplikacija. Softversko sučelje je jednostavno i intuitivno, s knjižnicama koje podržavaju različite zadatke vizije.



Slika 4. Open MV Kamera

[Preuzeto sa: https://dfimg.dfrobot.com/store/data/DFR0833/DFR0833_564x376.jpg]

U mom projektu, OpenMV H7+ kamera je jedan od najbitnijih dijelova robota. Koristio sam je za snalaženje u okolini i lokalizaciju robota, te za ciljanje na gol pomoću prepoznavanja boja. Kamera je montirana na robota i programirana da prepozna određene boje koje označavaju različite ciljeve i prepreke na terenu. Na primjer, prepoznavanje boje lopte i golova omogućuje

robotu da se orijentira i donosi odluke o smjeru kretanja i šutiranju. U programu algoritam za prepoznavanje golova omogućuje robotu da precizno cilja prema голу. Dodatno, nabavio sam male produžne kablčice za OpenMV kameru koji su mi omogućili fleksibilno postavljanje objektiva na bilo koje mjesto na robotu, što je značajno unaprijedilo mogućnost prilagodbe i učinkovitost sustava. Različiti testovi su pokazali da je OpenMV H7+ kamera vrlo učinkovita u detekciji i praćenju objekata. Performanse su testirane u različitim uvjetima osvjetljenja i s različitim vrstama objekata. Na primjer, prepoznavanje lica je testirano u stvarnom vremenu s brzinom do 30 FPS, što pokazuje visoku efikasnost ovog uređaja.

OpenMV H7+ kamera košta otprilike 70-ak eura, što je čini pristupačnom opcijom za mnoge hobiste i inženjere. Postoje i druge alternative na tržištu, kao što su Raspberry Pi Camera Module koja nudi manju fleksibilnost za zadatke računalnog vida, ali veću procesorsku snagu; Pixy2 CMUcam5 koja je specijalizirana za brzu detekciju objekata, ali je ograničena u mogućnostima prilagodbe i programiranja; te JeVois Smart Camera koja je slična OpenMV kameri po funkcionalnostima, ali nešto skuplja i kompleksnija za korištenje.

OpenMV H7+ kamera je moćan alat za zadatke računalnog vida, s visokom fleksibilnošću i jednostavnim softverskim sučeljem. Njene sposobnosti u prepoznavanju objekata, praćenju pokreta i optičkom toku čine je idealnom za projekte u robotici i automatskoj inspekciji. Zahvaljujući svojim naprednim funkcionalnostima i jednostavnom programiranju, ova kamera predstavlja izvrsnu opciju za inženjere i hobiste koji žele implementirati računalni vid u svojim projektima. U mojem projektu za lokalizaciju robota i ciljanje na gol, pokazala se kao nezamjenjiv alat koji je značajno unaprijedio performanse mog robota.[5]

4.1.3. Baterija

Za napajanje robota koristio sam TATTU LiPo bateriju kapaciteta 2300mAh i naponom od 12.5V (3S1P). Ova baterija omogućuje robotu neprekidno djelovanje tijekom jednog poluvremena, koje traje 10 minuta. Baterija je povezana s PCB-om (Printed Circuit Board), koji dalje vodi do dva step-down regulatora napona na 5V i 3.3V. Ovi regulatori napajaju sve ostale komponente na PCB-u, ili su povezani putem JSTXH konektora na samom PCB-u.

Izbor baterije bio je ključan jer sam odlučio koristiti vrlo jake motore koji mogu povući mnogo struje. Zato je bilo važno odabrati bateriju s visokom strujnom sposobnošću. TATTU 2300mAh 12.5V 75C 3S1P LiPo baterija pruža potrebnu snagu, jer njen kapacitet od 75C osigurava da ne dođe do prekida napajanja kada motori povuku veliku struju.



Slika 5. Tattu LiPo baterija

[Preuzeto sa:

https://www.gensace.de/media/catalog/product/cache/2/image/800x800/9df78eab33525d08d6e5fb8d27136e95/t/a/taa23003s75x6_7.jpg]

Ova baterija je ključna za osiguravanje trajanja rada robota tijekom sportskih događaja, gdje se igraju dva poluvremena. Njezina visoka gustoća energije i strujni kapacitet omogućuju robotu da izdrži intenzivnu aktivnost tijekom cijelog natjecanja.

Kod korištenja LiPo baterija poput TATTU 2300mAh 12.5V 75C 3S1P, vrlo važno je pridržavati se sigurnosnih mjera, kao što su ispravno punjenje, pražnjenje i skladištenje. Balansiranje stanica tijekom punjenja preporučuje se kako bi se osigurala ravnomjerna punjenost i produžila životnost baterije. Ova baterija pruža pouzdanost i performanse potrebne za dugotrajno djelovanje robota tijekom natjecanja, omogućujući mu da se nosi s zahtjevima igre i ostane funkcionalan tijekom cijelog događaja.[6]

4.1.4. Step-Down Regulatori

Na robotu sam koristio dva step-down regulatora, D36V50F3 i D36V50F5, za napajanje različitih komponenti. Ovi regulatori generiraju niže izlazne napone iz ulaznih napona do 50 V, a koriste se kao preklopni regulatori (poznati i kao switched-mode power supplies (SMPS) ili DC-to-DC pretvarači), što ih čini znatno učinkovitijima od linearnih regulatora napona, posebno kada je razlika između ulaznog i izlaznog napona velika.

D36V50F3 regulator je korišten za generiranje fiksnog izlaznog napona od 3.3V, dok je D36V50F5 regulator korišten za generiranje fiksnog izlaznog napona od 5V. Ovi regulatori mogu podržati kontinuiranu izlaznu struju od 2 A do 9 A, ovisno o ulaznom i izlaznom naponu. Generalno, dostupna izlazna struja je nešto veća za verzije s nižim naponom i smanjuje se kako se ulazni napon povećava.



Slika 6. Step-down regulator

[Preuzeto sa: <https://a.pololu->

[files.com/picture/OJ10722.1200.jpg?e868bd91bd502df78cdc68780e16bd93](https://a.pololu-files.com/picture/OJ10722.1200.jpg?e868bd91bd502df78cdc68780e16bd93)]

Oba regulatora imaju ugrađenu zaštitu od povratnog napona do 40 V, zaštitu od prenapona i podnapona na izlazu, zaštitu od prekomjerne struje i kratkog spoja, te funkciju termalnog

isključivanja koja pomaže spriječiti oštećenja od pregrijavanja. Funkcija mekog starta ograničava struju pri uključivanju i postupno povećava izlazni napon prilikom pokretanja.

D36V50F5 regulator, na primjer, ima ulazni napon od 5.5 V do 50 V i izlazni napon od 5 V s točnošću od 4%. Tipična maksimalna kontinuirana izlazna struja varira od 3.5 A do 8 A, ovisno o ulaznom naponu, izlaznom naponu i opterećenju. Tipična učinkovitost regulatora je između 80% i 95%, što ovisi o ulaznom naponu, izlaznom naponu i opterećenju. Regulator ima i "power good" izlaz koji ukazuje kada regulator ne može adekvatno održavati izlazni napon.

Ovi regulatori također imaju način rada za uštedu energije s ultrazvučnim radom, koji povećava učinkovitost pri niskom opterećenju smanjenjem frekvencije preklapanja, ali je održava iznad razine koju možemo čuti (20 kHz). Tipična struja mirovanja bez opterećenja je između 2 mA i 4 mA, a EN pin se može koristiti za isključivanje regulatora u stanje niske potrošnje struje, smanjujući struju mirovanja na otprilike 10 μ A do 20 μ A po voltu na VIN-u.

Ovi su regulatori ključni za napajanje robota, jer omogućuju preciznu i učinkovitu konverziju napona iz glavne baterije na potrebne razine napona za različite komponente, uključujući elektroniku upravljanja, senzore i druge kritične sustave.[7]

4.1.5. Motori

U svom radu koristio sam Joinmax Motore, model „JMP-BE-3561“, koji se pokazao kao idealan izbor za natjecateljske aplikacije zbog svoje visoke snage i pouzdanosti. Veoma je bitno napomenuti da sam Joinmax motore izabrao jer su bili „najjači“ i najbolji motori koje je udruga "Robofreak" imala u tom trenutku. Imao sam i opciju izabrati motore bez četkica, kao što je ML-R BLDC Motor 12 V, 16x36 mm, 1:107 (mrm-blbc16x36-107), kao i motore s četkicama, poput ML-R BDC Motor 12 V, 24 mm, 1:34, 175 RPM (mrm-bdc24-34). No ipak sam na kraju odlučio kako su Joinmax motori puno bolja opcija zbog veće snage.



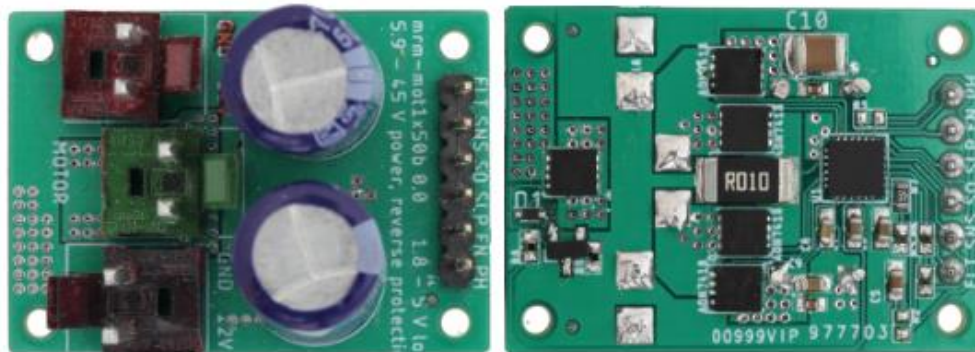
Slika 7. Joinmax motori

[Preuzeto sa: <https://www.javanelec.com/CustomAjax/GetAppDocument/ae6511dc-a918-4f0e-b3a0-fdf8293930d5?type=1&inlineName=True>]

Joinmax motor je dizajniran za zahtjevne uvjete u robotskim natjecanjima, pružajući potrebnu snagu i preciznost za složene zadatke. Standardna dvožična veza omogućava jednostavnu integraciju s visokoučinskim kontrolnim pločama, što je ključno za stabilan rad sustava. Motor radi na nazivnom naponu od DC 12V, pri čemu postiže brzinu bez opterećenja od $1700 \pm 8\%$ okretaja u minuti, uz struju bez opterećenja od 700 mA. Ove karakteristike osiguravaju brz i precizan rad robota, dok impresivan moment u mirovanju od 20 kgf.cm pri 12VDC omogućava izvrstan vučni kapacitet. Motor dolazi u kompletu s jednim visokoučinkovitim motorom i četiri para vijaka dimenzija 3*8 mm s maticama, što omogućava jednostavnu i sigurnu montažu. Ove specifikacije čine Joinmax Motor 3561 izvanrednim izborom za projekte koji zahtijevaju visoku pouzdanost i učinkovitost u dinamičkim uvjetima natjecanja.[8]

4.1.6. Kontroleri motora

U sklopu ovog rada koristim "ML-R Motor Driver 1x50A reverse protection (mrm-mot1x50r)" kontrolere motora. Ovi kontroleri jedini su dostupni na tržištu u Republici Hrvatskoj koji su dovoljno snažni da podrže zahtjeve Joinmax motora. Posebnost ovih kontrolera je zaštita od povratnog napona, koja je ključna za sprječavanje oštećenja ostalih komponenti robota uslijed povratnog napona jakih motora.



Slika 8. Kontroleri motora

[Preuzeto sa: https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-mot1x50r]

Kontroler motora podržava široki raspon ulaznih napona, od 5.9 V do 45 V, što omogućuje upotrebu različitih izvora napajanja, uključujući LiPo baterije. Za rad sa snažnijim motorima preporučuje se napajanje od 11.1 V ili više kako bi se osigurala stabilnost pri visokim opterećenjima. Elektronički je ograničena kontinuirana izlazna struja na 20 A, iako se fizički može dostići do 50 A uz odgovarajuće modifikacije. U kratkim trenucima može dostići vršnu izlaznu struju od 200 A, što je korisno prilikom pokretanja motora.

Osim zaštite od obrnutog ulaznog napona, kontroler uključuje zaštitu od podnapona, prekomjerne struje i pregrijavanja, čime se osigurava dugovječnost i sigurnost sustava. Kontrola brzine i smjera rotacije motora ostvaruje se putem Ph/En sučelja, koje koristi širinu pulsa (PWM) i digitalni ulaz, omogućujući preciznu kontrolu. U stanju hibernacije, kontroler motora troši minimalnu struju (9 μ A), što je bitno za produljenje trajanja baterije robota. Kontroler također posjeduje LED indikator za signalizaciju grešaka te pin za kontrolu grešaka. Koristi vanjski H-most s visokokvalitetnim tranzistorima koji podržavaju visoke struje (50/200 A), što djelomično objašnjava višu cijenu ovog uređaja. Iako ne podržava direktnu

komunikaciju s mikrokontrolerima niti mjerenje napona motora ili ulaze za enkodere, omogućuje mjerenje izlazne struje putem eksternih otpornika.[8]

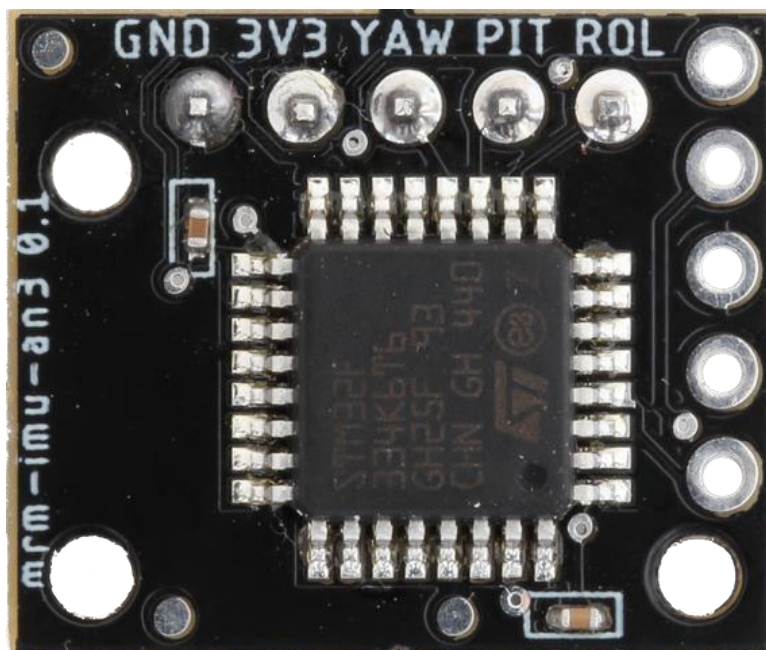
Sve u svemu, ML-R Motor Driver je robustan kontroler motora koji pruža napredne zaštitne funkcije, visoku izlaznu snagu i preciznu kontrolu, te je idealan za aplikacije koje zahtijevaju pouzdanost i visoke performanse.

4.2. Senzori

Senzori su uređaji ili komponente koji detektiraju i reagiraju na određene vrste ulaznih podražaja iz fizičkog okruženja. Ovi ulazi mogu uključivati različite vrste energija, poput svjetlosti, topline, vlage, tlaka, elektromagnetskih polja, zvuka, brzine i drugih. Kada senzori otkriju određeni podražaj, pretvaraju ga u električni signal koji može biti obrađen ili analiziran. Senzori su ključni u mnogim industrijskim, znanstvenim i potrošačkim aplikacijama, uključujući automatizaciju, robotiku, medicinsku dijagnostiku, meteorologiju i sigurnosne sustave. Na primjer, termometri su senzori koji mjere temperaturu, dok kamere služe kao senzori za snimanje slike ili videa. S razvojem tehnologije, senzori postaju sve sofisticiraniji i precizniji, omogućujući naprednu detekciju i analizu u realnom vremenu. Oni su neophodni za razvoj pametnih uređaja i sustava koji mogu reagirati na promjene u okolini. Isto tako su i neophodni za izradu robota koji će igrati nogomet. [9]

4.2.1. IMU Kompas

U prethodnim iskustvima s orijentacijskim uređajima nailazio sam na brojne izazove. U okviru ovog rada odlučio sam koristiti ML-R IMU senzor, koji se pokazao boljim u odnosu na BNO055 verziju senzora tvrtke Adafruit. Prednost ML-R IMU senzora leži u njegovoj robusnosti. Naime, prilikom korištenja BNO055 senzora često sam doživljavao gubitak i2c veze uslijed sudara. Smatram da je problem bio uzrokovan žicama koje su prilikom udarca izgubile dobar kontakt. Iako bi se ovaj problem mogao riješiti izradom PCB-a, odabrao sam sigurniju opciju korištenja ML-R IMU senzora. Ovaj senzor je također temeljen na Boschovom BNO055 čipu, ali nudi dodatne analogne izlaze za roll, yaw i pitch, što dodatno poboljšava njegovu funkcionalnost i pouzdanost u zahtjevnim uvjetima.



Slika 9. IMU senzor

[Preuzeto sa: <https://www.microline.hr/UploadedFilesmicpg/Images/mrm-imu-an3Back.png>]

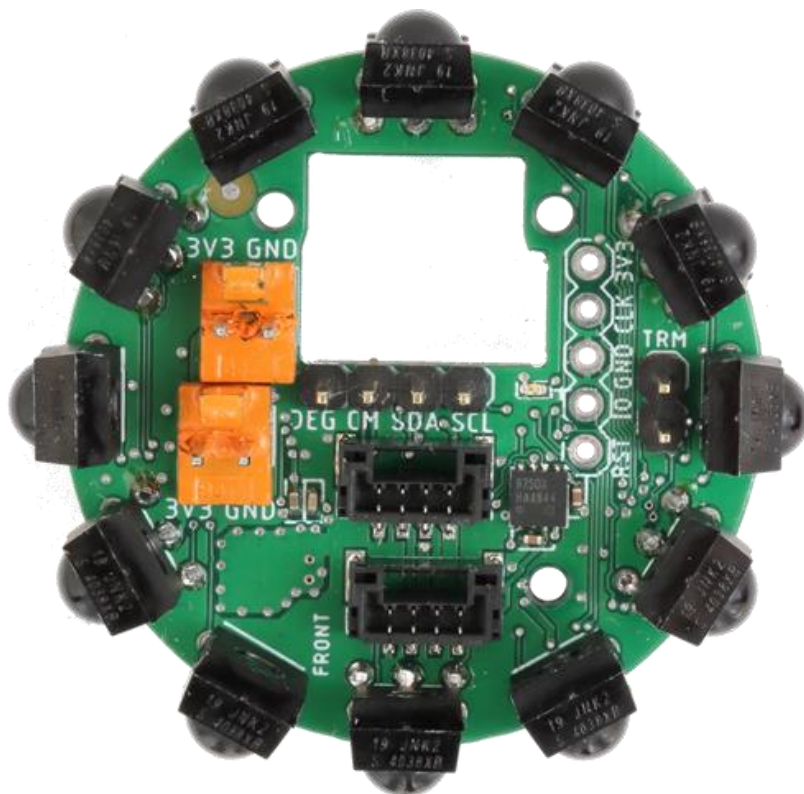
ML-R IMU senzor radi s ulaznim naponom od 3.3 V i ima logički napon također od 3.3 V. Senzor koristi sučelje s tri analogna izlaza, čiji raspon iznosi od 0 do 3.3 V, što ga čini kompatibilnim s mikrokontrolerima koji rade na 3.3 V i 5 V, uključujući Arduino platformu. Senzor omogućuje izlaz izračunatih veličina kao što su quaternion-i, Eulerovi kutevi, vektor rotacije, linearno ubrzanje, gravitacija te smjer iz naprednog troosnog 16-bitnog žiroskopa. Također uključuje vrhunski troosni 14-bitni akcelerometar i geomagnetski senzor. Pored toga, ML-R IMU senzor podržava modove štednje energije, kao što su normal, low power i suspend. Radni opseg akcelerometra kreće se od $\pm 2g$ do $\pm 16g$, dok žiroskop ima radni opseg od $\pm 125^\circ/s$ do $\pm 2000^\circ/s$. Magnetometar ima radni opseg magnetskog polja tipično $\pm 1300\mu T$ za x- i y-osi te $\pm 2500\mu T$ za z-os, s rezolucijom od približno $0.3\mu T$.

