

# Daljinski upravljani robot skakač

---

**Kaštelan, Pavao**

**Master's thesis / Diplomski rad**

**2024**

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:235:537314>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-08**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

**Pavao Kaštelan**

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Željko Šitum, dipl. ing.

Student:

Pavao Kaštelan

Zagreb, 2024.

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

Zahvaljujem se mentoru prof. dr. sc. Željku Šitumu za pomoć i savjete pri izradi diplomskog rada.

Također se zahvaljujem svojoj obitelji koja mi je bila podrška tijekom studiranja.

Pavao Kaštelan



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 24 - 06 / 1	
Ur.broj: 15 - 24 -	

## DIPLOMSKI ZADATAK

Student: **Pavao Kaštelan**

JMBAG: 0035210825

Naslov rada na hrvatskom jeziku: **Daljinski upravljani robot skakač**

Naslov rada na engleskom jeziku: **Remotely controlled jumping robot**

Opis zadatka:

Skakanje kao vrsta kretanja je sasvim normalna u životinjskom svijetu, ali se rijetko koristi kod mobilnih robota. Promatrajući životinje i insekte, istraživači diljem svijeta pronalaze načine oponašanja ili poboljšanja metoda kretanja mobilnih robota. Korištenje vertikalnog skakanja, omogućuje mobilnim robotima kretanje po terenima koji su teško savladivi za izvedbe s kotačima ili gusjenicama. Glavni zahtjev u razvoju mehanizma za skakanje je izgradnja lagane pogonske jedinice robota, u kojoj se može podešavati visina i kut skakanja. Kod malih robota za skakanje poželjno je najprije polako puniti energijom elastični element, a zatim koristiti mehanizam kao katapult za skok. Za ostvarenje skakanja mogao bi se koristiti elektromotor koji ima zadatak natezanja opruge u položaj za skok. Odvajanjem elektromotora od mehanizma za natezanje elastičnog elementa, postiže se naglo rastezanje elastičnog elementa i robot može napraviti skok. Roboti skakači mogu služiti za istraživanje nepoznatog urbanog ili teško pristupačnog terena, a korištenjem senzora i kamera mogu izravno slati podatke operateru.

U radu je potrebno:

- projektirati robota skakača koji ima mehanizam za natezanje elastičnog elementa i mogućnost ostvarenja skoka
- ugraditi komponente za daljinsko upravljanje robotom skakačem
- opisati korištene komponente mehaničkog, pogonskog, upravljačkog i mjernog dijela sustava
- izraditi robota skakača i provjeriti rad sustava u praksi.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

Datum predaje rada:

Predviđeni datumi obrane:

26. rujna 2024.

28. studeni 2024.

5., 6. i 9. prosinca 2024.

Zadatak zadao:

Prof. dr. sc. Željko Šitum

Predsjednik Povjerenstva:

  
Prof. dr. sc. Ivica Garašić

**SADRŽAJ**

SADRŽAJ .....	I
POPIS SLIKA .....	III
POPIS TABLICA.....	V
POPIS TEHNIČKE DOKUMENTACIJE .....	VI
POPIS OZNAKA .....	VII
SAŽETAK.....	VIII
SUMMARY .....	IX
1. UVOD.....	1
1.1. Terenska robotika.....	1
1.1.1. Područja primjene .....	2
1.1.2. Vertikalno skakanje .....	2
1.1.3. Mehanizam za skakanje .....	3
1.1.4. Tipovi mehanizma za skakanje .....	3
1.1.4.1. Mehanizam s oprugom.....	3
1.1.4.2. Mehanizam s pneumatskim cilindrom .....	4
1.1.4.3. Mehanizam s unutarnjim izgaranjem.....	5
2. PROJEKTIRANJE ROBOTA SKAKAČA.....	6
2.1. Mehanizam skakanja.....	8
2.1.1. Servo motor za namatanje užeta .....	10
2.1.2. Proračun maksimalne sile u opruzi .....	11
2.2. Pogon i tijelo robota.....	13
2.2.1. Tijelo robota.....	16
2.2.2. Pogon robota .....	19
2.3. Elektronika za izradu robota skakača.....	20
2.4. Arhitektura sustava .....	23
2.5. Programiranje robota.....	24
2.5.1. Kretanje pomoću dva servo motora .....	24
2.5.2. Upravljanje servo motorom za kompresiju opruge.....	24
2.5.3. Upravljanje spojke putem servo motora i prekidača.....	26
2.5.4. Komunikacija s Raspberry Pi-em .....	26
2.6. Instalacija OpenHD sustava .....	26
3. OPIS KOMPONENTI SUSTAVA.....	28
3.1. Komponente pogona .....	28
3.2. Komponente tijela.....	31
3.3. Komponente mehanizma za skakanje .....	32
3.3.1. Servo motor MG995 .....	32
3.3.2. Hirth spojka.....	33
3.3.3. Klizni ležaj .....	34
3.3.4. Kolotura za namatanje užeta .....	35
3.3.5. Uže .....	35
3.3.6. Osovine za vođenje opruge .....	36

---

3.3.7.	Opruge.....	36
3.3.8.	FS90R servo motor .....	36
3.3.9.	Odskočni podložak.....	37
3.3.10.	Prekidač za poziciju spojke.....	38
3.4.	Elektroničke komponente robota .....	39
3.4.1.	Mikrokontroler Arduino Nano .....	39
3.4.2.	Računalo robota .....	40
3.4.3.	Raspberry Pi kamera V3 .....	42
3.4.4.	Tp-link T3U Wi-Fi adapter .....	43
3.4.5.	Baterija litij-ionska 18650.....	43
3.4.6.	Pretvarač logičkih razina.....	44
3.4.7.	Akcelerometar i žiroskop MPU6050 .....	44
4.	IZVEDBA ROBOTA .....	46
4.1.	Konstrukcija i masa.....	46
4.2.	Spajanje robota na OpenHD sustav .....	48
4.3.	Mehanizam skakanja.....	50
5.	ZAKLJUČAK.....	51
	LITERATURA.....	52
	PRILOG – UPRAVLJAČKI KOD .....	53

## POPIS SLIKA

Slika 1. Grasshopper robot [1] .....	4
Slika 2. Boston Dynamics – Sand Flea [2].....	4
Slika 3. 3D model inicijalnog koncepta robota skakača .....	6
Slika 4. 3D model završne verzije robota .....	7
Slika 5. Mehanizam skakanja.....	8
Slika 6. Kolotura s integriranom Hirthovom spojkom.....	9
Slika 7. Upravljanje spojke servo motorom .....	9
Slika 8. Rotirajući element servo motora (2) .....	10
Slika 9. Raspored sila na koloturi.....	12
Slika 10. Simetričnost robota .....	14
Slika 11. Najveći kut skoka robota.....	15
Slika 12. Manji kut skoka robota .....	15
Slika 13. Kontakt kotačića i podloge .....	16
Slika 14. Osnovno tijelo robota.....	17
Slika 15. Klizni ležajevi s ojačanjem .....	17
Slika 16. Klizni ležaj s integriranim profilom.....	18
Slika 17. Shema pogona .....	19
Slika 18. Kotači u neopterećenom (lijevo) i opterećenom stanju (desno) .....	20
Slika 19. Lokacija baterija na robotu .....	21
Slika 20. Električna shema spajanja komponenti.....	21
Slika 21. Preporučeni Wi-Fi adapteri [5] .....	23
Slika 22. Shema arhitekture robota .....	24
Slika 23. Početni kut robota .....	25
Slika 24. Maksimalni kut robota .....	25
Slika 25. Sučelje programa ImageWriter .....	27
Slika 26. Raspberry Pi povezan s kamerom i Wi-Fi adapterom .....	27
Slika 27. Dimenzije FS90R servo motora [6] .....	29
Slika 28. Dimenzije kotača.....	30
Slika 29. Dimenzije polimernog okvira .....	31
Slika 30. Dimenzije spojnih cijevi .....	31
Slika 31. MG995 kontinuirani servo motor [7].....	32
Slika 32. Profil zuba Hirthove spojke .....	33
Slika 33. Dimenzije Hirth spojke .....	34
Slika 34. Dimenzije kliznog ležaja.....	34
Slika 35. Dimenzije koloture.....	35
Slika 36. Dimenzije osovine za vođenje opruge .....	36
Slika 37. Odskočni podložak u kontekstu robota .....	37
Slika 38. Mikroprekidač KW12-3 [8] .....	38
Slika 39. Prekidač u dodiru s rotirajućim elementom na servo motoru .....	39
Slika 40. Mikrokontroler Arduino Nano [9] .....	39
Slika 41. Računalo Raspberry Pi Zero 2 W [10].....	41
Slika 42. Raspberry Pi kamera V3 [11].....	42
Slika 43. Wi-Fi adapter Tp-link T3U [12] .....	43
Slika 44. Pretvarač logičkih razina tvrtke Soldered [12] .....	44
Slika 45. Izvedba robota.....	46
Slika 46. Mjerenje mase robota.....	48



---

Slika 47. Slika robota i njegovog okruženja .....	49
Slika 48. Snimka iz perspektive robota .....	49
Slika 49. Mehanizam skakanja sa sabijenom oprugom slikano iz profila .....	50
Slika 50. Mehanizam skakanja sa sabijenom oprugom.....	50

**POPIS TABLICA**

Tablica 1. Odabir elektromotora .....	11
Tablica 2. Usporedba osovine i kliznih ležaja.....	18
Tablica 3. Specifikacije FS90R servo motora .....	28
Tablica 4. Odabir zupčastog prijenosa .....	29
Tablica 5. Specifikacije 608-2Z ležaja .....	30
Tablica 6. Specifikacije MG995 servo motora.....	33
Tablica 7. Svojstva monofilamenta .....	35
Tablica 8. Specifikacija kompresijske opruge.....	36
Tablica 9. Specifikacije 623Z ležaja .....	38
Tablica 10. Specifikacije mikrokontrolera Arduino Nano .....	40
Tablica 11. Računalo Raspberry Pi Zero 2 W.....	41
Tablica 12. Specifikacije Raspberry Pi kamere .....	42
Tablica 13. Baterije 18650 litij-ionske.....	43
Tablica 14. Senzorski modul MPU6050 .....	45
Tablica 15. Popis svih izrađenih konstrukcijskih elemenata.....	46

## **POPIS TEHNIČKE DOKUMENTACIJE**

1.           Upravljački kod

**POPIS OZNAKA**

<b>Oznaka</b>	<b>Jedinica</b>	<b>Opis</b>
$a$	m	Razmak osovina
$C$	€	Cijena motora
$D_k$	m	Promjer koloture
$E_k$	J	Kinetička energija
$E_p$	J	Potencijalna energija
$F_{op}$	N	Sila opruge
$F_{servo}$	N	Sila servo motora
$F_u$	N	Sila u užetu
$g$	m/s <sup>2</sup>	Gravitacijsko ubrzanje
$h_s$	m	Visina skoka
$k$	N/m	Konstanta opruge
$l$	m	Duljina kraka
$m$	kg	Masa robota
$M_{servo}$	Nm	Moment servo motora
$S_i$	-	Relativna ocjena motora
$T$	Nm	Moment motora
$v$	m/s	Brzina robota pri otpuštanju opruge
$x$	m	Kompresija ili ekstenzija opruge
$z$	-	Broj zuba na zupčaniku

## **SAŽETAK**

Diplomski rad bavi se razvojem i konstrukcijom robota skakača, koji je osmišljen za istraživanje nepristupačnih terena i prevladavanje prepreka u prirodnim i strukturiranim okruženjima. Fokus je na projektiranju mehanizma skakanja koji koristi opruge za generiranje vertikalnog skoka. Robot je dizajniran s naglaskom na malu masu i jednostavnost konstrukcije pomoću 3D ispisa. Opisan je cjelokupni proces projektiranja: izbor materijala, konstrukcija tijela i komponenata poput servo motora, opruga i elektronike. Detaljno su opisani postupci programiranja sustava za daljinsko upravljanje i sustava za video prijenos pod nazivom OpenHD.

Ključne riječi: robot skakač, OpenHD, daljinsko upravljanje, istraživanje terena

## **SUMMARY**

The thesis focuses on the development and construction of a jumping robot designed for exploring inaccessible terrains and overcoming obstacles in natural and structured environments. The emphasis is on designing a jumping mechanism that uses springs to generate vertical jumps. The robot is designed with a focus on low weight and simplicity of construction using 3D printing. The entire design process is described, including material selection, body construction, and components such as servo motors, springs, and electronics. The programming processes for the remote-control system and the OpenHD video transmission system are described in detail.

Key words: jumping robot, OpenHD, remote control, terrain exploration

## 1. UVOD

Jedna od čestih i bitnijih primjena mobilnih robota je istraživanje nepoznate ili opasne okoline. Kretanje u stvarnim uvjetima koja su kompleksna i nepredvidljiva veoma otežava navigaciju i obavljanje zadataka. U takvim okolinama, roboti često nailaze na prepreke poput stepenica, stijena, ruševina ili drugih prepreka koje se ne mogu lako zaobići ili savladati konvencionalnim sredstvima kretanja poput kotača ili gusjenica. Razvijanje robota koji ima mogućnost vertikalnog skoka, omogućuje mu da prevlada vertikalne prepreke koje nisu moguće kod konvencionalnih robota. Dronovi također imaju mogućnost istraživanja nepoznatih ili opasnih okolina zahvaljujući svojoj sposobnosti da lete kroz kompleksne terene. Ali je mana dronova što imaju ograničeno trajanje leta zbog kapaciteta baterije, što ograničava vrijeme koje mogu provesti u zraku i domet koji mogu pokriti. Ovisno o modelu, većina dronova može letjeti između 20 i 60 minuta prije nego što im je potrebno ponovno punjenje ili zamjena baterije. Ovo može predstavljati izazov u misijama koje zahtijevaju dugotrajnu prisutnost ili pokrivanje velikih područja. Druga mana dronova je njihova osjetljivost na vremenske uvjete. Jaki vjetrovi, kiša, snijeg ili druge nepovoljne vremenske prilike mogu ometati ili potpuno onemogućiti let drona. Ovo ograničava njihovu operativnu učinkovitost u određenim vremenskim uvjetima i može utjecati na pouzdanost u kritičnim situacijama. Zbog toga se korištenje robota koji ima mogućnost vertikalnog skoka pokazuje poželjnijim u situacijama gdje su potrebna duža prisutnost, veća stabilnost i otpornost na vremenske uvjete. Ovi roboti mogu obavljati zadatke koji zahtijevaju dugotrajnu operativnost bez potrebe za čestim punjenjem, što im omogućava da pokrivaju veće područje i izvršavaju kompleksne zadatke.

### 1.1. Terenska robotika

Terenska robotika je grana robotike koja se bavi razvojem, implementacijom i primjenom robota za rad u vanjskim, često teškim i nepristupačnim uvjetima. Ovi roboti su dizajnirani da se autonomno ili polu-autonomno kreću i obavljaju zadatke u okolinama koje su previše opasne, udaljene ili zahtjevne za ljude.

### **1.1.1. Područja primjene**

Terenska robotika ima širok spektar primjena koje uključuju poljoprivredu, istraživanje, rudarstvo, građevinarstvo, vojnu i sigurnosnu upotrebu te operacije spašavanja i hitnih situacija. U poljoprivredi, terenski roboti igraju ključnu ulogu u optimizaciji procesa kao što su sadnja, navodnjavanje, prskanje pesticida i berba. Ovi roboti mogu se autonomno kretati kroz polja, detektirati bolesti biljaka i omogućiti efikasniju i precizniju obradu usjeva.

U istraživanju i ekspedicijama, terenski roboti su neophodni za pristup udaljenim i nepristupačnim područjima poput arktičkih regija, džungli, pustinja i planinskih lanaca. Oni omogućavaju prikupljanje podataka o okolišu, fauni i flori, čime doprinose znanstvenim istraživanjima i zaštiti okoliša. Slično tome, u rudarstvu, roboti se koriste za istraživanje rudarskih nalazišta, bušenje, sakupljanje uzoraka i nadzor rudarskih operacija. Njihova upotreba smanjuje rizik za ljudske radnike u opasnim rudarskim okruženjima, omogućavajući sigurniju i efikasniju eksploataciju resursa.

U građevinarstvu, terenski roboti pomažu u raznim građevinskim radovima, uključujući rušenje, izgradnju, transport materijala i inspekciju građevinskih objekata. Oni su posebno korisni u opasnim područjima kao što su visoke zgrade ili nestabilne strukture, gdje mogu smanjiti rizik za građevinske radnike. Vojna i sigurnosna primjena robota uključuje izviđanje, razminiranje, transport opreme i druge operacije u ratnim zonama ili područjima visokog rizika. Ovi roboti pružaju podršku vojnicima, smanjujući rizik od stradanja i povećavaju efikasnost vojnih operacija.

U operacijama spašavanja i hitnih situacija, terenski roboti su od ključne važnosti za pretrage i spašavanje u slučajevima prirodnih katastrofa kao što su potresi, poplave ili lavine. Mogu pretraživati ruševine, locirati preživjele i dostavljati hitnu pomoć, čime značajno povećavaju šanse za preživljavanje ugroženih osoba. Terenska robotika, sa svojim inovativnim rješenjima i tehnologijama, omogućava efikasnije i sigurnije izvođenje zadataka u teškim i nepristupačnim okolinama, čineći je ključnim alatom u mnogim industrijama i situacijama.

### **1.1.2. Vertikalno skakanje**

Skakanje je vrlo učinkovit način za prelazak preko prepreka u prirodnom svijetu. Većina životinja skače umjesto korištenja drugih metoda za prelazak prepreka. Uzimajući inspiraciju od životinja, predloženi su razni roboti za skakanje, kao što su roboti za skakanje s opružnim



odskokom, artikulirani, pneumatski, s unutarnjim izgaranjem te fleksibilni, za prilagodbu kompleksnom terenu i prelazak prepreka u strukturiranim i nestrukturiranim okruženjima. Štoviše, pojedinačni roboti za skakanje mogu lako prevladati prepreke mnogostruko veće od njihove vlastite veličine.

