

# Automatizirani robotski proces praćenja IR objekta unutar određenih granica

---

Kolarić, Ivan

Undergraduate thesis / Završni rad

2021

*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:211:866807>

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

*Download date / Datum preuzimanja:* **2025-01-29**



*Repository / Repozitorij:*

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



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

**Ivan Kolarić**

**Automatizirani robotski proces praćenja IR  
objekta unutar određenih granica**

**ZAVRŠNI RAD**

**Varaždin, 2021.**

**SVEUČILIŠTE U ZAGREBU**

**FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN**

**Ivan Kolarić**

**Matični broj: 0035207443**

**Studij: Primjena informacijske tehnologije u poslovanju**

**Automatizirani robotski proces praćenja IR objekta unutar  
određenih granica**

**ZAVRŠNI RAD**

**Mentor:**

Doc. dr. sc. Mario Konecki

**Varaždin, lipanj 2021.**

*Ivan Kolarić*

### **Izjava o izvornosti**

Izjavljujem da je moj završni/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/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi*

---

## **Sažetak**

U ovom radu sam se odlučio najviše fokusirati na povezivanje elektroničkih komponenti sa programiranjem u stvaranje jedne automatizirane cjeline koja funkcionalno odrađuje zadani cilj. No, ujedno ću i pratiti cijeli razvoj od nabave do elektroničkih komponenti kroz određenog dobavljača do izrade, te kasnije i programiranja robota.

**Ključne riječi:** Robot, Robotika, automatika, SolidWorks, arduino, Robocup

# Sadržaj

Sadržaj.....	iii
1. Uvod.....	1
2. Robotika.....	2
3. Elektronički dijelovi i njihova nabava.....	3
3.1. Mikrokontroleri .....	3
3.1.1. Teensy 3.2 specifikacije i način povezivanja .....	3
3.1.2. Razvodna pločica za mikrokontroler .....	5
3.2. Senzor za loptu.....	5
3.3. Senzori boje.....	6
3.4. IMU kompas.....	7
3.5. Motori .....	8
3.6. Kontroleri motora .....	9
3.7. Napajanje.....	10
3.8. Razvodna pločica .....	11
3.9. Ostali materijali i komponente .....	12
3.9.1. Tipkalo .....	12
3.9.2. Vodiči .....	12
3.9.3. Baterija i punjač za bateriju .....	13
3.9.4. Vijci, matice i odstojnici .....	14
3.9.5. Omni-Wheel kotači .....	15
4. Izrada šasije 3D printanjem i 3D dizajniranjem.....	16
4.1. SolidWorks dizajniranje .....	16
4.2. Printanje Zortrax M300 3D printerom.....	19
5. Izgradnja robota.....	21
6. Spajanje elektroničkih elemenata .....	25
7. Programiranje .....	28
7.1. Arduino IDE .....	28
7.2. Testiranje robota.....	28
7.3. Programiranje robota.....	31
8. Zaključak.....	33
Popis literature .....	34
Popis slika .....	36

# 1. Uvod

U današnjem svijetu gdje tehnologija konstantno napreduje brzinom munje, robotika je jedna od najperspektivnijih grana. Tako slobodno možemo reći da u robotici leži budućnost. Iz tog razloga sam odlučio napraviti robota koji će se snalaziti unutar određenih granica samog „nogometnog“ terena. Na predmetnu ideju sam došao uvidjevši kako postoji svjetsko robotičko natjecanje koje se održava pod krovnom organizacijom „Robocup“. Kako sam se i sam natjecao u mlađim uzrastima, oduvijek mi je želja bila okušati se u kategoriji „Soccer“.

Iz tog razloga odlučio sam istražiti i krenuti u izgradnju „nogometnog“ terena i robota koji će kasnije imati funkciju „igranja nogometa“, odnosno automatiziranog praćenja IR objekta unutar određenih granica, gdje granice čini nogometni teren obilježen bijelim linijama, a IR objekt je ustvari posebno izrađena nogometna lopta koja šalje infracrvene zrake u svim smjerovima. Takva lopta se sa običnim setom od više infracrvenih senzora može vrlo lako detektirati te odrediti u kojem smjeru se nalazi sama lopta. Također cilj robota je i ostati unutar granica, što znači da on mora imati i nekakve senzore za detektiranje linije. Ono što je možda najbitnije na robotu kojeg ću izraditi su motori i kotači. Odlučio sam se za varijantu sa četiri istosmjerna motora i svosmjerskim odnosno „Omni-wheels“ kotačima. I naravno, kako na kraju robot sam sebi ne bi „zabijao“ golove, potreban nam je kompas, kako bi robot stalno napadao na istu stranu.

Pravilnim spajanjem svih inputa i outputa na mikrokontroler, dobiva se jedna cjelina koja zapravo ništa ne radi, tako dugo dok se ju ne isprogramira. Svaki set senzora ima svoje zasebne biblioteke zajedno sa setovima naredbi. Jednom kada sve navedeno uspijem povezati u cjelinu, dobit ću funkcionalan sustav koji će biti u mogućnosti obavljati u naslovu navedeni proces.

## 2. Robotika

Robot „nogometaš“ kojeg ću u ovom radu napraviti, u cijelosti spada u granu znanosti pod nazivom robotika. Robotika je jedna grana znanosti koja obuhvaća znanja iz mnogo različitih područja znanosti, a kao što su to znanja iz mehanike u svrhu gradnje robota u konstrukcijsku cjelinu. Također, primjenjuju se i znanja iz elektronike, kako bi se povezali i koristili elektronički sklopovi u cjelinu te kroz znanja iz automatike primijenili zajedno sa znanjima iz programiranja u jednu funkcionalnu cjelinu koja ima mogućnosti zamijeniti neki ljudski rad. Robotika je ustvari jedna mlada tehnička grana sa plemenitim ciljem olakšavanja ljudskog posla, no ne bilo kakvog posla, već teških, repetitivnih, štetnih i opasnih po zdravlje poslova. Kao što je to i češki pisac Karel Čapek zamislio u svojoj drami R.U.R. gdje je u stvari prvi puta u povijesti iskorištena sama riječ robot. Riječ robot je u toj drami derivirana iz ruske riječi „rabota“ što u prijevodu znači rad. Stoga možemo zaključiti kako robotika zapravo zadire u skoro sva područja znanosti kao što su to područja filozofije, medicine, ekonomije i sociologije.[1]

Robotika se može podijeliti na dvije grane: mobilnu robotiku i industrijsku robotiku. Industrijska robotika koristi stacionarne robote koji su u principu robotske ruke koje teže ka preciznosti i brzini. Najčešća uporaba je u autoindustriji gdje se susrećemo sa robotima koji prenose dijelove, pričvršćuju i montiraju te iste dijelove itd., dok mobilna robotika, kao što i sama riječ govori, koristi robote koji imaju neku funkciju mobilnosti, koji se mogu micati i kretati od točke A do točke B. Najčešća uporaba mobilne robotike je u vojne svrhe, gdje imamo bespilotne letjelice ili robote koji razminiravaju minska polja. No, mobilna robotika koristi se i u istraživačke svrhe, kao naprimjer roveri na planetu Marsu ili na mjesecu. Robot kojeg ću ja izraditi spada u kategoriju mobilne robotike iz razloga što je pokretan, odnosno mobilan i kretat će se unutar određenih granica. Mobilni roboti se također mogu dalje razvrstati u dvije kategorije i to prema načinu kretanja i mjestu kretanja. Stoga, prema mjestu kretanja imamo podvodne robote, robote koji se kreću po zraku, robote koji se koriste u svemiru i zemljane robote, dok se s obzirom na način kretanja, dijele na robote gusjeničare, robote humanoide koji su nalik čovjeku i kreću se nogama, pužuci roboti koji gmižu i roboti na kotačima. Imajući u vidu sve podjele možemo zaključiti kako je robot kojega ću ja izraditi, mobilni zemljani robot na kotačima. No, kao što nam sama definicija robota govori, robot je u stvari programabilna tehnička tvorevina koja je u mogućnosti izvršavati seriju naredbi autonomno ili poluautonomno. S obzirom kako ćemo u ovom radu isprogramirati robota, prenijeti na njega program i pustiti ga da sam odrađuje neki niz naredbi, naš će robot biti autonoman.[2]



## 3. Elektronički dijelovi i njihova nabava

Što se tiče nabavke elektroničkih dijelova, dugogodišnjim volontiranjem u udruzi Hrvatsko društvo za robotiku, dobio sam uvid u nabavku elektroničkih komponenti. Velika većina elektroničkih komponenti se nabavlja putem jednog dobavljača, odnosno tvrtke Microline. Sama tvrtka Microline nam je dugi niz godina izlazila u susret, i čak modificirala neke dijelove u elektroničkom smislu, a kako bi nam omogućili što bolje elektroničke komponente koje odgovaraju Hrvatskom društvu za robotiku za daljnju primjenu u Robocup natjecanjima. Sama tvrtka je dugogodišnjim istraživanjem i radom u području robotike lansirala svoje verzije elektroničkih komponenata zvanom MRMS (Mobile robot modular system). No ipak, Microline ne drži sve elektroničke komponente, pa sam morao neke dijelove naručivati s vanjskih Webshopova, kao što sam to učinio i sa motorima. Ipak, na kraju poučen dobrim iskustvima i širokom ponudom, odlučio sam se za nabavu većine komponenata od tvrtke Microline.

Teensy 3.2 mikrokontroler je najbitniji dio mojeg robota s obzirom da je mikrokontroler u stvari glava samog robota te upravlja svim komponentama koje ćemo kasnije u radu navesti. No, prvo ćemo ukratko objasniti što su zapravo mikrokontroleri i čemu služe.

### 3.1. Mikrokontroleri

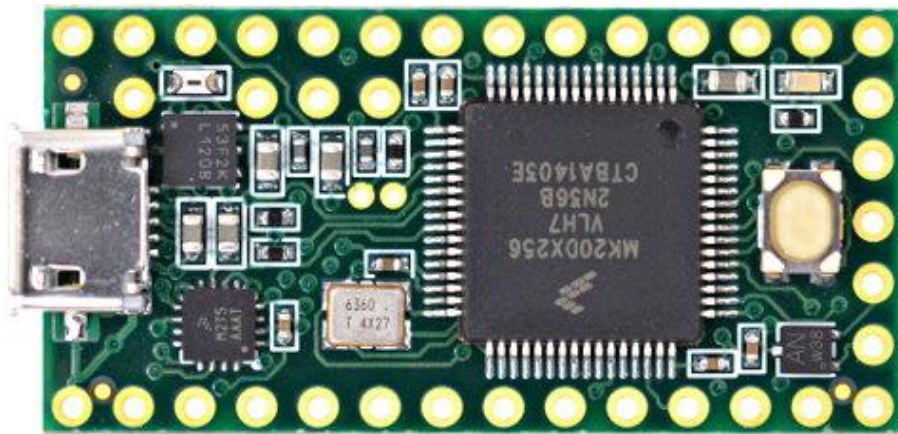
Mikrokontrolere možemo najlakše opisati kao „mala računala“, jer su zapravo skoro pa jednake građe. Najbitnija komponenta svakog mikrokontrolera je zapravo mikroprocesor. Mikroprocesor je ustvari centralna procesorska jedinica, isto kao i kod računala. Što se tiče ostalih komponenti, mikrokontroler uz mikroprocesor sadrži još i RAM memoriju, programabilnu memoriju, analogno-digitalni konverter, digitalne ulaze i izlaze, isto kao i analogne ulaze i izlaze. Ovakvi uređaji su ustvari namijenjeni za određenu namjenu, odnosno kako bi poslužili za neki određeni rad, pa se stoga i mogu nazvati namjenskim računalima.[3]

#### 3.1.1. Teensy 3.2 specifikacije i način povezivanja

Mikrokontroler kojeg ću koristiti u ovom radu je Teensy 3.2 zbog njegove kompaktnosti i lakoće korištenja, a ujedno i zbog razloga što tvrtka Microline ima unaprijed napravljenu razvodnu pločicu za taj mikrokontroler, a što će mi u daljnjem radu olakšati posao spajanja ulaza i izlaza s mikrokontrolera.