Senzor je opremljen Boschovim BNO055 čipom koji uključuje 32-bitni ARM Cortex M0+ mikrokontroler, čime se eliminira potreba za dodatnim računanjem. Osim toga, dodatno je integriran 32-bitni ARM Cortex M4 mikrokontroler. Senzor također uključuje zaštitu od obrnutog ulaznog napona i omogućuje nadogradnju firmwarea, pružajući dodatnu fleksibilnost i sigurnost. Nije potrebna plug and play funkcionalnost, što pojednostavljuje implementaciju senzora u različite sustave.[10]

Sve ove karakteristike čine ML-R IMU senzor idealnim izborom za precizno i pouzdano mjerenje orijentacije u robotici i drugim naprednim aplikacijama.

4.2.2. Senzor za loptu

Za praćenje infracrvene lopte u sklopu ovog rada koristio sam IR ball finder 3 senzor, koji se pokazao kao izvanredno rješenje za određivanje kuta i udaljenosti lopte. MRMS IR ball finder 3 nudi sučelje putem CAN Bus-a, I2C-a i analognih izlaza, što ga čini vrlo fleksibilnim za integraciju u različite sustave (mrm-ir-finder3).



Slika 10. Senzor za loptu

[Preuzeto sa: https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-ir-finder3]

Ovaj senzor je posebno razvijen za detekciju infracrvenih lopti u RoboCup nogometu, pružajući pokrivenost od 360 stupnjeva. Ključna prednost ovog senzora je njegova sposobnost ublažavanja problema povezanih s detekcijom bliskih izvora IC zračenja, što je čest problem kod standardnih senzora koji mogu primiti reflektirano infracrveno svjetlo od bliskih objekata, uzrokujući netočna ili fluktuirajuća mjerenja. IR ball finder 3 može istovremeno detektirati udaljene izvore do oko 10 metara, ovisno o vrsti izvora, što ga čini idealnim za upotrebu u RoboCup Soccer Lightweight Soccer natjecanjima. Ovaj senzor nije osjetljiv na smetnje koje mogu uzrokovati protivnički roboti, ljudi oko arene, golovi, zidovi ili podloga. Vrijeme integracije ovog senzora je manje od 3 ms, što mu omogućuje da postigne do 350 mjerenja u sekundi.

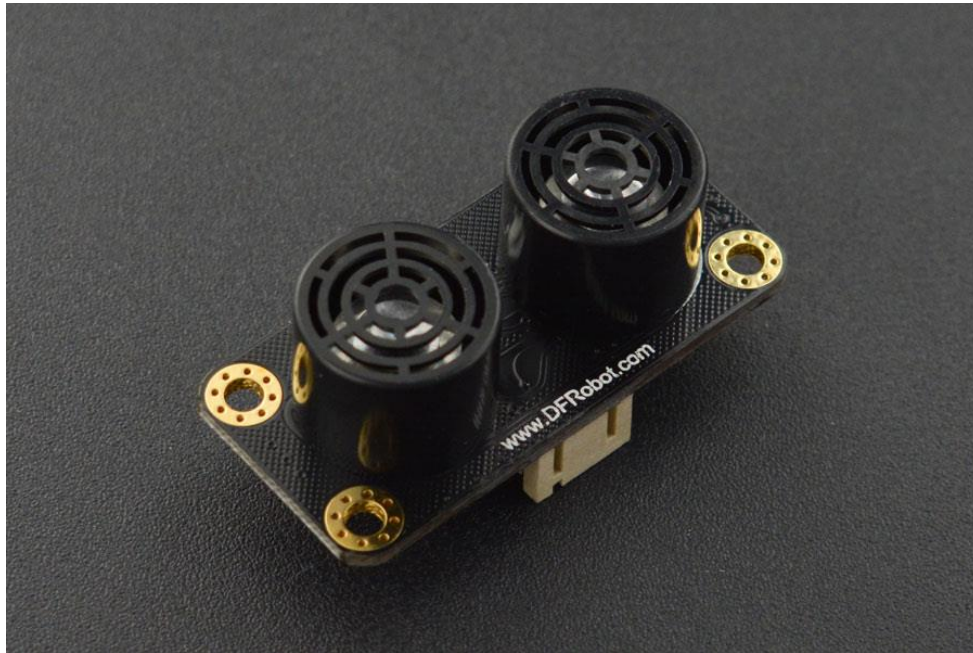
Podaci se šalju nešto rjeđe kako bi se izbjeglo preopterećenje sabirnice. Cijena senzora iznosi 28,96 € s uključenim PDV-om.

Osobine MRMS IR ball finder 3 senzora uključuju ulazni napon od 3.3 V, sučelje putem CAN Bus-a, dva analogna izlaza (za kut i udaljenost) i I2C. Senzor koristi 18 prijemnika za detekciju infracrvenog zračenja i omogućuje nadogradnju firmwarea, čime se pruža dodatna fleksibilnost. Senzor je također opremljen zaštitom od obrnutog ulaznog napona i LED statusnim indikatorom. Iako nije standardiziran formatom, senzor nudi plug and play funkcionalnost, što olakšava njegovo uključivanje u sustav. Za spajanje senzora, mogu se koristiti gotovi kablovi poput ML-R Cable KK254-KK254 10 cm (mrm-kk2.54-2.54-10) ako se napajanje uzima iz MRMS komponenti. Alternativno, mogu se koristiti standardni Dupont 0.1" priključci koji pristaju u Molex KK 254 konektore. Za izradu prilagođenih kablova, dostupan je set priključaka 0xKK396 + 2xKK254 (mrm-con0-2). Također postoji mogućnost korištenja standardnih MRMS kablova za CAN Bus, poput MRMS CAN Bus cable 10 cm (mrm-jst-can10), ili izrade prilagođenog kabela pomoću Adapter cable CAN Bus - 4 x Dupont (mrm-can-dup20) za spajanje na druge ploče. Ove opcije omogućuju povezivanje MRMS CAN Bus-a s standardnim Dupont 0.1" pinovima.

Postoje tri glavna načina spajanja senzora na sustav: preko analognog sučelja, I2C sučelja ili CAN Bus-a. Način koji ću ja koristiti je putem analognog sučelja iz razloga što je najjednostavniji za koristiti.[11]

4.2.3. Senzori udaljenosti

Kako bi se robot što bolje snalazio u prostoru, koristio sam ultrazvučne senzore udaljenosti. Prema pravilima igre, zidovi se nalaze na 10 cm od bijele linije, te se ti zidovi mogu koristiti kao referentne točke koje se mogu očitati pomoću senzora udaljenosti. Planirao sam razviti program koji će usporavati robota kada se približi zidu s lijeve ili desne strane, kako bi lakše detektirao liniju i izbjegao sudare.



Slika 11. Ultrazvučni senzor

[Preuzeto sa: <https://www.dfrobot.com/product-1862.html>]

Ultrazvučni senzori koje sam koristio su DF Robot URM09 analogni senzori. Ovi senzori mjere udaljenost do objekta koristeći zvučne valove. Princip rada senzora temelji se na slanju zvučnog vala na specifičnoj frekvenciji i praćenju vremena povratka odjeka tog vala. Budući da zvuk putuje kroz zrak brzinom od otprilike 344 m/s, vrijeme povratka zvučnog vala može se pomnožiti s 344 metra kako bi se dobila ukupna udaljenost puta zvučnog vala. Ova ukupna udaljenost uključuje dvostruku udaljenost do objekta jer zvučni val putuje od senzora do objekta i natrag do senzora. Za određivanje stvarne udaljenosti do objekta, potrebno je ukupnu udaljenost podijeliti s dva. DFRobot URM09 ultrazvučni senzor je posebno dizajniran za brzo mjerenje udaljenosti i izbjegavanje prepreka. Njegova frekvencija mjerenja može doseći do 30Hz, a senzor posjeduje ugrađenu temperaturnu kompenzaciju i analogni izlaz. Omogućuje precizno mjerenje udaljenosti unutar raspona od 2 cm do 500 cm. Ovaj senzor je kompatibilan s Arduino, Raspberry Pi i drugim glavnim kontrolnim pločama koje podržavaju logičke razine od 3.3V do 5V. Tehničke specifikacije DFRobot URM09 senzora uključuju napajanje od 3.3V do 5.5V DC, uz radnu struju od 20mA. Senzor može raditi u temperaturnom rasponu od -10°C do +70°C. Mjerenje udaljenosti moguće je u rasponu od 2 cm do 500 cm, s rezolucijom od 1 cm i točnošću od 1%. Maksimalna frekvencija mjerenja iznosi 30Hz. Dimenzije senzora su 47 mm x 22 mm.[12]

Ovi senzori su se pokazali iznimno korisnima za precizno očitavanje udaljenosti i omogućili su sigurno kretanje robota unutar definiranih granica terena.

4.2.4. Barijera

ML-R Light Barrier (mrm-barr) je uređaj koji detektira prekid svjetlosti koristeći zeleni LED kao izvor svjetlosti. Radi na ulaznom naponu od 3.3 V, dok je izlazni signal analogan i varira od 0 V do ulaznog napona. Opremljen je standardnim Dupont 0.1" konektorima i dolazi s kabelima duljine 25 cm. Uređaj ima jedan prijemnik i predajnik, koristi zeleno svjetlo, te nema potrebu za dogradnjom firmwarea jer nema čip. Zaštita od obrnutog ulaznog napona nije potrebna, a uređaj nije u standardiziranom formatu. LED svijetli zeleno, a uređaj je namijenjen za detekciju prekida svjetlosti, što je korisno u aplikacijama poput Robocup Soccer natjecanja.

Spajanje uključuje LED s crnom žicom na GND (0 V) i crvenom na 3.3 V (LED radi i na nižem naponu, ali s manjim intenzitetom), te fototranzistor s crnom žicom na GND (0 V), crvenom na 3.3 V, i bijelom žicom kao izlazom. Kao što se to može vidjeti na slici ispod. Cijena ovog uređaja iznosi 7,21 €.



Slika 12. Barijera i LED dioda

[Preuzeto sa: https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-barr]

Koristio sam barijeru na način da sam je postavio u kružno područje robota, koje je namijenjeno za prihvatiti loptu. Svaki put kada se svjetlost prekine, odnosno signal na barijeri se prekine, robot automatski puca loptu prema голу pomoću solenoida. [13]

4.2.5. Mehanizam za udaranje lopte

U okviru robotičkog sustava za natjecanja poput Robocup Soccer-a, mehanizam za udarac lopte, često nazivan "kicker," predstavlja ključnu komponentu za efikasnu manipulaciju loptom. U ovom radu, za kicker sam koristio solenoid ZMF 2551, poznat po svojoj pouzdanosti i učinkovitosti u takvim aplikacijama.

Solenoid ZMF 2551 je cilindrični magnet dizajniran za jednostavnu elektroničku kontrolu, što ga čini idealnim za robotičke aplikacije. Ovaj solenoid ima nazivnu snagu od 10 W, s maksimalnim hodom jezgre od 17 mm pri 100% napajanju, pri čemu generira silu veću od 25 N. Pri smanjenom napajanju od 10%, solenoid zadržava isti hod od 17 mm, ali generira veću silu, veću od 40 N. Solenoid dolazi s ugrađenom povratnom oprugom koja omogućava povrat jezgre u početni položaj, s opružnim hodom od oko 17 mm. Vanjska duljina hoda pritiskajućeg štapa iznosi također 17 mm, dok je magnetski moguć hod približno 28 mm.



Slika 13. Solenoid

[Preuzeto sa: <https://tremba.de/zylindermagnete/zylindermagnete.php>]

ZMF 2551 je posebno prilagođen za primjene gdje je potrebna robusnost i otpornost na stres, što ga čini pogodnim kao šutni magnet u robotičkim sustavima. Solenoid je dizajniran za "push" funkciju, što znači da se koristi za udaranje ili guranje predmeta, u ovom slučaju lopte. Njegova težina iznosi 180 g, a promjer mu je 25 mm, što omogućava kompaktno ugrađivanje u razne robotičke platforme. Tip priključka je žičana litza, a solenoid dolazi s povratnom oprugom, što osigurava automatski povrat u početni položaj nakon svakog udarca. Predviđena je mogućnost preopterećenja, što dodatno povećava sigurnost u radu. Za pričvršćivanje solenoida koristi se prednji navoj s priloženom matice za montažu.

Cijena solenoida ZMF 2551 iznosi oko 20,00 €, što ga čini ekonomičnim izborom za studentske projekte i istraživačke primjene. Njegova jednostavna konstrukcija i pouzdanost čine ga optimalnim izborom za mehanizme koji zahtijevaju preciznu kontrolu udarca, poput kicker sustava u robotičkom nogometu. Jedina mana ovog solenoida je što je dosta težak i zauzima dosta prostora što ga čini nepraktičnijim u primjeni kod kategorija koje su ograničene težinom, no unatoč tome uspio sam ga implementirati u ovog robota. [14]

4.3. Ostali potrebni materijali

Među ostalim materijalima koji su bili potrebni za rad a koristio sam ih pri izradi robota se nalaze JST-XH konektori koje sam koristio kako bi povezivao PCB pločice sa sensorima ili na primjer kontrolerima motora. Također koristio sam i jednostavne senzore dodira koji su zalemljeni direktno na pločici, te reagiraju na pritisak. Isto tako za uključivanje i isključivanje robota koristio sam I/O switch odnosno sklopku. Neke od senzora nisam želio odlemljivati kako bi ih spojio na PCB pločice, te sam ih povezao običnim dupont vodičima koji mogu biti u obliku muško muški, te muško ženski. Uz te materijale dolaze također i alati koje sam koristio za izradu, a to su lemilica, kliješta, odvijači i krimpalica. Krimpalica je potrebna iz razloga što se svi JST-XH konektori moraju ručno izraditi na način da se na vodiču otkine izolacijski sloj, te se taj vodič „zakrimpa“. Nakon „krimpanja“ taj vodič se postavlja u JST-XH konektor koji je tad spreman za korištenje. Odvijači i kliješta su mi potrebni kako bi pričvrstio sve printane dijelove zajedno. Neki od tih dijelova su povezani odstojećima koji se također nalaze među tim ostalim potrebnim materijalima, kao i vijci veličine M2 i M3 zajedno sa maticama. Svi odstojećici su veličine M3. Jedan od glavnih dijelova izrade PCB pločice je lemljenje budući da kad se pločica naruči ona dolazi potpuno prazna bez elektroničkih elemenata na njoj. Stoga sam koristio lemlicu i lem kako bih sve elemente povezao u pravilnoj formaciji na samu PCB pločicu.[15][16][17]

5. Izrada elektroničke pločice (PCB-a)

PCB pločice (Printed Circuit Board) su fizička osnova za elektroničke komponente i spojeve u gotovo svim modernim elektroničkim uređajima. Pločice su izrađene od nevodljivog materijala, najčešće fibreglasa, na kojem su tiskani tanki slojevi bakra koji tvore električne vodove. Ovi vodovi povezuju komponente poput otpornika, kondenzatora, tranzistora i integriranih krugova, omogućujući prijenos električnih signala između njih. PCB pločice dolaze u različitim slojevima, uključujući jednostrane, dvostrane i višeslojne varijante, ovisno o složenosti dizajna. Zahvaljujući PCB pločicama, proizvodnja elektronike postala je brža, jeftinija i učinkovitija jer eliminiraju potrebu za ručnim ožičavanjem svakog spoja.[18]

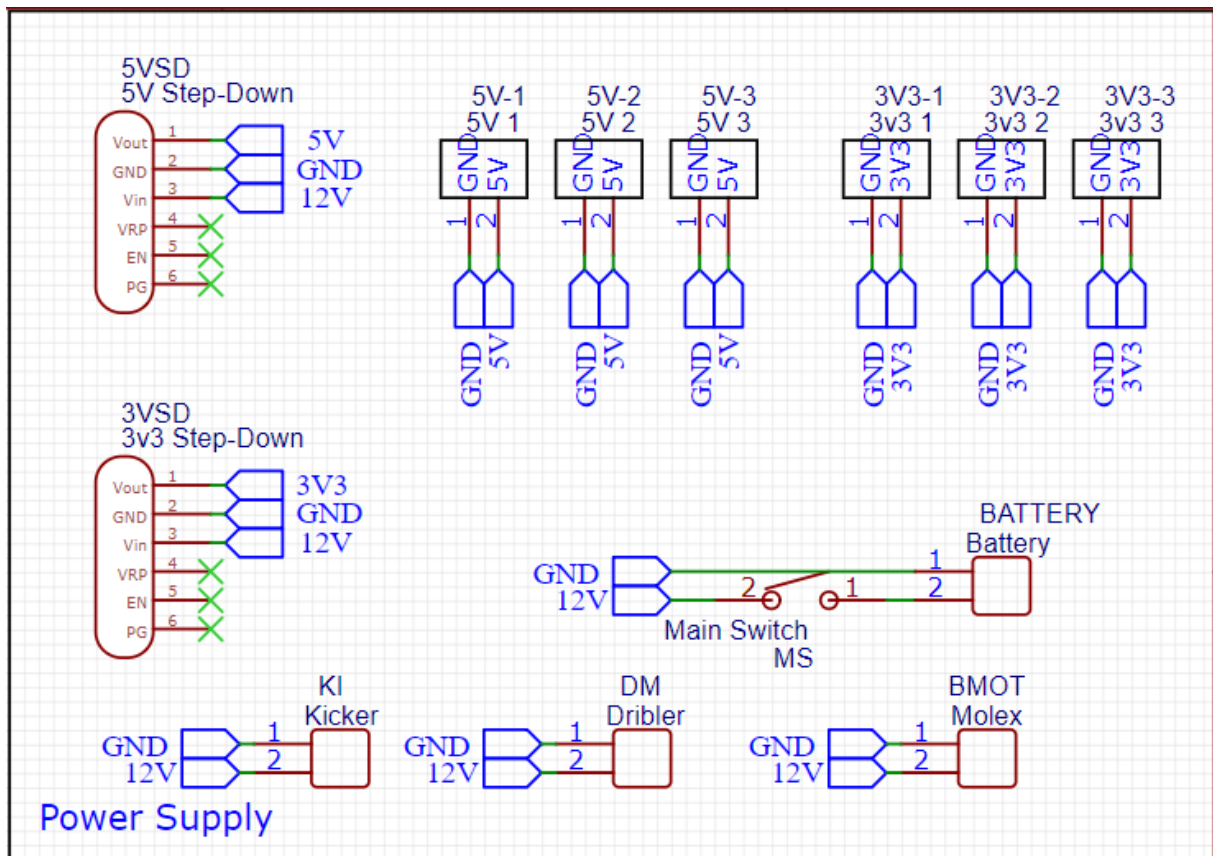
Potreba u izradi elektroničke pločice proizlazi iz toga što se u prijašnjim godinama učenicima često događalo da su se žice odspajale, što je često dovodilo do prestanka rada robota ili čak i do toga da je robota zahvatio plamen zbog kratkog spoja. Stoga kako bi se što manje smanjila mogućnost takvih spontanih događaja veoma je pametno zamijeniti vodove jednom takozvanom „matičnom pločom“ na koju se spaja većina Elektroničkih komponenata. Na ovom odlučio sam izraditi dvije PCB pločice, jednu senzorsku koja služi za detektiranje linije i jednu glavnu koja služi za povezivanje svih elektroničkih komponenata. Naravno, neke elektroničke elemente sam izostavio iz sigurnosnih razloga sa PCB pločice. Jedan od tih elemenata su kontroleri motora. Kontrolori motora često znaju povući jako puno struje što zna izazivati skokove u naponu, isto tako velike struje prolaze kroz njih što znači da stvaraju magnetno polje koje može smetati kompasu, a također oni imaju i najveće šanse „izgorjeti“ zbog povratnog napona motora budući da se u nogometu smjerovi kretanja motora vrlo naglo mijenjaju. Iz toga sam razloga odlučio njih izostaviti sa glavne PCB pločice, te ih povezati žicama na pločicu. Isto tako sam još nekoliko senzora odlučio izostaviti što ću objasniti u daljnjem tekstu. Same pločice sam izradio u EasyEDA programu, te sam naručio pločice sa JLCPCB web stranice od istoimene kompanije.