### ***1.1.3. Mehanizam za skakanje***

Odabir vrste pogona za skok sastoji se od četiri glavna faktora:

- Visina skoka
- Količina skokova
- Potrebno vrijeme za ponovni skok
- Kompaktnost mehanizma

### ***1.1.4. Tipovi mehanizma za skakanje***

Visina skoka u slučaju da je masa robota konstanta, određena je prvenstveno sa energijom koju mehanizam može pretvoriti u kinetičku energiju robota. Neki od načina s kojim se u praksi to može izvesti dani su u nastavku:

#### ***1.1.4.1. Mehanizam s oprugom***

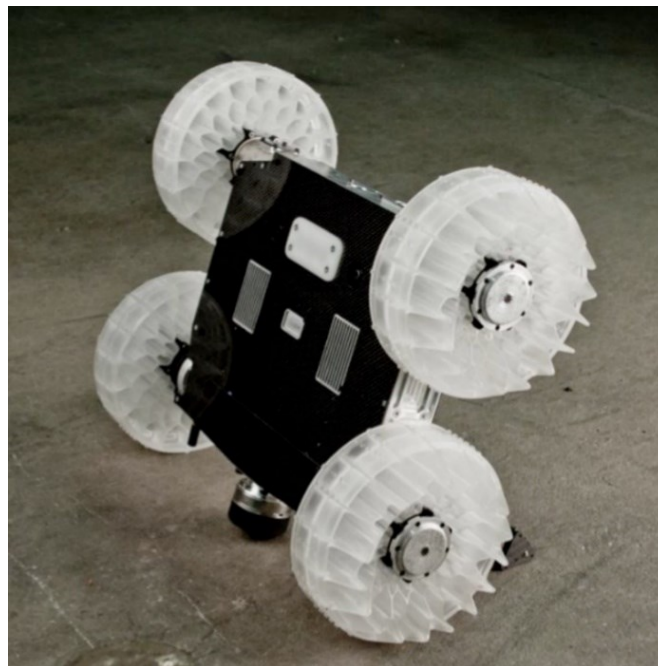
Kod mehanizma s oprugom energija se sprema kao potencijalna energija u opruzi koristeći neki sporiji sustav kao što je navojno vreteno pogonjeno motorom koje nateže oprugu u nekom vremenu koje je puno veće od vremena potrebnog za isprazniti energiju iz opruge. Većina sustava s oprugom je inspirirano prirodom, gdje se slični mehanizmi nalaze u živim bićima, kao što su skakanje skakavaca ili elastične strukture u mišićima i tetivama. Na slici 1 prikazan je robot koji koristi princip inspiriran skakavcem.



**Slika 1. Grasshopper robot [1]**

#### *1.1.4.2. Mehanizam s pneumatskim cilindrom*

Pneumatski cilindar koristi stlačeni plin (najčešće zrak) za stvaranje sile koja je potrebna za skok. Prednost pneumatskog sustava je vrlo brzo oslobađanje energije i puno kraće vrijeme između skokova jer sadrži spremnik stlačenog plina koji se treba puniti tek nakon određenog broja aktiviranja mehanizma. Kao primjer robota koji koristi pneumatiku za skok može se navesti Sand Flea robot američke tvrtke Boston Dynamics, prikazan na slici 2.



**Slika 2. Boston Dynamics – Sand Flea [2]**

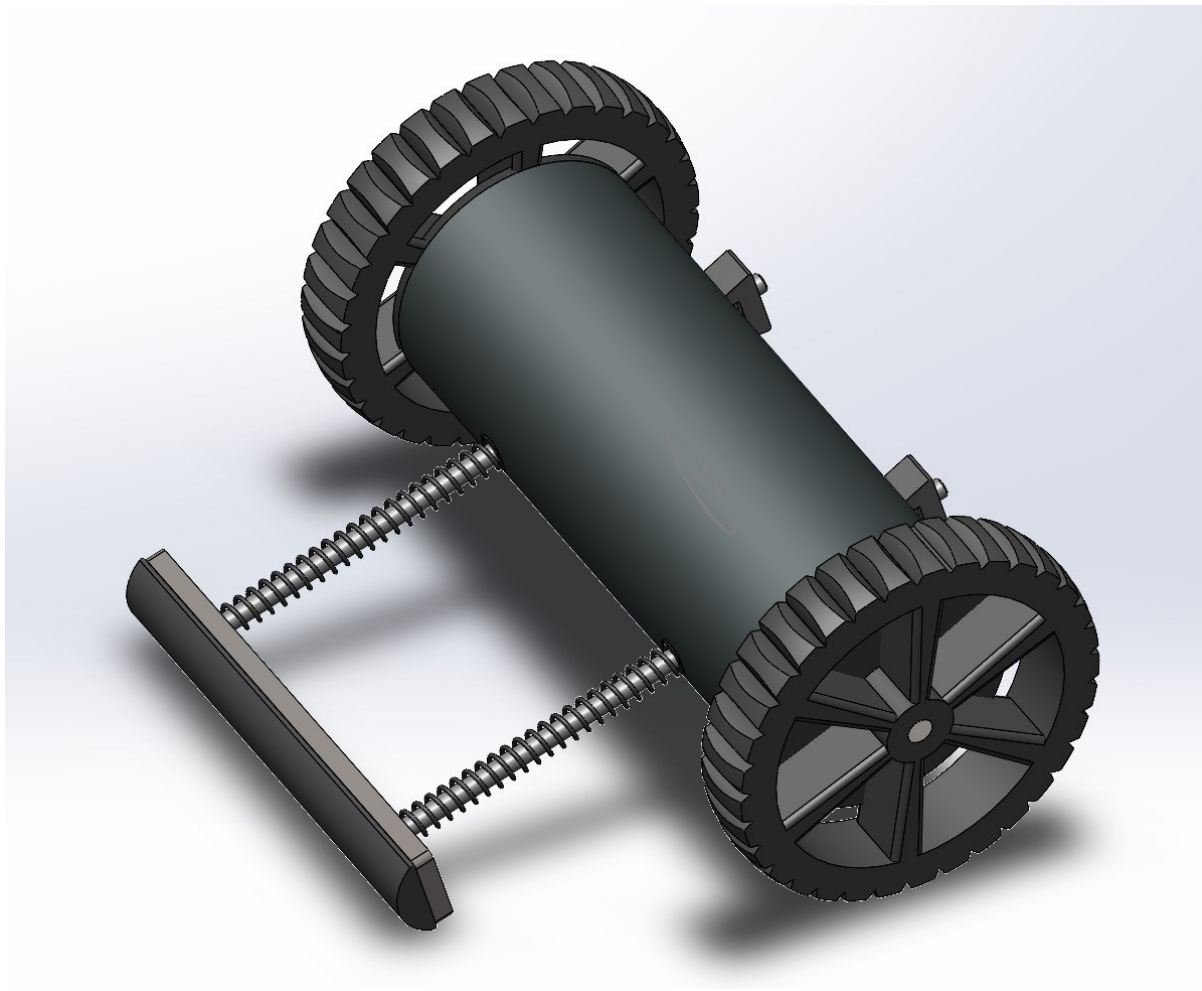
#### *1.1.4.3. Mehanizam s unutarnjim izgaranjem*

Koristi izgaranje goriva umjesto komprimiranog plina za generiranje sile skoka, što pruža neke prednosti u odnosu na pneumatski sustav, kao što je mogućnost većeg broja skokova prije ponovnog punjenja, ali uvodi dodatnu kompleksnost [3]. Neki od izazova bi bili:

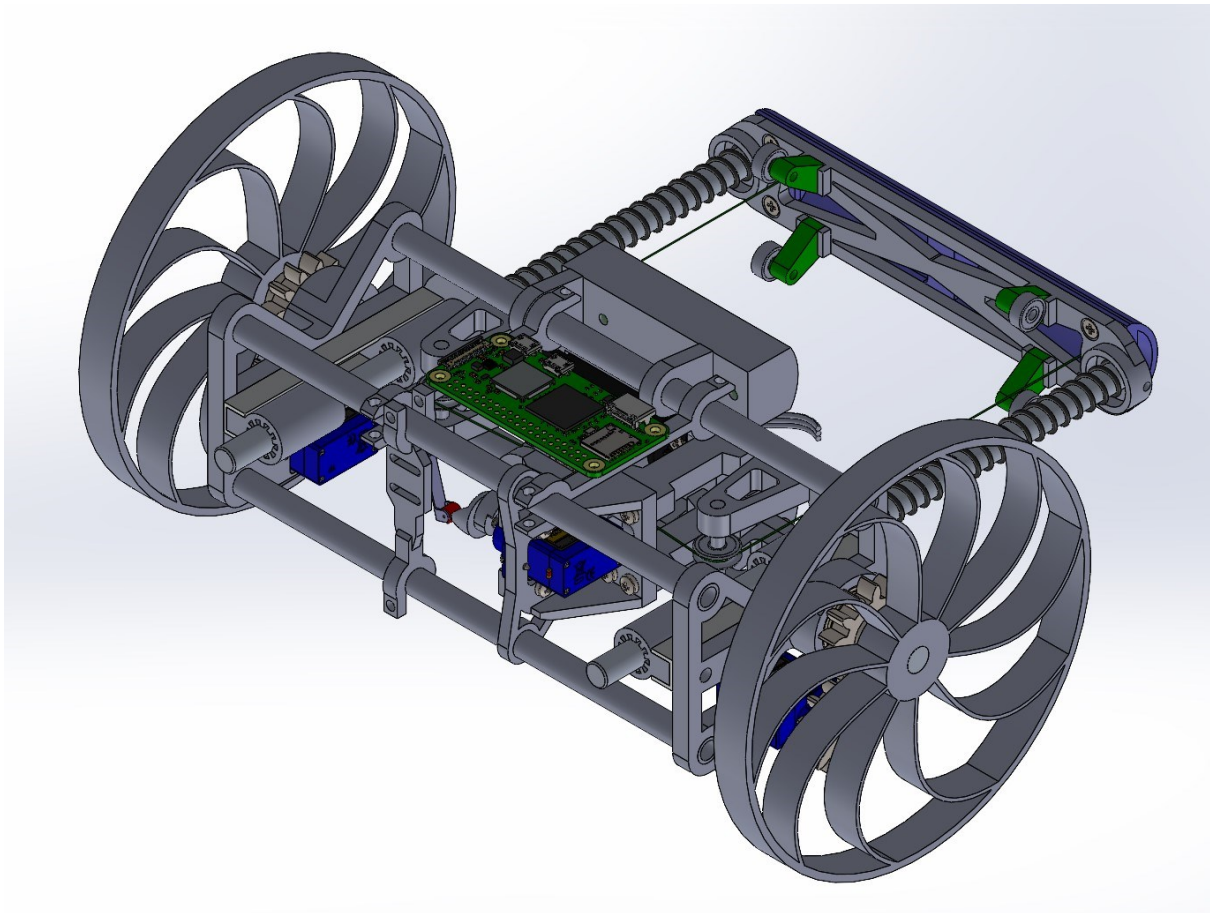
- Sustav za ubrizgavanje goriva
- Potreba za kontrolom procesa izgaranja
- Sigurnost kod korištenja zapaljivih goriva
- Veća masa zbog sustava za gorivo i izgaranje

## 2. PROJEKTIRANJE ROBOTA SKAKAČA

Pri projektiranju robota skakača prvobitni je fokus na mehanizam skakanja. Najveći faktori koji određuju sustav mehanizma skakanja su masa i potrebna visina skoka. Masa je vezana za mogućnosti robota, a sastoji se od motora za pogon, servo motora, kapaciteta baterije te materijala konstrukcije. Inicijalni koncept robota skakača koristi opruge kao metodu skoka i dva pogonska kotača za kretanje. Na slici 3 prikazan je 3D model koncepta, a na slici 4 prikazan je 3D model završne verzije robota.



Slika 3. 3D model inicijalnog koncepta robota skakača



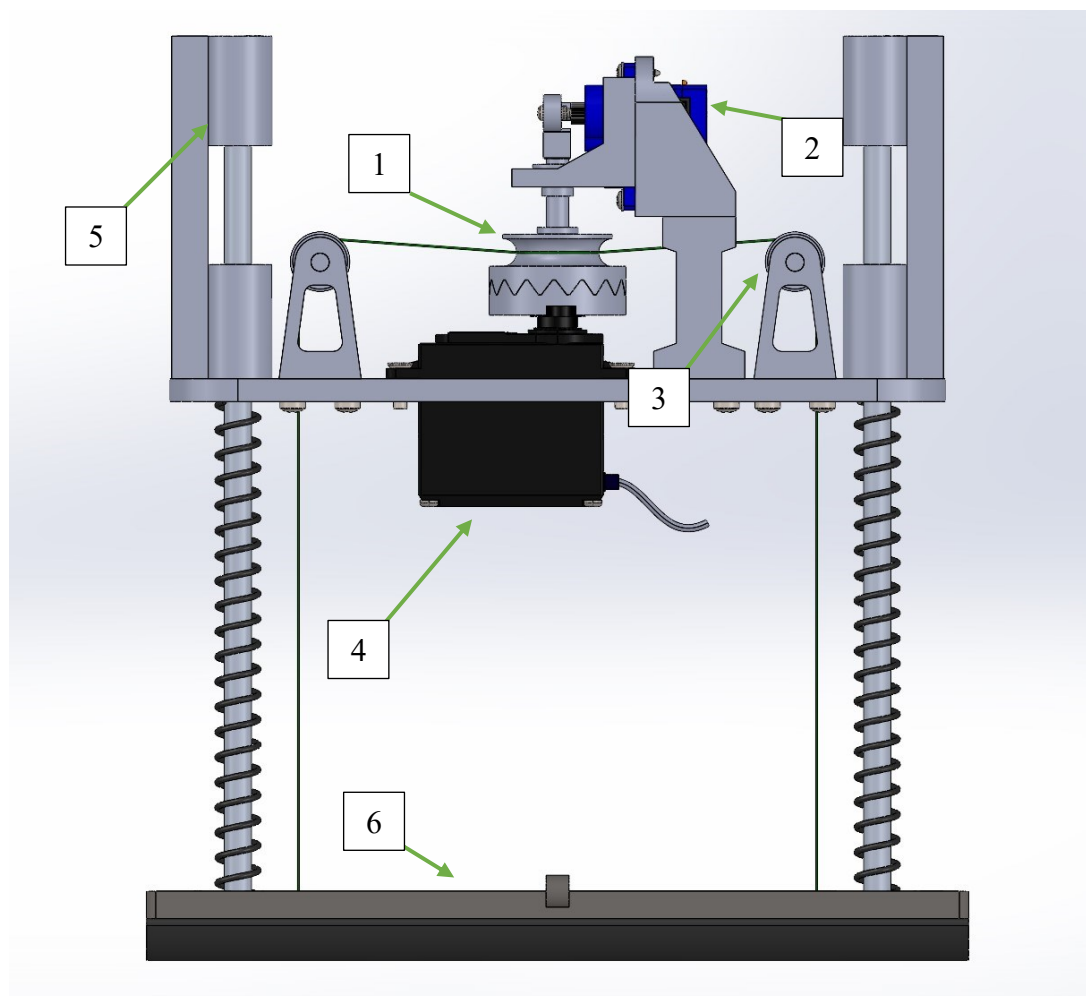
**Slika 4. 3D model završne verzije robota**

Početni korak za dizajniranje je određivanje parametara koje robot mora ostvariti.

- Visina skoka – robot mora moći ostvariti skok u vertikalnom smjeru od minimalno 1m
- Masa – robot velike mase bi bio trom i zahtijevao bi metalnu konstrukciju što bi dodatno povećavalo masu i otežalo razvoj prototipa, ako uzimamo u obzir da laganiji robot može biti konstruiran od polimera 3D pisačem. Ukupna masa robota treba biti ispod 500g, ukoliko to bude ostvarivo.

## 2.1. Mehanizam skakanja

Mehanizam skakanja zadužen je za kompresiju opruge i naglo otpuštanje iste. Potrebno je konstruirati metodu koja može ponavljati tu operaciju. Dizajn koji ima tu mogućnost osmišljen je na način da se opruga sabija pomoću užeta koje se namotava na koloturu i naglo otpušta. Na slici 5 prikazan je dizajn mehanizma skakanja.

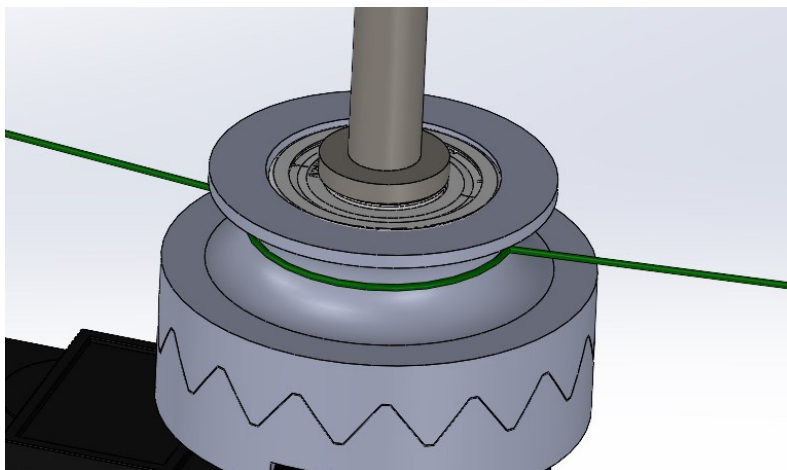


Slika 5. Mehanizam skakanja

Oznake na slici 5 su:

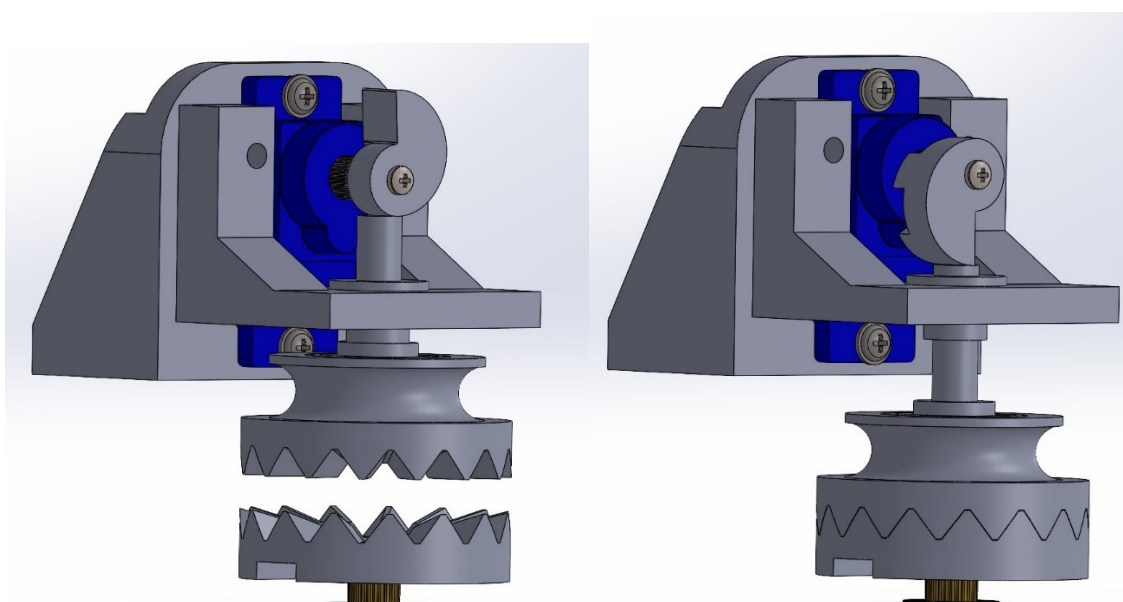
- 1) Kolotura s integriranom Hirthovom spojkom
- 2) Servo motor s mehanizmom za spajanje i odvajanje spojke
- 3) Pomoćne koloture za namatanje užeta
- 4) Servo motor za namatanje užeta
- 5) Klizni ležaj s osovinom
- 6) Odskočna podloga

Skok robota se postiže naglim odvajanjem servo motora (4) od koloture (1). Kolotura mora biti postavljena na ležaj koji osigurava slobodno otpuštanje užeta tj. opruge pri maksimalnoj brzini, a pri namatanju mora biti kruto spojena na elektromotor. Izvedba tog sustava sastojat će se od koloture kruto spojene na Hirthovu spojku i osovine spojene u ležaj. Na slici 6 prikazana je kolotura s integriranom Hirthovom spojkom.