Neke od specifikacija samog 32-bitnog Teensy 3.2 mikrokontrolera je mikroprocesor ARM Cortex-M4 sa brzinom takta od 72 MHz. Što se tiče memorije uređaja, ima 3 vrste memorije od kojih je 256K Flash memorije, 64K radne memorije i 2K programabilne EEPROM memorije. Jedna od bitnijih stavki koje su mi potrebne za ovaj rad su broj izlaznih i ulaznih pinova različitih vrsta. Pa tako ovaj uređaj ima 21 analogni input pin, što znači da se na uređaj može spojiti 21 analogni senzor ili neki uređaj koji šalje ulazne podatke. Kada se u obzir uzmu digitalni pinovi, tu imamo mogućnost spajanja 34 ulaznih ili izlaznih uređaja. Dok se na uređaj može spojiti maksimalno 12 motora odnosno PWM izlaza. Također, na uređaju se nalazi i USB kojim se najčešće na uređaj prenosi kod sa računala.[4]

Arduino IDE je možda najčešća okolina koja se koristi uz Teensy 3.2 mikrokontroler, no za nju je potrebno dodatno instalirati Teensyduino programsku podršku, a kako bi se biblioteke i programski kod lakše optimizirali za samu pločicu. Također Teensyduino ima širok spektar biblioteka koje se mogu koristiti za naš mikrokontroler. Što se tiče ostalih softverskih okruženja i platformi, tu su još Visual Micro, PlatformIO i Command Line with Makefile koje se mogu koristiti umjesto Arduino okruženja, no moje je mišljenje kako je Arduino IDE ipak možda najjednostavnije rješenje, te ću ga stoga i koristiti u ovom radu. [4]

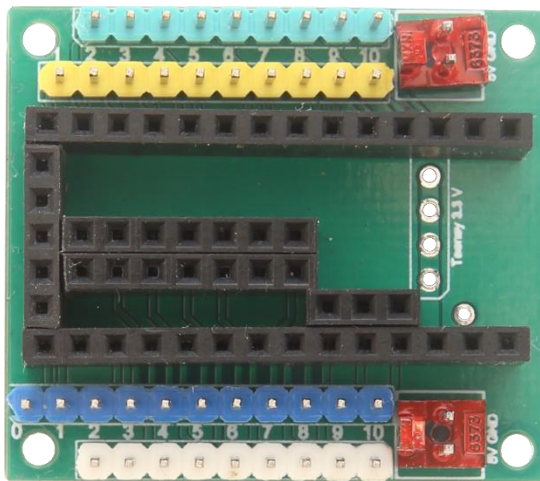


**Slika 1.** Teensy 3.2 mikrokontroler [4]

Nabava samog mikrokontrolera nije bila pretjerano zahtjevna. Teensy 3.2 sam naručio putem pjrc webshopa. Mikrokontroler je bio isporučen unutar nekoliko dana, a ukupno je trebalo otprilike 2 tjedna da ga dobijem na kućnu adresu. Sljedeći korak koji sam morao napraviti je pronaći razvodnu pločicu za teensy 3.2 mikrokontroler, kako bih lakše mogao doći do ulazno izlaznih pinova. Rješenje u tom problemu sam našao na Microlin-ovoj stranici u „MRMS Teen, board for Teensy 3.1/3.2“ razvodnoj pločici.[5]

### 3.1.2. Razvodna pločica za mikrokontroler

Potreba za razvodnom pločicom mikrokontrolera je ponajviše iz razloga što su pinovi na samom mikrokontroleru dosta blizu, što bi cijelu situaciju sa spajanjem više senzora i više komponenata dodatno otežalo. Iz tog razloga odlučio sam kako je pametan potez pribaviti razvodnu pločicu koja na sebi ima 2 Molex KK254 konektora koji služe za direktno napajanje samog mikrokontrolera, a uz to i pomno pripremljene i posložene pinove za kasniju upotrebu prilikom spajanja elektroničkih elemenata. [5]

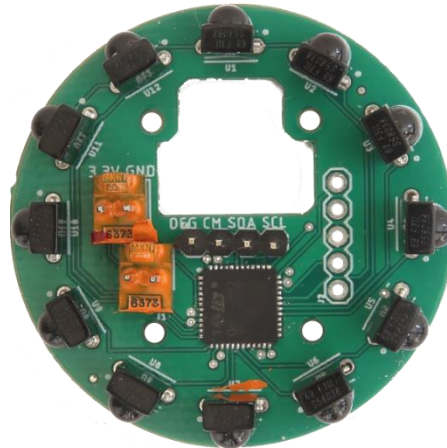


Slika 2. Razvodna pločica za mikrokontroler [5]

### 3.2. Senzor za loptu

Senzor za loptu je jedan od najbitnijih senzora koji mora biti pouzdan, a kako bi uspješno pronalazili loptu u svakom trenutku, odnosno kako bismo uvijek znali poziciju lopte, s obzirom na našeg robota. Senzor lopte koji ću ja koristiti je „MRMS IR ball finder 2“ koji je unaprijed napravljen kako bi hvatao točne zrake lopte u točnim intervalima. Taj senzor ima ulazni napon od 3.3 V i sastoji se od 12 prijamnika poredanih razmjerno u krug, te ima više izlaznih opcija od kojih su dva analogna izlaza. Jedan od dva analogna izlaza daje kut odstupanja lopte od prednje strane senzora u vrijednosti od 0 do 1023 koji mi kasnije u programu modificiramo na vrijednosti od 0 stupnjeva do 360 stupnjeva, dok nam drugi izlaz daje udaljenost lopte u centimetrima od 0 do 1023. Jedan od bitnijih detalja je taj što senzor ima ugrađen Molex KK254 konektor, a što nam olakšava daljnje spajanje senzora na napajanje. Također, na senzoru postoji i opcija  $I^2C$  gdje bi se taj senzor mogao i dodatno modificirati, no meni taj dio nije bio potreban, budući smatram kako senzor funkcionira idealno za moje potrebe.

Sam senzor funkcionira na način da pomoću 12 prijamnika prihvaća signal lopte, te kalkulira koji od prijamnika je u stvari dobio najjači signal, a kalkulira se i aritmetička sredina od nekoliko prijamnika sa najjačim signalom da bi se odredila točna pozicija lopte. Senzor ću postaviti na vrh robota kako bi mogao primati informacije sa cijelog terena, odnosno kako se između senzora i lopte ne bi nalazile prepreke koje bi blokirale signal. [6]



**Slika 3.** Senzor za loptu „MRMS IR ball finder 2“ [6]

### 3.3. Senzori boje

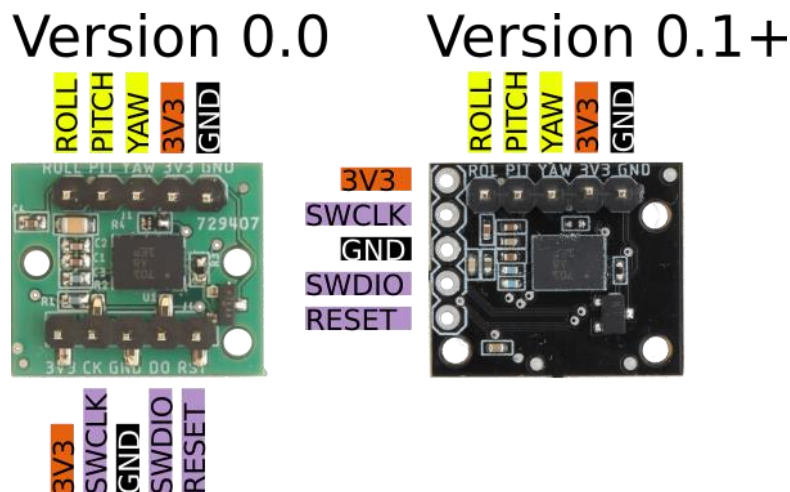
Kao senzor boje koristit ću MRMS Reflectance Sensor A, kojeg ću postaviti na dno robota. Kako bi senzor funkcionirao normalno, potrebno je da ga pozicioniram na udaljenosti od 10 milimetara od površine terena. Sam senzor može biti priključen na napon od 5V i 3.3V, no zbog bolje funkcionalnosti prilikom testiranja samog senzora, uočio sam kako senzor bolje funkcionira kada je spojen na 5V. Senzor radi na principu da šalje infracrvene zrake prema površini, gdje se te zrake odbijaju natrag do senzora koji pomoću infracrvenog prijamnika raspoznaje snop vraćenog infracrvenog zračenja u analognom obliku od 0 do 1023. Samo zračenje pokazuje različitu vrijednost iz razloga što su neke boje reflektivnije od drugih, te se tako od bijele boje odbija više svjetlosti, nego od zelene boje. [7]



**Slika 4.** Senzor za boju „MRMS Reflectance Sensor A“ [7]

### 3.4. IMU kompas

Kako bi se robot pravilno orijentirao, koristit ću pločicu MRMS IMU koja ima 3 analogna outputa od kojih su kompas, akcelerometar i žiroskop. MRMS IMU je baziran na Bosch-ovom čipu BNO055 koji je u današnjem svijetu jedan od najboljih izbora što se tiče kompasa, akcelerometra i žiroskopa u edukativnoj robotici i „Robocup“ organizacijama. Senzor se spaja na 3.3V napajanje, te također ima mogućnost spajanja na 5V, no nema bitne razlike između ta dva moda, pa sam ostavio napon na 3.3V. Što se tiče same pločice, koristim samo kompas koji ima izlazne vrijednosti od 0 do 1023, koje ću isto kasnije u programu pretvoriti u stupnjeve od 0 do 360 stupnjeva.[8]



Slika 5. Stara i nova verzija kompas uređaja „MRMS IMU“ [8]

Kao što možemo vidjeti na slici 3. , postoje dvije vrste kompasa: verzija 0.0 i verzija 0.1+ , odnosno starija verzija i novija verzija, a od kojih ću ja koristiti noviju verziju kompasa. Razlika između starije i novije verzije je najviše u preciznosti, no preciznost u nogometu i ne igra veliku ulogu, već da okvirno imamo konstantnu vrijednost kompasa s maksimalnim apsolutnim odstupanjem do 10 stupnjeva, a kako se robot ne bi okrenuo ka našem голу, odnosno kako bi uvijek ciljao otprilike prema naprijed. [8]

Prilikom nabave kompasa savjetovao sam se sa mentoricom iz udruge „Hrvatsko društvo za robotiku“ gdje sam obavio praksu, budući je kompas jedan od najkrucijalnih dijelova samog robota i ne bi smio imati veća odstupanja od zadanog smjera. Samu nabavu sam obavio putem već spomenutog dobavljača Microlin-a, gdje sam i nabavio većinu potrebnih komponenti.

### 3.5. Motori

Motori kao jedan od najbitnijih izlaza sa mikrokontrolera se koriste u svrhu gibanja robota. Većina robota je zamišljena sa samo 2 motora, no što se tiče nogometa, takvi motori bi samo dodatno zakomplicirali stvar jer bi roboti gubili vrijeme na okret prema lopti, pa će moj robot koristiti 4 četkana istosmjerna motora sa dva polariteta. Motori najbolje rade na 12V, no mogu raditi i na naponu od 3V do 12V. Ove motore sam izabrao iz razloga što su dosta lagani, a što je cilj takvog robota u kategorijama „Lightweight Robocup“. Pojedinačno motor teži 83 grama i ima brzinu od 730 okretaja po minuti. [9]



**Slika 6.** Istosmjerni četkani motor „Actobotics gear motor“ [9]

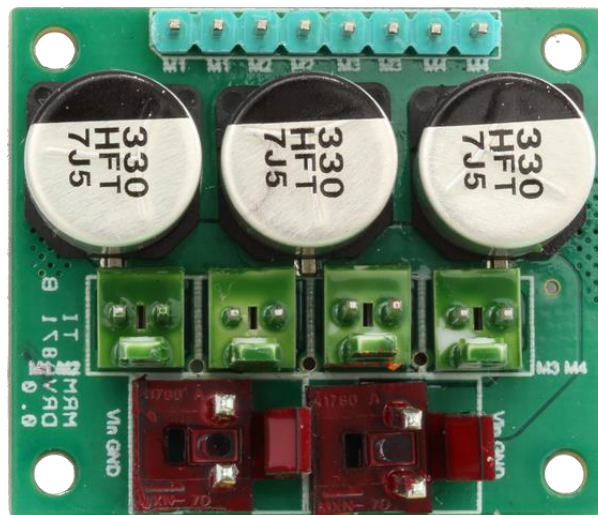
Ovo je i ujedno jedna od komponenta koju sam naručio kod drugog dobavljača putem Servocity webshopa. Motore i webshop mi je preporučila mentorica kod koje sam obavljao praksu, budući ona ima iskustva sa nabavom materijala sa istog webshopa. Motore je navela kao veoma pouzdane i jedne od najboljih prema omjeru cijene i kvalitete, pa oko motora nisam dugo dvojio jer su poprilično jeftini naspram drugih motora. [9]

Kod ove nabavke pribavio sam šest komada motora, iako je za mog robota potrebno četiri motora. Na taj potez sam se odlučio s obzirom se često zna dogoditi da motori jednostavno prestanu raditi, pogotovo u Robo-nogometu, gdje su konstantne promjene smjera vrtnje samog motora, a što može dovesti do značajnijih oštećenja motora.

### 3.6. Kontroleri motora

Što se tiče preostalih dijelova i elektroničkih komponenti koje sam nabavio za izradu mojeg robota, tu možemo navesti možda i jednu od najbitnijih komponenti tzv. „Motor drivere“ odnosno Kontrolere motora. Što se tiče samih kontrolera motora, najbitnije je izabrati prave motor drivere koji mogu upravljati određenim motorima.