5.1. Izrada pločica u programu EasyEDA

EasyEDA je besplatan alat za dizajn elektroničkih sklopova i tiskarskih pločica (PCB) koji se koristi online putem web preglednika. Omogućuje korisnicima da jednostavno kreiraju i simuliraju elektroničke sheme te dizajniraju složene PCB-ove. EasyEDA nudi intuitivno sučelje s opsežnom bibliotekom komponenti, što olakšava proces dizajniranja čak i onima s manje iskustva. Osim dizajna, program omogućuje i online naručivanje izrađenih PCB-ova, čime je proces razvoja elektroničkih projekata ubrzan i pojednostavljen. Pogodan je za hobiste, studente, ali i profesionalne inženjere.[19]

5.1.1. Izrada Glavne PCB pločice

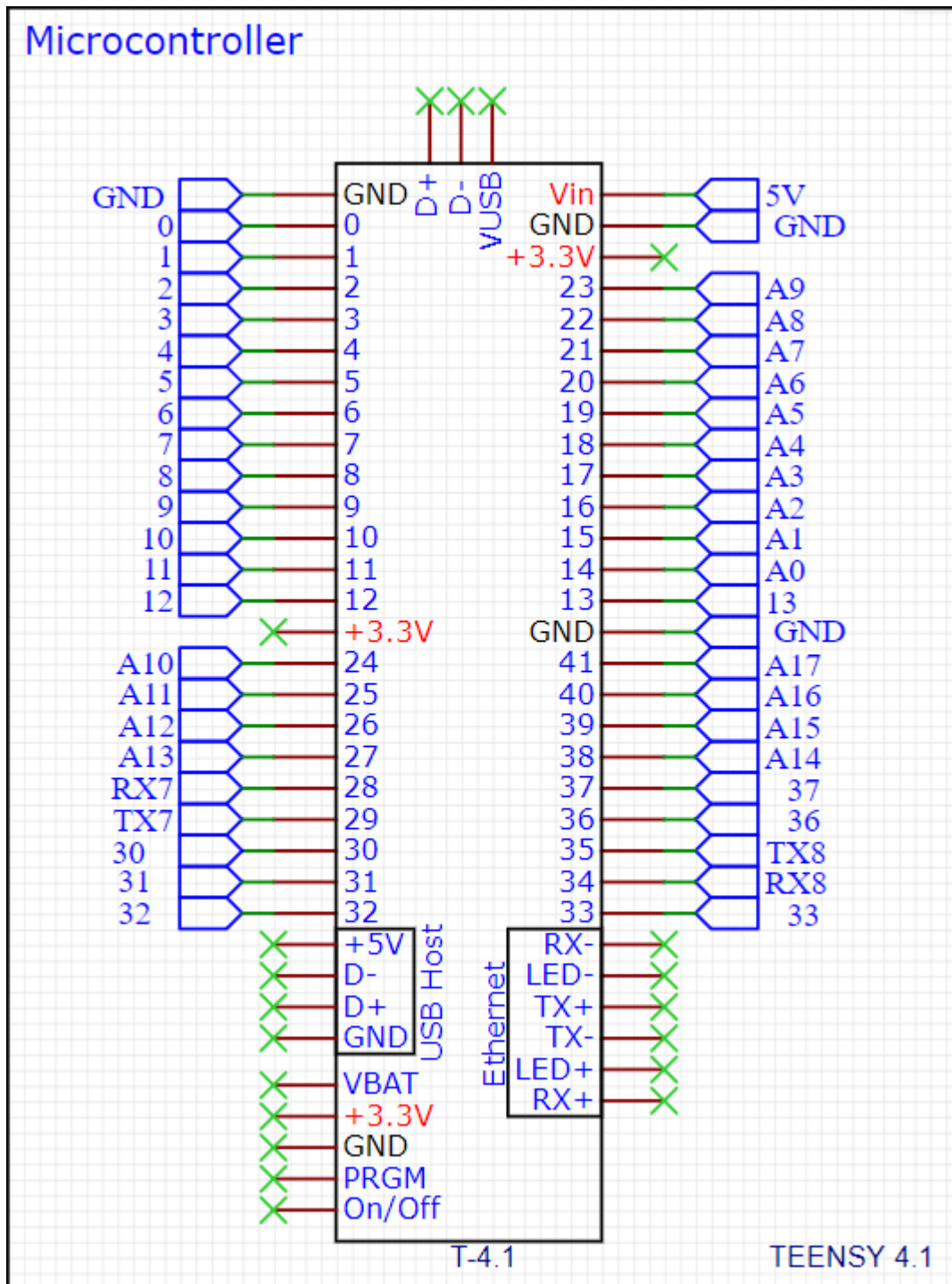
Glavnu pločicu svoga robota podijelio sam na nekoliko dijelova radi preglednosti. Izrađivati glavnu pločicu krenuo sam sa povezivanjem elemenata koji služe za napajanje, te sam tako kreirao glavni ulaz sa baterije pod nazivom BAT IN gdje 12V sa baterije ide na ostatak pločice preko jednog glavnog tipkala. Dalje sam ostavio nekoliko utora od 12V za tzv. kicker i dribbler. I ono najbitnije 12V sam spojio na step-down pretvornike koji spuštaju napon na 5V i 3.3V kako bi se mogle opskrbiti sve elektroničke komponente koje se nalaze na pločici. Ostavio sam i tri prazna 3.3V pinova, kao i tri 5V pinova kako bi se u slučaju potrebe moglo opskrbiti neke dodatne komponente. Ovaj dio pločice nazvao sam Power Supply budući da opskrbljuje pločicu strujom.



Slika 14. Power Supply dio PCB pločice

[Iz vlastite izrade]

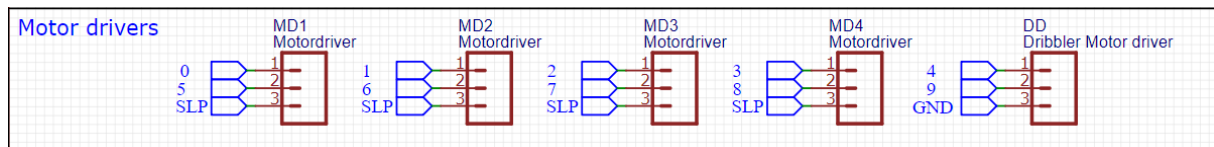
Sljedeći dio pločice je Microcontroller dio gdje se nalazi Teensy 4.1. Na teensy sam pažljivo spojio sve outpute prema schemi sa njihove web stranice za taj uređaj kako bih znao koji pin mi za šta služi prema načinu konekcije na taj pin.



Slika 15. Teensy 4.1. pinout na PCB pločici

[Iz vlastite izrade]

Sljedeći na redu dio je bio dio Motor drivers, gdje sam ostavio JST-XH utore kako bi kasnije mogli tzv. „krimpanjem“ žica izraditi kablčice kojima ćemo povezati pinove kontrole motordriversa sa pinovima na Teensy 4.1. mikrokontroleru.

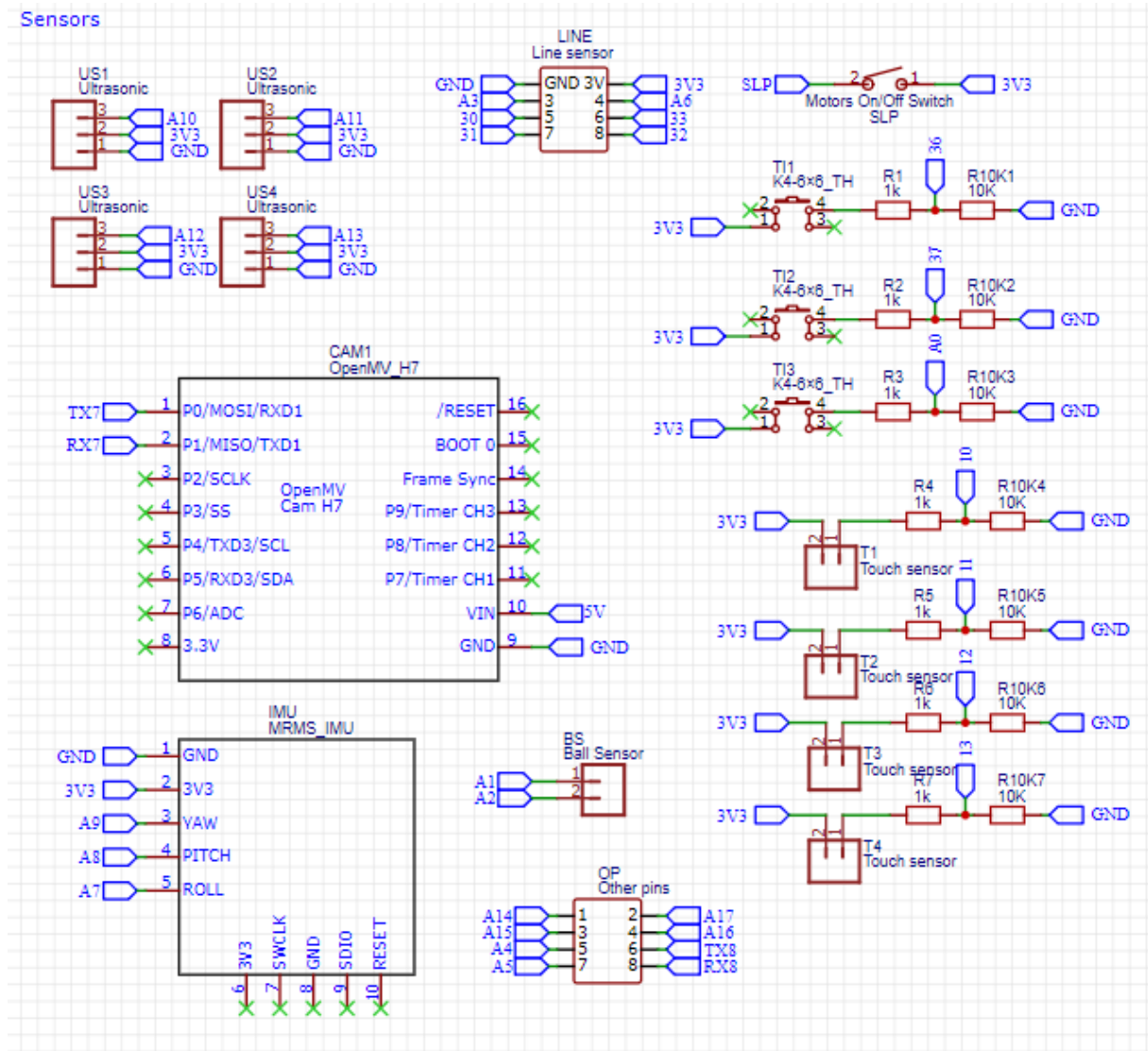


Slika 16. Pinout motor drivera na PCB pločici

[Iz vlastite izrade]

Na kraju nam je ostao najveći dio, a to je dio sa sensorima i kamerom. U tom dijelu je zasebno objašnjena povezanost svakog od elementa po redu kao i njihov način spajanja. Prvo imamo 4 JST-Xh konektora pod nazivom US1,US2,US3 i US4 što označava četiri ultrazvučna senzora. Na robotu su se na kraju koristila 3 ultrazvučna senzora, no ostavio sam i mogućnost dodavanja četvrtog. Ultrazvučni senzori su spojeni na način da su dvije žice spojene na napajanje odnosno na 3.3V i GND pinove, dok je treći pin povezan na Analogne pinove teensy 4.1. kontrolera. Nakon toga imamo Line sensor koji se također spaja na 3v3 i GND, te su mu potrebna 4 digitalna pina i 2 analogna pina. Kamera je spojena putem UART veze gdje se koriste RX7 odnosno TX/ kanali. Kameri je također potrebno i napajanje od 5V, kao i GND pin za uzemljenje. Nakon kamere imamo i kompas odnosno MRMS IMU sensor kojemu je potrebno napajanje od 3.3V i uzemljenje, te svoje signale šalje na tri analogna pina teensy kontrolera. Ostavio sam i dva analogna pina za ball sensor od kojih jedan služi za udaljenost a drugi služi za kut lopte. Također na pločici se nalazi i 8 switcheva, od kojih je jedan spojen direktno na pin od motordrivera, te služi za prekidanje rada motora u slučaju nužde, dok je ostalih 7 spojeno pomoću otpornika od 10kOhma i 1kOhma na digitalne pinove teensy 4.1. kontrolera kako bi se mogao očitavati signal. Ostatak pinova koji su slobodni sam posebno odvojio i označio u slučaju da je potrebno dodati neki sensor naknadno.

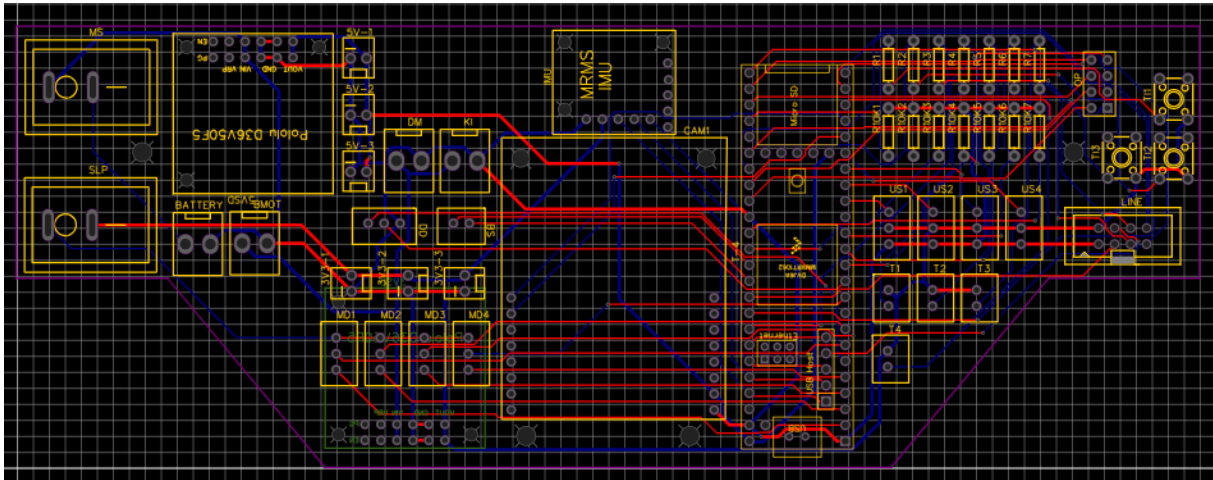
Sensors



Slika 17. Pinout senzora na PCB pločici

[Iz vlastite izrade]

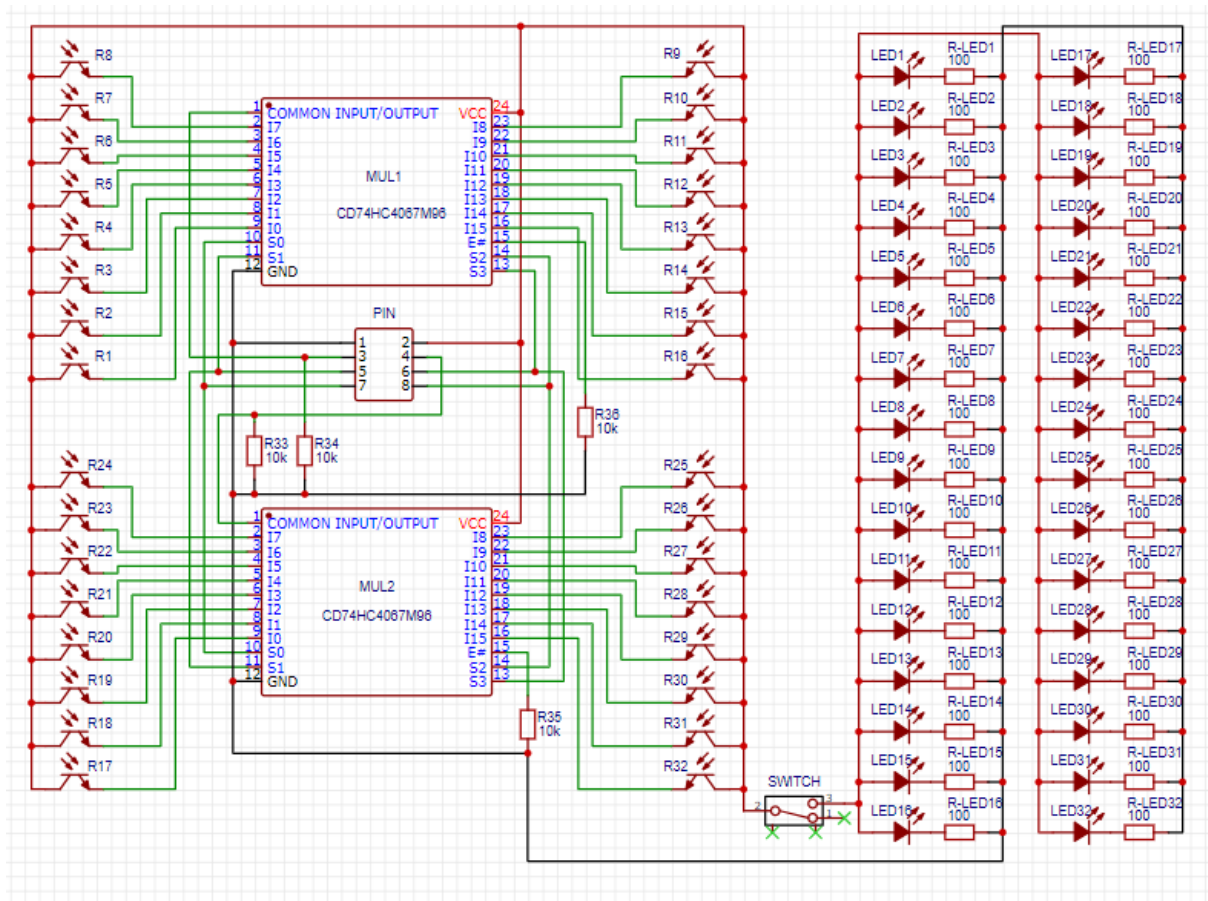
Nakon izrade scheme PCB pločice, krenuo sam u dizajniranje same pločice fizički, te sam odredio neke rupe kako bi pločica mogla biti kompatibilna sa robotom u onom trenutku kada ću dizajnirati robota. Samu pločicu sam dizajnirao tako da tipkala i gumbi budu dostupni sa strane, dok se u sredini nalazi kamera kako bi je kasnije mogao lakše iskoristiti u montaži za konusni pogled od 360 stupnjeva, te kompas kako bi bio što više u središtu robota, kako bi očitavao vrijednosti centralne osi okreta samog robota. Ostatak pločice je organiziran tako da vodovi što manje smetaju kompasu zbog magnetnih sila. Za povezivanje vodiča sam koristio tzv. autoRoute opciju iz EasyEDA programa uz nekoliko dodatnih naredba koje povećavaju vodiče koji provode struju odnosno sve izvođe iz power supplya-a.