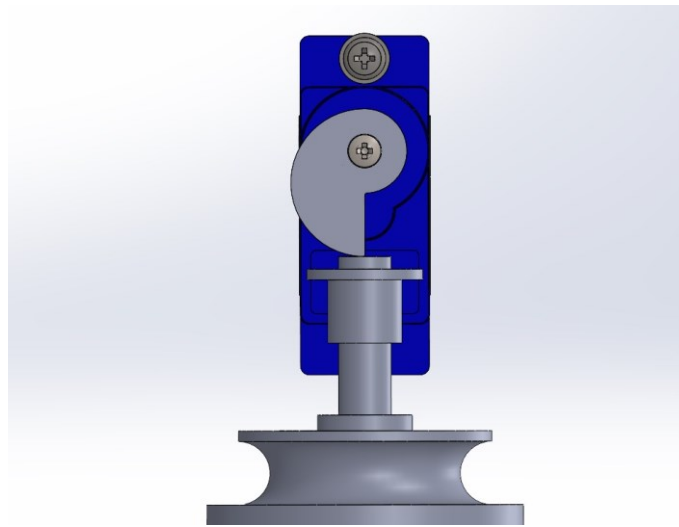
**Slika 6. Kolotura s integriranom Hirthovom spojkom**

Upravljanje mehanizmom spojke upotrebljava se servo motor (2) koji ima funkciju spajanja i odvajanja spojke. Na slici 7 prikazan je način upravljanja aktiviranjem i deaktiviranjem spojke. Za deaktivaciju spojke potrebno je imati silu za razdvajanje. Pozicioniranjem pomoćnih kolotura (3) iznad glavne koloture (1) stvara se sila koja razdvaja spojku.



**Slika 7. Upravljanje spojke servo motorom**

S obzirom da sustav zahtijeva naglo otpuštanje, dizajniran je rotirajući element na vratilu servo motora koji je prikazan na slici 8.



Slika 8. Rotirajući element servo motora (2)

Monofilament će se koristiti kao uže za namatanje, jer posjeduje elastična svojstva potrebna pri naglom vlačnom naprezanju za zaštitu komponenti od udarnih opterećenja. U slučaju nezadovoljavajućih performansi koristit će se kruće uže.

### 2.1.1. Servo motor za namatanje užeta

Pogonski elektromotor za kompresiju opruge mora se odabrati uzimajući u obzir masu, brzinu rotacije i moment. Razmatrat će se dva tipa elektromotora, kontinuirani servo motor i motor s četkicama s integriranim reduktorom. Odabir elektromotora bit će određen usporedbom dostupnih motora koji se veličinom mogu integrirati u robot. Korištenjem formule (2.1) dobije se relativna ocjena motora po kojoj se izabere motor:

$$S_i = \left( \frac{v_{max}}{v_i} \cdot w_v \right) + \left( \left( 1 - \frac{m_{max}}{m_i} \right) \cdot w_m \right) + \left( \frac{T_{max}}{T_i} \cdot w_T \right) + \left( \left( 1 - \frac{C_{max}}{C_i} \right) \cdot w_C \right) \quad (2.1)$$

gdje je:

$S_i$  – Relativna ocjena motora

$v_{max}$  – Najveća brzina rotacije [rpm]

$m_{max}$  – Najveća masa [kg]



$T_{max}$  – Najveći moment [Nm]

$C_{max}$  – Najviša cijena [€]

$w_v, w_m, w_T, w_c$  – Težinski faktori

Tablica 1. Odabir elektromotora

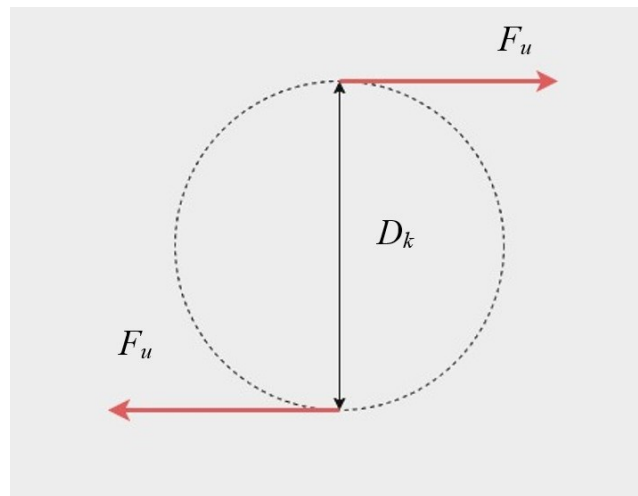
Motor	Masa ( $w_m = 0, 40$ )	Moment ( $w_v = 0, 40$ )	Brzina rotacije ( $w_T = 0, 10$ )	Cijena ( $w_c = 0, 10$ )	Relativna ocjena
N20	10,00	0,20	26,00	6,00	0,50
MG995	55,00	1,30	60,00	9,00	0,58
JGA25-370	96,00	0,82	60,00	6,00	0,41
JGB37-520	150,00	0,92	46,00	6,00	0,29
JGB37-520	155,00	1,30	37,00	6,00	0,33
JGB37-520	160,00	1,48	28,80	6,00	0,34
FS5106R	43,00	0,60	95,00	12,00	0,51
2000 Series	60,00	1,08	145,00	20,00	0,55
FS5113R	55,00	1,40	62,00	25,00	0,54
DF15RSMG	60,00	1,50	60,00	18,00	0,57
JGA25-310	60,00	0,70	57,00	8,00	0,47
JGY-370	160,00	2,40	30,00	8,00	0,49

Najbolji kandidati su istaknuti u tablici zelenom bojom. Servo motor MG995 lako je dostupan i ima nisku cijenu, što ga čini poželjnim izborom za izradu robota. Težinski faktori su odabrani iskustveno i u slučaju nezadovoljavajućeg učinka odabranog motora, potrebno je prilagoditi težinske faktore.

### 2.1.2. Proračun maksimalne sile u opruzi

Namatanje užeta za kompresiju opruge odvija se preko koloture radijusa 10 mm, s obzirom da namatamo dva užeta što odgovara kompresiji dvije opruge, iz maksimalnog momenta motora

možemo izračunati da je maksimalna sila po opruzi 65 N (2.2). Na slici 9 prikazana je shema sila na koloturi.



Slika 9. Raspored sila na koloturi

Gdje je:

$D_k$  – promjer koloture

$F_u$  – sila na užetu

$$2F_u = \frac{2T}{D_k} = \frac{2 \cdot 1,3}{0,02} = 130N$$

$$F_u = 65N \quad (2.2)$$

Poznavajući maksimalnu silu koju robot može postići, moguće je izračunati najviši teoretski mogući skok. Za izračun primjenjujemo osnovne jednadžbe o očuvanju energije, pri čemu se zanemaruju gubitci trenja s podlogom, trenja unutar kliznih ležaja i samih opruga.

Potencijalna energija u opruzi glasi:

1. Hooke-ov zakon opisuje silu opruge:

$$F_{op} = -kx \quad (2.3)$$

gdje je:

- $F_{op}$  - sila opruge (N),

- $k$  - konstanta opruge (N/m),
- $x$  - kompresija ili ekstenzija opruge (m).

2. Potencijalna energija pohranjena u opruzi iznosi:

$$E_p = \frac{1}{2} kx^2 \quad (2.4)$$

gdje je:

- $E_p$  - potencijalna energija (J)

3. Potencijalna energija opruge pretvara se u kinetičku energiju robota:

$$E_k = \frac{1}{2} mv^2 \quad (2.5)$$

gdje je:

- $E_k$  - kinetička energija (J),
- $m$  - masa robota (kg),
- $v$  - brzina robota pri otpuštanju opruge (m/s).

$$\begin{aligned} \frac{1}{2} kx^2 &= \frac{1}{2} mv^2 \\ v &= \sqrt{\frac{kx^2}{m}} \end{aligned} \quad (2.6)$$

4. Visina skoka:

$$\begin{aligned} \frac{1}{2} mv^2 &= mgh_s \\ h_s &= \frac{kx^2}{2mg} = 1,325 \text{ [m]} \end{aligned} \quad (2.7)$$

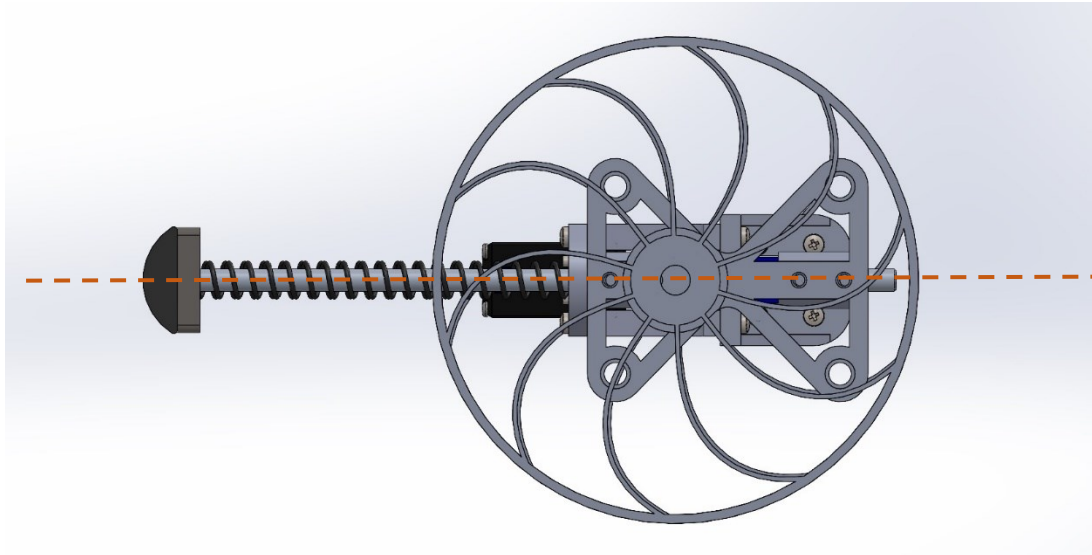
gdje je:

- $g$  - gravitacijsko ubrzanje ( $\text{m/s}^2$ )

## 2.2. Pogon i tijelo robota

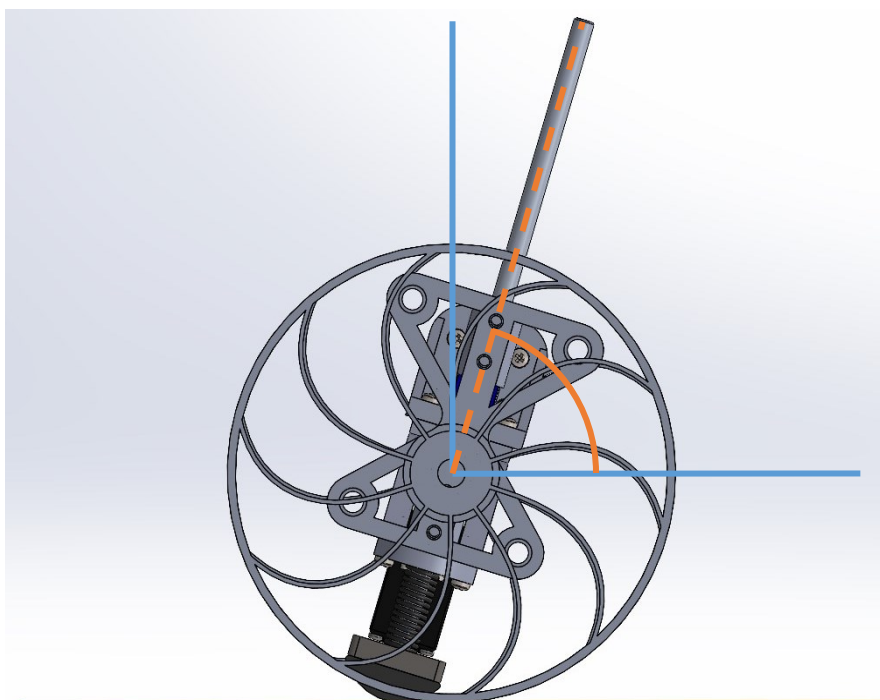
Kod projektiranja većine robota s mogućnošću skakanja nailazi se na problem doskoka. Tokom skoka roboti se mogu zarotirati oko svoje osi i doskočiti na suprotnu stranu, što u slučaju nesimetričnog robota bez mehanizma za vraćanje u početni položaj može robot onesposobiti za daljnju operaciju. Ovisno o vrsti i mehanizmu robota jedno od rješenja može biti projektiranje robota koji je simetričan neovisno o orijentaciji na koju doskoči. Jednostavno rješenje tog problema je robot s kotačima većeg profila od robota. Na slici 10 prikazan je dizajn robota iz

profila koji prikazuje neovisnost o doskočnoj orijentaciji. Centar ravnoteže treba biti iza kotača prema odskočnoj podlozi da se robot ne zarotira po osi kotača umjesto kretanja naprijed. Od manje je važnosti koliko centrirano u odnosu na simetričnu os, jer to određuje samo na koju stranu robot preferira doskočiti.



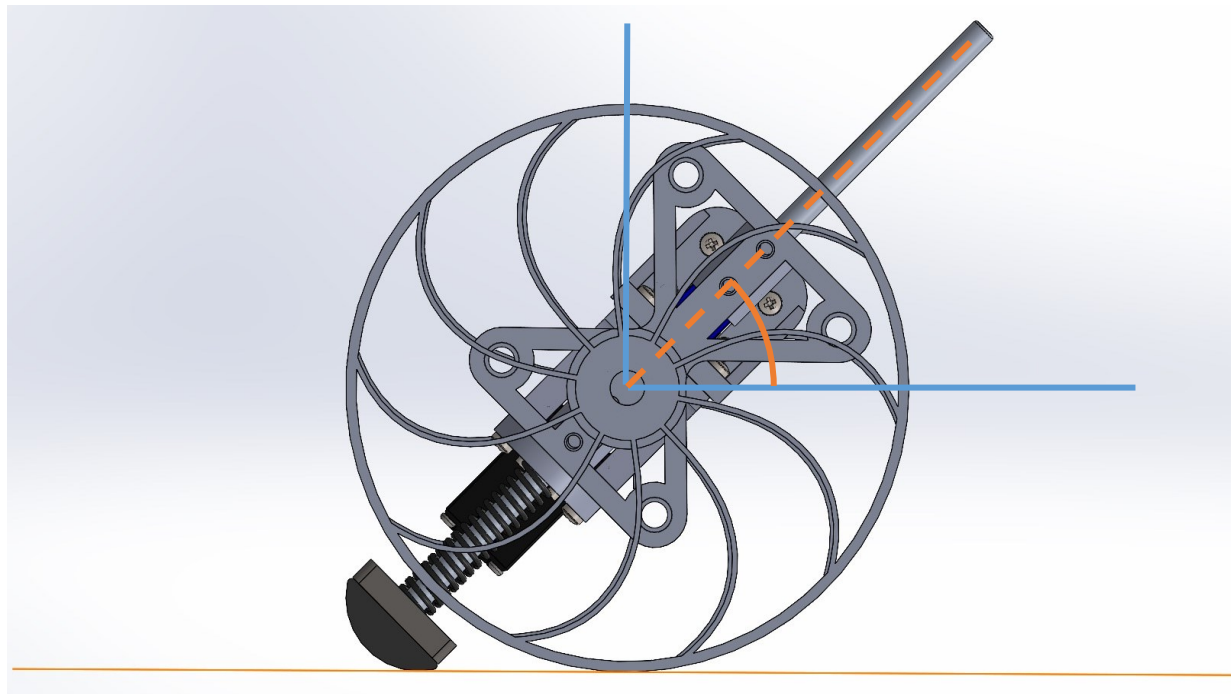
**Slika 10. Simetričnost robota**

Moguć je doskok i okomito na kotače ili na podlogu za skok, ali su to ravnotežno nestabilni položaji pa se mogu zanemariti. Robot koji ima mogućnost skakanja trebao bi imati i mehanizam za odabir kuta pod kojim skače. Izvedba tog mehanizma će se izvesti korištenjem akcelerometra koji nadzire orijentaciju robota, a sama kut skoka određuje se s pomakom tj. kompresijom opruge. Na slici 11 i 12 su prikazana dva primjera konfiguracije skoka s različitim kutovima za skok.



**Slika 11. Najveći kut skoka robota**

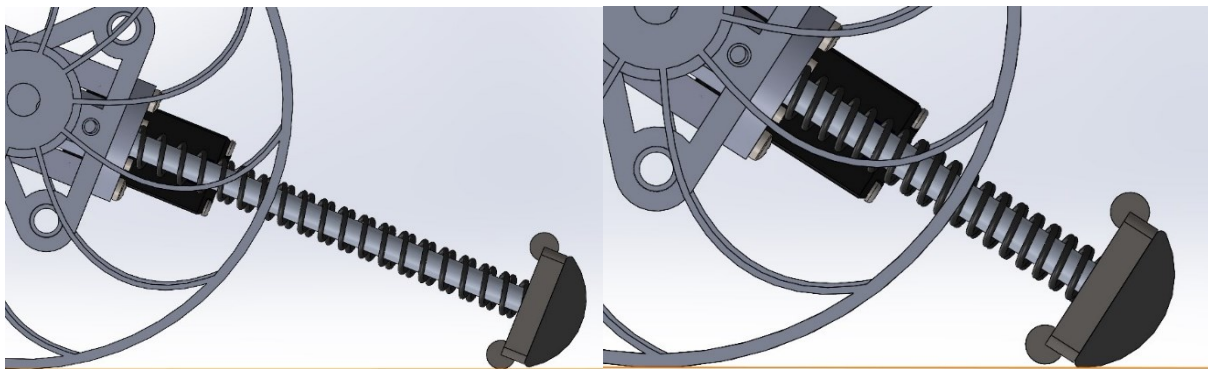
Slika 11. prikazuje potpunu kompresiju opruge i najveći mogući kut skoka robota.