Prema našim potrebama smatrao sam kako su „MRMS Motor Driver 4x3.6A B“ idealno rješenje za već prije izabrane motore. Ovi kontroleri motora imaju mogućnost spajanja 4 motora s četkicama, te mogućnost ulaznog napona između 6.5 i 45 V. Izabrani kontroleri motora imaju ugrađene Molex KK396 konektore koji omogućuju lakše spajanje na napajanje, odnosno zaobići ćemo spajanje na razvodnu pločicu. Nažalost, ovi kontroleri motora nemaju zaštitu od obrnutog napona motora. Obrnuti napon motora se događa prilikom nagle promjene smjera motora gdje se kotač i motor još uvijek vrte u jednu stranu i stvaraju napon, a kontroleri motora šalju struju u obratnom smjeru. Takav događaj može dovesti do ozbiljnog oštećenja elektroničkih komponenti, a iz moje prakse najčešće dolazi do oštećenja samih kontrolera koji su nakon toga potpuno neupotrebljivi. Također, važno je i napomenuti kako sami kontroleri dopuštaju vršnu struju od 3.6 ampera, no možda je bolje da se ne koriste do same vršne struje kako se ne bi pregrijavali, pa ću stoga kasnije u programu definirati maksimalnu brzinu motora na otprilike 40% kako bih izbjegao veći obrnuti napon, a i pregrijavanje kontrolera motora. [10]



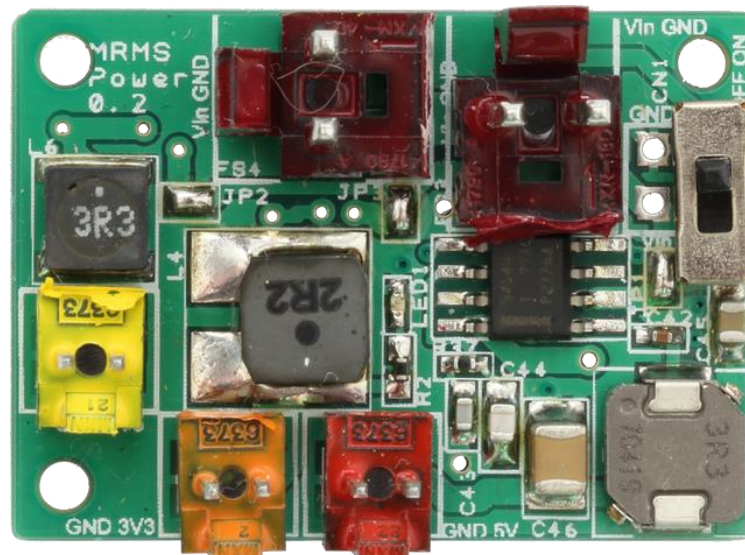
Slika 7. Kontroler motora „MRMS Motor Driver 4x3.6A B“ [10]

MRMS sistem kontrolera motora mi se činio kao najbolji mogući izbor iz razloga što ima unaprijed napravljene biblioteke koje se mogu koristiti za pokretanje samih motora zajedno sa nekim, već ugrađenim funkcijama.

### 3.7. Napajanje

Napajanje je jedan od najbitnijih dijelova iz razloga što se sva energija dobivena iz baterije u njemu pretvara step-down ili step-up metodom u nama potreban napon, a u našem konkretnom slučaju, potreban nam je napon od 5 V i 3.3 V što nam izabrano „MRMS Power Supply 3x (mrm-ps3x)“ napajanje omogućava. Također, treba nam i opcija step-up na 12V, no sa izabranim napajanjem smo je zaobišli, budući da ću izabrati bateriju od 12V, a izabrano napajanje ima pin ulaza na bateriju kojeg ću koristiti za napajanje motora. [11]

Neke od karakteristika samog napajanja su 2 ulazna konektora Molex KK396, dok kod izlaza imamo 3 konektora Molex KK254 koji služe za izlaz na 5V, 3.3V i izlaz varijabilnog napona kojeg možemo podesiti prema potrebi. Također, napajanje ima i prekidač, a efikasnost samog step-down pretvarača za 3.3V iznosi 91%, za 5V 85 – 96% i za varijabilni napon 89% što znači da su gubitci u principu niski. [11]



**Slika 8.** Napajanje „MRMS Power Supply 3x (mrm-ps3x)“ [11]

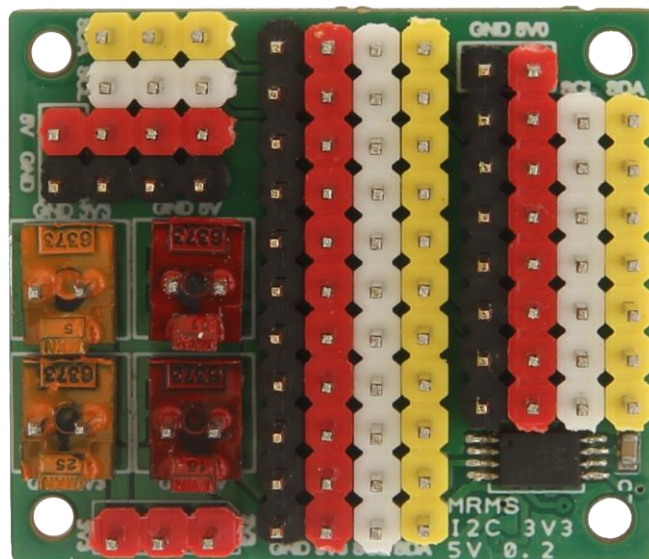
Napajanje sam također uzeo kod istog dobavljača iz razloga što sistem MRMS ima jako dobro razrađenu distribuciju napajanja preko Molex KK254 i Molex KK396. Takva distribucija samog napajanja uvelike olakšava preglednost i samu urednost robota. No, jedan od najbitnijih dijelova distribucije napajanja je taj što se Molex konektori mnogo teže mogu odvojiti jedan od drugoga pa prilikom nekakve potencijalne nogometne utakmice između dva robota manja je mogućnost od odvajanja vodiča i dolaska prekida struje.



## 3.8. Razvodna pločica

Razvodna pločica je bitna iz razloga što kroz nju provodimo napon, odnosno razvodimo napon ka svakom zasebnom senzoru koji unaprijed nema Molex KK254 ili MolexKK396 konektore. Jedan od tih senzora je senzor boje, kojih sam pripremio 8, pa ću svakog zasebno morati spojiti na razvodnu pločicu. Kompas također nema ugrađeni Molex konektor, pa ću ga morati spajati na razvodnu pločicu.

Neke od karakteristika same razvodne pločice su ta da ona ustvari ima I2C sabirnicu koju je moguće koristiti kao dodatne izvore napajanja od 3.3V i 5V, no dodatnih pinova mi neće trebati, pa ću koristiti obične razvodne pinove namijenjene za razvod napajanja. Pločica unaprijed ima spojene Molex KK254 konektore, jedan za 3.3V i drugi za 5V koji su povezani na skup razvodnih pinova od 5V i 3.3V preko kojih ću napajati senzore. Ova razvodna pločica je možda prevelika za moj robot, no kada sam naručivao, razmišljao sam na način da je bolje da imam nekoliko praznih pinova i da pločica bude pregledna, nego da imam premalo pinova što bi me natjeralo na nabavu još jedne razvodne pločice.[12]



**Slika 9.** Razvodna pločica „MRMS I2C 3V3 5V Distribution Pins (mrm-distrib-a)“ [12]

Samu nabavu razvodne pločice sam izvršio putem Microlin-a iz razloga što mi je bilo jednostavnije naručivati od istog dobavljača više komponenata, a i zbog Molex konektora, radi lakše distribucije napajanja.

## 3.9. Ostali materijali i komponente

Što se tiče ostalih materijala i komponenata, ovdje ću nabrojiti i napisati sve materijale koji su mi bili potrebni pri izradi, spajanju i puštanju robota u pogon, od čega su to tipkalo, baterije, punjači za baterije, vodiči, vijci, matice, odstojnici, Omni-Wheel kotači i Isprintani dijelovi koje ću kasnije detaljnije razraditi u temi 3D printanje i 3D dizajniranje.

### 3.9.1. Tipkalo

Još jedan od senzora koje ću koristiti je senzor pritiska, odnosno tipkalo. Njegova namjena će mi u ovom radu služiti samo za pokretanje mog programa na robotu, odnosno kako se robot ne bi počeo micati istog trena kada bude priključen na bateriju. Iz tog razloga sam nabavio „MRMS switch 2x“ elektroničku komponentu koja je u principu elektronička pločica sa ugrađena 2 tipkala namijenjena za ulazni i izlazni, odnosno logički napon od 3,3 V. [13]

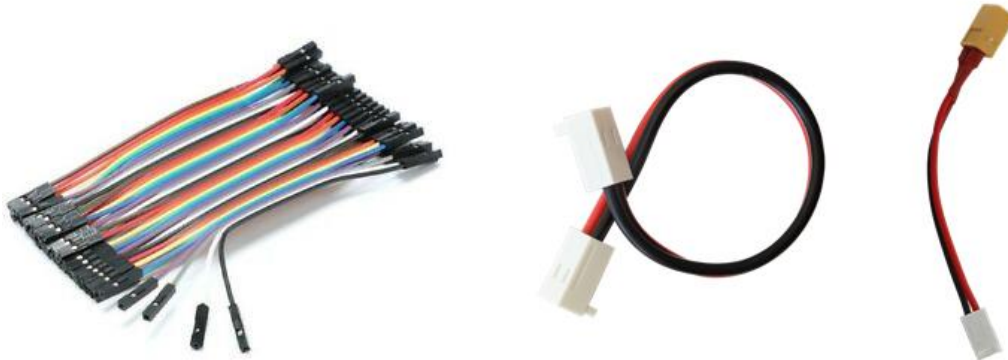


Slika 10. Tipkalo „MRMS switch 2x“ [13]

### 3.9.2. Vodiči

Od ostataka materijala koji su mi potrebni za izradu robota, ostali su još vodiči kojima ću povezivati sve komponente, no nisu svi jednaki vodiči. Za povezivanje Molex konektora trebaju mi vodiči koji na krajevima imaju Molex priključke, a njih sam također nabavio od istog dobavljača i to 10 komada „MRMS Cable KK254-KK254 15 cm (mrm-kk2.54-2.54-15)“ i 5 komada „MRMS Cable KK396-KK396 15 cm (mrm-kk396-396-15)“. Za spojiti bateriju na napajanje potreban mi je vodič sa XT60 konektorom na jednoj strani i Molex KK396 konektorom na drugoj strani, pa sam naručio dva takva konektora „MRMS Cable XT60-KK3.96 10 cm (mrm-xt60-kk3.96-10)“. Naručio sam ih 2 iz razloga kako bi jedan uvijek imao u pričuvi, jer iz prijašnjih iskustva se događalo da je vodič znao pregorjeti ili se jednostavno odvojiti od Molex konektora, a s obzirom kako kod kuće ne posjedujem alate za popraviti takav konektor, smatrao sam boljim rješenjem naručiti jedan dodatan vodič. [14]

Zadnji vodiči koje sam naručio, ujedno i u najvećoj količini su tzv. „Dupont female to female“ vodiči. Njih sam naručio sa E-bay webshopa zbog jeftinije cijene. Dupont vodiči su najjednostavniji vodiči i naručio sam količinu od 80 komada kako bih imao zalihe, ukoliko se na nekim vodičima dogodi prekid ili jednostavno prestanu funkcionirati. [15]



**Slika 11.** Vodiči s lijeve na desnu(Dupont, MRMS Cable KK396-KK396, MRMS Cable XT60-KK3.96) [14], [15]

### 3.9.3. Baterija i punjač za bateriju

Baterija je jedan od najosnovnijih dijelova robota i kao takav može ga se pronaći u svakom robotu, dronu i ostalim elektroničkim napravama koje koriste bežični način rada. Poučen prethodnim iskustvima na praksi u udruzi „Hrvatsko društvo za robotiku“, uvidio sam kako je jedna od najboljih baterija u omjeru cijene i efikasnosti „Turnigy High discharge Li-Po Battery“ koja ima 3 ćelije i daje napon od 11.1V.

Minimalni kapacitet baterije je 1500 mAh što može robota konstantno uključenog sa visokim intenzitetom potrošnje motora i senzora može pokretati otprilike pola sata. No u pravilu se robot uključuje i isključuje konstantno na nekih minutu do dvije prilikom testiranja i programiranja što nam zapravo daje rezultat kako prilikom testiranja i programiranja imamo nekih dva do tri sata mogućnosti rada s jednom baterijom. Naručio sam dvije baterije kako bi mogao konstantno raditi na robotu na način da se jedna baterija puni dok se sa drugom radi na robotu. [16]



Slika 12. Turnigy Li-Po baterija [16]

Za samo punjenje baterije potreban nam je punjač. Što se tiče punjača, smatrao sam kako nema potrebe za naručivanjem posebne vrste punjača, pa sam se odlučio za najjednostavniju varijantu, odnosno naručivanje kod istog dobavljača kao i većine komponenata. Iz tog razloga sam naručio „LiPo battery 2S/3S balance charger (mrm-lipo-char)“ koji ima mogućnost punjenja Li-Po baterija sa dvije ili 3 ćelije i vrlo je jednostavan za korištenje, malen i kompaktan. [17]



Slika 13. Punjač baterije „LiPo battery 2S/3S balance charger (mrm-lipo-char)“ [17]

### 3.9.4. Vijci, matice i odstoynici

Kako bi povezali sve dijelove robota zajedno, potrebni su nam vijci, matice i odstoynici, a najlakši način mi je bio uzeti set vijaka, matica i odstoynika putem E-bay webshopa. Među navedenim vijcima i maticama koristio sam dvije vrste, a to su M2 i M3 vijci. Kako bi povezoao motore i neke robusnije dijelove robota, bili su mi potrebni deblji vijci, odnosno veći, tj. M3 vijci, dok se MRMS komponente povezuju M2 vijcima i u većini slučajeva se prema potrebi postavljaju na odstoynike, kako ne bi došlo do kratkog spoja između elektroničke komponente i podloge na kojoj se ta komponenta nalazi. Toj mjeri predostroynosti se poseže najčešće kada je šasija izrađena od aluminija, no s obzirom kako ću ja šasiju 3D printati PLA materijalom koji nije provodljiv, kratki spoj je gotovo nemoguć. No unatoč tome, postaviti ću komponente na odstoynike radi preglednosti. [18]