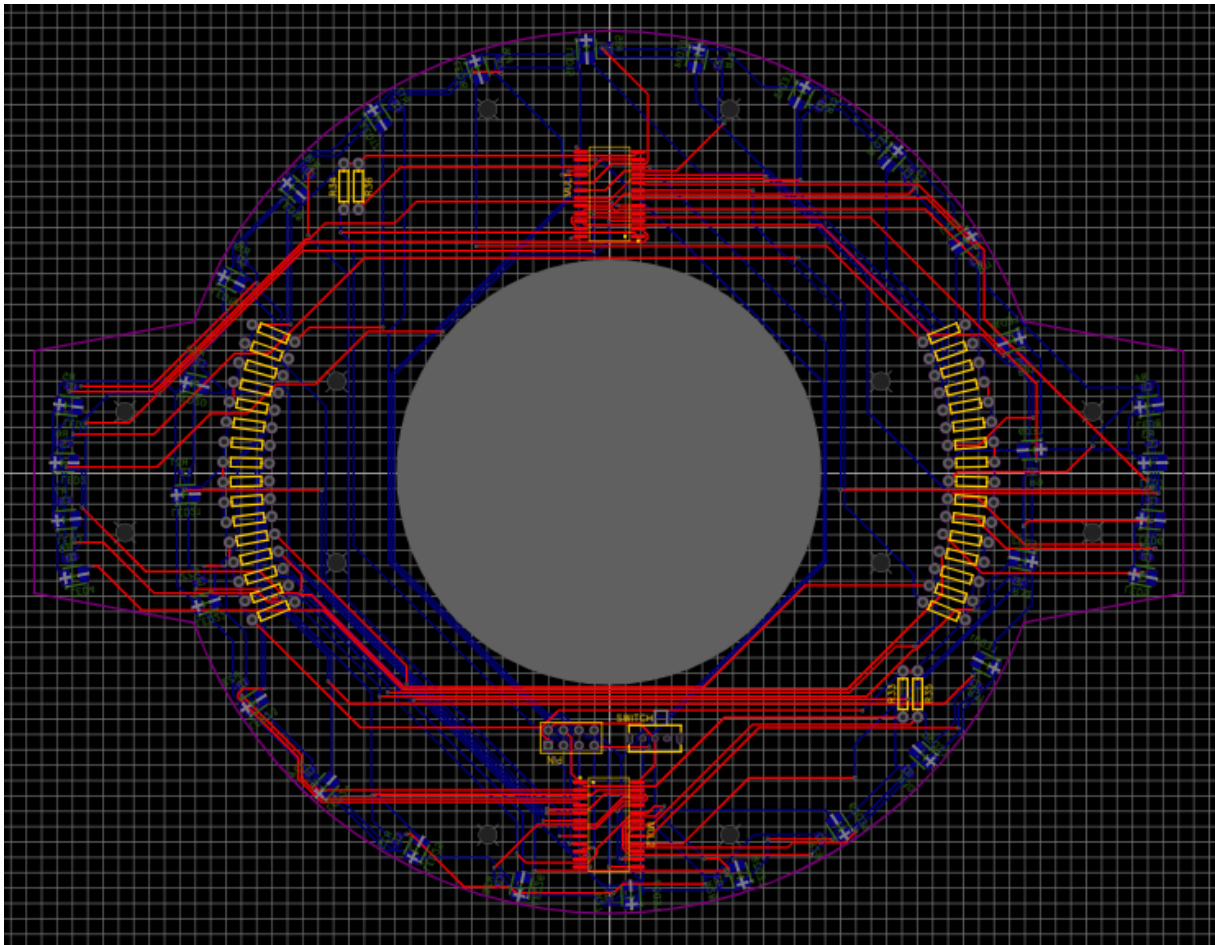
Slika 18. Dizajn PCB pločice prije narudžbe
[Iz vlastite izrade]

5.1.2. Izrada Linijske PCB pločice

Izrada linijske pločice je malo manje kompleksna budući da se sastoji od malih reflektivnih senzora, multipleksera, otpornika i LED dioda. Povezanost između tih komponenata je takva da za napraviti taj senzor potrebno mi je bilo 32 mala reflectance senzora, kao i led dioda za svaki od tih senzora kako bi on bolje čitao. Prilikom izbora led diode proučio sam i koja je boja će najbolje reagirati na podlogu koja je zelena, te na bijelu liniju. Ispostavilo se kako je to crvena boja, stoga sam izabrao crvene led diode. Budući da na pločici imamo 32 senzora koji šalju analogne vrijednosti iskoristio sam dva multipleksora koje sam kasnije programski iskoristio na način da sam birao kojim sensorima pristupiti. Svrha toga je u tome što mi to omogućuje da umjesto 32 analogna pina iskoristim 2 analogna i 4 digitalna pina, budući da teensy 4.1. niti nema 32 analogna pina to mi je omogućilo korištenje više dodatnih senzora za sigurnost čitanja linije.



Slika 19. Schema PCB pločice za liniju
[Iz vlastite izrade]



Slika 20. Dizajn PCB pločice za liniju
[Iz vlastite izrade]

6. Izrada robota

Izrada robota kompleksan je i složeni proces koji se sastoji od nekoliko koraka. Nakon što sam prikupio sve potrebne materijale za rad, proces sam započeo sa dizajniranjem samog robota. Dizajn robota sam odradio u SolidWorks programu. Dizajn nije bio jednostavan i robot se ne sastoji od samo jednog dijela nego od više isprintanih manjih dijelova koji se sklapaju u cjelinu. Nakon dizajniranja robota slijedi 3D printanje istih dijelova. 3D printanje sam izvršio pomoću Prusa mk3 printera koji je u vlasništvu udruge „Robofreak“. Nakon printanja potrebno je zalemiti sve elektroničke komponente na PCB pločicu i na kraju sklopiti pomoću vijaka cijeli robot u cjelinu. Kada je robot sastavljen, zadnji korak je bio još povezati preostale elektroničke komponente žicama i tada je robot spreman za programiranje.

Izrada robota se sastoji od nekoliko dijelova, a to su:

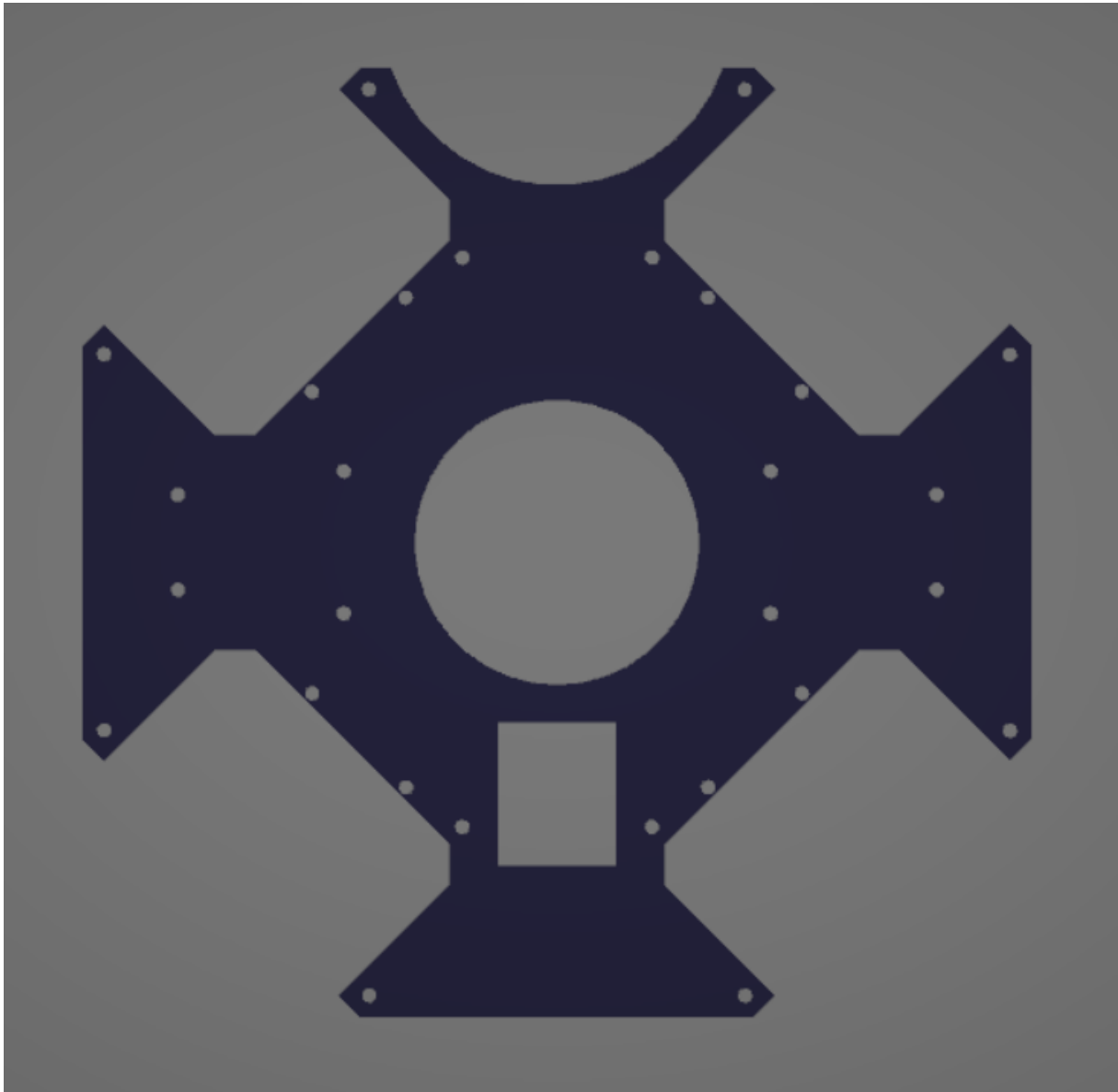
- Dizajn 3D dijelova
- 3D printanje dijelova
- Lemljenje PCB dijelova
- Sklapanje robota

6.1. Dizajn 3D dijelova

Za dobar dizajn potrebno je imati viziju na koji način kreirati samog robota. Vrlo je važno zamisliti u glavi na koji će način takav robot izgledati. Ja sam ga zamislio da ne bude klasično okrugli robot kao i svaki drugi u RoboCup Junior ligama već sam u zamislio oktagonalni oblik. Odlučio sam kako će se robot sastojati od Gornje i Donje šasije kao dva glavna dijela robota, te od ostalih dijelova koji su pomoćni.

6.1.1. Donja šasija

Donja šasija je tako dio koji se sastoji od 4 rupe koje se nalaze na pozicijama od 45 i -45 stupnjeva, kao i 135 i -135 stupnjeva. Na taj način motori najbolje funkcioniraju s kretanjama koje se onda mogu manifestirati u svaku stranu robota. Moguće je naravno napraviti program i sa drugačijim zamakom no onda bi se gubila snaga prema X ili Y osi kretanja robota. Na prednjem kraju donje šasije nalazi se prihvatnište za loptu koje je udubljeno 29mm budući da je prema pravilima dopušteno prihvatiti loptu do 30mm.

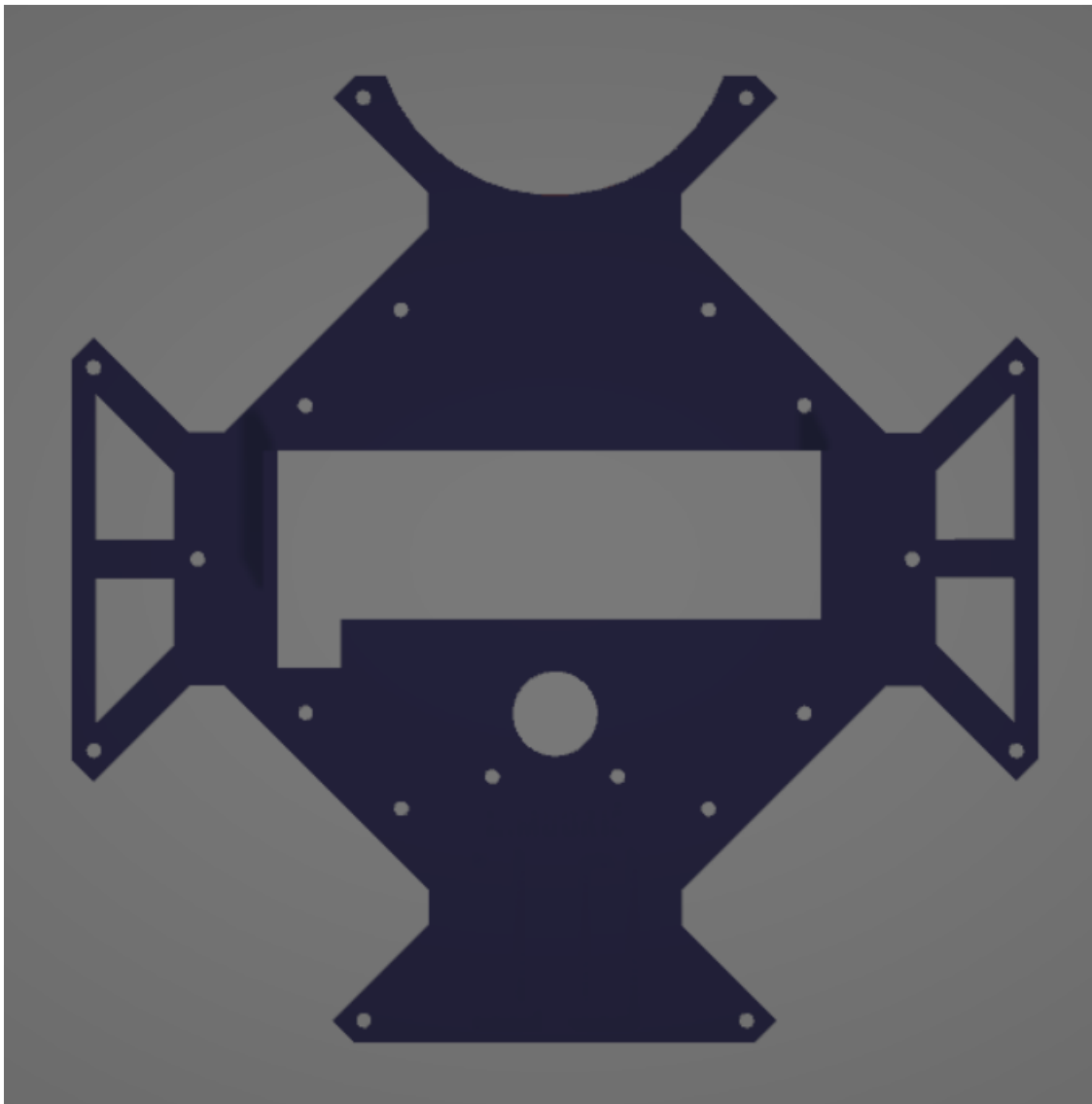


Slika 21. 3D model donje šasije
[Iz vlastite izrade]

Kao što još to možemo vidjeti nalaze se na donjoj šasiji još dvije rupe jedna pravokutna i druga okrugla, te one obje služe kako bi se vodiči mogli provući kroz šasiju do glavne PCB pločice. Na rubovima šasije nalaze se male rupe od 3mm koje služe za povezivanje donje i gornje šasije sa odstojećima, te se još na otvorima za motore nalaze po dvije rupe koje služe za montiranje motora na šasiju. Na prednjem kraju robota kod prihvaćišta za loptu se također nalaze dvije rupe od 3mm kako bi se mogao montirati tzv. kicker. Preostale rupe od 3mm koje se nalaze po sredini šasije služe za pričvršćivanje linijske PCB pločice na šasiju kao i kontrolera motora.