**Slika 12. Manji kut skoka robota**

Konfiguracija skoka na slici 12 može služiti u slučaju da robot mora preskočiti rupu ili neprohodan teren. Podloga za skakanje jedini je kontakt robota s podom u procesu skakanja,

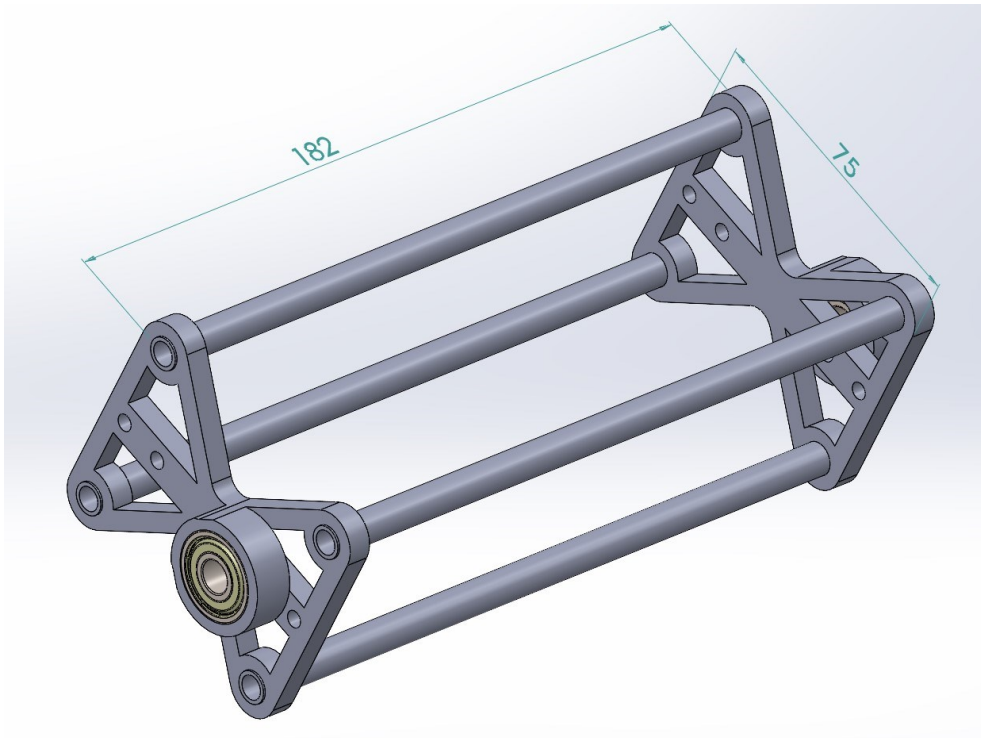
što zahtijeva da ima veliki koeficijent trenja. Lako dostupan i efikasan materijal za taj zadatak je prirodni gumeni lateks. Trenje s podlogom je korisno pri skakanju robota, ali predstavlja problem pri kretanju robota. Taj problem se može riješiti dodavanjem kotačića na podlogu za skakanje koji preuzima kontakt s podlogom kad je opruga neopterećena tj. robot je u početnom stanju s ispruženom oprugom. Na slici 13 može se primijetiti kako kotačić sprječava dodir podloge i gume.



Slika 13. Kontakt kotačića i podloge

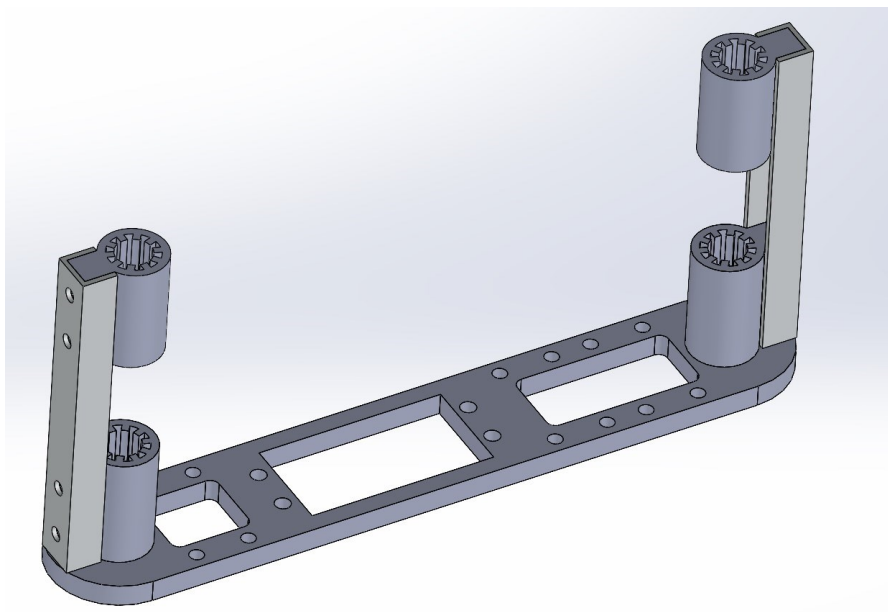
### 2.2.1. Tijelo robota

Relativno veliki zahtjevi za krutosti robota uslijed sila u opruzi odražavaju se na potrebu čvrstog tijela robota. Korištenje isključivo polimera za konstrukciju tijela robota može predstavljati problem kod kliznih ležaja i mehanizma za aktivaciju i deaktivaciju spojke. Korištenjem čeličnih cijevi vanjskog promjera 8 mm i unutarnjeg promjera 6 mm kao osnovu za tijelo kompenziramo nedostatke polimera uz relativno malo uvećanje mase robota. Na slici 14 prikazano je osnovno tijelo robota.



**Slika 14. Osnovno tijelo robota**

Ostatak konstrukcije povezan je polimerima s integriranim kućištem za ležaj pogonskog kotača. Kod linearnih kliznih ležaja mehanizma za skakanje koriste se aluminijski profili za osiguravanje koncentričnosti. Na slici 15 prikazani su klizni ležajevi s ojačanjem od aluminijskih profila.



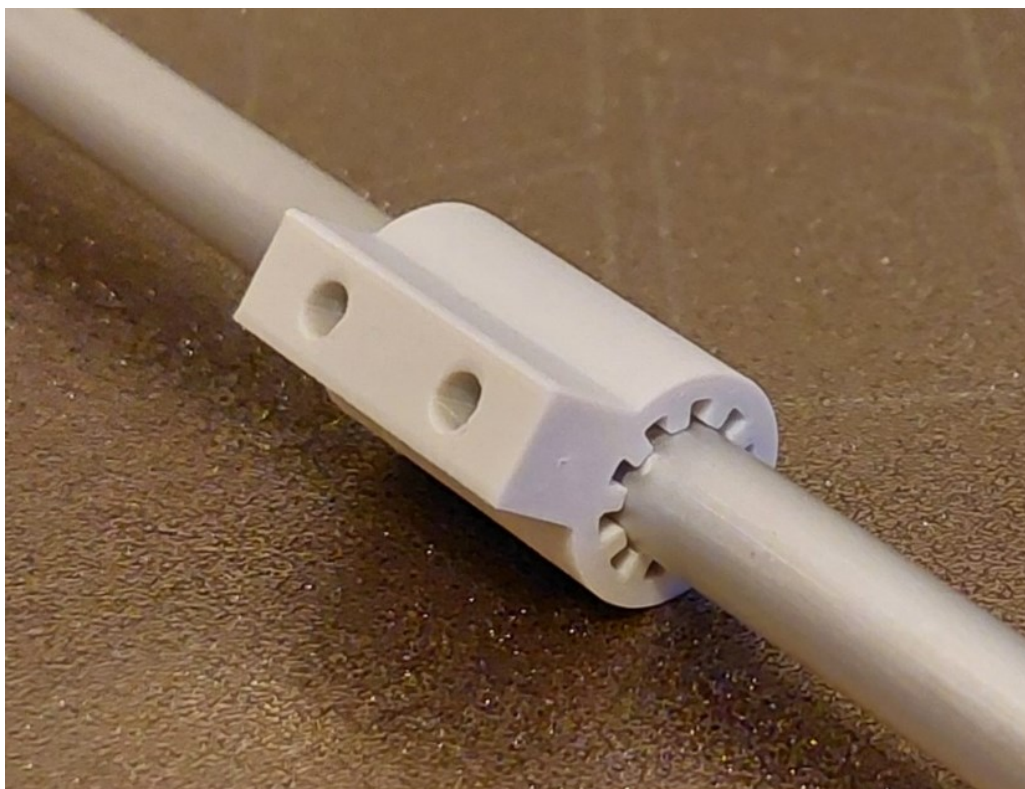
**Slika 15. Klizni ležajevi s ojačanjem**

Na slici se može primijetiti da su aluminijski profili predstavljeni kao svijetlo siva boja, dok su polimerni dijelovi tamno sive boje. Odabir vrste kliznih ležaja i osovine za oprugu prikazat će se u tablici 2.

**Tablica 2. Usporedba osovine i kliznih ležaja**

Materijal osovine	Masa [g]	Kompatibilni ležaji	Ukupna masa [g]
Čelik	156,74	Polimerni/Čelični	164,70/186,10
Aluminij	54,26	Polimerni	62,22

S obzirom da se sila ne prenosi preko osovine za oprugu, može se odabrati aluminijska osovina jer je znatno lakša u odnosu na čeličnu. Polimerni ležajevi se mogu kupiti u standardnim dimenzijama i oblicima ili isprintati koristeći 3D filamente pogodne za klizne ležajeve. S obzirom da se koristi aluminijski „U profil“ poželjno je integrirati u ležajeve oblik profila na montažnom mjestu, što se može izvesti u printanoj izvedbi ležaja. Na slici 16 prikazan je isprintani klizni ležaj koristeći optimalan filament.

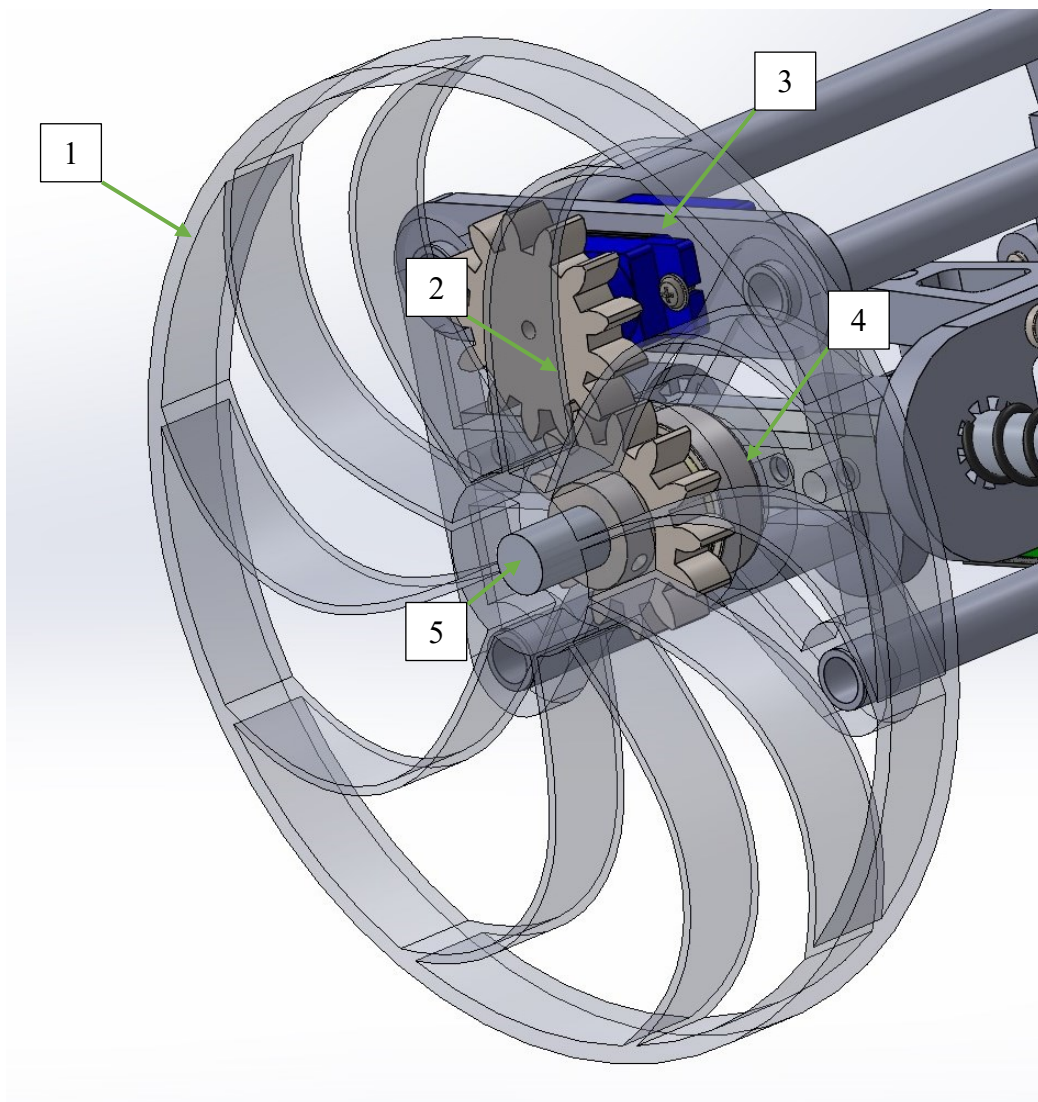


**Slika 16. Klizni ležaj s integriranim profilom**



### 2.2.2. Pogon robota

Kod udarnih opterećenja poželjno je odvojiti vratilo motora od elementa na koji djeluju opterećenja, što je osobito slučaj kod malih elektromotora. Kod robota skakača to bi bio ekvivalent kotaču i njegovom pogonskom elektromotoru. Kontinuirani servo motor male mase poput onog korištenog za mehanizam spojke bit će i ovdje korišten. Servo motor ima prednost u odnosu na standardni elektromotor u smislu integrirane mogućnosti kočenja koja je potrebna pri kompresiji opruge. Prijenos momenta na kotač može se izvesti putem remena, zupčanika ili tarenica. Za ovaj slučaj koristit će se zupčasti prijenos radi istog razloga kao i polimerni ležajevi, jednostavna varijanta pogodna za 3D ispis. Na slici 17 prikazana je shema pogona.

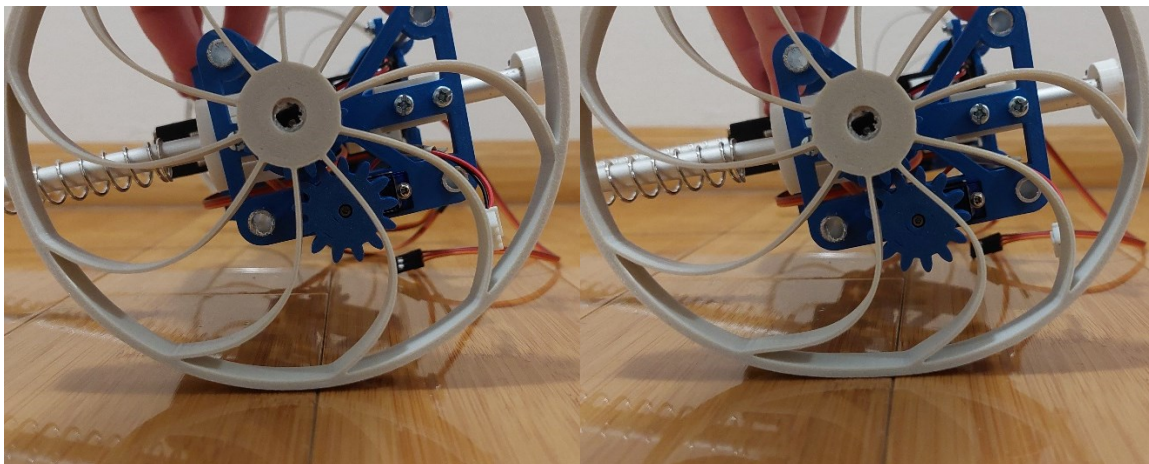


Slika 17. Shema pogona

Oznake na slici 17 su:

- 1) Kotač
- 2) Zupčanici
- 3) Servo motor
- 4) Ležaj
- 5) Vratilo

U slučaju potrebe za većim momentom ili brzinom od osnovne specifikacije servo motora, može se promijeniti odnos zupčastog prijenosa tako da se smanji brzina i poveća moment ili obratno. Kotač robota izradit će se od PETG polimera, jer ima poželjna svojstva na udarna opterećenja i znatno veću deformaciju u elastičnom području. Na slici 18 prikazani su izrađeni kotači u neopterećenom stanju i deformiranom stanju.

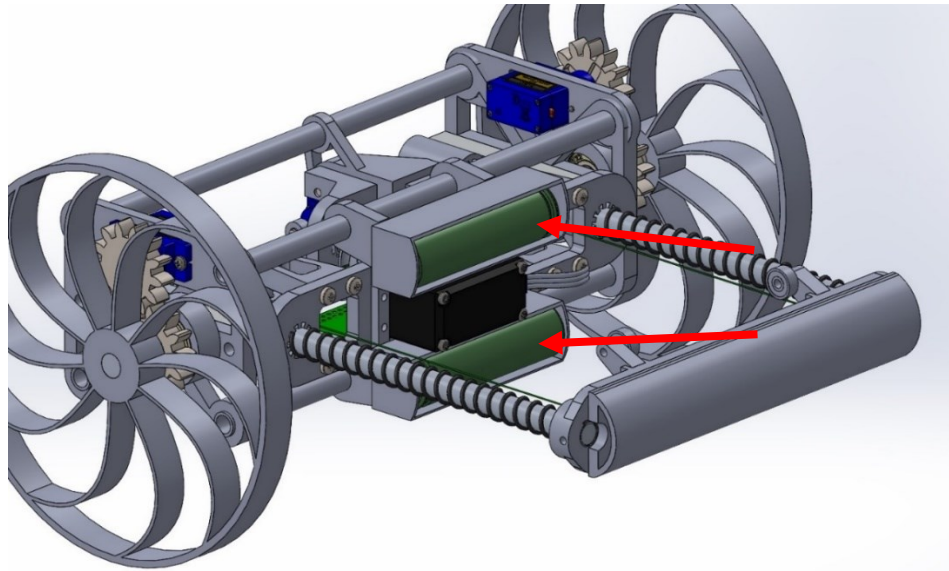


**Slika 18. Kotači u neopterećenom (lijevo) i opterećenom stanju (desno)**

### 2.3. Elektronika za izradu robota skakača

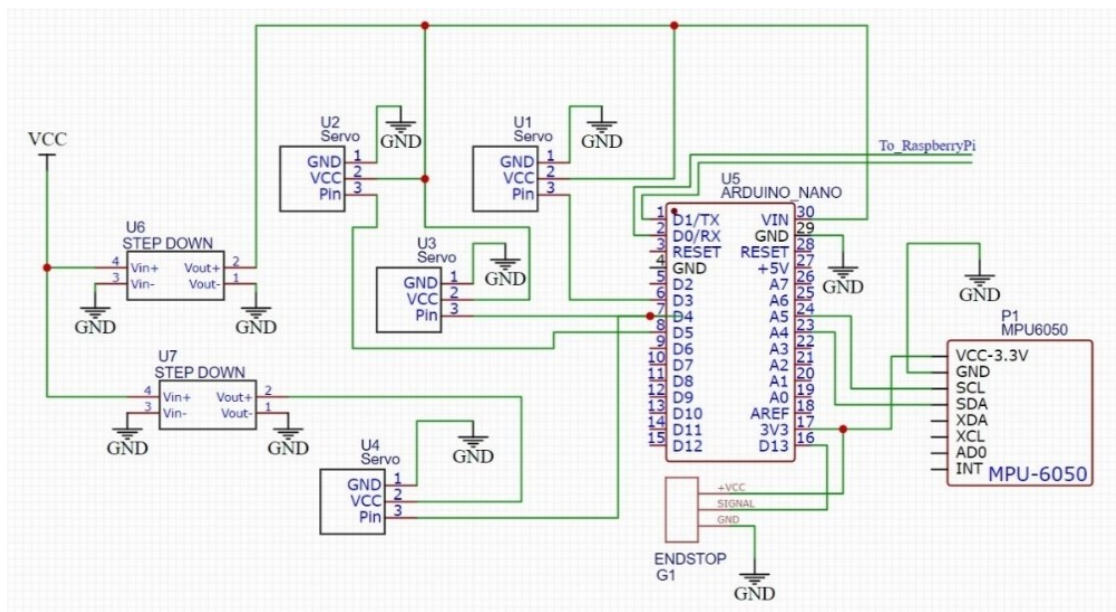
Daljinski upravljani roboti uglavnom koriste baterije kao izvor energije. U većini slučajeva izbor baterije ovisno o masi, izuzev dronova, ne radi veliku razliku u performansama robota. Kod robota skakača masa baterije utječe na visinu skoka. Prvi korak kod odabira baterije bi bio potreban napon, a iza toga kapacitet u odnosu na masu. S obzirom da servo motor za kompresiju opruge radi na 7.2V poželjno je koristiti LiPo baterije u serijskom spoju „2S“ koje daju 7.4V nazivnog napona. Prilagođavanje napona za ostale komponente može se izvršiti putem „Step-down“ DC-DC konvertera, kao što je to slučaj kod ostalih servo motora, mikrokontrolera i

Raspberry Pi-a. Visok omjer Ah/g pružaju 18650 ćelije koje će se ujedno i koristiti kao izvor energije. Montaža baterija na robota će se izvesti imajući u vidu potrebu za centrom mase iza kotača prema odskočnoj podlozi. Na slici 19 prikazana je crvenom strelicom lokacija baterija na robotu.