**Slika 14.** Set vijaka, matica i odstojnika [18]

### 3.9.5. Omni-Wheel kotači

Omni-wheel kotači, odnosno svosmjerski kotači su ključni u cijelom ovom procesu iz razloga što sam na početku odlučio izraditi robota sa četiri motora razmaknuta jedan od drugog pod kutem od 90 stupnjeva. Sa takvim načinom postavljanja motora bilo je potrebno uključiti svosmjerske kotače, jer se inače robot ne bi mogao kretati. Svosmjerski kotači koje sam ja izabrao su „Nexus Robotics Omni Wheel for Lego NXT/EV3“. Izabrao sam ih iz razloga što su jednostavni za montiranje na gore već odabrane motore, no i zbog same činjenice kako su i sami kotači dobre izrade. Također, iste kotače mi je i preporučila mentorica sa prakse iz razloga što su otporni na udarce i dosta izdržljivi.[19]



**Slika 15.** Svosmjerski kotači „Nexus Robotics Omni Wheel for Lego NXT/EV3“ [19]

Nabavu samih kotača izvršio sam putem Generation robots webshopa i naručio sam 8 komada kotača kako bih imao i rezervne kotače u slučaju da se neki od kotača unište. Ono što je još važno napomenuti je i nabava tzv. „hubova“ ,odnosno priključaka od 3mm kojima sam omogućio spajanje kotača sa motorom direktno pomoću navedenog „hub-a“. Njih sam nabavio sa Nexus robot webshopa. U paket su bili uključeni i vijci za pričvršćivanje „Hub-a“ i kotača na motore. [20]

## **4. Izrada šasije 3D printanjem i 3D dizajniranjem**

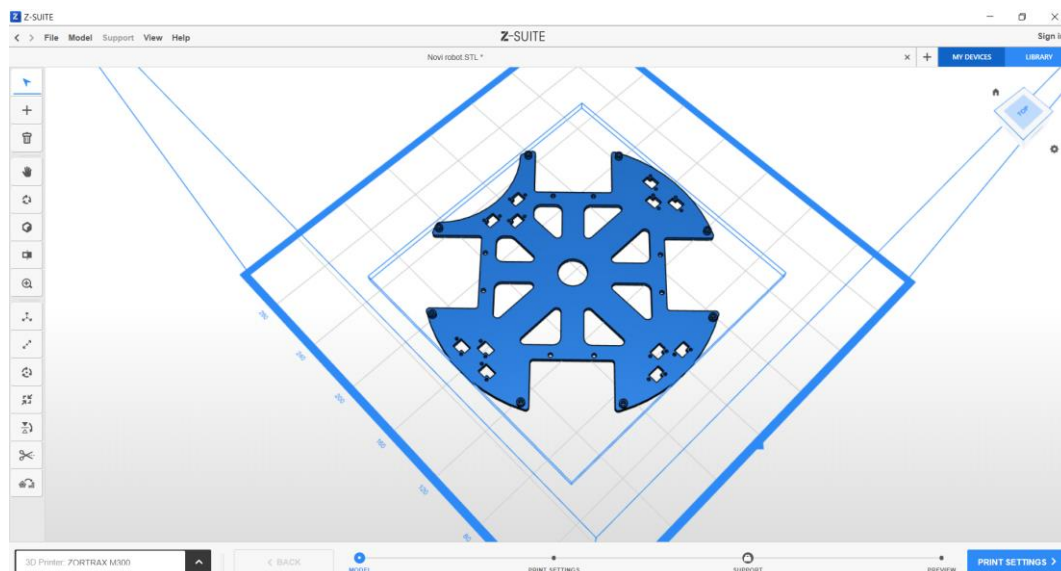
Šasija robota koji će imati namjenu pratiti infracrveni objekt je glavni dio robota na kojeg će se kasnije povezati svaki dodatni dio, kao i svaka moguća elektronička komponenta pomoću vijaka, matica i odstojnika. Prilikom izrade i dizajna same šasije, glavni fokus mi je bio kako što efikasnije i efektivnije iskoristiti prostor, a da sama šasija zadrži čvrstoću. Prilikom izrade i dizajna, također je bilo potrebno paziti i na stvaranje otvora za senzore boje i naravno izraditi pravilno otvore pomoću kojih će se pričvrstiti motori. Sama šasija se sastoji od dva dijela, gornjeg i donjeg.

### **4.1. SolidWorks dizajniranje**

Kao jedan od alata sa kojim sam se već ranije susreo je SolidWorks, a koristio sam ga u svrhe dizajniranja i 3D printanja manjih dijelova, kao npr. manje robotske ruke koje su se mogle pričvrstiti na manje robote. No, nikad nisam 3D modelirao kompletnu šasiju, što mi je bio izazov. No, najprije je potrebno odgovoriti na pitanje što je to uopće SolidWorks? SolidWorks je jedan od najrasprostranjenijih i najkorištenijih CAD (Computer aided design) programa u industrijskom svijetu. Čak bi se moglo reći i kako je SolidWorks ustvari standard u inženjerstvu. Izrađen je od strane francuske tvrtke Dassault Systemes i fokusira se primarno na inženjerskom dizajnu pomoću geometrijskog parametarskog modeliranja. [21]

Prije no što sam krenuo dizajnirati u programu SolidWorks, potrebno je bilo razmisliti na koji način ću dizajnirati samu šasiju. Tada sam shvatio kako ću šasiju morati podijeliti na dva dijela i kako ću graditi i dizajnirati od donjeg dijela prema gornjem. Uzevši u obzir dimenzije samog robota, uzeo sam promjer same šasije 21 cm i nacrtao u alatu SolidWorks skicu kruga od 21 cm koji sam tada naredbom „Extrude“ pretvorio u jedan veliki kružni element iz kojeg sam daljnjim postupcima izrezivao dijelove koji su mi bili potrebni.

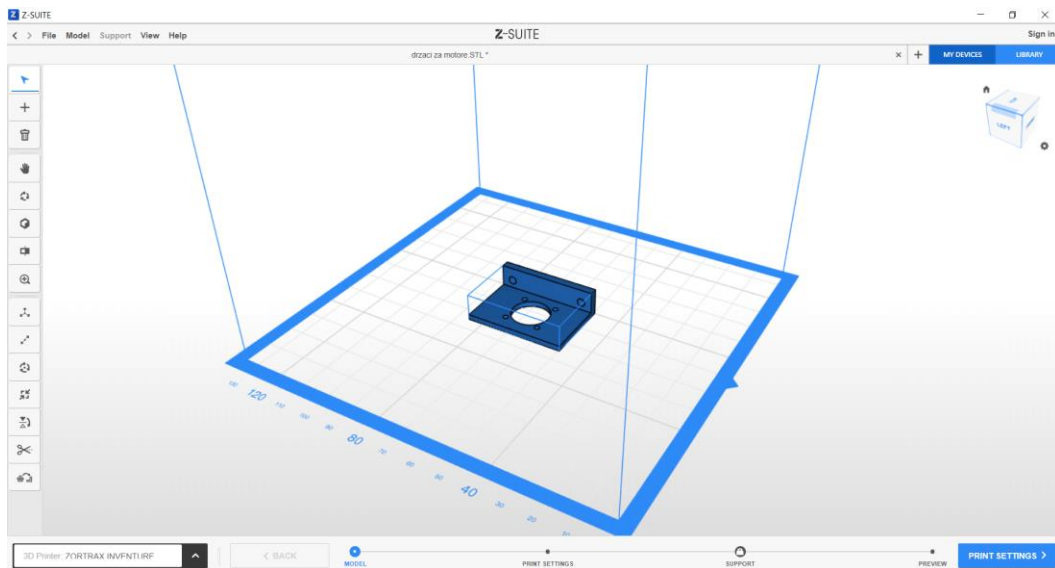
Kada sam imao kružni element iz kojeg sam mogao izrezivati dijelove, prvo sam izrezao prednju stranu polukružno naredbom „Cut“. Prednja strana služi tome kako bi lopta mogla ući u posjed robota. Nakon što sam odredio prednju stranu, tada sam mogao pozicionirati motore, gdje su prednja dva motora pod kutom od 45 i -45 stupnjeva naspram prednje strane i stražnja dva motora pod kutom od 135 i -135 stupnjeva. Kada sam to uzeo u obzir, sukladno tome sam mogao dalje skicirati rupe gdje će se nalaziti kotači. Nakon rupa za kotače krenuo sam na skiciranje provrta za pričvršćivanje motora i šasije, budući su to jedni od najbitnijih dijelova koji moraju biti dosta pritegnuti, a kako ne bi došlo do pucanja, napravio sam provrte koji odgovaraju M3 vijcima. Pažljivo sam izmjerio razmak između dva provrta, kako odstojnici kasnije ne bi bili u dodiru sa motorima. Nakon toga je slijedilo podizanje rubova vanjskih provrta, kako bi se visine izjednačile sa držačima motora koje ću kasnije dizajnirati. Imajući na umu kako ću senzore za očitavanje boje podloge morati montirati na donju podlogu, krenuo sam u skiciranje izgleda senzora zajedno sa njihovim provrtima. Skicirao sam tako svaki senzor na svakoj strani zasebno i kada sam skicirao sve senzore, naredbom „Extruded cut“ sam izrezao sve odjednom. Tada sam već imao gotovu podlogu, no htio sam smanjiti potrošnju print materijala, pa sam izrezao nekolicinu rupa oko sredine koje neće oslabiti čvrstoću same donje šasije, a opet smanjiti će količinu PLA plastike potrebne za print.



**Slika 16.** Donji dio šasije prikazan u programu Z-Suite

Nakon završetka donjeg dijela šasije, krenuo sam u izradu držača za motore koji će povezivati gornji i donji dio šasije. Pažljivo sam uračunao visinu odstojnika, šasije i motora, kako bi se sve moglo savršeno poklopiti gornjim dijelom šasije, a opet da kotači budu na podu, uz idealnu visinu donjeg dijela šasije od podloge do nje od otprilike 5mm – 10mm.

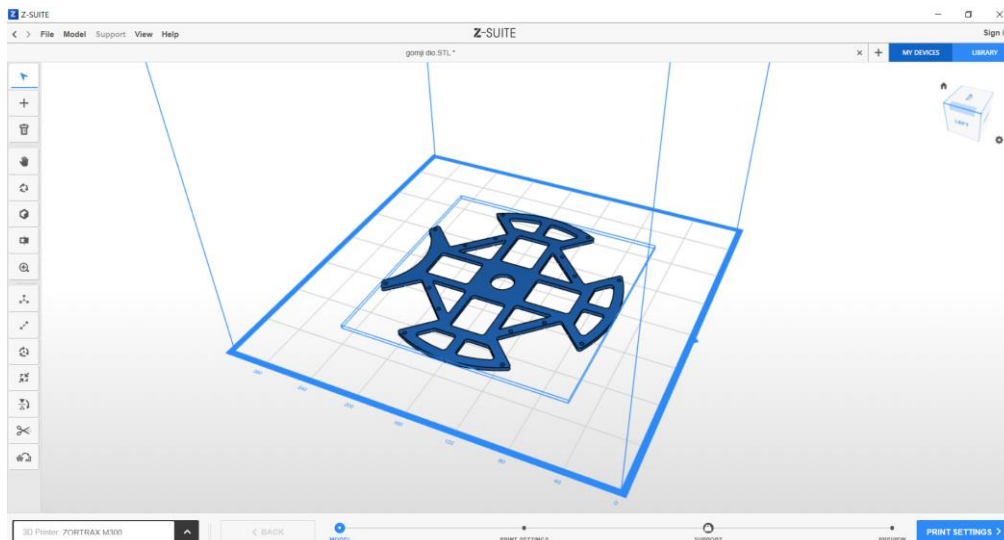
Kada sam krenuo raditi držače za motore, započeo sam izradom skice većeg dijela podloge dužine 35mm i širine 25mm, nakon toga sam naredbom „Extruded cut“ pretvorio skicu u 3D model visine 2mm. Zatim sam skicirao 4 provrta za M3 vijke i 1 veliki provrt kroz koji će motor biti spojen na kotače. Naredbom „Extruded cut“ izrezao sam sve provrte odjednom i preostalo mi je samo još skicirati i podići stijenku koja će biti pričvršćena na donji dio šasije naredbom „Extrude“. Nakon šta sam napravio stijenku, još je bilo potrebno napraviti 2 provrta koja će biti u istoj ravnini sa provrtima na donjem dijelu šasije i naredbom „Extruded cut“ izrezati.



**Slika 17.** Držać za motor prikazan u programu Z-Suite

Jedini dio koji mi je tada preostao za dizajnirati je bio gornji dio šasije. Gornji dio šasije započeo sam na isti način kao i donji dio, tako što sam skicirao krug i naredbom „Extrude“ izvukao visinu kruga 2mm da se stvori 3D objekt. Taj 3D objekt sam prema već zadanim mjerama od donjeg dijela naredbom „Extruded cut“ izrezao na način da mi ostanu rupe za kotače i prednja rupa. Nakon toga sam, također po već odrađenim mjerama sa donjeg dijela šasije napravio vanjske M3 provrte, kojima sam dobio jednake provrte kao i na donjem dijelu, a iz razloga kako bi se podudarali. Isto sam učinio i za unutarnje M3 provrte, odnosno provrte koji služe za pričvršćivanje motora. Nakon toga je slijedio novi dio dizajniranja šasije, gdje sam skicirao rupe po cijeloj preostaloj šasiji kako bih sačuvao što više materijala prilikom provedbe 3D printa. To se najbolje može vidjeti na sljedećoj slici, gdje sam napravio nekoliko rupa na način da one ne narušavaju onu glavnu strukturu koja se sastoji od 2 tzv. stupa koji drže svaku stranu robota. Na taj način sam postigao čvrstoću, a ujedno i uštedio na materijalu.