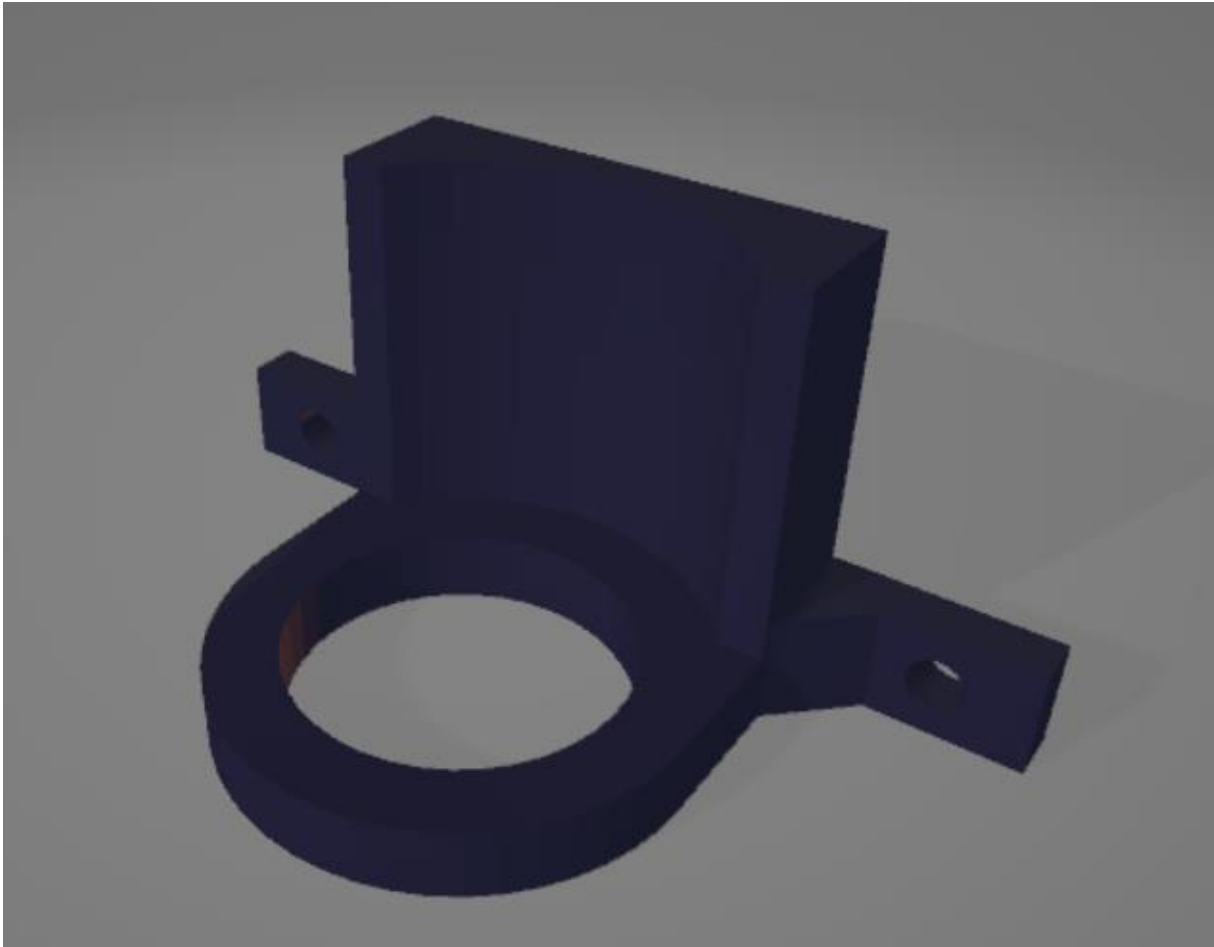
6.1.2. Gornja šasija

Gornja šasija kao jedan od bitnijih dijelova je veoma slična donjoj šasiji, no ipak ima neke promjene. Ono što je jednako na gornjoj i donjoj šasiji su oblik, rupe za motore, rupe za odstoynike i prihvatiste za loptu. Na gornjoj šasiji je u sredini jedna velika rupa koja ima zidove sastrane, te je ona dizajnirana tako da će baterija ići u tu rupu kako bi težište bilo što više u sredini robota. Imamo dvije rupe od 3mm sa lijeve i desne strane koje služe za montiranje glavne PCB pločice, te imamo jednu veću okruglu rupu koja služi za vodiče koji se nalaze na donjoj šasiji, a trebaju se spojiti na glavnu PCB pločicu koja se nalazi na gornjoj šasiji. Na kraju ono što se primjećuje su 4 izreza s lijeve i desne strane koji su stavljeni kako bi se mogle povući žice od kontrolera motora s lijeve i desne strane.



Slika 22. 3D model gornje šasije
[Iz vlastite izrade]

6.1.3. Držač pucaljke (kickera)



Slika 23. 3D model držača pucaljke
[Iz vlastite izrade]

Držač pucaljke je dizajniran tako da se može sa MR vijcima pričvrstiti na donji dio šasije. Sam solenoid koji služi za pucanje je pričvršćen pomoću velike matice koja je zamišljena da drži solenoid kroz jedan otvor kao na slici gore. Kao dodatnu potporu za pucaljku odlučio sam dodati 20 mm kanal koji služi kako bi minimizirao mogućnost gibanja pucaljke, te dodao čvrstinu samom sklopu.

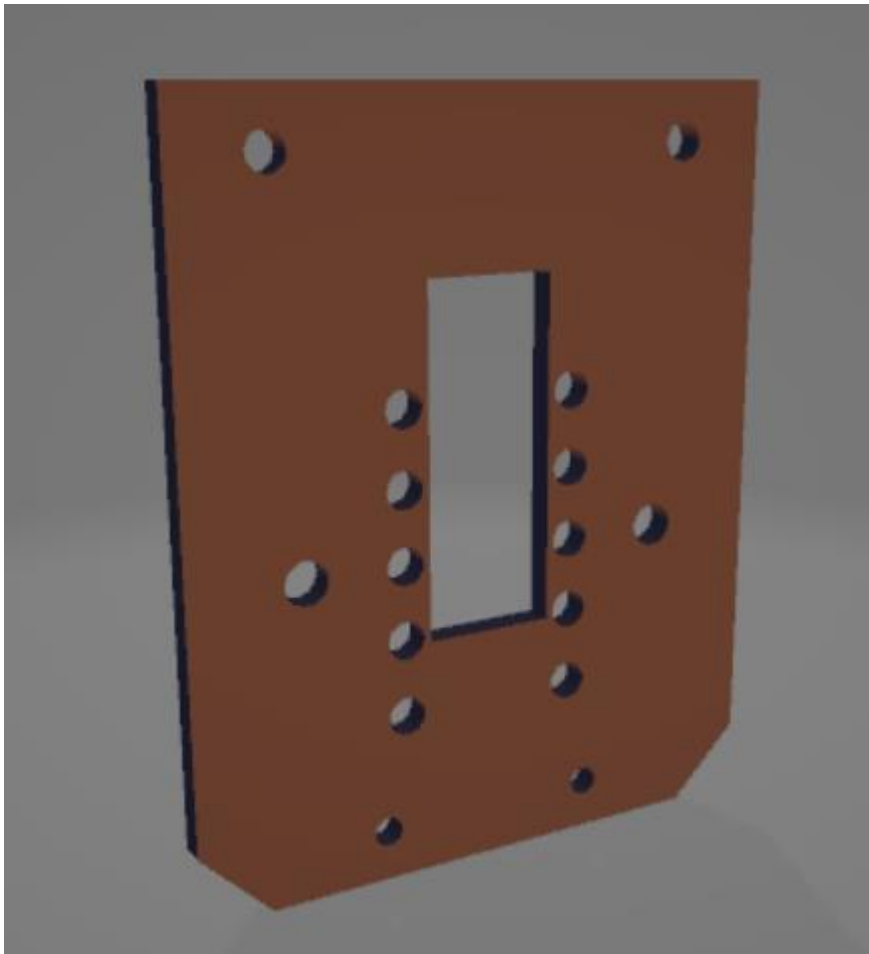
6.1.4. Udarač



Slika 24. 3D model udarača
[Iz vlastite izrade]

Udarač pucaljke je veoma bitan budući da on mora biti dovoljno čvrst da izdrži mnogo udaraca od plastičnu loptu. Pričvršćen je jednom M5 maticom koja je udubljena tako da ne udara direktno u loptu.

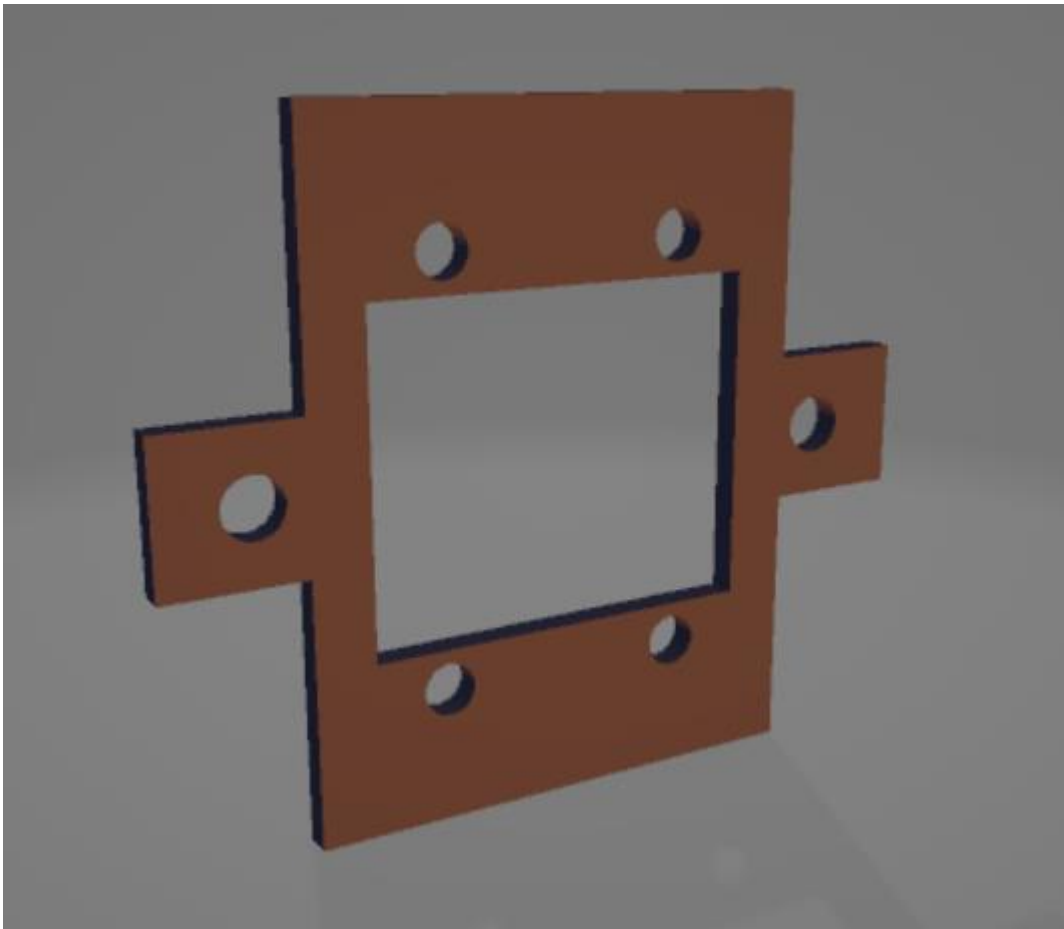
6.1.5. Držać senzora za loptu



Slika 25. 3D model držaća senzora za loptu
[Iz vlastite izrade]

Držać senzora za loptu je jedan od važnijih dijelova koji se nalazi pričvršćen na Glavnoj PCB pločici. U sredini ima jedan otvor koji služi za provođenje vodiča od senzora kako bi urednost robota bila bolja, te kako vodiči ne bi stršali. Na sebi on još ima četiri vanjske M3 rupe koje se pričvršćuju na glavnu PCB ploču, kao i 10 M2 rupe koje služe kako bi se senzor za loptu mogao pričvrstiti za njega.

6.1.6. Držáč kamere



Slika 26. 3D model držača kamere
[Iz vlastite izrade]

Držáč kamere se direktno nadovezuje na senzor lopte, te je dizajniran na način da se može pričvrstiti na sami senzor putem četiri rupe M2 širine. Također u sredini je ostavljen prazan prostor kako bi vodič od kamere mogao proći do samog objektiva.

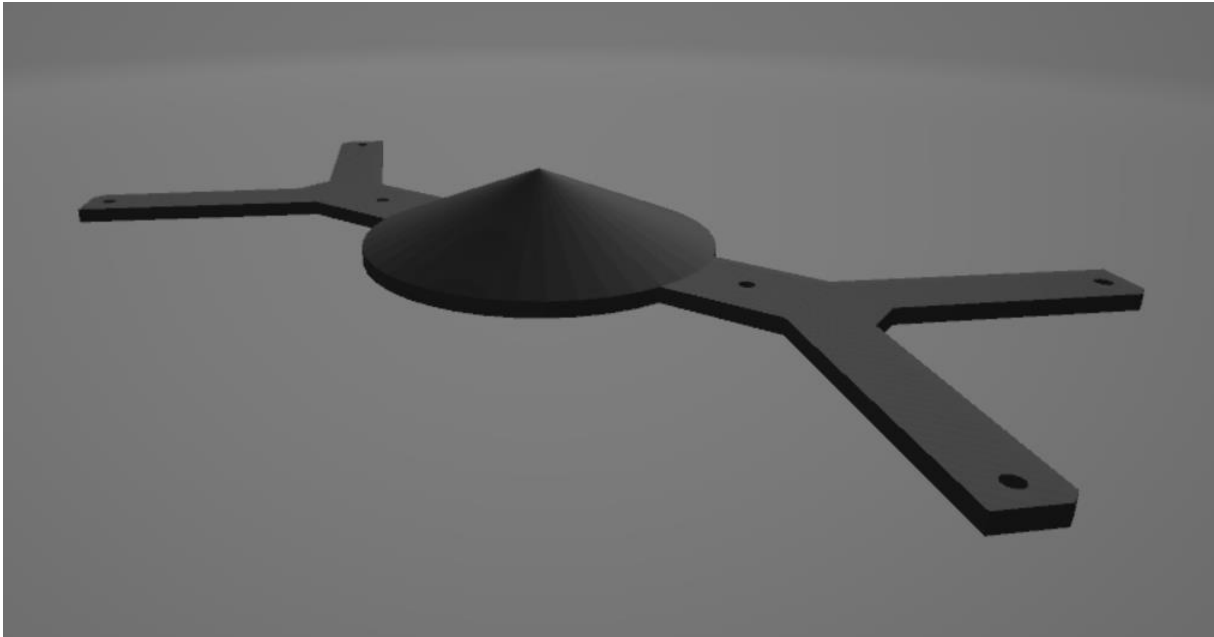
6.1.7. Držač ultrazvučnih senzora



Slika 27. 3D model držača ultrazvučnih senzora
[Iz vlastite izrade]

Držač ultrazvučnih senzora je kreiran kako bi se mogao montirati na lijevu, desnu ili stražnju stranu gornje šasije. Na njemu se nalaze četiri M2 rupe koje služe za montažu samog senzora, kao i prazan prostor kroz kojeg taman sjedaju prijamnik i predajnik ultrazvučnog senzora. Također na dnu samog držača za ultrazvučni sensor se nalaze i dvije M3 rupe koje služe za povezivanje gornje šasije i samog držača.

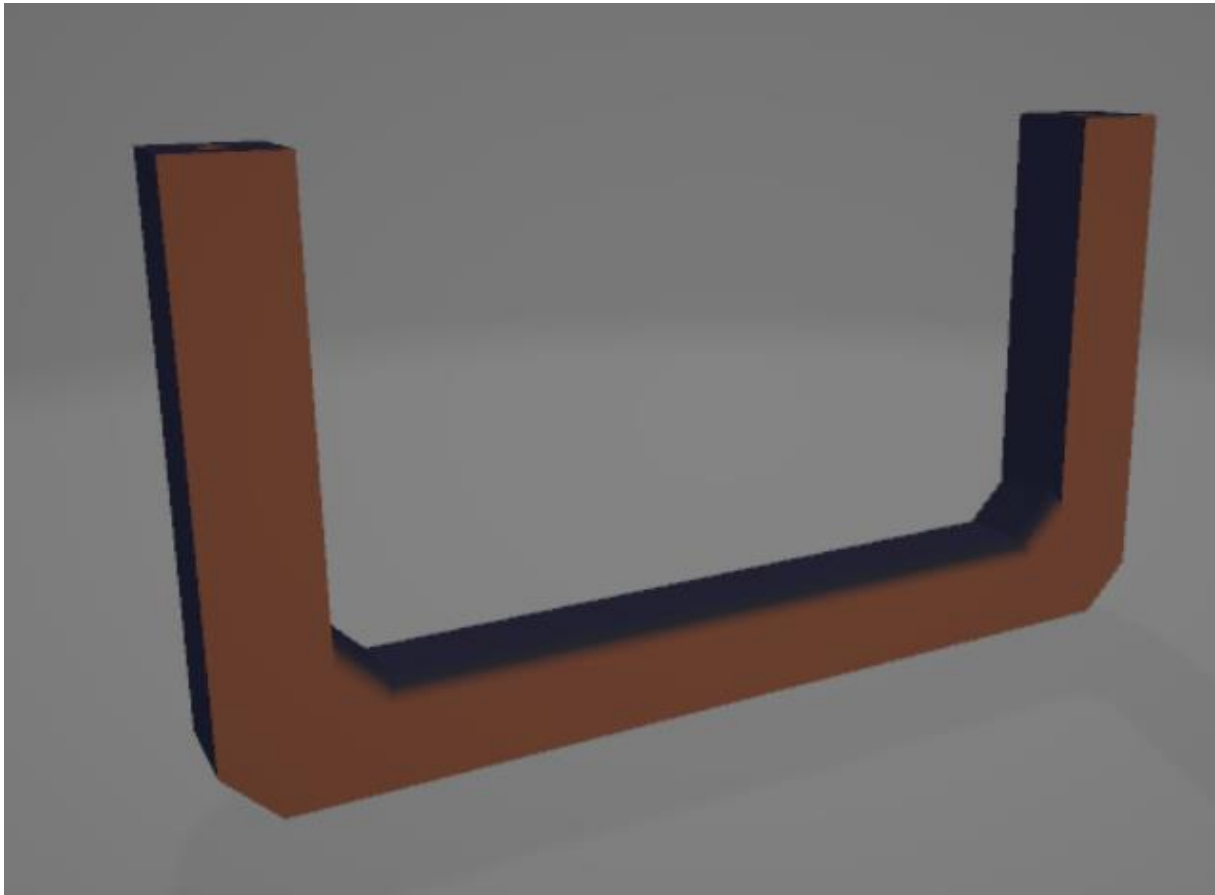
6.1.8. Zrcalo i ručka



Slika 28. 3D model zrcala

[Iz vlastite izrade]

Zrcalo i ručka su dva posebna dijela koja su napravljena tako da se prvo zajedno pričvršćuju, pa tek naknadno montiraju na samog robota. Samo ime zrcalo je zbog toga što se na konus ovog elementa pričvršćuje aluminijski omotač. Taj omotač tada služi kao zrcalo kameri koja gleda direktno u njega, te na taj način robot ima 360 stupnjeva pregleda. Odnosno robot prema pravilu vidi cijeli teren. Samo zrcalo je povezano na gornji dio šasije putem 4 M3 vijka koji se nalaze na krajnjim rubovima zrcala, dok se sama drška nalazi povezana putem dva M3 vijka koji su pokraj konusa. Drška je kreirana dosta robusno s namjerom, kako bi mogla podržati nošenje samog robota koji je težak do 1400 grama.



Slika 29. 3D model ručke za držanje
[Iz vlastite izrade]

6.2. 3D Printanje dijelova

3D printanje, poznato i kao aditivna proizvodnja, predstavlja inovativnu tehnologiju koja omogućava izradu fizičkih objekata slojevitim dodavanjem materijala na temelju digitalnog modela. Od svojih prvih koraka tijekom 1980-ih godina, kada je Charles Hull komercijalizirao ovu tehnologiju, 3D printanje je doživjelo ekspanziju i postalo ključna komponenta u mnogim industrijskim sektorima. Danas se ova tehnologija koristi za širok spektar primjena, uključujući izradu medicinskih uređaja poput umjetnih srčanih pumpi, proizvodnju složenih komponenti u zrakoplovnoj industriji, kao i kreiranje prilagođenih proizvoda u automobilske industriji.