Slika 19. Lokacija baterija na robotu

Upravljanje elektroničkim komponentama će biti izvršeno putem mikrokontrolera spojenog na Raspberry Pi, a koji prima naredbe od operatera. Na slici 20 prikazana je električna shema spajanja komponenti.



Slika 20. Električna shema spajanja komponenti

Oznake na slici 20 su:

U1, U2 – Pogonski servo motori

U3 – Servo motor za upravljanje spojkom

U4 – Servo motor za kompresiju opruge

U5 – Arduino Nano

U6 – Step-Down modul (7.4V na 5V)

U7 – Step-Down modul (7.4 na 6V)

G1 – Prekidač za položaj U3 Servo motora

S obzirom da se ne može smatrati da će robot biti cijelo vrijeme u vidokrugu operatera, potrebno je omogućiti video prijenos putem kamere s robota za jednostavnije upravljanje. Za taj zadatak može se koristiti OpenHD. OpenHD (Open High Definition) je softverski projekt koji omogućuje visokokvalitetan video prijenos uživo putem radio-veza. Njegova primjena je najčešća u područjima poput daljinskog upravljanja dronovima (FPV - First Person View), bežičnog video nadzora i drugih situacija gdje je potreban prijenos HD videa na velike udaljenosti. OpenHD koristi Wi-Fi adaptere koji podržavaju način rada s proširenim dometom. Video signal s kamere kodira se na uređaju kao što je Raspberry Pi, prenosi putem Wi-Fi veze i dekodira na prijemnom uređaju (obično drugi Raspberry Pi ili računalo). Softver osigurava nisku latenciju, ključnu za FPV aplikacije. Na slici 21 prikazani su preporučeni Wi-Fi adapteri.

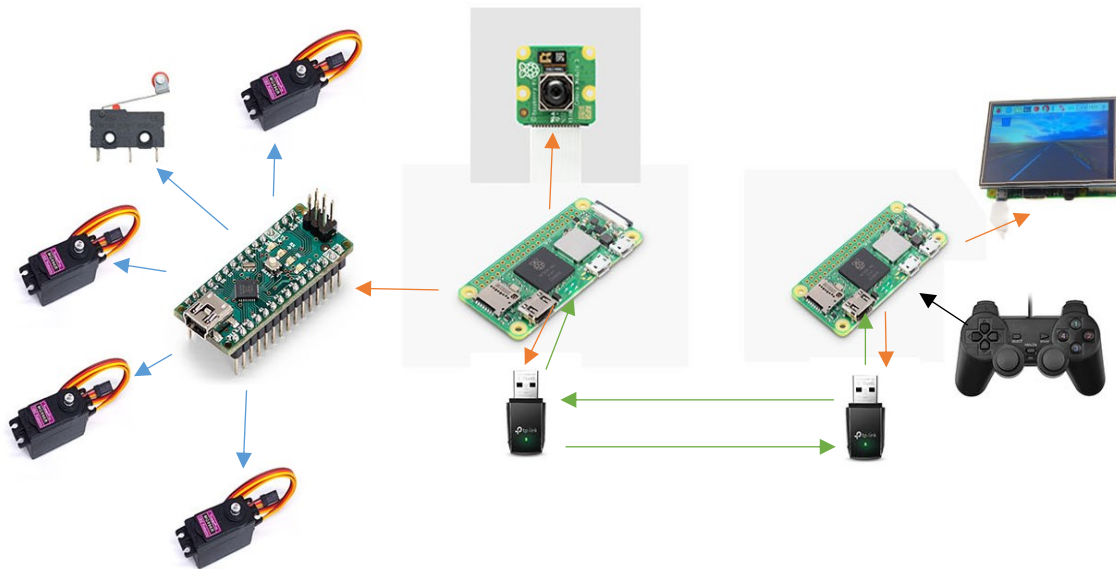
## Recommended WiFi Dongles

Name	Band	TX Power	Chip	Antennas
ALFA AWUS036ACH	5.8 - 2.4*	500mW	RTL8812AU	2x RP-SMA
ASUS USB-AC56	5.8 - 2.4*	500mW	RTL8812AU	1x RP-SMA 1x MS156
"Taobao Card"	5.8 - 2.4*	500mW	RTL8812AU	2x u.fl
ALFA AWUS1900	5.8 - 2.4*	500mW	RTL8814AU	4x RP-SMA
Tenda U12	5.8 - 2.4*	60mW	RTL8812AU	2x MS156 2x internal
Cudy AC 1300	5.8	<50mW	RTL8812BU	internal
Aigital AC1200	5.8	? low power	RTL8812BU	internal
COMFAST 1300Mbps	5.8	? low power	RTL8812BU	internal
D-Link DWA-182	5.8 - 2.4*	70mW	RTL8812BU	2x internal
TP-LINK T3U Plus	5.8 - 2.4*	40mW	RTL8812BU	2x u.fl 1x RP-SMA
TP-Link T3U	5.8 - 2.4*	80mW	RTL8812BU	2x MS156 2x internal

Slika 21. Preporučeni Wi-Fi adapteri [5]

### 2.4. Arhitektura sustava

Sustav je izveden kao komunikacijski sustav između Raspberry Pi računala, na kojem je instaliran OpenHD sustav za prijenos video signala, te Arduino mikrokontrolera koji upravlja izvršnim elementima robota. Sustav omogućava operateru daljinsko upravljanje robotom putem video povratne veze i slanjem upravljačkih signala. Komunikacija između Raspberry Pi računala i Arduino mikrokontrolera realizirana je putem serijske veze koja može biti implementirana na dva načina. Prvi način je korištenje USB povezivanja koje pruža jednostavnost implementacije i visoku pouzdanost. Alternativno, moguće je koristiti UART komunikaciju preko GPIO pinova Raspberry Pi računala, što je posebno korisno u situacijama kada je potrebno minimizirati broj korištenih kabela ili kada su USB portovi potrebni za druge komponente sustava. S obzirom da je korišten Raspberry Pi Zero koji sadrži samo jedan USB port za prijenos podataka, komunikacija će biti napravljena preko UART protokola, jer je USB port zauzet s Wi-Fi adapterom. Shema arhitekture sustava prikazana je na slici 22.



Slika 22. Shema arhitekture robota

## 2.5. Programiranje robota

Programiranje robota izvršava se u Arduino IDE sučelju. Za upravljanje robotom potrebno je isprogramirati:

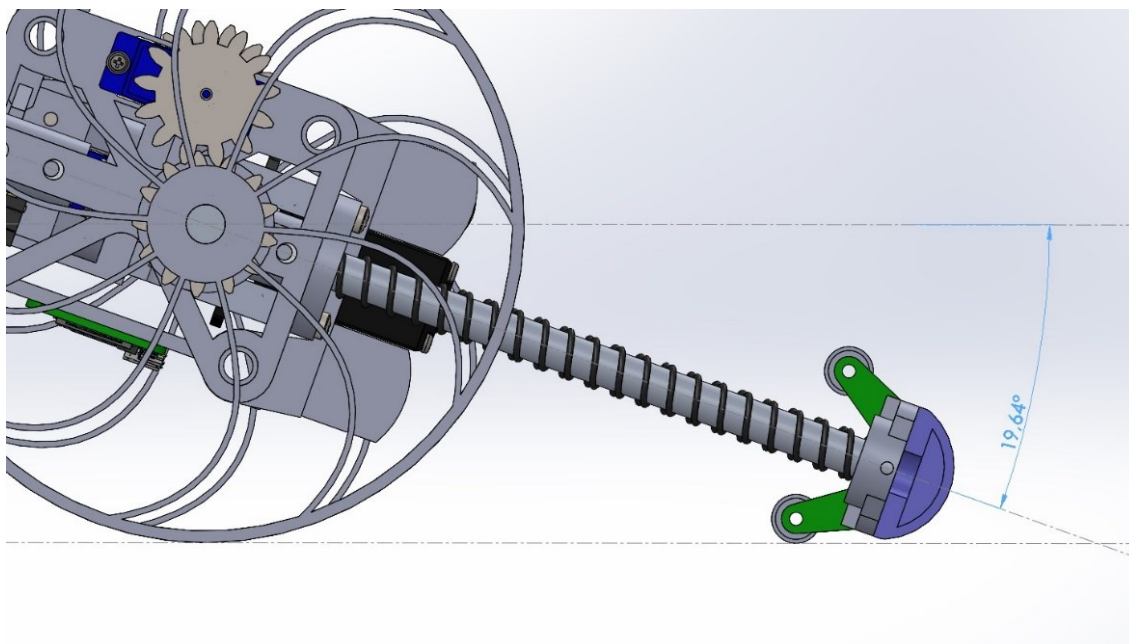
- Kretanje pomoću dva servo motora
- Upravljanje servo motorom za kompresiju opruge
- Upravljanje spojke putem servo motora i prekidača
- Komunikaciju s Raspberry Pi

### 2.5.1. Kretanje pomoću dva servo motora

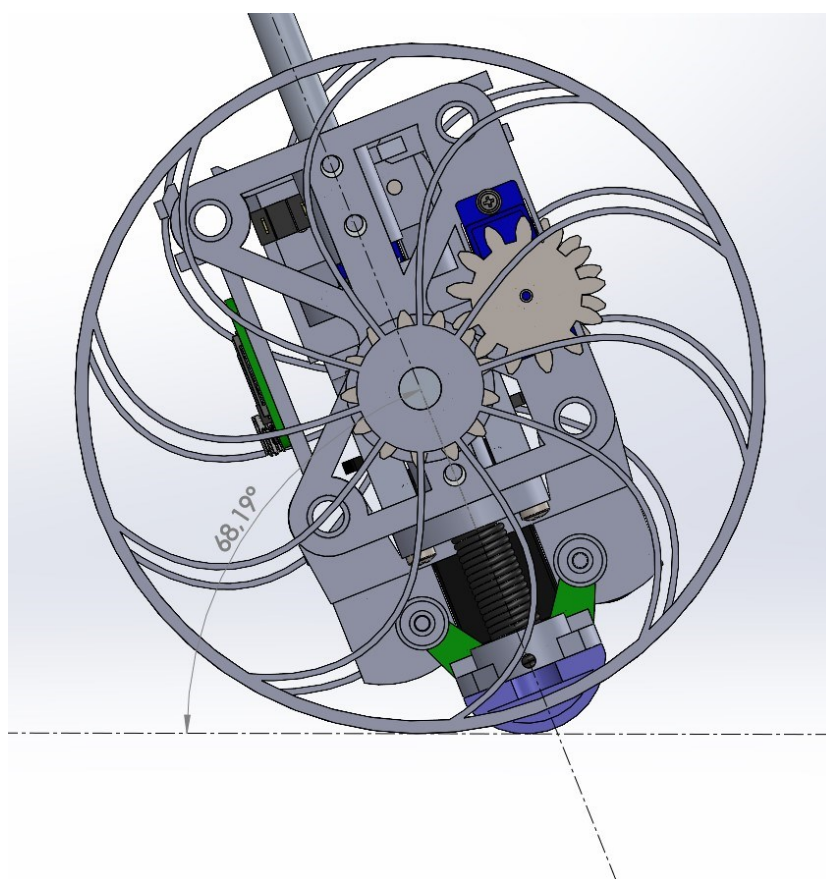
Kod programiranja kretanja, treba uračunati mogućnost robota da se preokrene oko osi i kontrole postanu invertirane. Uz pomoć žiroskopa se to može detektirati i u programskom kodu kompenzirati tako da se robotom može jednostavno upravljati. Upravljača jedinica za robota je upravljačka palica (eng. joystick), koja ima mogućnost slanja vrijednosti u velikom rasponu, što se može isprogramirati na mikrokontroleru kao raspon brzine i intenzitet skretanja.

### 2.5.2. Upravljanje servo motorom za kompresiju opruge

Servo motor za kompresiju opruge koristi žiroskop za određivanje položaja robota tj. kuta skoka i trenutnu duljinu opruge. Teoretski podatci dobiveni iz 3D modela prikazani su na slikama 23 i 24.



Slika 23. Početni kut robota



Slika 24. Maksimalni kut robota

Podatke o maksimalnom i minimalnom kutu robota potrebno je korigirati u stvarnoj izvedbi .

Servo motor se programira tako da prati podatke sa žiroskopa dok komprimira oprugu na željenu poziciju. U maksimalnoj točki potrebno je isprogramirati zaustavljanje servo motora za zaštitu robota i opruge.

### **2.5.3. Upravljanje spojke putem servo motora i prekidača**

Servo motor za upravljanje spojke mora poznavati lokaciju u kojoj se nalazi isključivo kod najviše točke rotirajućeg elementa, referirati se na sliku 7. Zbog geometrije rotirajućeg elementa servo motor se smije rotirati u samo jednom smjeru. Servo motor se programira tako da se rotira u najvišu točku u kojoj se aktivira prekidač i servo motor tada čeka upravljački signal za skok. Kada primi upravljački signal za skok nastavi se rotirati što aktivira mehanizam skoka, pa ponovno stane u točki kad se aktivira prekidač, pri čemu aktivira spojku i čeka servo motor za kompresiju opruge i upravljački signal za skok.

### **2.5.4. Komunikacija s Raspberry Pi-em**

Komunikacija s Arduinoom odvija se putem serijske veze (UART), gdje se mogu slati podatci jedan po jedan (bit po bit) preko komunikacijskog kanala. Osnovni parametri serijske veze:

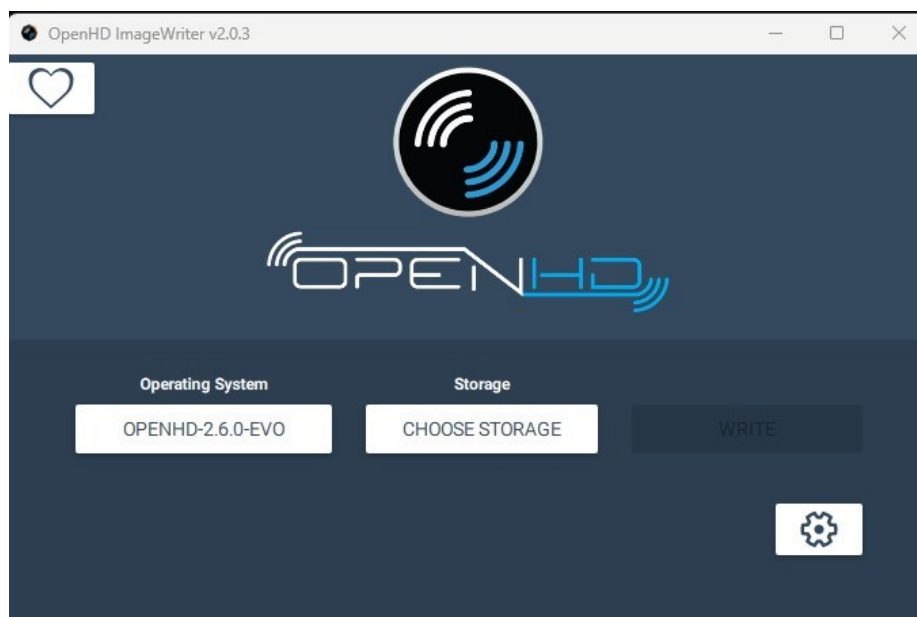
- Baud rate (brzina prijenosa)
  - Najčešće korištene brzine: 9600, 115200 baud
  - Definiira koliko bitova se prenosi u sekundi
  - Mora biti ista na oba uređaja
- Format podataka
  - [Start bit] [Data bitovi (5-9)] [Stop bit(ovi)]

Način rada je isti kao i kod spajanja Windows računala na Arduino.

## **2.6. Instalacija OpenHD sustava**

Instalacija na Raspberry Pi započinje preuzimanjem alata OpenHD ImageWriter, koji omogućuje jednostavno preuzimanje i konfiguraciju sustava. Uz pomoć ovog alata odabire se odgovarajuća slika sustava za određeni model Raspberry Pi uređaja, koja se zatim zapisuje na Micro SD karticu kapaciteta od najmanje 16 GB. Na slici 25 prikazano je sučelje programa ImageWriter.





Slika 25. Sučelje programa ImageWriter

Nakon postavljanja, Raspberry Pi se povezuje s potrebnim perifernim uređajima kao što su WiFi adapter (najčešće se preporučuje model s RTL8812AU ili RTL8812BU čipsetom zbog kompatibilnosti), kamera i, po potrebi, RC prijemnik za daljinsko upravljanje. Uređaj se zatim pokreće, pri čemu OpenHD automatski učitava konfiguracije.

Sustav omogućuje dodatnu prilagodbu postavki prijenosa videa, telemetrije i RC upravljanja putem uređivanja konfiguracijskih datoteka na Micro SD kartici. Aplikacije poput QGroundControl i MissionPlanner mogu se koristiti za integraciju i upravljanje sustavom. Na slici 26 prikazan je Raspberry Pi povezan s kamerom i Wi-Fi adapterom.