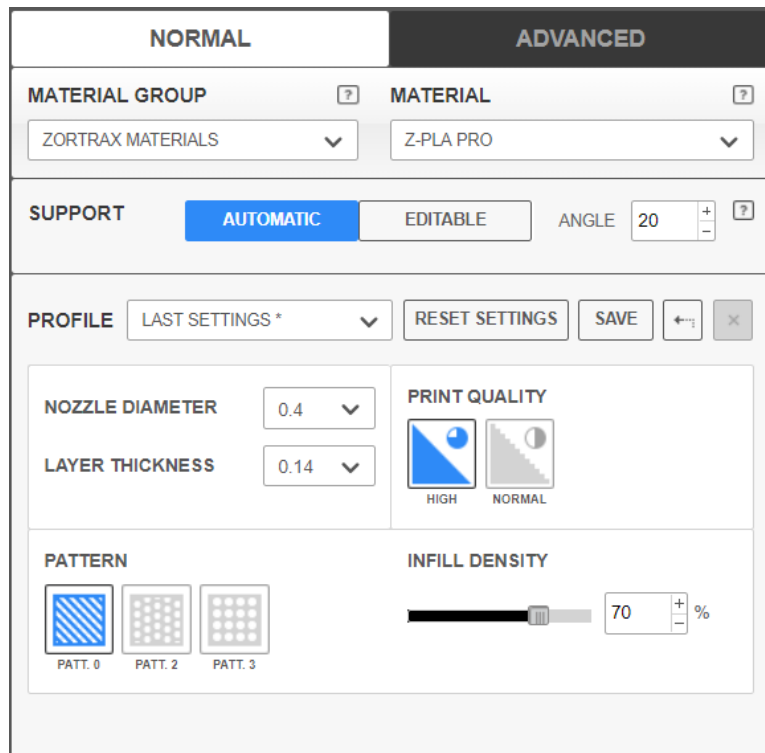
**Slika 18.** Gornji dio šasije prikazan u programu Z-Suite

## 4.2. Printanje Zortrax M300 3D printerom

3D printanje je jedna od grana koja se u današnjem svijetu počinje koristiti sve više i više. Sami 3D printeri se mogu već nabaviti po prihvatljivoj cijeni, pa i ljudi u običnim kućanstvima počinju sve više i više koristiti 3D printer u privatne svrhe.

Za moje svrhe, otišao sam u Varaždinsku udrugu „Robofreak“, gdje mi je predsjednik udruge Ivica Kolarić dozvolio da iskoristim printer „Zortrax M300“ u svrhu printanja već nacrtanih 3D modela. Na korištenje sam tako dobio program Z-suite pomoću kojeg sam namjestio postavke printanja svakog pojedinog elementa. Svaki od zadanih elemenata sam postavio na iste postavke, kako bi print bio jednak na sva 3 dijela šasije.

Postavke koje sam ja koristio odnosile su se na generalne postavke, gdje sam pod grupu korištenih materijala po defaultu ostavio na Zortrax materials iz razloga što printer nije imao dodatne externe materijale, već materijale namijenjene direktno za Zortrax printere. Od ponuđenih materijala u padajućem izborniku izabrao sam Z-PLA PRO Zortraxov PLA materijal koji se koristi u svrhe preciznog printanja, a ujedno se i već nalazio u printeru, pa nije bilo svrhe mijenjati materijal i u programu i još dodatno montirati novi materijal u printer. Što se tiče same debljine slojeva printanja, izabrao sam najmanji sloj u iznosu od 0.14mm iz razloga kako bi dijelovi bili što preciznije isprintani i što čvršći, dok sam recimo popunjavanje rupa unutar printa postavio na 70% kako ne bi došlo do pucanja zbog rupa unutar printa. Sve postavke printa se mogu vidjeti na sljedećoj slici:



**Slika 19.** Postavke printanja u programu Z-suite

Nakon što sam namjestio postavke printanja, program je sam pripremio datoteku tipa .zcode koju sam prenio na memorijsku karticu. Memorijsku karticu sam umetnuo u 3D printer, kojeg sam tada pokrenuo i pomoću digitalnog ekrana i nekoliko kontrola pritisnuo pokretanje printanja moje datoteke. Donjem dijelu šasije je bilo potrebno 8 sati i 17 minuta kako bi se isprintao i bilo je potrebno 92 grama PLA materijala. Dok je gornjem dijelu šasije bilo potrebno 6 sati i 28 minuta za printanje i 73 grama PLA materijala. Jednom držaču za motore je bilo potrebno 32 minute kako bi se isprintao i 5 grama PLA materijala.



**Slika 20.** Izvješće o potrebnom vremenu i materijalu za print gornjeg dijela šasije

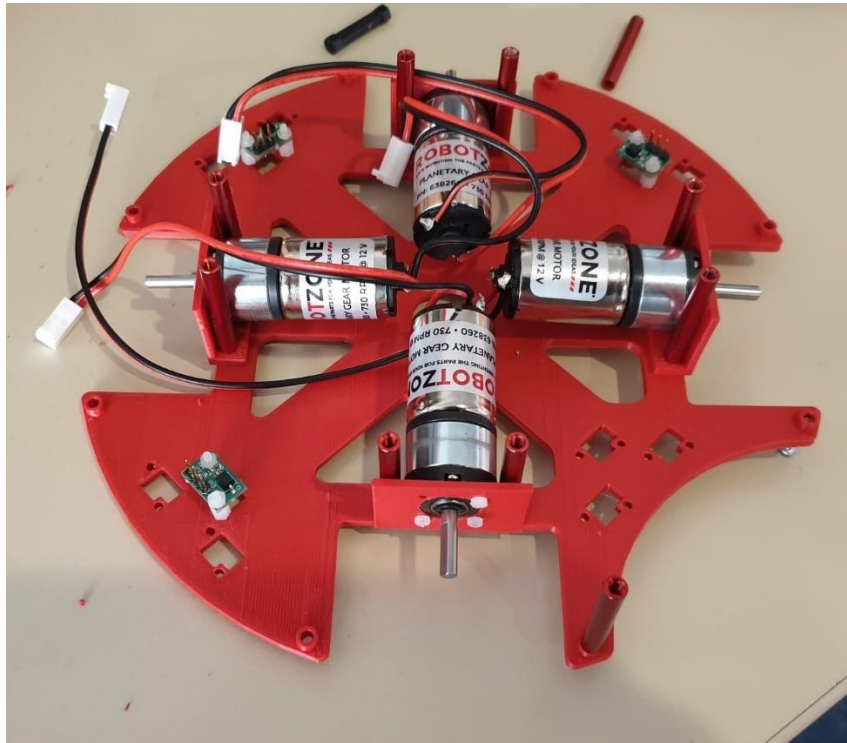
## 5. Izgradnja robota

Za izgradnju robota i spoj elektroničkih elemenata bilo je potrebno najviše vremena. Sama izgradnja trajala je otprilike tjedan dana iz razloga što su dijelovi sitni, pa je u većini slučajeva teško doći do mjesta gdje bi vijak i matica trebali biti. Iz tog razloga sam se snalazio i koristio dodatnim alatom kao što je Hemostat, a kako bi mi na nezgodnim mjestima bilo lakše pridržavati maticu, dok uvijam vijke odvijačem. Kada se print završio i kada su svi dijelovi bili spremni, tada sam krenuo na pričvršćivanje dijelova.



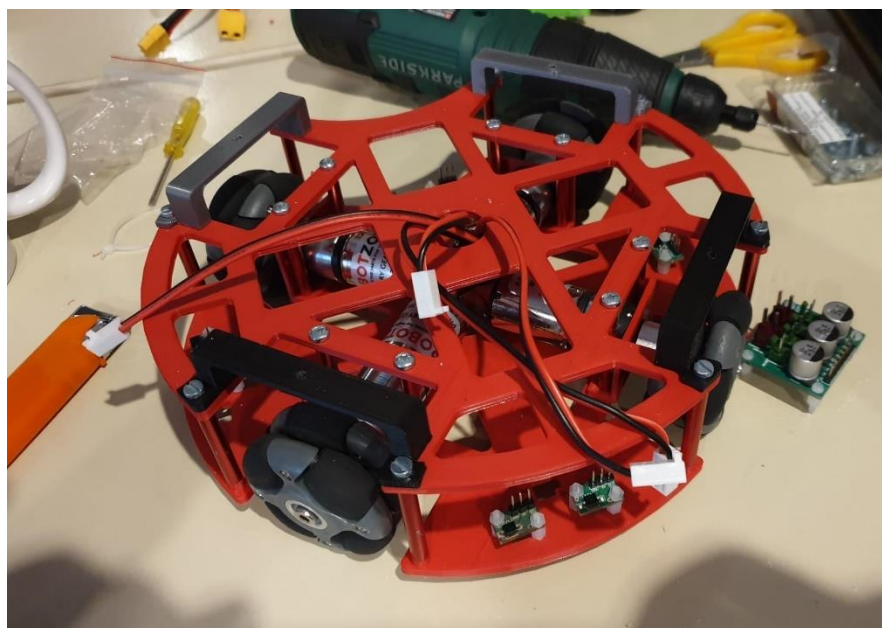
**Slika 21.** Isprintani dijelovi šasije

Rad sam započeo tako što sam najprije zalemio na motore vodiče koji na jednom kraju sadrže Molex KK254 konektore, a kako bih motore mogao kasnije spojiti na kontrolere motora. Razlog što sam prvo lemio je taj kako ne bi kasnije oštetio šasiju lemilicom, a i bilo bi nespretno lemiti kada su motori između gornjeg i donjeg dijela šasije. Nakon lemljenja, motore sam pričvrstio sa četiri M2 vijka na držače za motore. Nakon toga sam držače zajedno sa motorima pričvrstio pomoću dva velika crvena odstoynika i M3 vijaka za donji dio šasije. Tada su motori bili montirani, i sve što je još trebalo je postaviti odstoynike i pričvrstiti ih vijcima na vanjske provrte donjeg dijela šasije. U tom trenu sam se sjetio kako bi bilo pametno pričvrstiti i senzore boje, a upravo iz razloga da kasnije ne bih imao problema sa pričvršćivanjem između gornjeg i donjeg dijela šasije.



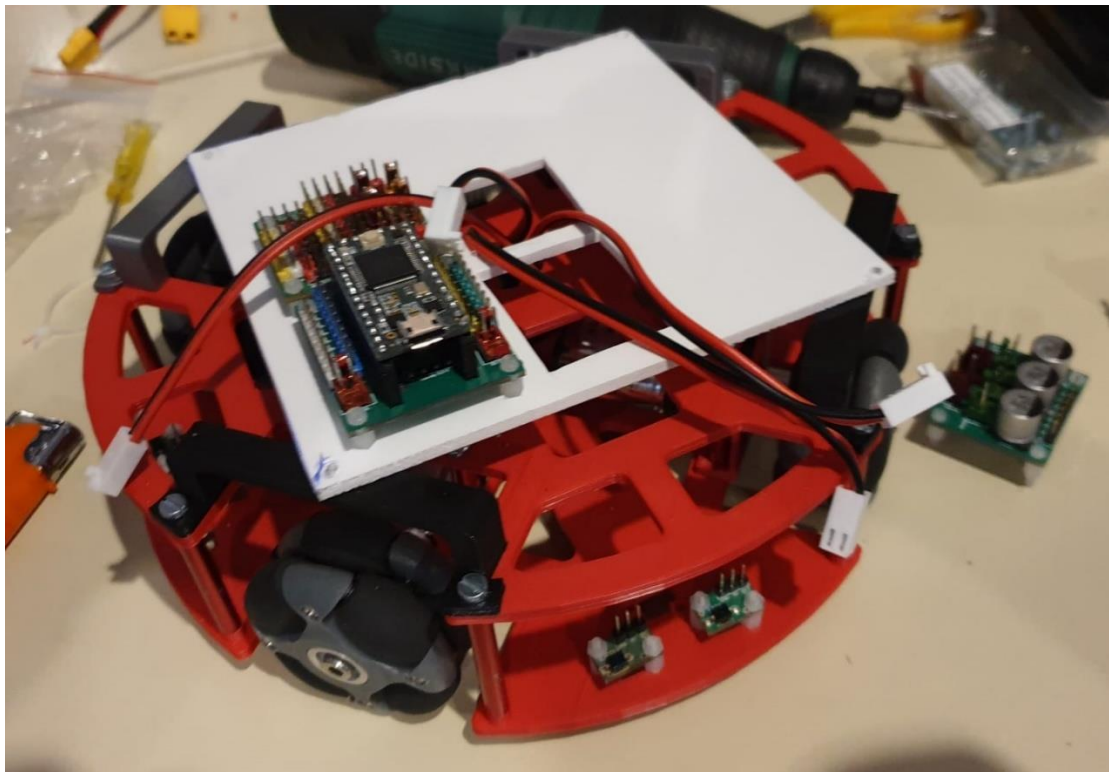
**Slika 22.** Pričvršćeni motori i senzori na donji dio šasije

Jednom kada sam pričvrstio sve potrebne senzore i motore na donji dio šasije, tada sam prvo provukao vodiče motora kroz gornji dio šasije, a tek onda sam krenuo na pričvršćivanje gornjeg dijela šasije zajedno sa donjim dijelom šasije. Pažljivo sam pričvrstio odvijačem sve vanjske dijelove šasije kako bi kompletna šasija bila što kompaktnija i nakon toga sam krenuo na pričvršćivanje motora sa gornjim dijelom šasije.