Razvoj 3D printanja obuhvaća različite tehnologije, koje se prema ASTM standardima svrstavaju u sedam osnovnih kategorija: vezivanje prašine, direktno deponiranje energije, ekstruziju materijala, jetovanje materijala, fuziju prašina, laminaciju listova i fotopolimerizaciju u kadama. Svaka od ovih tehnologija ima svoje specifične prednosti i primjene, omogućujući

proizvodnju prototipova, ali i potpuno funkcionalnih proizvoda od različitih materijala. Materijali koji se koriste u 3D printanju obuhvaćaju širok spektar opcija, od metala poput titanija i niki-legura, koji su popularni u zrakoplovnoj industriji, do polimera koji se koriste za medicinske implantate. Također, keramika, kompoziti i pametni materijali nalaze svoju primjenu u različitim industrijama, dok specijalni materijali kao što su hrana i tekstil otvaraju nove mogućnosti za prilagodbu proizvoda specifičnim potrebama korisnika. Primjene 3D printanja su široke i obuhvaćaju različite industrijske sektore. U zrakoplovnoj industriji, ova tehnologija omogućuje proizvodnju laganih i kompleksnih dijelova koji smanjuju potrošnju goriva i optimiziraju performanse zrakoplova. U automobilskoj industriji, 3D printanje se koristi za izradu prototipova, alata i dijelova motora, omogućujući brzu iteraciju dizajna i smanjenje troškova. U prehrambenoj industriji, ova tehnologija omogućuje proizvodnju prilagođene hrane koja zadovoljava specifične dijetetske potrebe, dok u zdravstvenoj industriji omogućuje izradu personaliziranih medicinskih rješenja, uključujući printanje kože, kostiju i organa.

Iako 3D printanje donosi mnoge prednosti, kao što su energetska učinkovitost, smanjenje troškova i mogućnost prilagodbe proizvodnje, postoje i izazovi. Na primjer, mogućnost proizvodnje oružja i drugih opasnih predmeta zahtijeva dodatnu regulaciju i kontrolu. Unatoč tim izazovima, 3D printanje nastavlja mijenjati način na koji se proizvodi izrađuju i distribuiraju, otvarajući nove mogućnosti za inovacije u industriji i društvu.[20]

3D printanje modeliranih dijelova za robot je vrlo bitan dio iz razloga što je potrebno izabrati materijale koji su dovoljno čvrsti, a opet dovoljno lagani kako bi robot bio unutar parametrima težine. Iz tog razloga vrlo je bitan način izrade. Mogao sam, na primjer, izraditi šasiju od čelika koja bi bila mnogo čvršća, no ona bi također bila preteška. Stoga sam 3D printao s nekoliko različitih materijala. Sveo sam dijelove na komponente koje moraju biti čvrste i na komponente koje neće doći u situaciju gdje mogu biti polomljene. Budući da roboti prilikom igre udaraju jedan od drugoga, shvatio sam kako su dvije najbitnije komponente zapravo gornji i donji dio šasije, budući da oni mogu imati udarne kontakte s drugim robotima s bilo koje strane. Iz tog razloga njih sam odlučio printati čvršćim materijalom. Manje bitne dijelove, poput držača senzora, kamere i zrcala, isprintao sam s manje čvrstim materijalom, što mi je doprinijelo lakoci robota, omogućujući mi dodavanje većeg broja senzora, kamere i udaraljke. Samo printanje provodio sam putem PrusinoG Mk3 printera koji je u vlasništvu udruge Robofreak.

6.2.1.Filamenti

Prilikom izrade robota, izbor materijala igra ključnu ulogu u postizanju optimalne ravnoteže između čvrstoće i lakoće, što direktno utječe na performanse i funkcionalnost konačnog proizvoda. U ovom kontekstu, koristio sam dva različita materijala za 3D printanje komponenti robota: PLA i Carbon PLA. Svaki od ovih materijala ima specifična svojstva koja ih čine prikladnima za različite dijelove robota, omogućujući postizanje idealne kombinacije čvrstoće i lakoće.

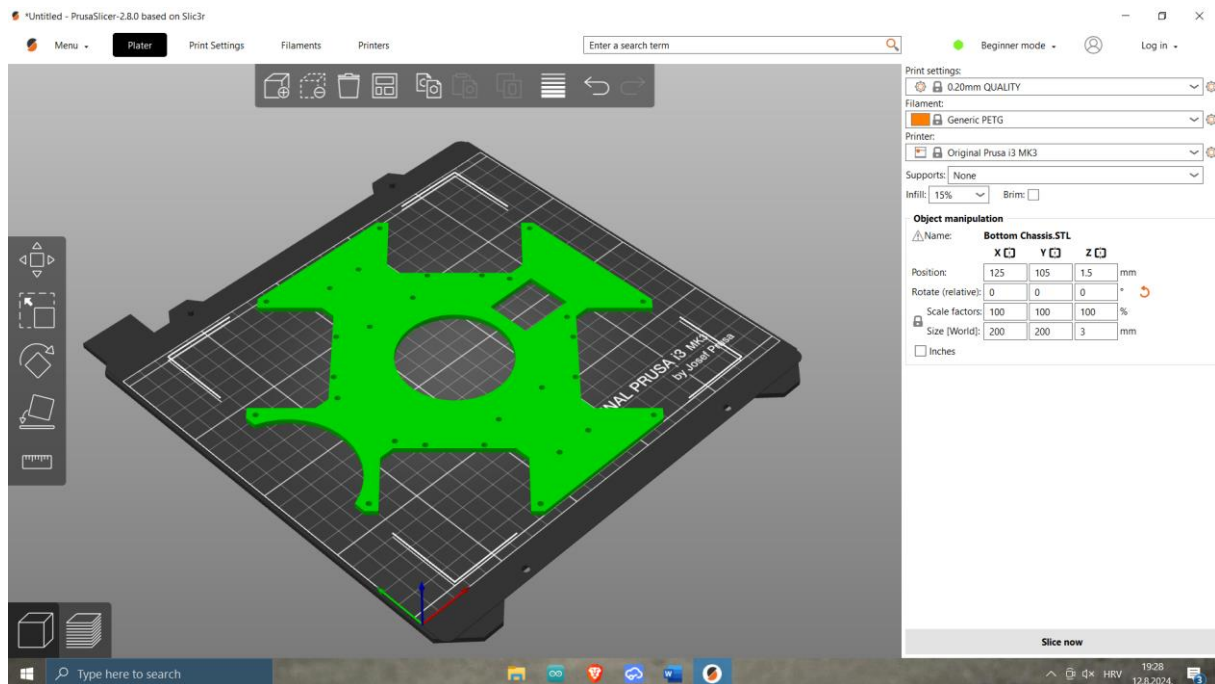
PLA (poli mliječna kiselina) je bio-plastika koja se proizvodi iz kukuruza, a koristi se kao jedan od najpopularnijih materijala u 3D printanju. PLA filament koji sam koristio proizveden je od strane Verbatima, renomiranog proizvođača poznatog po visokoj kvaliteti svojih materijala. Ovaj PLA filament je potpuno ekološki prihvatljiv, jer ne sadrži teške metale, ftalate ni BPA, što ga čini sigurnim za korištenje u raznim aplikacijama. Osim toga, PLA je materijal koji ne zahtijeva grijanu radnu podlogu, što značajno olakšava proces 3D printanja, čineći ga pristupačnim i za početnike u ovom području. PLA se pokazao idealnim za izradu dijelova robota koji nisu izloženi velikim mehaničkim opterećenjima ili udarcima, ali su ipak bitni za funkcioniranje robota. Primjeri takvih dijelova uključuju nosače senzora, kućišta kamera i druge manje komponente. Zahvaljujući svojoj lakoći, PLA je omogućio smanjenje ukupne težine robota, što je direktno doprinijelo njegovoj pokretljivosti i energetske učinkovitosti. Lagani dijelovi robota su posebno važni kada je potrebno integrirati više komponenti kao što su senzori i udaraljke, bez da se naruši ukupna funkcionalnost i performanse robota.

S druge strane, Carbon PLA predstavlja napredniji materijal koji kombinira osnovu PLA s dodatkom sitnih ugljičnih vlakana. Ova kombinacija rezultira materijalom koji ima znatno poboljšana mehanička svojstva, uključujući veću krutost, bolju dimenzijsku stabilnost i otpornost na UV zračenje. Carbon PLA je dizajniran za primjene koje zahtijevaju visoku čvrstoću i otpornost, čime je idealan za izradu dijelova koji moraju izdržati veća opterećenja i udarce. Koristio sam Carbon PLA za izradu kritičnih dijelova robota, poput gornjeg i donjeg dijela šasije, koji su najizloženiji udarcima tijekom rada. Ovi dijelovi robota moraju biti dovoljno čvrsti da izdrže eventualne sudare i udarce, a istovremeno ne smiju biti preteški kako bi se očuvala pokretljivost robota. Carbon PLA je omogućio izradu dijelova koji zadovoljavaju ove zahtjeve, osiguravajući dugotrajnost i pouzdanost robota u zahtjevnim uvjetima rada. Međutim, upotreba Carbon PLA materijala zahtijeva posebnu pažnju tijekom procesa 3D printanja, s obzirom na njegovu abrazivnu prirodu. Preporuča se korištenje mlaznica otpornih na habanje kako bi se izbjeglo začepljenje i oštećenje 3D printera, čime se osigurava kontinuitet i kvaliteta procesa ispisa.

Kombinacija PLA i Carbon PLA materijala pokazala se izuzetno učinkovita u postizanju balansa između težine i čvrstoće robota. PLA je korišten za dijelove koji nisu izloženi visokim mehaničkim stresovima, čime je postignuta smanjena težina i poboljšana agilnost robota. S druge strane, Carbon PLA je osigurao potrebnu čvrstoću i otpornost za dijelove koji trpe najveća opterećenja. Ovaj pristup omogućio je izradu robota koji zadovoljava sve ključne zahtjeve u pogledu performansi, trajnosti i funkcionalnosti, čime se pokazao kao optimalan izbor za ovu aplikaciju.[21][22]

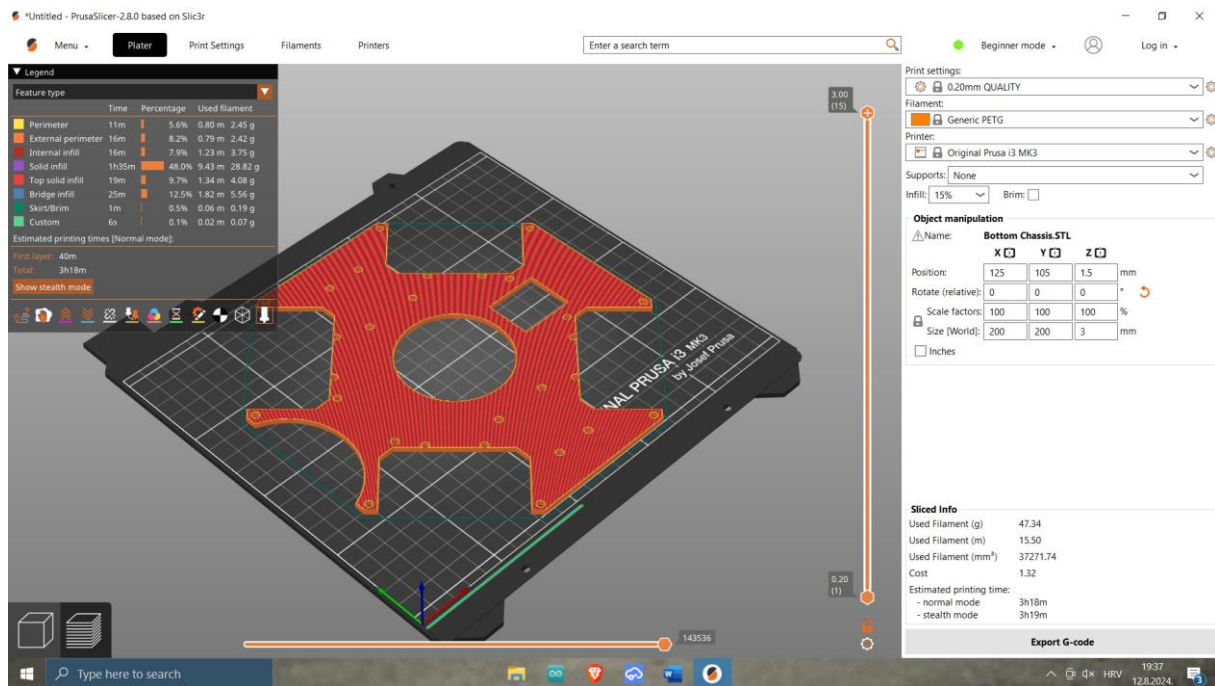
6.2.2.Prusa slicer

Prusa slicer je radna okolina koja je open source, te je dostupna za skinuti putem interneta. Ta okolina je prilagođena za sve vrste prusinih printera, pa tako i za verziju Mk3 koju sam ja koristio za printanje. U prusa sliceru se prvo odabire verzija printera koju imamo, te se nakon toga dodaje STL oblik datoteke 3D modela kojeg smo nacrtali. Na sljedećoj slici se nalazi 3D model donje šasije unutar prusinog slicera verzije 2.8.0. U tom programu postoji mnogo opcija, pa tako i namještanje 3D modela na podlogu, te ono najbitnije postavljanje odgovarajućeg materijala zbog toga što svaki materijal ima vlastitu temperaturu na kojoj se tali što omogućava lakše printanje. Također moguće je namjestiti i neke druge opcije kako bi printer printao npr. većom brzinom ili pak boljom kvalitetom. Također postoje još i brojne druge opcije. Ja sam se držao prilikom printanja onih najosnovnijih opcija gdje sam odabrao materijal, te sam u nekim situacijama birao popunjenost materijala kako bi neki dijelovi imali manje težine.



Slika 30. 3D model donje šasije u prusa sliceru
[Iz vlastite izrade]

Nakon što sam odredio sve postavke, pritisnuo sam gumb Slice now. Tada program stvara svoj G-code kojeg je kasnije potrebno putem SD memorijske kartice prebaciti na sam printer. Kada se program uploada na printer, potrebno je izabrati taj program, te dalje printer odrađuje sav posao samostalno. Na samom printeru nalaze se još neke opcije kao upravljanje brzinom printanja, te kalibracija Z-osi no na ovom printeru nije bilo potrebe budući da sam imamo vremena isprintati sve dijelove. Također, još sam na printeru ovisno o dijelovima mijenjao materijale sa kojima printam kako fizički tako i u prusa sliceru. Sljedeća slika prikazuje sliceanu donju šasiju.



Slika 31. Eksportiranje G-code-a u prusa sliceru
[Iz vlastite izrade]

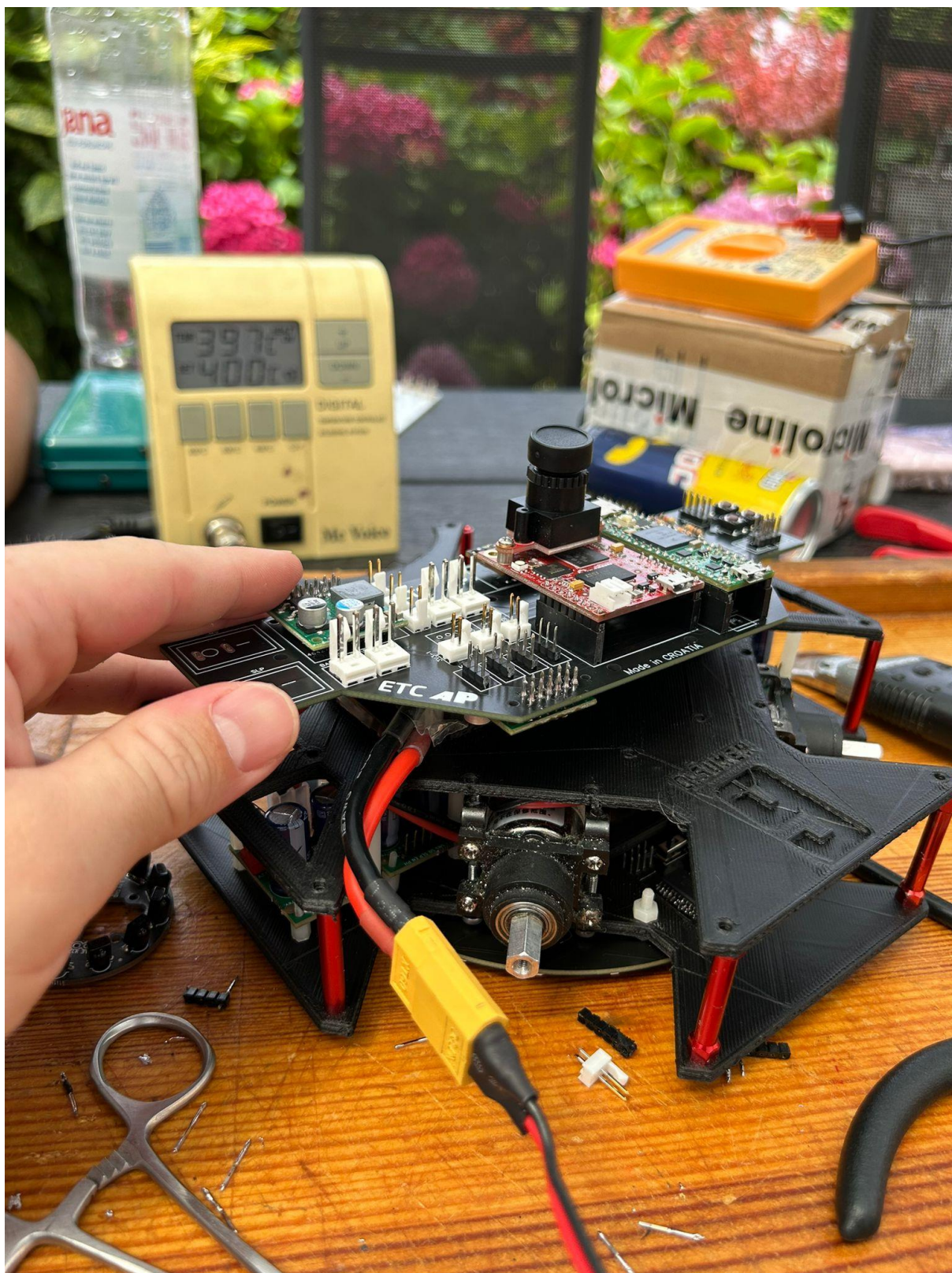
6.3. Sklapanje robota

Sklapanje samog robota je bio najzahtjevniji dio ovog rada, te mi je bilo potrebno nekoliko dana da ga sastavim. Za početak sam prikupio sve materijale, te krenuo sa montiranjem motora na donju šasiju. Motore sam pričvrstio sa osam M3 vijka. Nakon toga bilo je potrebno montirati i senzor linije kojeg sam također pričvrstio sa 4 M3 vijka na donju stranu donje šasije kako bi gledao u podlogu. Kada sam to napravio sljedeći korak je bilo postavljanje 4 kontrolera motora na donju šasiju. Njih sam pričvrstio sa 8 M2 vijka, te sam odmah nakon toga krenuo na sastavljanje udaraljke. Udaraljku sam prvo pričvrstio na držač udaraljke, te sam ju tako sklopljenu montirao na donju šasiju sa dva M3 vijka. Tada sam napravio 4 JST-XH konektora koje sam povezao sa kontrolerima motora i žice provukao prema gore kako bi bile spremne za montiranje gornje šasije. Također dodao sam i 8 crveni odstoynika po rubovima kako bi se gornja šasija mogla pričvrstiti, te povezao donju šasiju sa odstoynicima preko 8 M3 vijka, a nakon toga i pričvrstio gornju šasiju također sa M3 vijcima na odstoynike. Preostalo je još povezati gornju šasiju putem M3 vijaka na motore, te krenuti na lemljenje PCB pločice.