Slika 26. Raspberry Pi povezan s kamerom i Wi-Fi adapterom

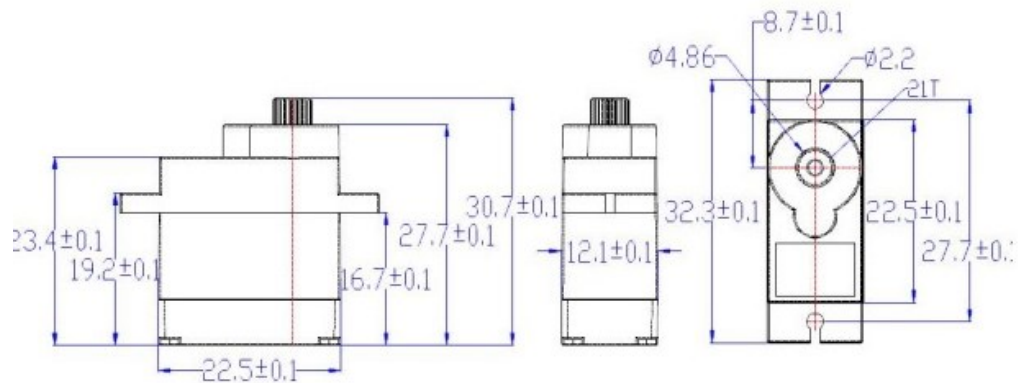
### 3. OPIS KOMPONENTI SUSTAVA

#### 3.1. Komponente pogona

Komponente pogona sastoje se od dva kontinuirana servo motora FS90R koji zupčastim prijenosom pokreću robot s kotačima promjera  $\varnothing 130\text{mm}$ . FS90R je servo motor dizajniran za kontinuiranu rotaciju, što ga čini idealnim za robotske projekte, poput pogonskih sistema s kotačima ili pomicanje platformi. Za razliku od standardnih servomotora koji se koriste za pozicioniranje, FS90R omogućava kontrolu brzine i smjera rotacije pomoću PWM signala. U tablici 3. prikazane su specifikacije FS90R servo motora. Na slici 27 prikazan je servo motor s dimenzijama.

**Tablica 3. Specifikacije FS90R servo motora**

Specifikacija	Opis
Tip	Kontinuirani servo motor
Napajanje	4.8V - 6.0V
Maksimalna brzina	130 RPM (pri 6V)
Upravljački signal	PWM (1-2 ms)
Okretni moment	1.5 kg·cm (pri 6V)
Dimenzije	22.5 mm × 12.0 mm × 29.5 mm
Težina	9 g
Konektor	Standardni 3-pinski (JR konektor)
Radna temperatura	-10°C do 50°C
Materijal kućišta	Plastika



Slika 27. Dimenzije FS90R servo motora [6]

Međuosovinski razmak servo motora i ležaja iznosi 26.5 mm. Za prijenosni omjer 1:1 odabiru se zupčanici putem formule:

$$a = \frac{m(z_1 + z_2)}{2} \quad (3.1)$$

gdje je:

$a$  – razmak osovina – 26.5 mm

$z_1 = z_2$  – broj zuba

S obzirom da će se zupčanici 3D printati, potrebno je imati relativno velike zube radi boljeg printanja zbog toga će se postaviti uvjet da je  $z < 16$ . U tablici 4 prikazan je način odabira zupčanika.

Tablica 4. Odabir zupčastog prijenosa

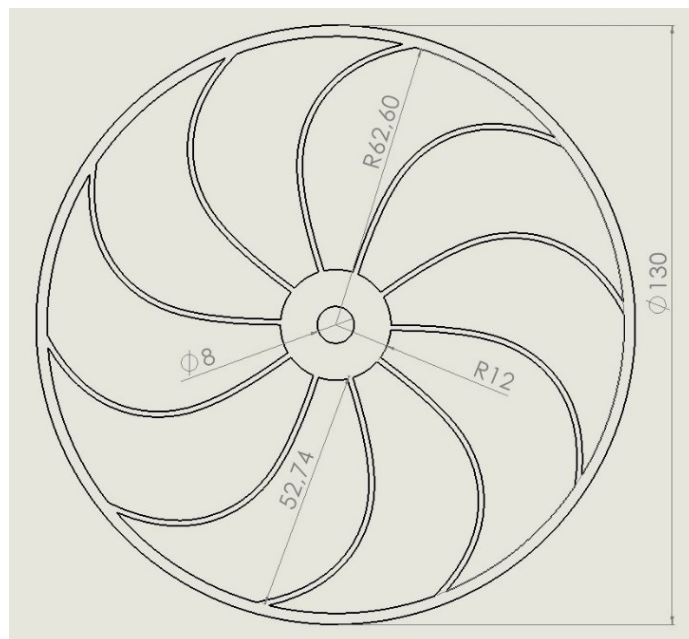
$z$ (broj zuba)	$m = 53 / (2z)$	Standardni modul	Razmak osovina ( $a$ )
15,00	1,77	1,75	26,25
14,00	1,89	1,88	26,25
13,00	2,04	2,00	26,00
12,00	2,21	2,25	27,00
11,00	2,41	2,50	27,50
10,00	2,65	2,50	25,00

Optimalni izbor:

- $z=14, m=1.875$

Standardni modul 1.875 daje razmak osovina 26.25 mm, vrlo blizu ciljanih 26.5 mm. S najmanjim brojem zuba.

Kotač robota s laticama za elastičnu deformaciju pri doskoku s dimenzijama (u milimetrima) prikazan je na slici 28. Materijal kotača je PETG polimer s optimalnim svojstvima za udarna opterećenja.



Slika 28. Dimenzije kotača

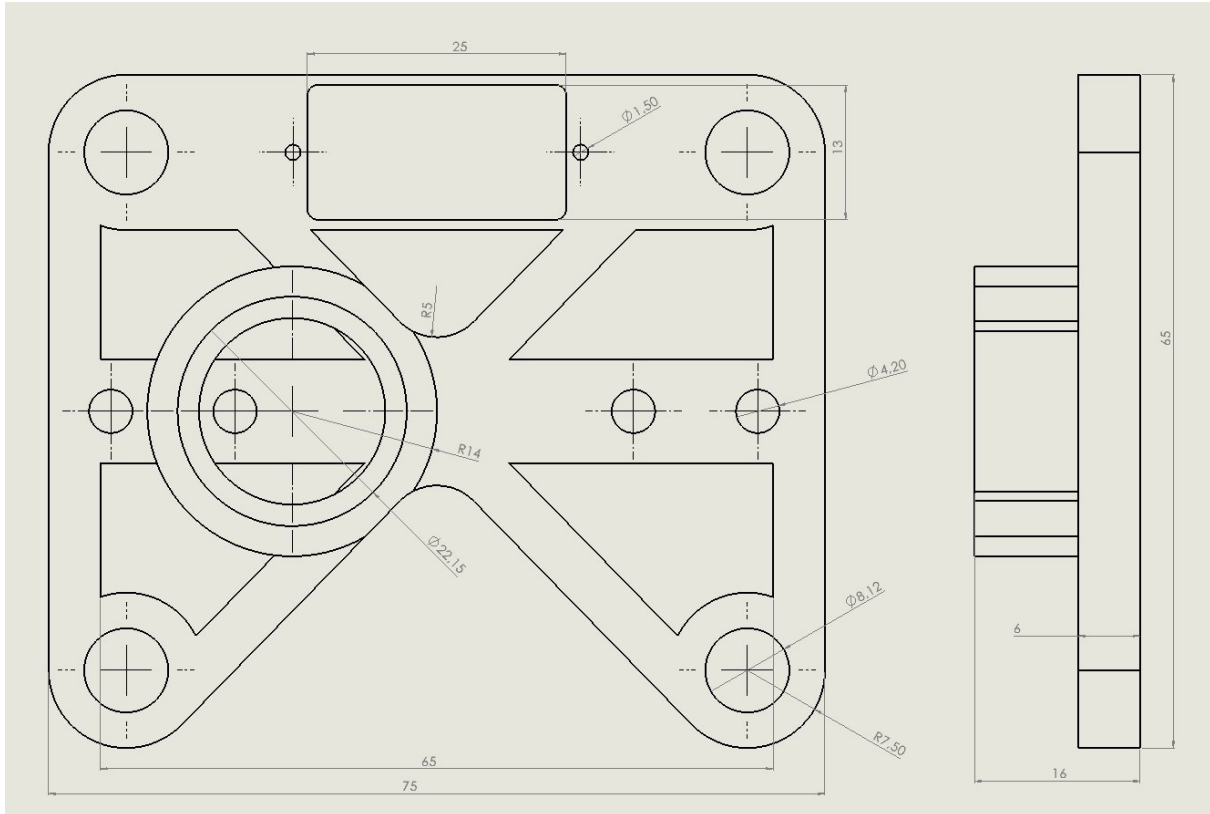
Ležajevi kotača su oznake SKF 608-2Z. Specifikacije ležaja prikazane su u tablici 5.

Tablica 5. Specifikacije 608-2Z ležaja

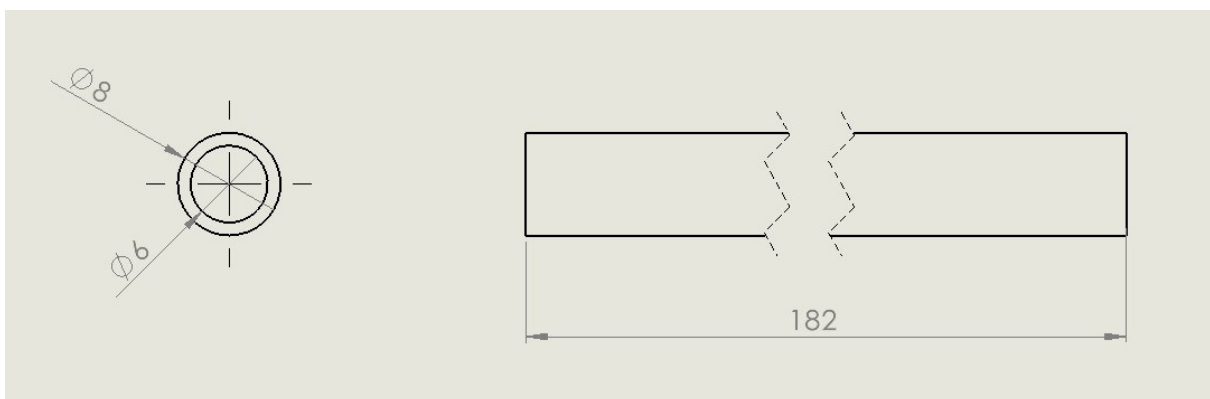
Kategorija	Jednoredni kuglični ležaj
<b>Redovi kuglica</b>	Jednoredni ležaj
<b>Materijal</b>	Standardni čelik za ležaje
<b>Vanjski prsten</b>	Bez utora
<b>Zatvaranje</b>	Zatvoren sa obje strane
<b>Kavez</b>	Kavez od čelika
<b>Raspon temperature</b>	-20°C do +120°C
<b>Unutarnji promjer (mm)</b>	8
<b>Vanjski promjer (mm)</b>	22
<b>Širina (mm)</b>	7

### 3.2. Komponente tijela

Kao što je već spomenuto u poglavlju 3.2.1, tijelo robota sastoji se od čeličnih cijevi i printanih polimera. Detaljnije dimenzije (u milimetrima) tih komponenti prikazani su na slikama 29 i 30.



Slika 29. Dimenzije polimernog okvira



Slika 30. Dimenzije spojnih cijevi

### 3.3. Komponente mehanizma za skakanje

Mehanizam za skakanje sastoji se od

- MG995 Servo motora
- Hirthove spojke
- Kliznih ležaja
- Kolotura za namatanje užeta
- Uže
- Osovine za vođenje opruge
- Opruge
- FS90R servo motora
- Odskočna podloga
- Prekidač za poziciju spojke

#### 3.3.1. Servo motor MG995

MG995 je snažan servo motor s mogućnošću kontinuirane rotacije, široko korišten u robotskim aplikacijama, poput robotiziranih ruku, vozila i sustava za kretanje. Uz kontinuiranu rotaciju, MG995 koristi PWM signal za kontrolu brzine i smjera rotacije. Zahvaljujući izdržljivom dizajnu, može raditi u raznim uvjetima i nositi se s većim potrebama za momentom zakreta. Na slici 31 prikazan je MG995 servo motor.



Slika 31. MG995 kontinuirani servo motor [7]

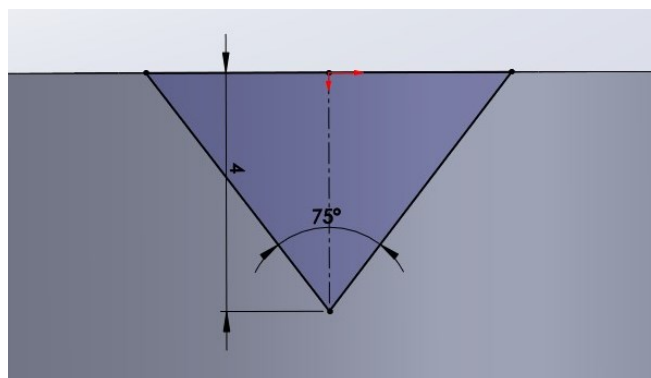
U tablici 6 prikazane su specifikacije MG995 servo motora.

**Tablica 6. Specifikacije MG995 servo motora**

Specifikacija	Vrijednost
Tip	Kontinuirani servo motor
Napajanje	4.8V - 7.2V
Maksimalna brzina	60°/0.2 s (pri 4.8V)
Okretni moment	10 kg·cm (pri 4.8V), 13 kg·cm (pri 7.2V)
Upravljački signal	PWM (1-2 ms)
Zupčanci	Metalni
Dimenzije	40.7 mm × 19.7 mm × 42.9 mm
Težina	55 g
Konektor	Standardni 3-pinski (JR konektor)
Radna temperatura	-10°C do 60°C
Materijal kućišta	Plastika s metalnim zupčanicima

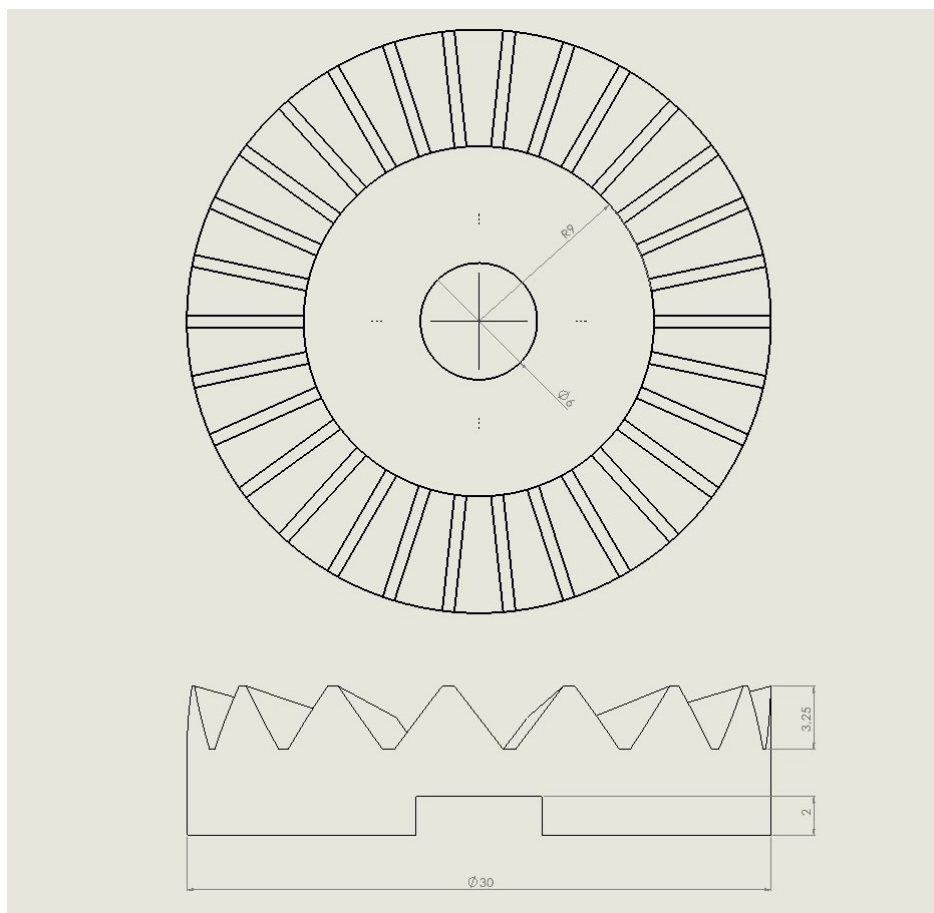
### 3.3.2. Hirth spojka

Hirth spojka se sastoji od međusobno urezanih i nazubljenih površina (često u obliku zubaca na krajevima osovine), koje omogućuju precizno poravnanje i prijenos velikih zakretnih momenata. Zubi su raspoređeni radijalno pod točno određenim kutovima, što osigurava visoku preciznost i ponovljivost spajanja. Na slici 32 prikazan je profil zuba povećan na 75° u odnosu na standardi kut od 60° za lakše odvajanje spojke pri otpuštanju užeta.



**Slika 32. Profil zuba Hirthove spojke**

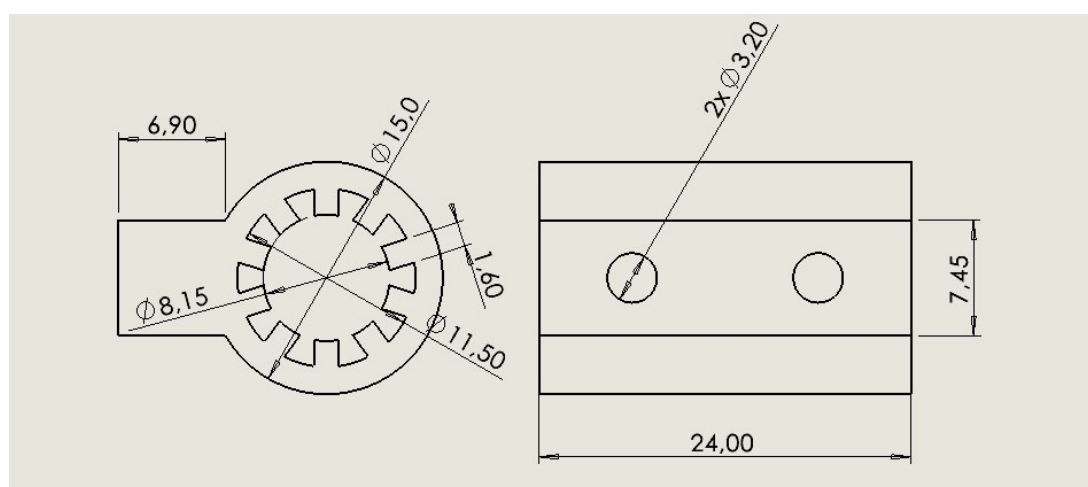
Glavne dimenzije (u milimetrima) spojke prikazane su na slici 33.



Slika 33. Dimenzije Hirth spojke

### 3.3.3. Klizni ležaj

Klizni ležajevi služe za linearno kretanje osovine koja vodi kompresijsku oprugu. Dimenzije (u milimetrima) kliznih ležajeva prikazani su na slici 34.