**Slika 23.** Spoj gornjeg i donjeg dijela šasije

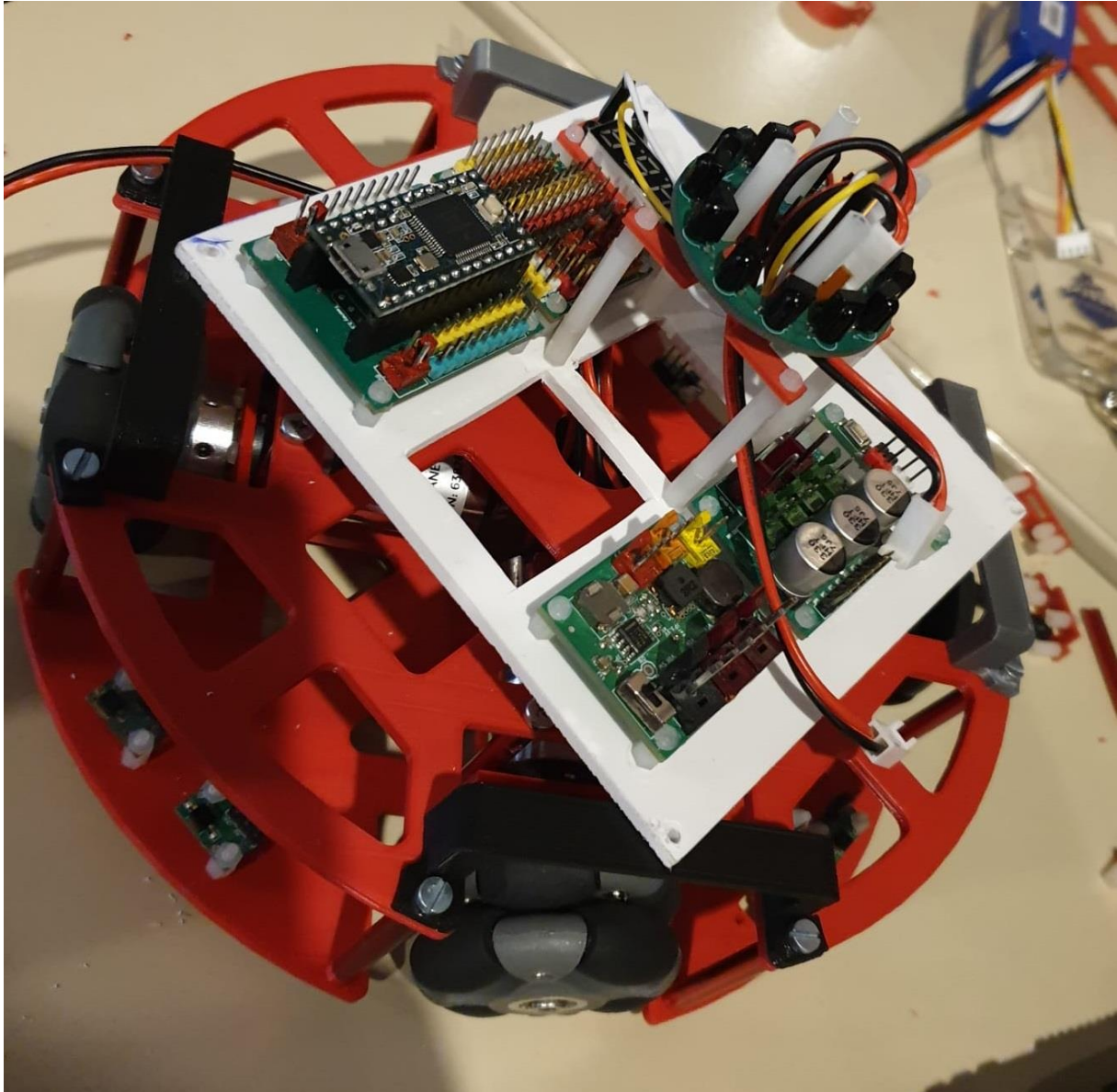
Sljedeći korak u izgradnji robota je bio postavljanje elektroničkih elemenata. Kako bih izbjegao kratki spoj, a i dugotrajno printanje materijala, odlučio sam nabaviti plastičnu mekanu pločicu u obliku A4 papira, koju sam izrezao prema dimenzijama samog robota. Izabrao sam je iz razloga što je jednostavna za izrezivanje i stvaranje provrta bušilicom, te kako bih mogao prema potrebi stvarati nove provrte za pričvršćivanje samih elektroničkih elemenata. Kao što možete vidjeti na slici 20. najprije sam izradio jednu rupu u sredini pločice kako bi kroz nju provodio vodiče, a druga rupa je bila namijenjena za pridržavanje baterije. Sljedeći korak je bio izbušiti provrte na krajevima, kako bi montirao pločicu na šasiju. Sa montiranjem pločice na šasiju sam još pričekao iz razloga kako bih lakše mogao postaviti sve elektroničke komponente na pločicu.



**Slika 24.** Plastična pločica i elektronički elementi

Kako bih lakše napravio provrte za elektroničke pločice, najprije sam na sve elektroničke pločice odvijačem pričvrstio vijke i odstoynike. Malo jačim pritiskom odstoynika na plastičnu pločicu, ostali su tragovi pomoću kojih sam finom bušilicom napravio provrte M2 veličine kako bi odstoynike mogao pričvrstiti na pločicu pomoću matice. Prvo sam montirao Mikrokontroler Teensy 3.2 zajedno sa razvodnom pločicom za mikrokontroler, a nakon toga sam montirao razvodnu pločicu za napajanje.

Istim principom sam nakon toga pričvrstio i kontrolere motora, napajanje i kompas. Nakon toga sam uzeo senzor za loptu na kojeg sam pričvrstio nekoliko odstoynika, a kako bih ga montirao na veću visinu. Senzor lopte bilo je tako potrebno montirati iz razloga da mu ostali elektronički elementi ne bi smetali prilikom prijema infracrvenih zraka koje lopta šalje, a iz istog razloga je sam taj senzor u sredini šupalj, dakle kako bi se vodiči mogli provesti kroz sredinu. Jednom kada sam pričvrstio sve elemente na šasiju, robot je bio spreman za povezivanje elektroničkih dijelova sa mikrokontrolerom.



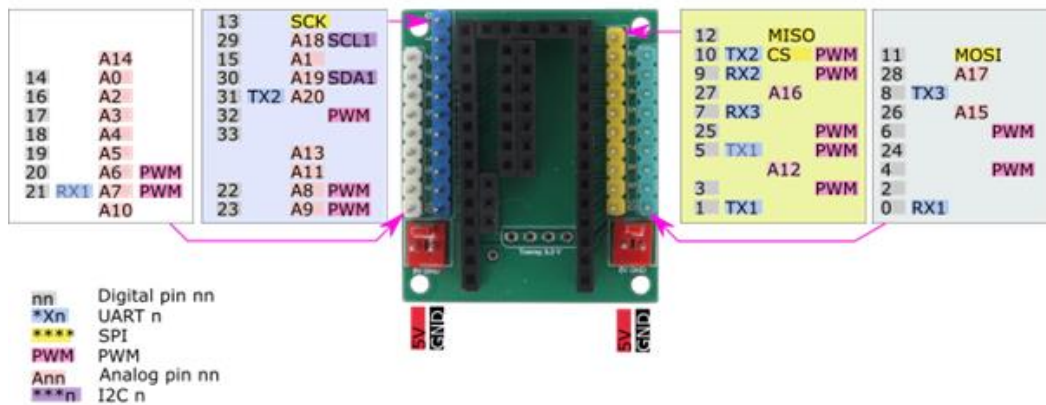
**Slika 25.** Mehanički spoj svih elemenata na šasiju

## 6. Spajanje elektroničkih elemenata

Spoj elektroničkih dijelova je jedan od najtežih dijelova izrade robota iz razloga što iziskuje potpunu koncentraciju. Mjesta za pogrešku nema, jer svaki krivo spojeni vodič može prouzrokovati kratki spoj koji značajno može oštetiti elektroničke elemente. Takav scenarij bi me vratio skoro pa na početak iz razloga što bih morao većinu elektroničkih elemenata ponovno nabaviti, a i sam proces ponovne montaže i demontaže elemenata bi bio veoma kompliciran. Iz tog razloga sam krenuo veoma pažljivo u proces spajanja vodiča.

Najprije sam krenuo u spoj svih napajanja pomoću Molex KK254 kablova. Molex KK254 konektori na pojedinim elektroničkim pločicama su podijeljeni na narančaste koji služe za napon od 3.3V i crvene koji služe za napon od 5V. Sa napajanja sam tako prvo krenuo sa crvenim Molex KK254 konektorima. Napajanje sam povezao sa crvenim molex konektorom na razvodnoj pločici za mikrokontroler, a nakon toga sa izlaznog Molex konektora sam povukao vodič prema razvodnoj pločici za napajanje. Tako sam završio niz elemenata kojima je potrebno napajanje od 5V. Nakon toga krenuo sam na narančaste Molex 254 konektore gdje sam sa napajanja krenuo prema ulaznom Molex 254 konektoru od senzora za loptu. Sa izlaznog konektora sam vodič dalje spojio na ulazni konektor razvodne pločice za napajanje gdje je završio niz elemenata kojima je potrebno 3.3V. Sljedeći korak je bio povezati motore sa napajanjem putem većih Molex KK396 konektora, što sam i uradio i tako završio niz elemenata kojima je potrebno napajanje direktno iz baterije. Ovime sam završio spajanje napajanja pomoću Molex konektora i preostalo mi je sa razvodne pločice za napajanje spojiti preostale elemente koji nemaju Molex konektore. Kada smo Molex konektorima doveli napon na razvodnu pločicu, tada možemo pomoću razvodne pločice dalje dupont vodičima spojiti preostale elemente na napajanje. Tako sam redom spojio dupont kablove na +5V i GND sa razvodne pločice na senzor boje na istoimene pinove. Učinio sam tako za svaki od 8 senzora boje. Nakon toga je još preostalo spojiti napajanje za tipkalo i kompas na isti način kao i senzore boje. Tada je u stvari napajanje svih komponenata bilo u potpunosti spremno, no preostalo je samo spojiti vodič sa XT60-Molex KK396 konektorima spojiti na napajanje kako bi na XT60 kraj mogli spojiti bateriju. U principu, jednom kada smo spojili sve elemente napajanjem i povezali u jednu cjelinu, tada je najpametnije izvršiti provjeru kako ne bi došlo do kratkog spoja. Nakon provjere je tehnički moguće uključiti bateriju kako bi provjerili dolazi li do svakog elektroničkog elementa napon. Najčešće elektronički elementi na sebi imaju nekakav LED indikator pomoću kojeg možemo vidjeti kada je uključen.

Nakon napajanja, slijedi povezivanje ulazno izlaznih pinova elektroničkih elemenata sa mikrokontrolerom. To je jedan od najbitnijih dijelova budući da senzori mogu raditi, no u potpunosti su neiskoristivi ukoliko nam ne daju signal koji kasnije upotrebljavamo za snalaženje u okolini. Ista stvar je i sa izlaznim elementima, koji nemaju svrhu ukoliko ne možemo upravljati njima.