Slika 32. Spajanje gornje i donje šasije prilikom izrade robota
[Iz vlastite izrade]

Na PCB pločicu sam prvo zalemio step-down pretvarače, te molex konektore kako bih mogao spojiti bateriju i provjeriti ima li pločica struju. Nakon toga sam odspojio bateriju, te krenuo lemiti po redu komponente. Prvi na redu je bio teensy, pa nakon toga kamera. Nakon toga sam zalemio otpornike od 10kOhma i 1kOhma zajedno sa pritisnim tipkama i sklopkama za struju i motore. Na kraju sam zalemio preostale JST-XH konektore, te je pločica bila spremna za montažu na robota. Pločicu sam montirao na odstojnike putem M3 vijaka.



Slika 33. Pričvrščivanje PCB pločice na gornju šasiju
[Iz vlastite izrade]

Nakon montaže pločice krenuo sam sa montiranjem senzora za liniju i kamere, te ultrazvučnih senzora. Njih sam povezivao sa M2 vijcima. Na samom kraju preostala su 4 velika odstoynika koji drže ogledalo i dršku. Ogledalo i dršku sam povezao M3 vijcima, no prije toga sam zalijepio aluminijsku foliju na ogledalo kako bi kamera imala pregled od 360 stupnjeva. Nakon toga sam povezao preostale senzore putem JST-XH konektora na PCB pločicu, te dodao barijeru i tako završio konstrukciju robota.

7. Programiranje i testiranje robota

7.1. Arduino okolina

Arduino je otvorena platforma za razvoj elektronike koja pruža nekoliko segmenata prilagođenih različitim korisnicima, uključujući hobiste, edukatore i poslovne korisnike. Ovi segmenti su osmišljeni kako bi omogućili pristup naprednim tehnologijama i olakšali njihovu primjenu u svakodnevnom životu, obrazovanju i poslovanju.

Prvi segment je Maker segment, koji je usmjeren na kreativne pojedince i hobiste. Cilj ovog segmenta je demokratizirati najnaprednije tehnologije, poput povezivanja IoT uređaja, naprednih senzora, Cloud aplikacija, strojnog učenja i umjetne inteligencije, čineći ih dostupnima svima. Arduino u ovom segmentu pojednostavljuje korištenje složenih alata, omogućujući korisnicima da pronađu kreativna rješenja za svakodnevne izazove.

Drugi segment je Education segment, koji je namijenjen obrazovnim institucijama, uključujući srednje škole, sveučilišta i druge obrazovne ustanove. Arduino Education nudi alate i resurse za provođenje STEAM lekcija, koje su relevantne, zabavne i kreativne. Programi obrazovanja uključuju kitove, pakete i pločice s putanjama za učenje koje potiču individualni i kolaborativni pristup učenju. Ovaj segment je otvorenog koda i nudi sadržaje koji se mogu prilagoditi različitim obrazovnim potrebama, omogućujući učenicima da napreduju kroz svoje obrazovanje.

Treći segment je PRO segment, koji je namijenjen poslovnim korisnicima. Ovaj segment omogućuje tvrtkama svih veličina da iskoriste potencijal umjetne inteligencije (AI) i Interneta stvari (IoT). PRO segment pruža platformu za brzo i sigurno povezivanje udaljenih senzora s poslovnim sustavima putem jednostavne IoT aplikacije. Arduino PRO podržava cijeli razvojni ciklus, od hardvera i firmvera do aplikacija za Cloud i mobilne uređaje, omogućujući tvrtkama transformaciju poslovnih modela uz pomoć robusnog i razumljivog IoT hardvera i SaaS platformi. [23]

U svom radu koristio sam Maker segment, konkretno Arduino softver, kao ključni alat za programiranje mog robota. Arduino softver, koji je temeljen na C++ programskom jeziku, omogućio mi je da intuitivno i učinkovito upravljam svim funkcijama robota. Uz pomoć Teensyduina, povezo sam Arduino sa svojim robotom, čime sam osigurao stabilnu komunikaciju i preciznu kontrolu nad hardverom robota. Ova kombinacija alata omogućila mi

je da iskoristim sve prednosti koje Arduino platforma nudi za kreiranje i upravljanje naprednim elektroničkim sustavima.

7.2. OpenMV okolina

OpenMV je inovativna platforma za računalni vid koja pruža jednostavne i pristupačne alate za razvoj aplikacija temeljenih na vizualnoj obradi. Ova platforma omogućava korisnicima programiranje malih, moćnih kamera pomoću MicroPython jezika, što omogućuje brzo i jednostavno postavljanje i upravljanje različitim vizualnim projektima. OpenMV Cam moduli dizajnirani su za pojednostavljenje upotrebe algoritama za računalni vid, omogućujući korisnicima da obavljaju složene zadatke poput prepoznavanja lica, praćenja objekata i dekodiranja bar kodova s minimalnim tehničkim znanjem.

U mom projektu koristio sam OpenMV H7+ kameru zajedno s extension kabelom kako bih omogućio fleksibilniju primjenu kamere u različitim dijelovima robota, odnosno kako bih je mogao montirati direktno ispod konusa za integraciju 360 pregleda. Ova konfiguracija omogućila mi je da proširim mogućnosti kamere i integriram računalni vid u moj robotski sustav na način koji je bio prilagođen mojim potrebama. Zahvaljujući jednostavnosti korištenja MicroPython jezika i fleksibilnosti OpenMV modula, bio sam u mogućnosti brzo razviti i implementirati sustav za praćenje objekata. OpenMV je omogućio integraciju računalnog vida u moj projekt, bez potrebe za složenim konfiguracijama ili dubokim tehničkim znanjem, čime je značajno ubrzao razvojni proces. [24]

7.3. Program

Program koji koristim na robotu napravio sam u arduino software-u, te se temelji na nekoliko glavnih funkcija koje omogućavaju robotu da se kreće, izbjegava liniju, kalkulira putanju kretanja prema robotu, proračunava kut okreta prema голу i puca prema голу.

Funkcija za kretanje robota je najbitniji dio robota, budući da omogućava osnovnu funkciju potrebnu za igranje robotskog nogomet. Funkcija `go()` upravlja kretanjem robota kontroliranjem brzine i smjera svakog od njegovih četiri motora. Ona prima četiri parametra: brzinu, kut kretanja, rotaciju i ograničenje maksimalne brzine. Prvo, funkcija prilagođava kut kretanja koji se koristi za određivanje smjera robota. Kut se transformira kako bi odgovarao specifičnom koordinatnom sustavu robota, a zatim se pretvara iz stupnjeva u radijane, što je potrebno za daljnje matematičke izračune. Nakon toga, izračunavaju se projekcije brzine na osi motora uzimajući u obzir i linearnu brzinu i rotaciju robota. Na temelju tih izračuna, određuje se brzina

za svaki od četiri motora. Sljedeći korak je normalizacija brzina motora. Funkcija prvo identificira najveću brzinu među motorima. Ako je ova brzina veća od dopuštene maksimalne brzine, sve brzine se proporcionalno smanjuju kako bi se osiguralo da nijedan motor ne premašuje zadano ograničenje. Ako je maksimalna brzina postavljena na nulu, svi motori se zaustavljaju. Cilj ove funkcije je omogućiti preciznu kontrolu kretanja robota, tako da se može kretati u željenom smjeru i pri željenoj brzini, dok se istovremeno kontrolira rotacija i ograničava maksimalna brzina za sigurno i stabilno kretanje.

```
void go(float speed, float angle, float rotation, int speedLimit = 10) {
    delayMicroseconds(10);
    LastDirection = angle;
    angle += 135;
    angle *= -1;
    angle = toRad(angle);
    float si = sin(angle);
    float co = cos(angle);
    float motorSpeed[4];

    motorSpeed[0] = -speed * si - rotation;
    motorSpeed[1] = -speed * co - rotation;
    motorSpeed[2] = speed * si - rotation;
    motorSpeed[3] = speed * co - rotation;

    float maxMotorSpeed = abs(motorSpeed[0]);
    for (int i = 0; i < 4; i++) {
        if (abs(motorSpeed[i]) > maxMotorSpeed)
            maxMotorSpeed = abs(motorSpeed[i]);
    }
    for (int i = 0; i < 4; i++) {

        if (speedLimit == 0)
            motorSetSpeed(i, 0);
        else if (maxMotorSpeed > speedLimit)
            motorSetSpeed(i, motorSpeed[i] / maxMotorSpeed * speedLimit);
        else
            motorSetSpeed(i, motorSpeed[i]);
    }
}

void Uread(){
    sensity_t2 = analogRead(Uleft);
    sensity_t3 = analogRead(Uright);
```

```

dist_t2 = sensity_t2 * MAX_RANG / ADC_SOLUTION;//
dist_t3 = sensity_t3 * MAX_RANG / ADC_SOLUTION;//

if(dist_t2 < 20){
  left = true;
}
else{
  left = false;
}
if(dist_t3 < 30){
  right = true;
}
else{
  right = false;
}
//Serial.println(dist_t2);
delay(1);
}

```

Sljedeća na redu je funkcija koja izračunava odmak od kompasa koji je jedan od ključnih faktora koji se koristi u funkciji go() kao treći parametar odnosno rotacija. Bez te funkcije robot bi bio dezorijentiran, te bi nakon nekog vremena počeo pucati na vlastiti gol. Funkcija rotationToMaintainHeading() očitava vrijednost iz kompasa putem analognog senzora. Ova vrijednost se mapira u raspon od 0 do 360 stupnjeva, što predstavlja orijentaciju robota u odnosu na sjever. Kako bi se osiguralo da kompas uvijek daje konzistentne vrijednosti unutar tog raspona, dodatno se koristi funkcija koja osigurava da rezultat ostane unutar 0-360 stupnjeva, čak i ako dođe do preklapanja preko granice (npr. prelazak s 359 na 0 stupnjeva). Nakon što se utvrdi trenutna orijentacija robota, funkcija izračunava razliku između željene orijentacije (headingToMaintain) i stvarnog trenutnog smjera očitano s kompasa. Ta razlika označava koliko se robot treba okrenuti kako bi se usmjerio prema zadanom smjeru. Budući da je ova razlika smjera (heading error) ključna za određivanje potrebne korekcije, ona se normalizira na raspon od -180 do 180 stupnjeva. To osigurava da robot uvijek bira najkraći put za okretanje, bilo u lijevo ili desno. Nakon normalizacije, razlika smjera se mapira u raspon od -150 do 150, što predstavlja brzinu rotacije motora koja će se primijeniti kako bi se robot usmjerio prema željenom smjeru. Ovaj raspon omogućuje da rotacija bude brža ili sporija, ovisno o veličini odstupanja od željenog smjera. Ako izračunata vrijednost rotacije bude negativna, što znači da se robot treba okrenuti u suprotnom smjeru, rotacija se dodatno usporava množenjem s faktorom 0,15. Ovaj faktor nije fiksna, te se po mogućnosti može

mijenjati, no u mojem slučaju je bio idealan. On omogućuje da robot lagano prilagodi smjer kada je potrebno vrlo malo korigirati putanju, izbjegavajući nagle promjene koje bi mogle destabilizirati njegovo kretanje. Funkcija `rotationToMaintainHeading()` prilagođava rotaciju robota tako da se on uvijek zadrži na zadanom smjeru kretanja. Kombiniranjem očitavanja iz kompasa s izračunima za korekciju, ova funkcija osigurava da robot može precizno održavati željeni smjer, bez obzira na vanjske faktore ili promjene u putanji. To je posebno korisno u situacijama gdje je važno da robot ostane orijentiran prema određenoj meti ili smjeru dok se kreće kroz prostor.

```
float rotationToMaintainHeading(float headingToMaintain) {
    float kompas = map(analogRead(A9), 0, 1023, 0, 360);
    float kompi = fmod((kompas + 360), 360);
    float rotation = map(normalized(headingToMaintain - kompi), -180, 180, -
150, 150);
    if (rotation < 0) {
        rotation = rotation * 0.4;
    }
    return rotation;
}
```

Sljedeća na redu je funkcija koja kalkulira smjer kretnje robota kako bi što lakše došao u poziciju da prihvati loptu sa prednjim krajem. Funkcija prvo očitava položaj lopte pomoću senzora, što se naziva "ležaj lopte" (ball bearing). Ova vrijednost se mapira u raspon od 0 do 360 stupnjeva, što omogućuje funkciji da utvrdi smjer u kojem se lopta nalazi u odnosu na robota. Također se očitava udaljenost do lopte i mapira u raspon od 0 do 100, gdje 0 predstavlja kada je blizu lopta, a 100 označava maksimalnu udaljenost unutar koje robot može detektirati loptu. Nakon određivanja osnovnog smjera prema lopti, funkcija izračunava "kut odstupanja" koristeći funkciju `calculateOffsetAngle()`. Ovaj kut odstupanja određuje kako će robot kružiti oko lopte, umjesto da se kreće ravno prema njoj. Na temelju ovih izračuna, robot odlučuje hoće li se kretati direktno prema lopti ili će započeti kruženje. Ako je lopta daleko (udaljenost manja od 75), robot će se kretati ravno prema lopti. Funkcija prilagođava smjer tako da se robot kreće ravno prema lopti, bez ikakvog odstupanja. Kut kretanja se podešava u slučaju da prelazi 180 stupnjeva, kako bi se osiguralo da robot prati najkraći put prema lopti. Ako je lopta relativno blizu, robot će početi kružiti oko nje. Kut kretanja se korigira prema izračunatom kutu odstupanja, omogućujući robotu da kruži oko lopte umjesto da se izravno kreće prema njoj. U svakom slučaju, funkcija koristi `go()` kako bi upravljala kretanjem robota, određujući brzinu, kut i rotaciju tako da robot ili slijedi ili kruži oko lopte, ovisno o situaciji.

CalculateOffsetAngle. Ova pomoćna funkcija služi za izračunavanje kuta odstupanja koji se koristi za kruženje oko lopte. Funkcija uzima u obzir položaj lopte i prilagođava kut odstupanja tako da robot može kružiti oko lopte s različitim odstupanjem, ovisno o tome na kojem se dijelu kruga oko lopte nalazi. Ako je lopta ispred robota (unutar prvih 180 stupnjeva), kut odstupanja raste proporcionalno kretanju robota prema naprijed, do maksimalnog odstupanja od 90 stupnjeva. Ako je lopta iza robota (između 180 i 360 stupnjeva), kut odstupanja postaje negativan, što omogućuje robotu da zadrži kretanje u suprotnom smjeru, opet do maksimalnog odstupanja od -90 stupnjeva. Na kraju, funkcija vraća izračunati kut odstupanja, koji se koristi u glavnoj funkciji OrbitBall() kako bi se upravljalo kruženjem robota oko lopte. Kombinirajući funkcije OrbitBall() i calculateOffsetAngle(), robot može inteligentno pratiti i kružiti oko lopte. Ove funkcije omogućuju fleksibilno kretanje, bilo da robot treba ići direktno prema lopti ili je zaobilaziti, što ga čini sposobnim za pronalaženje lopte na terenu.

```
void OrbitBall() {
    float ballBearing = map(analogRead(A1), 10, 850, 0, 360);
    float ballDistance = map(analogRead(A2), 0, 200, 100, 0);
    readLineSensors();
    ballDistance = ballDistance - 60;
    if (ballDistance < 0) {
        ballDistance = 0;
    }

    float offsetAngle = calculateOffsetAngle(ballBearing);
    float multiplier = calculateMultiplier(ballDistance);

    Serial.print("Offset Angle: ");
    Serial.print(offsetAngle);
    Serial.print(" angle: ");

    int angle = ballBearing + offsetAngle ;
    if (angle > 180) {
        angle = angle - 360;
    }
    Serial.println(angle);
    if(line){
        go(BRZINA, (angle), rotationToMaintainHeading(headingToMaintain),
BRZINA);
    }
    if(angle == 0){
        kickBall();
    }
}
```

```

    }
    if(ballDistance > 12){
        go(BRZINA, (-angle), rotationToMaintainHeading(headingToMaintain),
BRZINA);
    }
    else{
        angle = angle -offsetAngle;
        go(BRZINA, (-ballBearing), rotationToMaintainHeading(headingToMaintain),
BRZINA);
    }
}
float calculateOffsetAngle(float ballBearing) {
    float offsetAngle;

    if (ballBearing >= 0 && ballBearing <= 180) {
        offsetAngle = min(ballBearing * n, 90);
    } else if (ballBearing > 180 && ballBearing < 360) {

        offsetAngle = max((ballBearing - 360) * n2, -90);
    }

    return offsetAngle;
}