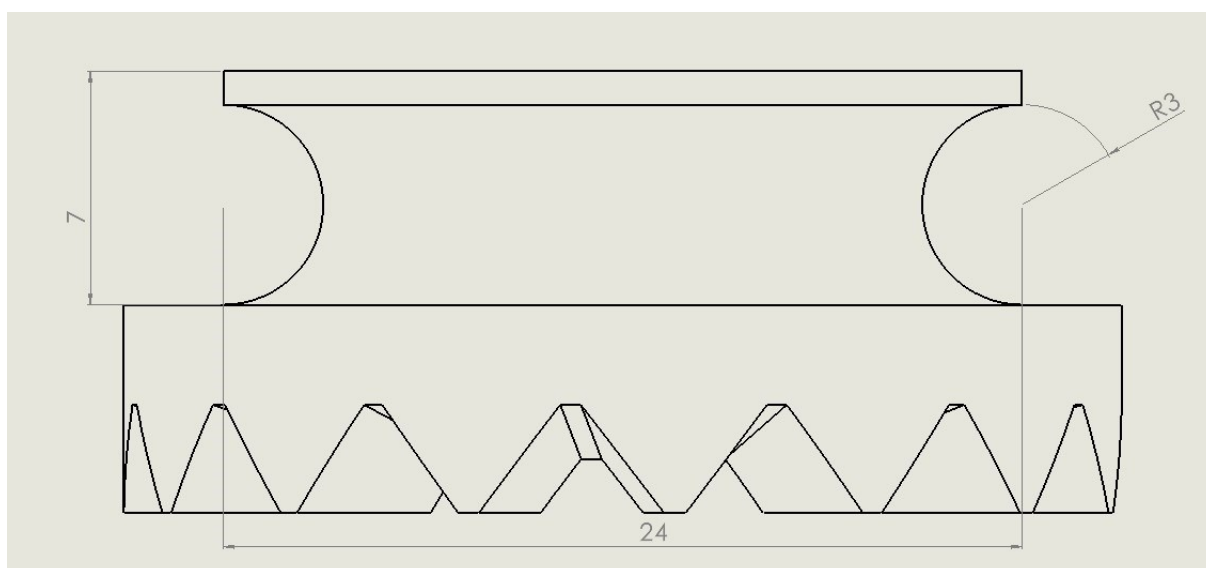
Slika 34. Dimenzije kliznog ležaja



Unutarnji promjer ležaja je uvećan u odnosu na osovinu radi skraćivanja polimera pri hlađenju nakon printanja.

### 3.3.4. Kolotura za namatanje užeta

Kolotura za namatanje užeta spojena je s drugom polovicom Hirthove spojke jer se zajedno okreću na ležaju pri odmatanju užeta. Na slici 35 prikazane su dimenzije (u milimetrima) koloture.



Slika 35. Dimenzije koloture

### 3.3.5. Uže

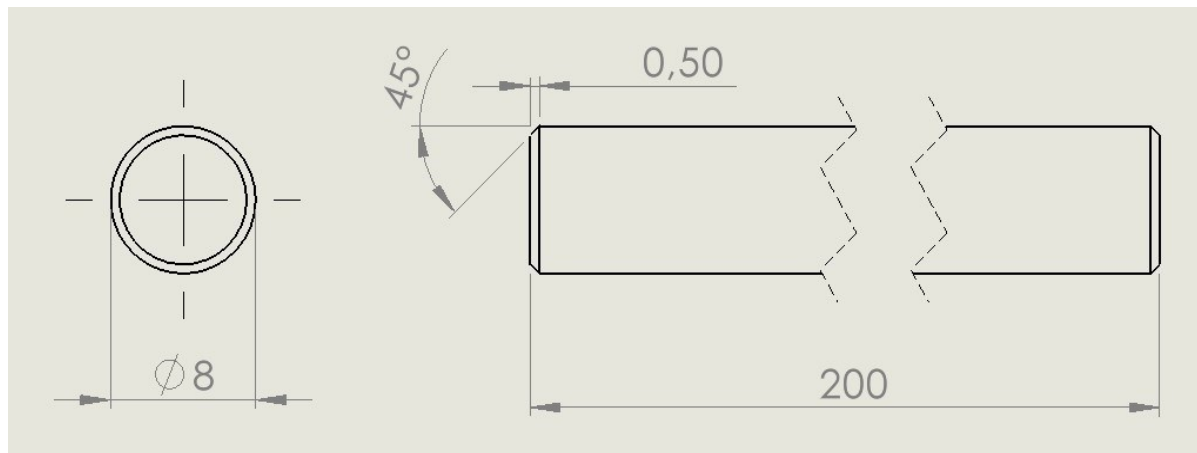
Materijal za uže je monofilament. Monofilament je materijal izrađen od jedne čvrste niti sintetičkog polimera, najčešće najlona, polietilena ili polipropilena. Kada se koristi kao uže, monofilament ima specifična svojstva koja ga čine pogodnim za određene aplikacije, ali i ograničenja koja treba uzeti u obzir. U tablici 7 prikazana su svojstva korištenog monofilamenta.

Tablica 7. Svojstva monofilamenta

Specifikacija	Vrijednost
Nosivost	13,12 kg
Promjer	0,4 mm
Rastezanje	6%

### 3.3.6. Osovine za vođenje opruge

Osovine za vođenje opruge su aluminijske šipke izrezane na potrebnu duljinu. Dimenzije (u milimetrima) osovine prikazane su na slici 36.



Slika 36. Dimenzije osovine za vođenje opruge

### 3.3.7. Opruge

Robot ima dvije opruge od kojih svaka može imati maksimalnu silu od 65 N pri najvećem sabijanju. Specifikacija korištene opruge prikazana je u tablici 8.

Tablica 8. Specifikacija kompresijske opruge

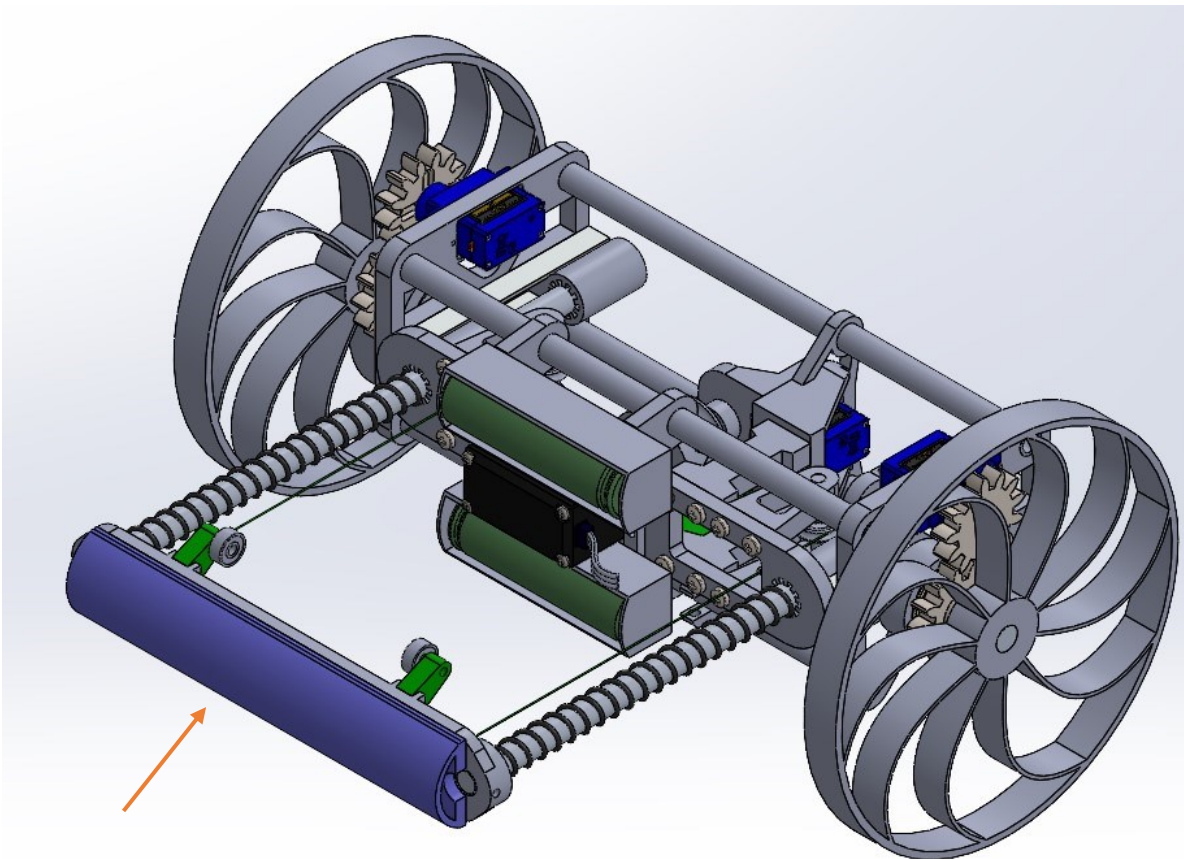
Specifikacija	Vrijednost
Vrsta opruge	Tlačna (linearno elastična)
Ukupna dužina	120 mm
Minimalna dužina	20 mm
Maksimalna kompresija	100 mm
Maksimalna sila	65 N
Konstanta opruge (k)	0,65 N/mm
Materijal	Opružni čelik

### 3.3.8. FS90R servo motor

Koristi se isti servo motor FS90R za upravljanje spojke kao i kod pogona robota.

### 3.3.9. Odskočni podložak

Odskočni podložak ključna je komponenta za skok robota. Mora biti kruta da se ne deformira prilikom skoka robota, ali u isto vrijeme mora osigurati zadovoljavajuće trenje s podlogom jer u suprotnom robot ne može skočiti. U slučaju nedovoljnog trenja opruga samo gura podložak po podlozi što može slomiti graničnike ili ležajeve uslijed razvijanja prevelike brzine pri neopterećenom mehanizmu za skakanje. Odskočni podložak također pomaže osovina da održe jednak međuosovinski razmak kao i u ležajevima, što osigurava neometano kretanje osovine u ležajima. Također je potrebno da ima kotačiće radi lakšeg kretanja pri ispruženom mehanizmu kao što je bilo prikazano na slici 12. Na slici 37. prikazan je odskočni podložak u kontekstu cijelog robota.



Slika 37. Odskočni podložak u kontekstu robota

Kao kotačići, koriste se 623Z kuglični ležajevi. S obzirom da služe isključivo za razdvajanje podloška pri ispruženom mehanizmu za skakanje, ne treba dodavati na njih materijal većeg trenja jer bi negativno utjecali pri skretanju robota. Specifikacije 623Z ležaja nalaze se u tablici 9.

Tablica 9. Specifikacije 623Z ležaja

Specifikacija	Vrijednost
Unutarnji promjer (d)	3 mm
Vanjski promjer (D)	10 mm
Širina (B)	4 mm
Tip zaštite	Z (jedan metalni zaštitni prsten)
Materijal	Kromirani čelik (AISI 52100)
Dinamičko opterećenje (C)	600 N
Statičko opterećenje (C <sub>0</sub> )	200 N
Maksimalna brzina rotacije	45,000 okr/min (ovisno o podmazivanju)
Težina	2 g

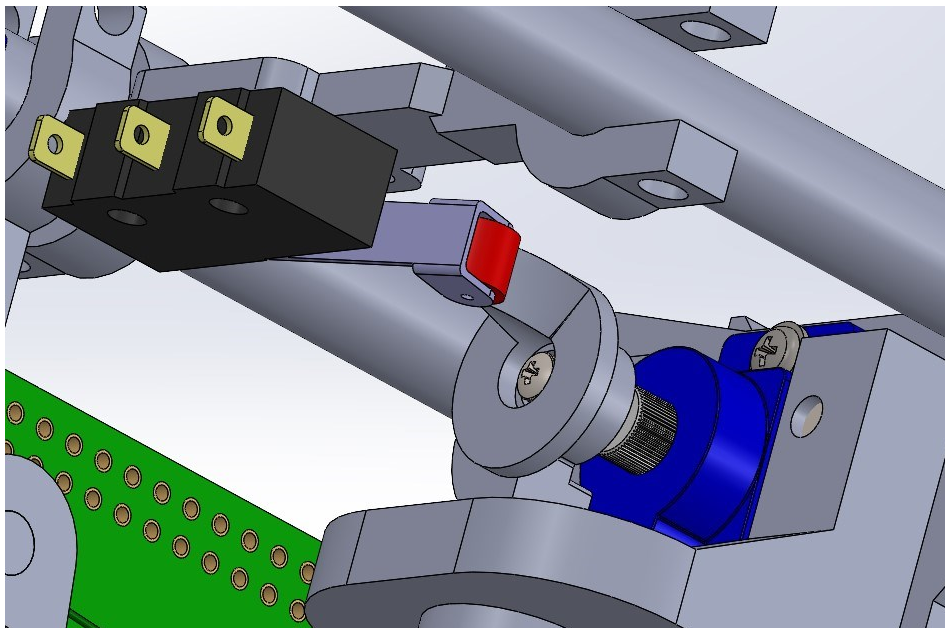
### 3.3.10. Prekidač za poziciju spojke

Za upravljanje spojkom koristi se kontinuirani servo motor, koji ne daje povratne informacije o položaju. Za upravljanje spojkom mora biti poznato kada se servo motor nalazi u poziciji prije odvajanja spojke tj. na vrhu mehanizma. Na slici 38 prikazan je prekidač za poziciju spojke.



Slika 38. Mikroprekidač KW12-3 [8]

Prekidač se montira na konstrukciju robota u ravnini s rotirajućim elementom na servo motoru. Na slici 39 prikazan je prekidač u kontekstu mehanizma.

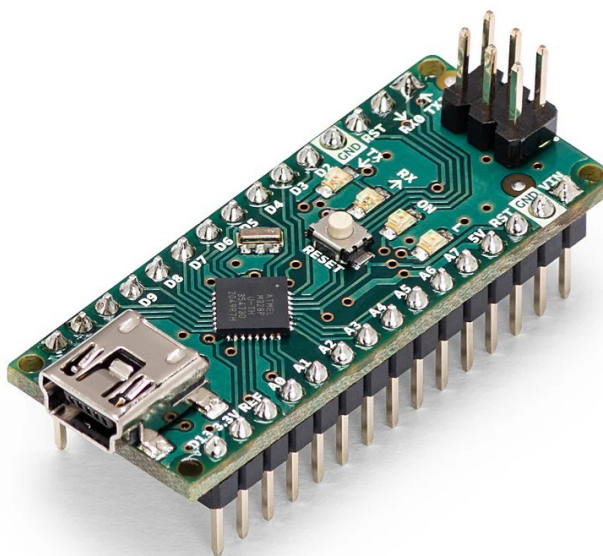


Slika 39. Prekidač u dodiru s rotirajućim elementom na servo motoru

### 3.4. Elektroničke komponente robota

#### 3.4.1. Mikrokontroler Arduino Nano

Arduino Nano je kompaktna mikrokontrolerska ploča bazirana na ATmega328P procesoru, dizajnirana za projekte koji zahtijevaju malu veličinu i nisku potrošnju energije. Nano podržava sve osnovne funkcionalnosti Arduino platforme, a programira se pomoću Arduino IDE-a koristeći USB vezu. Na slici 40 prikazan je Arduino Nano.



Slika 40. Mikrokontroler Arduino Nano [9]

U tablici 10 prikazane su specifikacije mikrokontrolera.

**Tablica 10. Specifikacije mikrokontrolera Arduino Nano**

Specifikacija	Opis
<b>Mikrokontroler</b>	ATmega328P
<b>Radni napon</b>	5V
<b>Ulazni napon (preko VIN)</b>	7-12 V
<b>Digitalni I/O pinovi</b>	14 (od kojih 6 podržavaju PWM izlaz)
<b>Analogni ulazni pinovi</b>	8
<b>Flash memorija</b>	32 KB (od čega 2 KB koristi bootloader)
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1 KB
<b>Brzina procesora</b>	16 MHz
<b>USB konektor</b>	Mini USB
<b>Dimenzije</b>	45 mm x 18 mm
<b>Težina</b>	7 g

### 3.4.2. Računalo robota

Raspberry Pi Zero 2 W je kompaktno i energetski učinkovito računalo razvijeno za aplikacije gdje su veličina, cijena i funkcionalnost ključni faktori. Predstavlja napredniju verziju popularnog Raspberry Pi Zero W modela, s boljim performansama zahvaljujući višestruko jačem procesoru. Kao i ostali Raspberry Pi uređaji, Zero 2 W koristi Raspberry Pi OS ili druge operacijske sustave zasnovane na Linuxu, a programira se pomoću Python-a, C-a, ili drugih jezika kompatibilnih s ARM arhitekturom. Na slici 41 prikazan je Raspberry Pi Zero 2 W.



Slika 41. Računalo Raspberry Pi Zero 2 W [10]

U tablici 11 prikazane su specifikacije Raspberry Pi Zero 2 W računala.

Tablica 11. Računalo Raspberry Pi Zero 2 W

Specifikacija	Opis
<b>Procesor (CPU)</b>	Broadcom BCM2710A1, Quad-core ARM Cortex-A53 (64-bit), 1 GHz
<b>Grafički procesor (GPU)</b>	VideoCore IV
<b>Radna memorija (RAM)</b>	512 MB LPDDR2
<b>Bežična povezanost</b>	Wi-Fi 4 (802.11 b/g/n), Bluetooth 4.2, BLE
<b>USB portovi</b>	1 x Micro USB za podatke, 1 x Micro USB za napajanje
<b>HDMI izlaz</b>	Mini HDMI
<b>GPIO</b>	40-pin header (nelemljen)
<b>Konektor za kameru</b>	CSI-2 kamere (kompatibilno s Raspberry Pi kamerama)
<b>Operativni sustav</b>	Raspberry Pi OS (prethodno Raspbian)
<b>Napajanje</b>	5V/2.5A preko Micro USB
<b>Dimenzije</b>	65 mm x 30 mm x 5 mm
<b>Težina</b>	9 g

### 3.4.3. Raspberry Pi kamera V3

Raspberry Pi kamera V3 opremljena je s 12-megapikselnim Sony IMX708 senzorom, s mogućnošću automatskog fokusiranja, te dolazi u verzijama s različitim kutovima vidnog polja (standardna i širokokutna) i NoIR varijantama za infracrvenu osjetljivost. Na slici 42 prikazana je Raspberry Pi kamera V3.



Slika 42. Raspberry Pi kamera V3 [11]

U tablici 12 prikazane su specifikacije Raspberry Pi kamere.

Tablica 12. Specifikacije Raspberry Pi kamere

Specifikacija	Opis
<b>Senzor</b>	Sony IMX708, 12 MP
<b>Rezolucija</b>	4608 x 2592 piksela
<b>Automatski fokus (AF)</b>	Standardna: 10 cm do $\infty$ , Širokokutna: 5 cm do $\infty$
<b>Kut vidnog polja (FoV)</b>	Standardna: 75°, Širokokutna: 120°
<b>Podržani video načini</b>	1080p (50 fps), 720p (100 fps), 480p (120 fps)
<b>HDR</b>	Do 3 MP rezolucije



#### 3.4.4. *Tp-link T3U Wi-Fi adapter*

Tp-link T3U Wi-Fi adapter koristi RTL8812BU chipset. RTL8812BU chipset je postao standard za OpenHD sustave zbog nekoliko važnih karakteristika. Njegova ključna prednost je podrška za Monitor Mode koji omogućava izravno praćenje Wi-Fi prometa bez potrebe za povezivanjem na mrežu, što je nužno za FPV video streaming. RTL8812BU podržava rad na dvije frekvencije (2.4 GHz i 5 GHz) te može postići brzine prijenosa do 867 Mbps, uz izvrsnu osjetljivost prijemnika. Na slici 43 prikazan je Tp-link T3U Wi-Fi adapter.