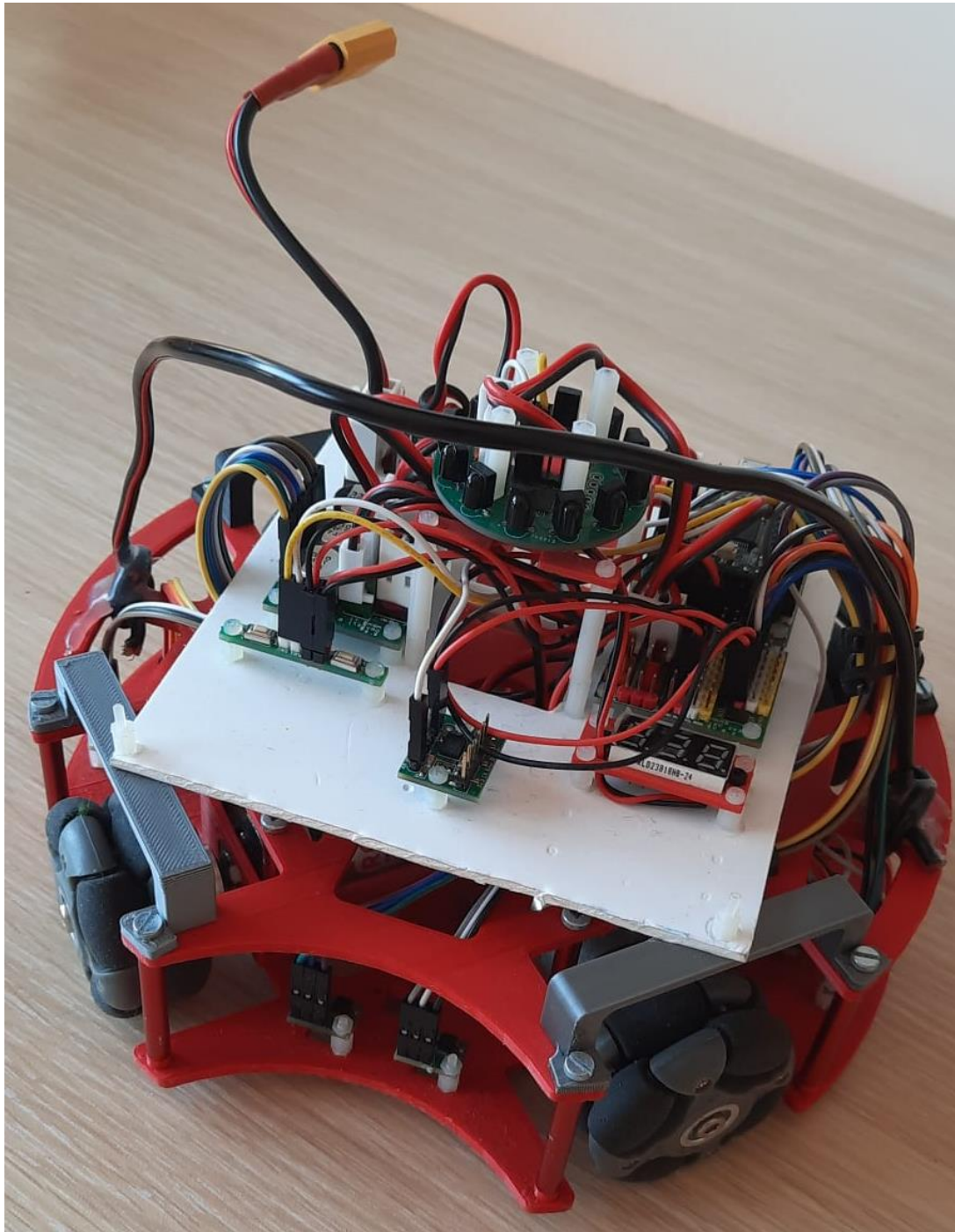


Slika 26. Prikaz pinova razvodne pločice [5]

Krenuo sam od izlaznih elemenata odnosno motora iz razloga što se oni spajaju na posebne PWM pinove kako ih ne bi slučajno zauzeo sensorima gdje bi se onda kasnije doveo u neugodnu situaciju premještanja pinova. Na kontrolerima motora postoje označeni pinovi za motore sa oznakama M1, M2, M3, i M4 koji redom označavaju motore, pa sam tako krenuo redom i spajati. Dva M1 pina sam spojio na pinove 4 i 6 razvodne pločice mikrokontrolera. Sljedeća dva M2 pina sam spojio na pinove 20 i 21 razvodne pločice mikrokontrolera. Sljedeća dva M3 pina sam spojio na pinove 10 i 9 razvodne pločice mikrokontrolera. I na kraju zadnja dva M4 pina sam spojio na pinove 23 i 3 razvodne pločice mikrokontrolera čime sam završio spoj motora na mikrokontroler i na napajanje, što znači da su spremni za rad. Sljedeći korak je bio spojiti sve ulazne elemente, odnosno senzore. Jedan od najbitnijih senzora je zapravo senzor za praćenje lopte, pa sam tako njega spojio prvog. Na senzoru lopte se nalaze 2 pina koja služe za ispis analognih vrijednosti od kojih pin DEG služi za ispis vrijednosti kuta lopte, dok drugi pin pod nazivom CM služi za ispis udaljenosti senzora od lopte. DEG pin sam spojio na A19 pin razvodne pločice, dok sam CM pin spojio na A1 pin razvodne pločice. Sljedeći senzor kojeg sam spajao je bio kompas gdje sam koristio ROLL pin koji sam spojio na analogni A8 pin razvodne pločice.



Sljedećih 8 senzora sam sve spojio na isti način, a to su senzori boje. Senzori boje imaju pin OUT koji je namijenjen za ispisivanje analognih vrijednosti, gdje sam taj pin na svakom od 8 senzora povezivao redom s lijeve strane robota prema zadnjoj strani robota u smjeru kazaljke na satu na pinove A10, A5, A4, A3, A2, A0, A14, A12. Nakon senzora linije preostalo mi je još samo spojiti oba tipkala, što sam i učinio spojivši OUT pinove na digitalne pinove 27 i 7 sa razvodne pločice mikrokontrolera. [5]



**Slika 27.** Finalna verzija robota

## 7. Programiranje

Programiranje je po svojoj definiciji informatička disciplina koja se temelji na osmišljavanju i izrađivanju računalnih programa. Potrebno je poznavati sintaksu određenog programskog jezika kako bi se moglo njime koristiti te doći do rješenja zadanog problema. Kako bi se neki programski jezik koristio, potrebno je susreti se sa problemom. Kada se susretnemo sa problemom, tek tada možemo iskoristiti programiranje u svoju korist u svrhu rješavanja samog problema pomoću računala. No, prije rješavanja problema potrebno je shvatiti što je to u stvari programski jezik. Prema definiciji, programski jezik je formalni jezik kojim se u računalu opisuju algoritmi, dok se sam algoritam može smatrati zapisanim programom u određenom programskom jeziku. Programski jezik koji ću ja koristiti u svome radu je Arduino, koji je u principu C++ sa nekim dodatcima i dodatnim funkcijama. Sam programski jezik se koristi u Arduino IDE razvojnoj okolini. [22], [23]

### 7.1. Arduino IDE

Arduino razvojna okolina je u principu jedan minimalistički uređivač teksta u koji se upisuje kod prije samog prenašanja teksta na željeni mikrokontroler. Arduino okruženje ima opciju provjere koda, kako bi ustanovili da kod nema greške i da je spreman za prijenos na mikrokontroler. Također, ukoliko se pritisne gumb za prijenos podataka tada se program prvo kompajlira i nakon toga prenosi na mikrokontroler. Kada dođe do greške, arduino sam izbacuje liniju koda u kojoj se dogodila greška ( u nekim slučajevima pokazuje na liniju ispod ) pa se na taj način vrlo jednostavno greške mogu ispraviti. U samom okruženju ima još dosta mogućnosti kao što je to na primjer Serial monitor na kojeg možemo ispisivati direktno sa mikrokontrolera što može puno pomoći prilikom rješavanja problema, ili ispitivanja senzora. [23]

### 7.2. Testiranje robota

Kada sam spojio sve elektroničke komponente i pričvrstio sve mehaničke dijelove na šasiju, robot je bio spreman za prvo testiranje. Priključio sam bateriju na napajanje i uzeo robota u ruku kako se ne bi kojim slučajem dogodilo, da je na Teensy 3.2 mikrokontroleru prenesen neki testni kod koji bi igrom slučaja pokrenuo motore. Pokrenuo sam robota pomoću tipkala na elektroničkoj pločici za napajanje i uključio robota. Kada sam ustanovio da svi senzori rade pomoću led indikatora na sensorima i kako se robot ne miče, spustio sam ga na stol i isključio.

Kako bi dalje istestirao čitaju li senzori vrijednosti, morao sam u arduino programskom jeziku pokrenuti svaki senzor zasebno i na serijski monitor ispisati vrijednost svakog pojedinog senzora naredbom `Serial.print()`; No prije samog ispisa bilo je potrebno pripremiti biblioteke za same senzore, stoga sam skinuo biblioteke za MRMS sustave zajedno sa početnim programom koji unaprijed uvrštava sve potrebne biblioteke za MRMS komponente. [24]

Jednom kada sam imao spremne biblioteke, tada sam krenuo sa pisanjem koda. Prvo što sam učinio je bilo inicijaliziranje ulaza i izlaza u funkciji „`setup`“. Pomoću funkcija iz biblioteka dodao sam svaku ulaznu i izlaznu komponentu zasebno. Pa sam tako u funkciju `motors.add()` upisivao 4 argumenta od kojih su prva dva argumenta bili pinovi prvog motora, dok su druga dva argumenta bila tipa `bool` i određivali su smjer vrtnje motora. Kasnije sam pomoću druga dva argumenta mijenjao smjer vrtnje motora kako bi ih kalibrirao za daljnje funkcije. Sljedeći po redu je bio senzor lopte kojega sam dodao pomoću funkcije `irFinders.add()` gdje su se upisivala 2 argumenta, a to su oba pina senzora za loptu, gdje je prvi argument kut između lopte i senzora, dok je drugi argument udaljenost između lopte i senzora. Nakon toga sam dodao kompas pomoću funkcije `imu.add()` koja prima jedan argument odnosno pin na koji je imu spojen. Pomoću funkcije `reflectanceSensors.add()` sam dodao svih 8 senzora boje, funkcija prima samo jedan argument, a to je pin na koji je senzor spojen. Pomoću direktne arduino funkcije `pinMode()` sam dodao još dva tipkala. Funkcija `pinMode()` ima 2 argumenta, prvi je pin na koji je senzor spojen, dok je drugi tip senzora odnosno je li on `input` ili `output`.

```
void setup() {

    motors.add(4, 6, false, true); //45° (relative, 0 degrees is the robot's front, angle increases counterclockwise)
    motors.add(20, 21, false, true); //135°
    motors.add(10, 9, true, true); //-135°
    motors.add(23, 3, true, true); //-45°

    //Add IR ball sensors - unchecked
    irFinders.add(A19, A1); // Angle and distance - check function declaration

    //Add IMU
    imu.add(A8);

    //Line sensors
    reflectanceSensors.add(A10); // 0, 90°
    reflectanceSensors.add(A5); // 1, 180°
    reflectanceSensors.add(A4); // 2, -90°
    reflectanceSensors.add(A3); // 0, 90°
    reflectanceSensors.add(A2); // 1, 180°
    reflectanceSensors.add(A0); // 2, -90°
    reflectanceSensors.add(A14); // 1, 180°
    reflectanceSensors.add(A12); // 2, -90°
    reflectanceSensors.eepromRead();//Reading EEPROM calibration data into RAM

    pinMode(27, INPUT);
    pinMode(7, INPUT);
}
```

**Slika 28.** Inicijalizacija senzora

Nakon inicijalizacije Funkcijom Serial.print() smo mogli posebno prikazati na zaslonu vrijednost svake komponente koju imamo osim motora, no njih smo mogli upravljati funkcijom motors.go() koja ima 2 argumenta, a to su brzine lijevih motora odnosno desnih motora, što baš i nije praktično prilikom korištenja 4 motora i svosmjerskih kotača, stoga sam koristio funkciju motors.goOmni koja ima 4 argumenta od kojih je jedan koji određuje brzinu kretnje, drugi koji određuje smjer kretnje u odnosu na prednju stranu robota, treći argument koji služi za uvrštavanje kompasa kako ne bi zastranio sa puta i četvrti argument koji služi za maksimalnu moguću brzinu. Kako bih kalibrirao kretnje motora i uskladio ih sa funkcijom za početak, u prazan program koji je imao samo inicijalizaciju sam u glavnu petlju utipkao motors.goOmni(BRZINA, 0, 0, BRZINA); gdje sam postavio smjer prema naprijed, bez kompasa kako mi ne bi ometalo kretnje motora. Kod brzine sam stavio konstantu BRZINA koju sam postavio na 40. Kada sam ustanovio koji motori se kreću u pogrešnom pravcu u inicijalizaciji sam pomoću 3. i 4. argumenta funkcije motors.add() promjenio početni smjer vrtnje motora. Kroz metodu pokušaja i pogrešaka sam na kraju uspio usmjeriti sve motore kako bi se podudarali sa funkcijom motors.goOmni(). Kroz naredbu Serial.print() sam kasnije još isprintao sirove vrijednosti svakog senzora pomoću funkcije analogRead(). To mi je omogućilo pročitati direktnu vrijednost svakog senzora na serijskom monitoru.

```
Serial.print(analogRead(A10));  
Serial.print(" ");  
Serial.print(analogRead(A5));  
Serial.print(" ");  
Serial.print(analogRead(A4));  
Serial.print(" ");  
Serial.print(analogRead(A3));  
Serial.print(" ");  
Serial.print(analogRead(A2));  
Serial.print(" ");  
Serial.print(analogRead(A0));  
Serial.print(" ");  
Serial.print(analogRead(A14));  
Serial.print(" ");  
Serial.print(analogRead(A12));  
Serial.print(" ");  
Serial.print(analogRead(A19));  
Serial.println(" ");
```

**Slika 29.** ispisivanje sirovih vrijednosti svakog senzora

Kada sam ispitao sve vrijednosti svih senzora, robot je bio spreman za daljnje programiranje i daljnji rad sa sensorima i motorima.