```

Zadnja od glavnih funkcija na kojima se temelji rad je funkcija `readLineSensors4()`. Ona se koristi se za upravljanje kretanjem robota na temelju očitavanja senzora koji detektiraju linije na podlozi. Ova funkcija omogućava robotu da prepozna prisutnost linija i reagira u skladu s tim kako bi ostao unutar zadanog terena i izbjegao prelazak preko linija. Na početku, funkcija postavlja četiri digitalna izlaza na različite kombinacije nula i jedinica koje su direktno povezane na multiplekser senzora linije, te na taj način program bira kojim analognim sensorima će pristupiti. Za svaku kombinaciju, funkcija očitava vrijednosti s dva analogna senzora i pohranjuje te vrijednosti u niz `lin`. Tijekom ovog procesa, funkcija prati jesu li očitavanja iznad određenog praga, što ukazuje na prisutnost linije. Ako je linija detektirana, bilježi se da je linija pronađena. Nakon što su svi senzori očitani, funkcija analizira rezultate kako bi utvrdila gdje je linija detektirana. Senzorska očitavanja grupirana su u četiri glavne kategorije: linija gore, linija lijevo, linija desno i linija dolje. Svaka od ovih kategorija označava prisutnost linije u specifičnom smjeru na osnovu određenih senzorskih očitavanja. Ovisno o tome gdje su linije detektirane, funkcija odlučuje o smjeru i načinu kretanja robota. Ako se detektira linija na specifičnim pozicijama u nizu senzorskih očitavanja, robot izvršava određene manevre poput

promjene smjera kretanja prema gore-lijevo, gore-desno, dolje ili se potpuno zaustavlja. Ako su senzori na pozicijama koje ukazuju na detekciju linije na lijevoj strani, robot može odlučiti za pomicanje u lijevu stranu ili izvršiti rotaciju kako bi prilagodio svoj položaj. Ako su linije detektirane na desnoj strani, robot može odlučiti za kretanje udesno ili rotaciju prema suprotnoj strani. Kada se linija detektira prema gore ili dolje, robot može prilagoditi svoj smjer kretanja kako bi ostao na stazi i izbjegao prelazak preko linije. Ako funkcija ne detektira liniju u nijednom smjeru, robot će se zaustaviti kako bi spriječio nepredviđene pokrete ili napuštanje terena. Funkcija `readLineSensors4()` omogućava robotu da se autonomno kreće i prilagođava prema očitanim linijama na podlozi. Analizom senzorskih podataka i primjenom logike odlučivanja, robot može zadržati optimalan smjer kretanja ili prilagoditi svoje ponašanje kako bi se učinkovito kretao unutar zadanog okruženja i izbjegao izlazak izvan terena.

```
void readLineSensors4() {
    int counter = 0;

    // Iterate through all sensor combinations
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            for (int k = 0; k < 2; k++) {
                for (int l = 0; l < 2; l++) {
                    digitalWrite(30, i);
                    digitalWrite(31, j);
                    digitalWrite(32, k);
                    digitalWrite(33, l);

                    delay(1);

                    // Read analog values
                    lin[counter] = analogRead(A3);
                    lin[counter + 16] = analogRead(A6);
                    if (lin[counter] > LINI || lin[counter + 16] > LINI) found = true;
                    counter++;
                }
            }
        }
    }

    // Determine line detection based on thresholds
    bool lineUp = lin[15] > LINI || lin[16] > LINI || lin[11] > LINI || lin[3]
> LINI || lin[7] > LINI;
```

```

    bool lineLeft = lin[24] > LINI || lin[28] > LINI || lin[18] > LINI ||
lin[26] > LINI || lin[14] > LINI || lin[2] > LINI || lin[6] > LINI || lin[10]
> LINI || lin[22] > LINI;
    bool lineRight = lin[0] > LINI || lin[4] > LINI || lin[8] > LINI || lin[12]
> LINI || lin[1] > LINI || lin[5] > LINI || lin[3] > LINI || lin[9] > LINI
|| lin[27] > LINI;
    bool lineDown = lin[17] > LINI || lin[21] > LINI || lin[25] > LINI ||
lin[29] > LINI || lin[19] > LINI;

    // Movement logic based on sensor readings with priority
    if (lin[30] > LINI) {
        while(true){
            go2(BRZINA , 45, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left
            readlineSensors5();
            if(!found3){
                break;
            }
        }
    } else if (lin[23] > LINI) {
        while(true){
            go2(BRZINA , -45, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left
            readlineSensors5();
            if(!found3){
                break;
            }
        }
    } else if (lin[13] > LINI) {
        while(true){
            go2(BRZINA , -135, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left
            readlineSensors5();
            if(!found3){
                break;
            }
        }
    } else if (lin[20] > LINI) {
        while(true){
            go2(BRZINA , 135, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left

```

```

        readlineSensors5();
        if(!found3){
            break;
        }
    }

} else if (lineUp) {
    int kut = 180;
    while(true){

        go2(BRZINA , kut, rotationToMaintainHeading(headingToMaintain),
BRZINA -5); // Up-Left
        readlineSensors5();
        Uread();
        if(dist_t3 > dist_t2){
            kut = 135;
        }
        else if(dist_t2 > dist_t3){
            kut = -135;
        }
        else{
            kut = 180;
        }
        if(!found3){
            break;
        }
    }
} else if (lineLeft) {
    int kut = 180;
    while(true){
        go2(BRZINA , 90, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left
        readlineSensors5();
        if(!found3){
            break;
        }
        if(lineRight2){
            while(true){

                go2(BRZINA , kut, rotationToMaintainHeading(headingToMaintain),
BRZINA -5); // Up-Left
                readlineSensors5();

```



```

    Uread();
    if(dist_t3 > dist_t2){
        kut = 135;
    }
    else if(dist_t2 > dist_t3){
        kut = -135;
    }
    else{
        kut = 180;
    }
    if(!found3){
        break;
    }
}
}
if(lineDown2){
    int kut = 0;
while(true){
    go2(BRZINA, kut, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left
    readlineSensors5();
    Uread();
    if(dist_t3 > dist_t2){
        kut = 45;

    }
    else if(dist_t2 > dist_t3){
        kut = -45;
    }
    else{
        kut = 0;
    }
    //Serial.println(kut);
    if(!found3){
        break;
    }
}
}
}

} else if (lineRight) {
    int kut = 180;

```

```

while(true){
    go2(BRZINA , -90, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left
    readlineSensors5();
    if(!found3){
        break;
    }
    if(lineLeft2){
        while(true){

            go2(BRZINA , kut, rotationToMaintainHeading(headingToMaintain),
BRZINA -5); // Up-Left
            readlineSensors5();
            Uread();
            if(dist_t3 > dist_t2){
                kut = 135;
            }
            else if(dist_t2 > dist_t3){
                kut = -135;
            }
            else{
                kut = 180;
            }
            if(!found3){
                break;
            }
        }
    }
    if(lineDown2){
        int kut = 0;
        while(true){
            go2(BRZINA, kut, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left
            readlineSensors5();
            Uread();
            if(dist_t3 > dist_t2){
                kut = 45;

            }
            else if(dist_t2 > dist_t3){
                kut = -45;
            }
        }
    }
}

```

```

else{
    kut = 0;
}
//Serial.println(kut);
if(!found3){
    break;
}
}
}
}

} else if (lineDown) {
    int kut = 0;
    while(true){
        go2(BRZINA, kut, rotationToMaintainHeading(headingToMaintain), BRZINA
); // Up-Left
        readLineSensors5();
        Uread();
        if(dist_t3 > dist_t2){
            kut = 45;

        }
        else if(dist_t2 > dist_t3){
            kut = -45;
        }
        else{
            kut = 0;
        }
        //Serial.println(kut);
        if(!found3){
            break;
        }
    }
} else {
    go2(0, 0, rotationToMaintainHeading(headingToMaintain), BRZINA); // Stop
if no line is detected
    found = false;
}

// Debugging output
delay(1); // Ensure a small delay for stability, but no longer unnecessary
delays}

```

Posljednji korak u ovom radu je implementacija ciljanja sa kamerom. Ovaj Python kod implementira praćenje boje pomoću OpenMV kamere, koristeći UART za serijsku komunikaciju i LED diode za vizualnu indikaciju. Kod omogućuje prepoznavanje objekata određene boje (npr. crvena ili zelena) na temelju zadatih pragova boja i određivanje njihove pozicije na slici. Na početku se definiraju pragovi boja u LAB prostoru za crvenu i zelenu boju. Kamera se zatim postavlja da radi u RGB565 formatu sa rezolucijom QVGA (320x240 piksela), dok su automatsko podešavanje jačine svjetla i balans bijele boje isključeni kako bi detekcija boje bila stabilnija. Tijekom rada, kamera uzima slike u stvarnom vremenu i koristi funkciju `find_blobs` kako bi pronašla objekte koji zadovoljavaju kriterije detekcije boje. Za svaki pronađeni objekt (blob), kamera određuje njegovu poziciju na slici, pri čemu koristi X-koordinatu središta objekta za klasifikaciju njegovog položaja. Ako je objekt centriran, rezultat je "1", ako je na lijevoj strani slike, rezultat je "2", a ako je na desnoj, rezultat je "3". Podaci o poziciji objekta šalju se putem UART serijske komunikacije na drugu komponentu za daljnju obradu. Također, crvena i zelena LED dioda na kameri svijetle kada je detektiran objekt, dok se gase kada nema detekcije. Na taj način, kod omogućuje učinkovito praćenje objekata određene boje i njihovu lokalizaciju u prostoru slike.

```
import sensor, image, time, math, pyb
from pyb import UART

threshold_index = 1 # 0 for red, 1 for green, 2 for blue

# Color Tracking Thresholds (L Min, L Max, A Min, A Max, B Min, B Max)
# The below thresholds track in general red/green/blue things. You may wish
to tune them...
thresholds = [(0, 28, -12, 26, 19, 47), #(30, 100, 15, 127, 15, 127)
              (33, 46, 5, 127, 0, 39)] # green_thresholds (13, 100, -128, -
20, 76, -37) (20, 100, -122, -29, -128, 127)

uart = UART(3, 115200)
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.set_auto_gain(False)
sensor.set_auto_whitebal(False)
clock = time.clock()

r = pyb.LED(1)
g = pyb.LED(2)
b = pyb.LED(3)
```

```

#ir = pyb.LED(4)
r.off()
g.off()
b.off()
l = 0
d = 0
#ir.on();

while(True):
    clock.tick()
    img = sensor.snapshot()
    blob = img.find_blobs([thresholds[threshold_index]],
pixels_threshold=200, area_threshold=200, merge=True)
    l = 0
    for b in blob:
        img.draw_rectangle(b.rect())
        img.draw_cross(b.cx(), b.cy())
        if b.cx() > 105 and b.cx() < 215:
            l = 1
        elif b.cx() <= 100:
            l = 2
        elif b.cx() >= 100:
            l = 3
        r.on()
        g.on()
    if l == 0:
        r.off()
        g.off()
    uart.write(str(l) + "#")
    print(l)

```

8. Zaključak

Ovaj diplomski rad detaljno je opisao proces izrade robota prilagođenog za RoboCup Junior Soccer natjecanje, s posebnim naglaskom na dizajn, integraciju elektroničkih i mehaničkih komponenti te programiranje sustava. Tijekom izrade robota prošao sam kroz sve ključne faze od dizajniranja elektroničkih pločica (PCB), 3D modeliranja dijelova robota, do samog sklapanja i testiranja. Kroz ovaj proces, stekao sam vrijedna praktična znanja u području robotike, elektronike i programiranja, što su ključne vještine za daljnji profesionalni razvoj u ovom području.

Jedan od glavnih izazova bio je integracija različitih senzora i aktuatora kako bi robot mogao autonomno navigirati terenom, prepoznati loptu, detektirati prepreke te donositi brze i točne odluke tijekom igre. Za tu svrhu koristio sam napredne senzore, poput infracrvenog senzora za praćenje lopte, kompasa za orijentaciju te ultrazvučnih senzora za detekciju udaljenosti od prepreka. Svaki senzor i aktuator zahtijevali su posebnu pažnju prilikom implementacije i programiranja kako bi sustav radio u stvarnom vremenu, što je zahtijevalo optimizaciju koda i testiranje. Poseban fokus rada bio je na izradi PCB pločica, koje su omogućile stabilnije i pouzdanije povezivanje elektroničkih komponenti u odnosu na tradicionalno ožičenje. Korištenjem softverskog alata EasyEDA uspješno sam dizajnirao i proizveo glavnu i linijsku PCB pločicu koje su se pokazale ključnima za stabilnost i funkcionalnost robota. Ovaj korak omogućio je smanjenje mogućnosti grešaka povezanih s odspajanjem žica i kratkim spojevima, što je često problem na natjecanjima. Što se tiče mehaničkog aspekta, 3D modeliranje u programu SolidWorks omogućilo mi je da dizajniram prilagođene dijelove robota koji su kasnije isprintani pomoću 3D printera. Ovaj proces pružio je značajnu fleksibilnost u kreiranju dijelova koji su savršeno odgovarali konstrukciji robota. 3D printanje omogućilo je izradu kompleksnih dijelova brzo i učinkovito, što je bitno za projekte ove vrste. Programiranje je odigralo ključnu ulogu u funkcionalnosti robota. Korištenjem Arduino platforme i OpenMV kamere, razvijeni su algoritmi za prepoznavanje lopte, kretanje robota i interakciju s okolinom. Kroz ovaj proces, stekao sam duboko razumijevanje rada s mikrokontrolerima, senzorima i aktuatorima, što će mi koristiti u budućim projektima i istraživanjima.

Konačno, iskustvo stečeno kroz rad na ovom projektu donijelo je duboko razumijevanje kompleksnosti izrade autonomnih sustava. Ovo iskustvo ne samo da je proširilo moje tehničke vještine, već mi je pružilo i mogućnost da doprinesem unapređenju RoboCup Junior natjecanja. Nadalje, ovaj rad predstavlja čvrstu osnovu za daljnji profesionalni razvoj u području robotike i programiranja, s potencijalom za sudjelovanje u naprednijim natjecanjima

kao što je RoboCup Major. Vjerujem da će mi stečene vještine pomoći u budućim profesionalnim izazovima te u kreiranju inovativnih rješenja u području autonomnih robotskih sustava.

Popis literature

- [1] - Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., & Osawa, E. "Robocup: The Robot World Cup Initiative." U: Proceedings of the First International Conference on Autonomous Agents, pp. 340-347, veljača 1997.
- [2] - Sklar, E., Eguchi, A., Johnson, J. "RoboCupJunior: Learning with Educational Robotics." U: Kaminka, G.A., Lima, P.U., Rojas, R. (ur.) RoboCup 2002: Robot Soccer World Cup VI. Lecture Notes in Computer Science, vol. 2752. Berlin, Heidelberg: Springer, 2003. https://doi.org/10.1007/978-3-540-45135-8_18
- [3] - RoboCupJunior. "RoboCupJunior Official Website." RoboCupJunior. <https://junior.robocup.org/> (pristupljeno 17. kolovoza 2024.).
- [4] - PJRC. "Teensy 4.1." PJRC. <https://www.pjrc.com/store/teensy41.html> (pristupljeno 17. kolovoza 2024.).
- [5] - OpenMV. "OpenMV Cam H7 Plus." OpenMV. <https://openmv.io/products/openmv-cam-h7-plus> (pristupljeno 17. kolovoza 2024.).
- [6] - Gens Ace. "Tattu 2300mAh 11.1V 75C 3S1P LiPo Battery Pack with XT60." Gens Ace. <https://www.gensace.de/tattu-2300mah-11-1v-75c-3s1p-lipo-battery-pack-with-xt60.html> (pristupljeno 17. kolovoza 2024.).
- [7] - Javan Elec. "Specifikacija motora." Javan Elec. <https://www.javanelec.com/CustomAjax/GetAppDocument/ae6511dc-a918-4f0e-b3a0-fdf8293930d5?type=1&inlineName=True> (pristupljeno 17. kolovoza 2024.).
- [8] - Microline. "Motor 1x50R." Microline. https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-mot1x50r (pristupljeno 17. kolovoza 2024.).
- [9] - Wilson, J.S. Sensor Technology Handbook. Amsterdam: Elsevier, 2004.
- [10] - Microline. "IMU AN3." Microline. https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-imu-an3 (pristupljeno 17. kolovoza 2024.).
- [11] - Microline. "IR Finder 3." Microline. https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-ir-finder3 (pristupljeno 17. kolovoza 2024.).
- [12] - DFRobot. "Line Tracking Sensor for Arduino (QTR-8RC)." DFRobot. <https://www.dfrobot.com/product-1862.html> (pristupljeno 17. kolovoza 2024.).

- [13] - Microline. "MRM Barr." Microline. https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-barr (pristupljeno 17. kolovoza 2024.).
- [14] - Tremba. "Zylindermagnete." Tremba. <https://tremba.de/zyndermagnete/zyndermagnete.php> (pristupljeno 17. kolovoza 2024.).
- [15] - Bright Components. "Dupont Jumper Wires." Bright Components. <https://store.brightcomponents.co.uk/product-category/connectors/dupont-jumper-wires/> (pristupljeno 17. kolovoza 2024.).
- [16] - JST. "XH Connector." JST. <https://www.jst.com/products/crimp-style-connectors-wire-to-board-type/xh-connector/> (pristupljeno 17. kolovoza 2024.).
- [17] - Chipoteka. "Prekidači i tasteri." Chipoteka. <https://www.chipoteka.hr/prekidaci-i-tasteri> (pristupljeno 17. kolovoza 2024.).
- [18] - Gupta, R. Printed Circuit Boards: Design, Fabrication, Assembly, and Testing. New York: McGraw-Hill, 2006.
- [19] - EasyEDA. "EasyEDA Online PCB Design Tool." EasyEDA. <https://easyeda.com/> (pristupljeno 17. kolovoza 2024.).
- [20] - Olakanmi, E.O., Cochrane, R.F., Dalgarno, K.W. "A review on selective laser sintering/melting (SLS/SLM) of aluminium alloy powders: Processing, microstructure, and properties." *Additive Manufacturing*, vol. 22, pp. 318-332, 2018. <https://www.sciencedirect.com/science/article/pii/S2351978919308169> (pristupljeno 17. kolovoza 2024.).
- [21] - 3DJake. "Verbatim PLA Zelena 2." 3DJake. <https://www.3djake.hr/verbatim/pla-zelena-2> (pristupljeno 17. kolovoza 2024.).
- [22] - 3DJake. "R3D Carbon Black." 3DJake. <https://www.3djake.hr/r3d/carbon-black> (pristupljeno 17. kolovoza 2024.).
- [23] - Arduino. "About Arduino." Arduino. <https://www.arduino.cc/en/about> (pristupljeno 17. kolovoza 2024.).
- [24] - OpenMV. "OpenMV Official Website." OpenMV. <https://openmv.io> (pristupljeno 17. kolovoza 2024.).

Popis slika

| | |
|---|----|
| Slika 1. IR lopta korištena na RCJ natjecanjima..... | 9 |
| Slika 2. Teren na kojem se igra Robotski nogomet..... | 10 |
| Slika 3. Teensy 4.1 Mikrokontroler..... | 13 |
| Slika 4. Open MV Kamera..... | 14 |
| Slika 5. Tattu LiPo baterija | 16 |
| Slika 6. Step-down regulator | 17 |
| Slika 7. Joinmax motori | 19 |
| Slika 8. Kontroleri motora | 20 |
| Slika 9. IMU senzor | 22 |
| Slika 10. Senzor za loptu..... | 23 |
| Slika 11. Ultrazvučni senzor | 25 |
| Slika 12. Barijera i LED dioda | 26 |
| Slika 13. Solenoid..... | 27 |
| Slika 14. Power Supply dio PCB pločice | 31 |
| Slika 15. Teensy 4.1. pinout na PCB pločici..... | 32 |
| Slika 16. Pinout motor drivera na PCB pločici | 33 |
| Slika 17. Pinout senzora na PCB pločici..... | 34 |
| Slika 18. Dizajn PCB pločice prije narudžbe..... | 35 |
| Slika 19. Schema PCB pločice za liniju | 36 |
| Slika 20. Dizajn PCB pločice za liniju..... | 37 |
| Slika 21. 3D model donje šasije | 39 |
| Slika 22. 3D model gornje šasije..... | 40 |
| Slika 23. 3D model držača pucaljke..... | 41 |
| Slika 24. 3D model udarača | 42 |
| Slika 25. 3D model držača senzora za loptu | 43 |
| Slika 26. 3D model držača kamere..... | 44 |
| Slika 27. 3D model držača ultrazvučnih senzora..... | 45 |

| | |
|---|----|
| Slika 28. 3D model zrcala..... | 46 |
| Slika 29. 3D model ručke za držanje | 47 |
| Slika 30. 3D model donje šasije u prusa sliceru | 51 |
| Slika 31. Eksportiranje G-code-a u prusa sliceru | 52 |
| Slika 32. Spajanje gornje i donje šasije prilikom izrade robota | 53 |
| Slika 33. Pričvršćivanje PCB pločice na gornju šasiju..... | 54 |