## 7.3. Programiranje robota

Samo programiranje robota u svrhu ispunjavanja zadatka je potrebno postepeno odrađivati. Isto kao što sam i do sad postepeno odradio testiranje robota, tako sam odradio i programiranje samog robota. Krenuo sam top-down metodom gdje mi je za početak najvažnije bilo kako napraviti da robot prati infracrveni objekt odnosno loptu. Shvatio sam da ću to postići kombinacijom dvije funkcije, a to su funkcija `motors.goOmni()` i `irFinders.irSource().angle`. Funkciju `irFinders.irSource().angle` sam zapisivao u varijablu `smjer` konstantno, no ispitivanjima sam shvatio kako mi vrijednost same funkcije izbacuje sirove vrijednosti između 0 i 950. Slijedom uvrštavanja same varijable, `smjer` samo uspio iskorigitirati na vrijednost smjera lopte tako da uvijek bude između -180 i 180. Takvom korekcijom sam uspio vrijednost senzora lopte prilagoditi funkciji `motors.goOmni()` kako bi kasnije u 2. argument mogao uvrstiti varijablu `smjer`, budući da 2. argument čita vrijednosti između -180 i 180 i kreće se prema tom smjeru. Kada sam uvrstio `smjer` u funkciju `motors.goOmni` robot je pratio loptu, no shvatio sam kako se robot nakon otprilike minute i sudaranja sa drugim objektima, gotovo uvijek rotira na kompletno drugu stranu. Iz tog razloga, morao sam u program uvrstiti kompas. Budući kako MRMS sistem već ima ugrađene funkcije za poravnanje po kompasu, samo sam uvrstio funkciju `rotationToMaintainHeading(headingToMaintain)` u treći argument, jedino što je bilo potrebno je bilo zapamtiti u varijablu `headingToMaintain` vrijednost 0 prilikom pokretanja robota. Na taj način je robot znao koji mu je početni smjer kojeg se treba pridržavati. Kasnije sam se igrao sa tipkalima i uvrstio mogućnost napadanja na suprotni gol gdje bih na početku programa zapravo postavio vrijednost `headingToMaintain` varijable na 180.

```
smjer =irFinders.irSource().angle;
map(smjer, 0, 950, -180, 180);
smjer = smjer -30;
normalized(smjer);
smjer = -smjer;
Serial.print(smjer);
Serial.print(" ");

if (digitalRead(27) == 1){
  pali = 1;
  headingToMaintain = 180;
}
else if (digitalRead(7) ==1){
  pali = 1;
  headingToMaintain = 0;
}
```

**Slika 30.** Korekcija smjera lopte i postavljanje kompasa na početnu vrijednost

Kada sam uspio korigirati kompas i smjer dolaska lopte sa usmjerenjem kretanja motora, tada je robot već pratio loptu i sve što je dalje trebalo je u već postojeći program pomoću nekoliko grananja uvrstiti senzore za očitavanje boje. Kako bi senzori boje funkcionirali normalno, na početku je trebalo zapamtiti vrijednosti svakog pojedinog senzora, što sam i učinio upisavši njihove vrijednosti u varijable kojima sam zbrojio konstantu SENZORI, koja se kasnije prema potrebi može mijenjati, no u vrijeme pisanja ovog rada iznosila je 200. Konstantu SENZORI sam zbrojio iz razloga kako se bijela boja očitava na vrlo visokim vrijednostima od preko 1000, dok je zelena podloga površine terena na otprilike 400. Kako robot uvijek započinje na zelenoj površini, ideja mi je bila kako bi na početku robot zapamtio vrijednosti površine + toleranciju koju sam morao uvrstiti zbog robotove brze kretnje. Te početne vrijednosti onda kasnije provjeravam sa trenutnim vrijednostima senzora u svakom trenutku. Na taj način provjeravam postoji li bijela linija. Senzori su raspoređeni taktički na 4 strane robota, kako bi znao sa koje strane je učitana linija i u koju stranu je potrebno odmaknuti robota od linije. Zbog urednosti programa, provjeru postoji li linija, izvršavam u funkciji senzoriLinije(), gdje postavljam varijable svakog senzora posebno na true odnosno false. Kasnije u kodu grananjem vršim provjeru tih istih varijabli i ukoliko je linija očitana, robot staje na 100 milisekundi i nakon toga se kreće unatrag za konstantu LINIJAMS iz razloga jer u većini slučajeva robot ne stigne zakočiti i prijeđe liniju. Nakon toga, robot kreće u stranu još za jednu konstantu VRATI, no kada se vraća, svaku milisekundu provjerava je li učitao bilo koji od preostalih 6 senzora, a ukoliko ih je očitao, robot staje nakon 300 milisekundi i izlazi iz petlje.

```

if (SS1 || SS2){
    motors.goOmni(0, 0, 0, 0);
    delay(100);
    motors.goOmni(BRZINA, 180, rotationToMaintainHeading(headingToMaintain), BRZINA);
    delay(LINIJAMS);
    for(int i = 0; i<VRATI; i++){
        motors.goOmni(BRZINA-20, 180, rotationToMaintainHeading(headingToMaintain), BRZINA);
        senzoriLinije();
        if (ZS1 || ZS2 || LS1 || LS2 || DS1 || DS2){
            motors.goOmni(0, 0, 0, 0);
            delay(300);
            break;
        }
        delay(1);
    }
}
}

```

**Slika 31.** Jedno od grananja za ispitivanje linije

Kada sam riješio problem linije, tada je program u potpunosti bio gotov i funkcionalan, na kraju sam se malo još poigravao sa robotom i brzinama i ustanovio kako robot najbolje funkcionira na nižim brzinama, no i kako se zna dogoditi propust prilikom očitavanja linije zbog brzine robota.

## 8. Zaključak

Izgradnja jednog ovakvog funkcionalnog sustava je vrlo komplicirana stvar, no gradnjom kompliciranih sustava ljudi poboljšavaju svoje sposobnosti i vještine i postižu rezultate o kojima nisu mogli ni sanjati. S namjerom da naučim nešto novo, krenuo sam u izradu ovog kompleksnog sustava koji ustvari ima vrlo jednostavnu svrhu, a to je igranje nogometa. Tehnologija i znanje koje sam koristio za izradu ovog robota mogli bi se iskoristiti i u drukčije svrhe, kao na primjer izradu robota koji može služiti u istraživačke svrhe na mjesecu ili čak i Marsu. Sama izrada je bila veoma dugotrajan proces gdje sam od početka do kraja naučio neke bitne lekcije te stekao znanja i potrebne vještine.

Kroz nabavu sam shvatio kako je najbitnije gledati poseban omjer cijene i kvalitete i kako je uvijek važno unaprijed isplanirati sav potreban materijal, a kako bi se kasnije lakše organizirao prilikom izrade samog robota. Od početka nabave gledao sam kako ću većinu proizvoda nabaviti od jednog dobavljača. Prilikom dizajniranja, kada sam uključio SolidWorks kako bih izradio šasiju, nisam imao viziju o konačnom proizvodu, no kada sam krenuo sa početnim radnjama, sve više i više sam pred očima imao sliku šasije i kako bi ona ustvari trebala izgledati. Prilikom 3D printanja shvatio sam kako je s današnjim 3D printerima potrebno biti veoma strpljiv i kako se ništa ne može izgraditi na brzinu. Shvatio sam i kako je većina 3D printera automatizirana i sve što je potrebno za jedan print je dobar dizajn i malo finog podešavanja. Kada sam mehanički krenuo spajati robota, mislio sam kako nikad neću uspjeti završiti pričvršćivanje malim M2 vijcima, no unatoč tome, nakon prvih par dana moje vještine su se mnogo poboljšale, a što mi je omogućilo lakše i brže pričvršćivanje šasije. Prilikom spajanja elektronike naučio sam kako je najbitnije provjeriti svaki vodič više puta, a kako bi se uvjerio da je spojen na pravo mjesto. Programirajući robota shvatio sam kako je najbitnije graditi program postepeno, a kako bih izbjegao previše problema kasnije.

Povezavši sva prethodna poglavlja u jednu cjelinu, shvatio sam da je najbitnije naučiti imati kontrolu nad samim robotom. Kontrola nad robotom se provodi pomoću programiranja, gdje čovjek nad robotom ne mora imati direktnu kontrolu, već unaprijed određenim instrukcijama upravlja njime. Smatram da u automatiziranom načinu programiranja leži budućnost i da će se na tržištu u budućnosti pojavljivati sve više i više automatiziranih robota.

# Popis literature

- [1] - Z. Kovačić, S. Bogdan, V. Krajči, Osnove robotike, Graphis Zagreb, 2002.
- [2] - Osnove robotike(mag.ing.mech Tomislav Pavlič),  
[https://vub.hr/images/uploads/3314/podjela\\_robotike\\_industrijska\\_i\\_mobilna\\_robotika.pdf](https://vub.hr/images/uploads/3314/podjela_robotike_industrijska_i_mobilna_robotika.pdf)
- [3] - <http://tnt.etf.rs/~oo1ue/predavanja/P6.pdf>
- [4] - <https://www.pjrc.com/store/teensy32.html>
- [5] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-tee](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-tee)
- [6] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-ir-finder2](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-ir-finder2)
- [7] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-ref-a](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-ref-a)
- [8] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-imu-an3](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-imu-an3)
- [9] - <https://www.servocity.com/730-rpm-premium-planetary-gear-motor/>
- [10] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-mot4x3.6b](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-mot4x3.6b)
- [11] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-ps3x](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-ps3x)
- [12] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-distrib-a](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-distrib-a)
- [13] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-switch](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-switch)
- [14] - <https://www.microline.hr/HTML.ashx?name=PriceList&key=kab>
- [15] - 40Pcs Dupont Wire Jumper Cables 10cm Female To Female  
<https://www.ebay.com/itm/162739208243?hash=item25e4034033:g:nTcAAOSwXxhZ~7-H>
- [16] - [https://hobbyking.com/en\\_us/turnigy-1500mah-3s-25c-lipo-pack.html?queryID=efec36bf1cadd1457b4b2679a8ec41c2&objectID=18180&indexName=hbk\\_live\\_products\\_analytics](https://hobbyking.com/en_us/turnigy-1500mah-3s-25c-lipo-pack.html?queryID=efec36bf1cadd1457b4b2679a8ec41c2&objectID=18180&indexName=hbk_live_products_analytics)
- [17] - [https://www.microline.hr/HTML.ashx?\\_command=showItem&itemKey=mrm-lipo-char](https://www.microline.hr/HTML.ashx?_command=showItem&itemKey=mrm-lipo-char)
- [18] -  
<https://www.ebay.com/itm/173947197198?hash=item28800f970e:g:qhoAAOSwfgBdFcE3>
- [19] - <https://www.generationrobots.com/en/402779-58mm-omni-wheel-for-lego-nxt.html>
- [20] - <http://www.nexusrobot.com/product/3mm-aluminum-mounting-hub-18027.html>
- [21] - <https://all3dp.com/2/what-is-solidworks-simply-explained/>
- [22] - B. Motik, J. Štibar: Demistificirani C++, Element 1997



[23] - <https://www.circuito.io/blog/arduino-code/>

[24] - <https://github.com/PribaNosati/MRMS>

# Popis slika

<b>Slika 1.</b> Teensy 3.2 mikrokontroler [4].....	4
<b>Slika 2.</b> Razvodna pločica za mikrokontroler [5] .....	5
<b>Slika 3.</b> Senzor za loptu „MRMS IR ball finder 2“ [6] .....	6
<b>Slika 4.</b> Senzor za boju „MRMS Reflectance Sensor A“ [7].....	6
<b>Slika 5.</b> Stara i nova verzija kompas uređaja „MRMS IMU“ [8] .....	7
<b>Slika 6.</b> Istosmjerni četkani motor „Actobotics gear motor“ [9] .....	8
<b>Slika 7.</b> Kontroler motora „MRMS Motor Driver 4x3.6A B“ [10].....	9
<b>Slika 8.</b> Napajanje „MRMS Power Supply 3x (mrm-ps3x)“ [11].....	10
<b>Slika 9.</b> Razvodna pločica „MRMS I2C 3V3 5V Distribution Pins (mrm-distrib-a)“ [12] .....	11
<b>Slika 10.</b> Tipkalo „MRMS switch 2x“ [13].....	12
<b>Slika 11.</b> Vodiči s lijeve na desnu(Dupont, MRMS Cable KK396-KK396, MRMS Cable XT60-KK3.96) [14], [15] .....	13
<b>Slika 12.</b> Turnigy Li-Po baterija [16] .....	14
<b>Slika 13.</b> Punjač baterije „LiPo battery 2S/3S balance charger (mrm-lipo-char)“ [17] .....	14
<b>Slika 14.</b> Set vijaka, matica i odstojnika [18] .....	15
<b>Slika 15.</b> Svosmjerski kotači „Nexus Robotics Omni Wheel for Lego NXT/EV3“ [19].....	15
<b>Slika 16.</b> Donji dio šasije prikazan u programu Z-Suite .....	17
<b>Slika 17.</b> Držač za motor prikazan u programu Z-Suite .....	18
<b>Slika 18.</b> Gornji dio šasije prikazan u programu Z-Suite.....	19
<b>Slika 19.</b> Postavke printanja u programu Z-suite .....	20
<b>Slika 20.</b> Izvješće o potrebnom vremenu i materijalu za print gornjeg dijela šasije .....	20
<b>Slika 21.</b> Isprintani dijelovi šasije.....	21
<b>Slika 22.</b> Pričvršćeni motori i senzori na donji dio šasije .....	22
<b>Slika 23.</b> Spoj gornjeg i donjeg dijela šasije .....	22
<b>Slika 24.</b> Plastična pločica i elektronički elementi .....	23
<b>Slika 25.</b> Mehanički spoj svih elemenata na šasiju .....	24
<b>Slika 26.</b> Prikaz pinova razvodne pločice [5].....	26
<b>Slika 27.</b> Finalna verzija robota .....	27
<b>Slika 28.</b> Inicijalizacija senzora.....	29
<b>Slika 29.</b> ispisivanje sirovih vrijednosti svakog senzora .....	30
<b>Slika 30.</b> Korekcija smjera lopte i postavljanje kompasa na početnu vrijednost .....	31
<b>Slika 31.</b> Jedno od grananja za ispitivanje linije.....	32