Slika 43. Wi-Fi adapter Tp-link T3U [12]

#### 3.4.5. *Baterija litij-ionska 18650*

Baterije 18650 su cilindrične litij-ionske punjive baterije koje su popularne zbog svoje energetske gustoće, dugog vijeka trajanja i široke primjene u raznim uređajima. Njihovo ime dolazi iz dimenzija: 18 mm promjera i 65 mm duljine. To su jedne od najčešće korištenih baterija u prijenosnoj elektronici, električnim vozilima, svjetiljkama, alatima. Tipične značajke 18650 baterija prikazane su u tablici 13.

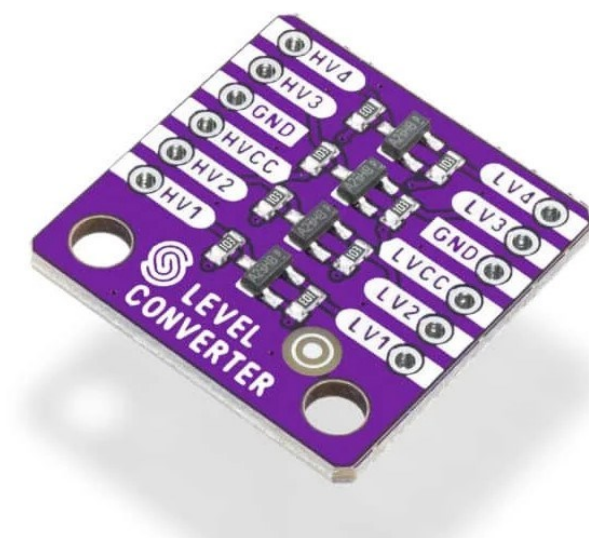
Tablica 13. Baterije 18650 litij-ionske

Specifikacija	Opis
<b>Kemija</b>	Litij-ionska (Li-ion)
<b>Napon</b>	Nominalni napon: 3.6V - 3.7V; puni napon: ~4.2V; minimalni napon: ~2.5V
<b>Kapacitet</b>	Obično između 1200 mAh i 3500 mAh, ovisno o modelu i kvaliteti.
<b>Energetska gustoća</b>	Visoka, što omogućuje dug radni vijek uređaja uz malu težinu.

<b>Punjenje</b>	Specijalizirani punjači; koristi se CC/CV metoda (konstantna struja/napon).
<b>Vijek trajanja</b>	Oko 300-500 ciklusa punjenja i pražnjenja uz pravilno korištenje.
<b>Masa</b>	Oko 45 grama, ovisno o modelu i kapacitetu

### 3.4.6. Pretvarač logičkih razina

Pretvarač logičkih razina je elektronička komponenta ili sklop koji omogućuje međusobnu komunikaciju uređaja s različitim logičkim naponima. Mnogi moderni mikrokontroleri i senzori rade na 3.3V logičkim razinama, u ovom slučaju Raspberry Pi Zero 2 W, dok stariji uređaji, poput Arduino Nano koriste 5V logiku. Pretvarači logičkih razina omogućuju siguran prijenos podataka između takvih uređaja bez rizika od oštećenja zbog nekompatibilnih napona. Na slici 44 prikazan je pretvarač logičkih razina tvrtke Soldered.



Slika 44. Pretvarač logičkih razina tvrtke Soldered [12]

### 3.4.7. Akcelerometar i žiroskop MPU6050

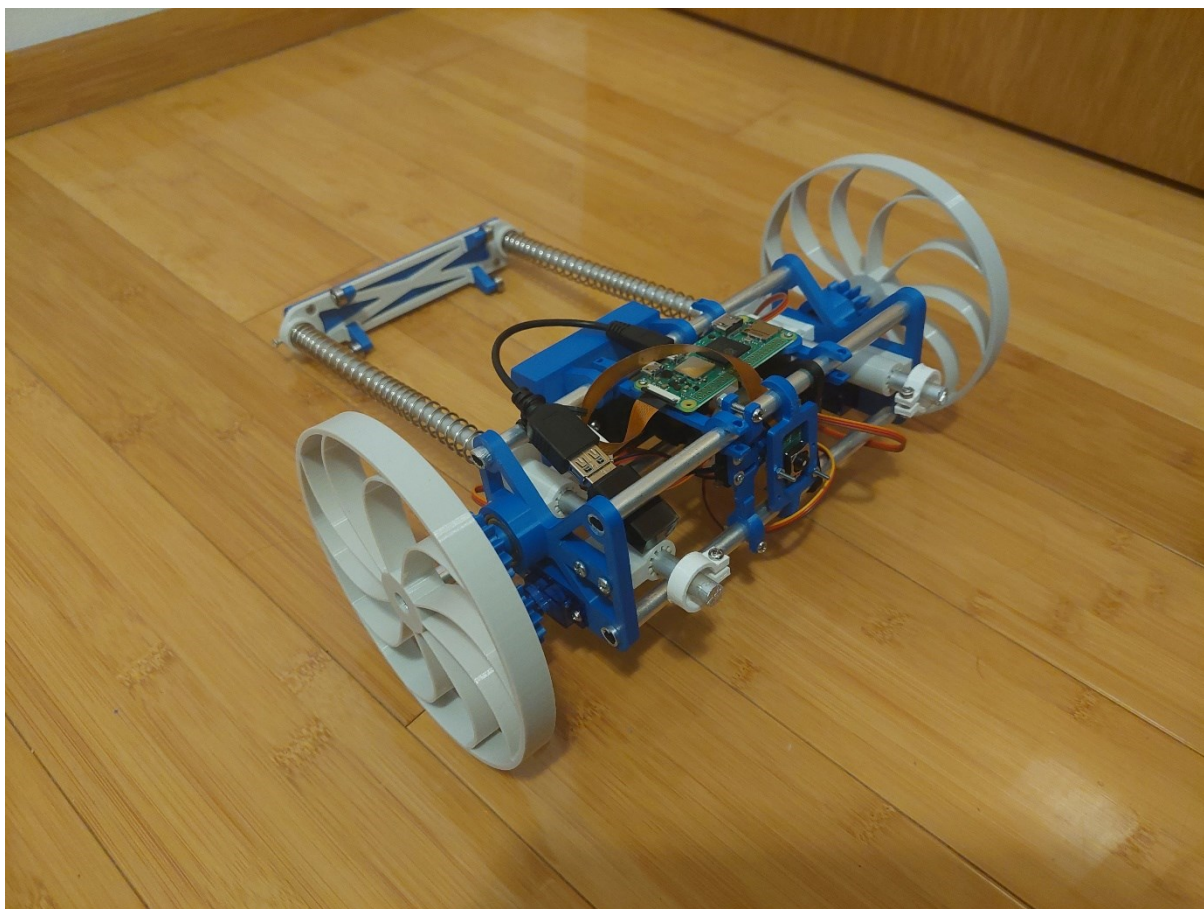
MPU6050 je senzorski modul koji kombinira troosni akcelerometar i troosni žiroskop u jednom integriranom čipu, što ga čini idealnim za aplikacije koje zahtijevaju mjerenje pokreta, orijentacije i ubrzanja. Također uključuje DMP (Digital Motion Processor), koji omogućuje obradu podataka unutar senzora prije nego što ih korisnik pročita, smanjujući opterećenje mikrokontrolera. U tablici 14 prikazane su specifikacije senzorskog modula MPU6050.

Tablica 14. Senzorski modul MPU6050

Specifikacija	Opis
<b>Akcelerometar</b>	Troosni (X, Y, Z); raspon: $\pm 2g$ , $\pm 4g$ , $\pm 8g$ , $\pm 16g$ .
<b>Žiroskop</b>	Troosni (X, Y, Z); raspon: $\pm 250^\circ/s$ , $\pm 500^\circ/s$ , $\pm 1000^\circ/s$ , $\pm 2000^\circ/s$ .
<b>Napajanje</b>	3.3V ili 5V (ovisno o modulu).
<b>Komunikacija</b>	I2C protokol (adresa: 0x68 ili 0x69).

## 4. IZVEDBA ROBOTA

Na slici 45 prikazana je izvedba robota koji je opremljen svim dijelovima za rad.



Slika 45. Izvedba robota

### 4.1. Konstrukcija i masa

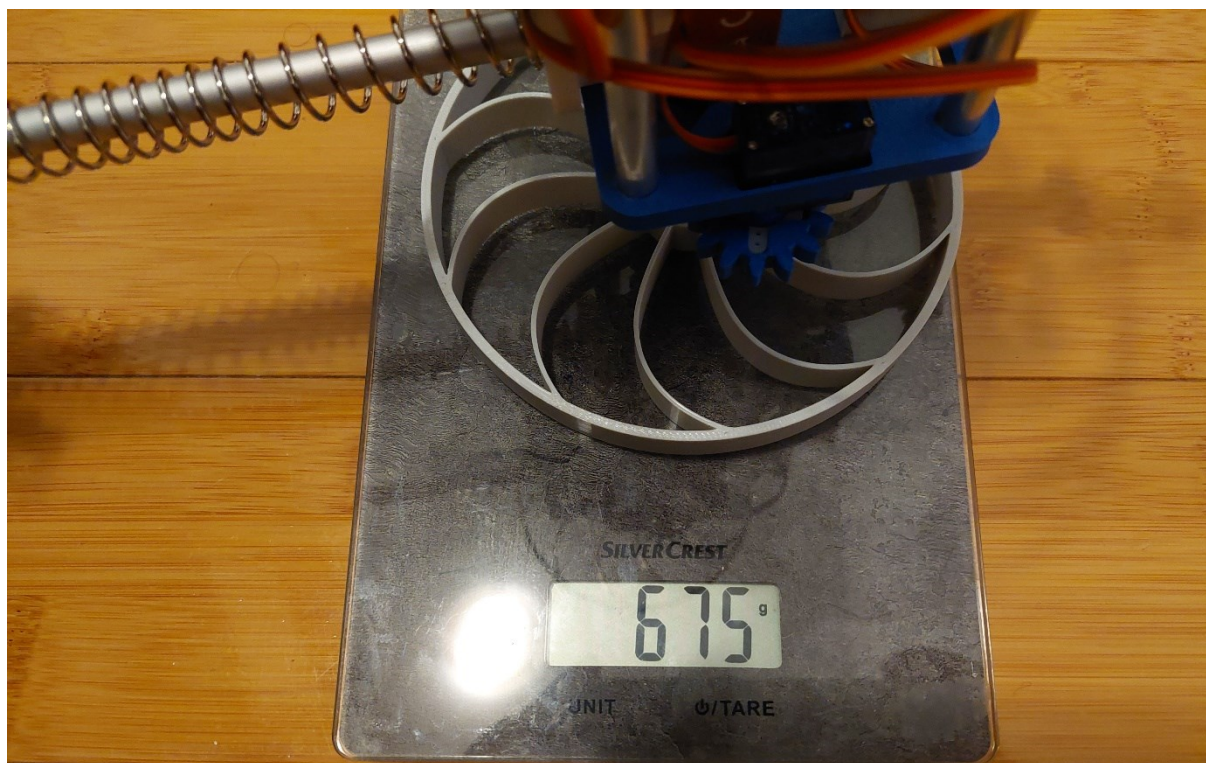
Većina konstrukcije robota izrađena je od PLA i PETG polimera. Ojačanja su izrađena od aluminijskih šipki i čeličnih cijevi. U tablici 15 prikazan je popis svih izrađenih konstrukcijskih elemenata s materijalom, količinom i masom.

Tablica 15. Popis svih izrađenih konstrukcijskih elemenata

Element	Materijal	Količina	Masa
Nosač servo motora i pomoćnih elemenata	PLA	1	23,63 g
Držac kolotura	PLA	4	2,43 g
Hirthova spojka - servo motor	PLA	1	3,09 g

<b>Hirthova spojka - kolotura</b>	PLA	1	4,76 g
<b>Spojnica servo motora koloture</b>	PLA	2	3,19 g
<b>Držač servo motora spojke</b>	PLA	1	9,58 g
<b>Oslonac s ležajem za osovinu koloture</b>	PETG	1	5,77 g
<b>Rotirajući element koloture</b>	PLA	1	0,94 g
<b>Klizni ležajevi osovine za oprugu</b>	PETG	4	3,97 g
<b>Kotač</b>	PETG	2	28,34 g
<b>Okvir odskočnog podloška</b>	PETG	1	13,36 g
<b>Zakrivljeni dio odskočnog podloška</b>	PLA	1	18,68 g
<b>Nosač kotačića odskočnog podloška</b>	PLA	4	0,81 g
<b>Ojačanje glavne konstrukcije</b>	PLA	2	18,21 g
<b>Zupčanik</b>	PLA	4	4,01 g
<b>Držač baterije</b>	PLA	2	6,61 g
<b>Oslonac baterije</b>	PLA	2	3,72 g
<b>Držač Raspberry Pi</b>	PLA	2	2,24 g
<b>Držač Arduino Nano</b>	PLA	1	2,24 g
<b>Držač prekidača</b>	PLA	1	2,70 g
<b>Nosač kamere</b>	PLA	1	3,10 g
<b>Stopice za vrh osovine</b>	PLA	2	1,40 g
<b>Osovina opruge</b>	Aluminij	2	40,81 g
<b>Cijev za ojačanje</b>	Aluminij	4	12,03 g
<b>Vratilo kotača</b>	Aluminij	2	4,31 g
<b>UKUPNO</b>		49	398,51 g

Mase polimernih djelova estimirana je putem alata BambuStudio prilikom printanja. Ukupna masa robota s elektronikom i baterijama izmjerit će se putem kuhinjske vage preciznosti unutar dva grama. Na slici 46 prikazan je rezultat mjerenja.

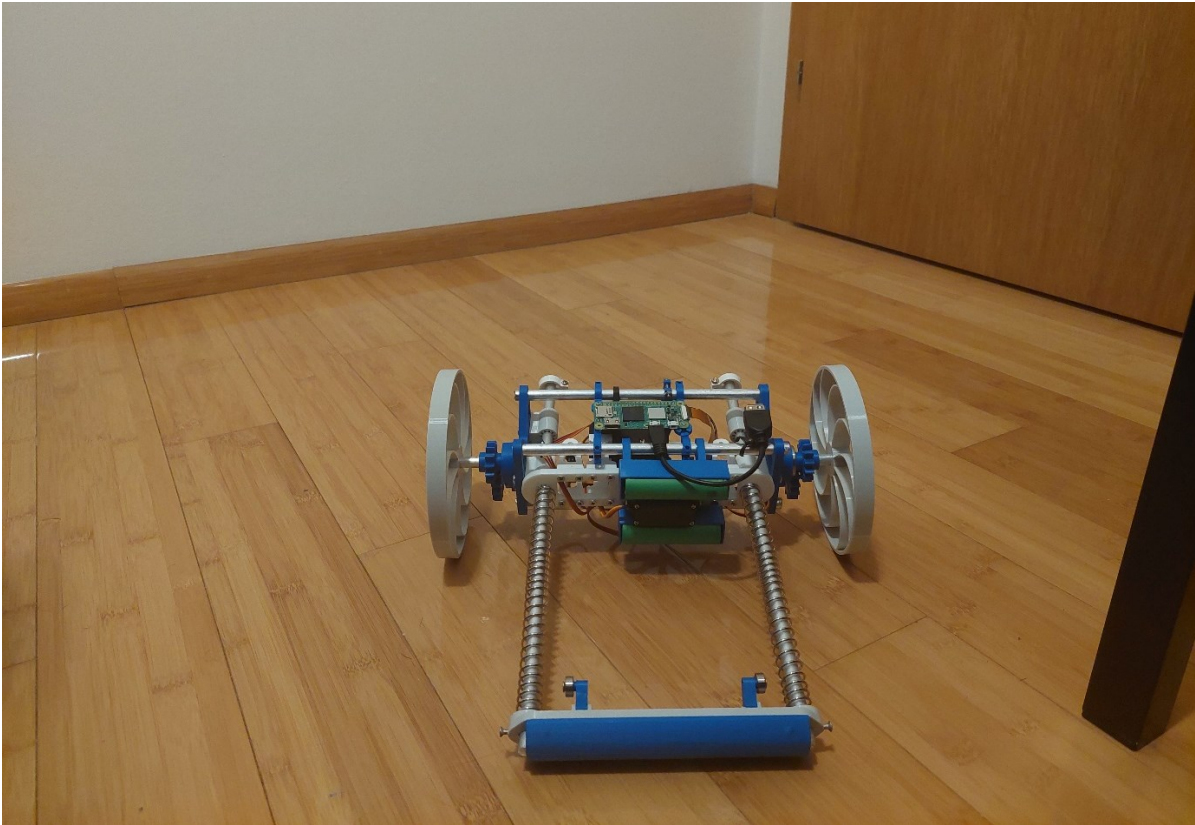


**Slika 46. Mjerenje mase robota**

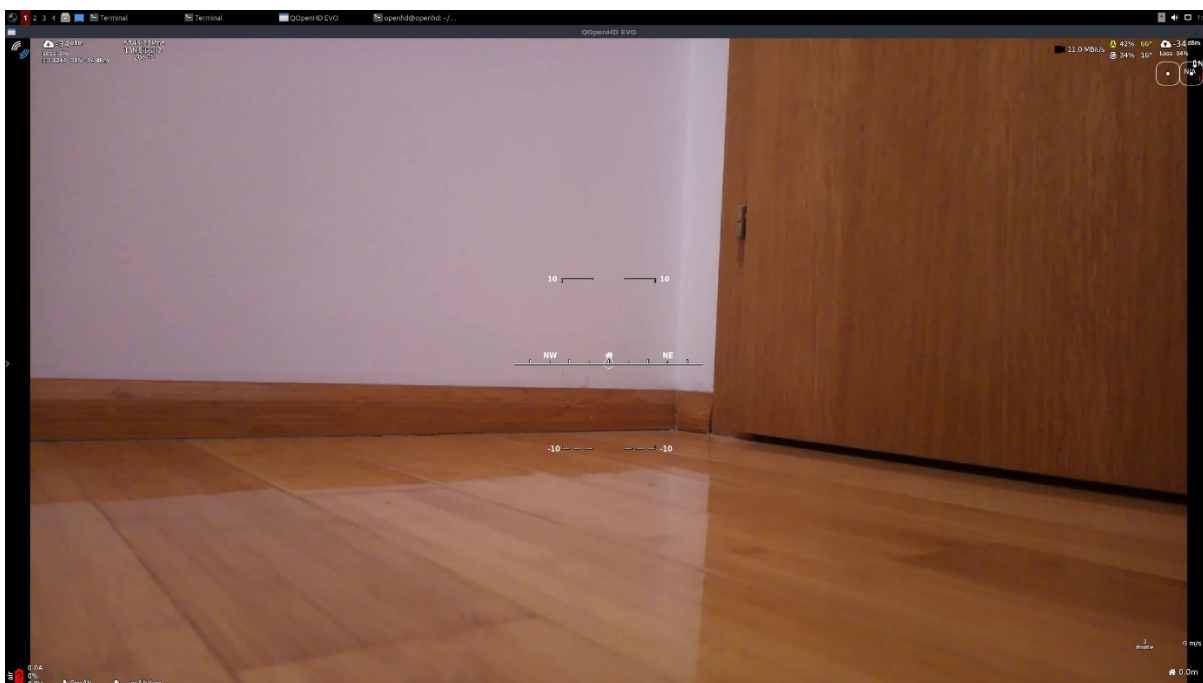
Iz ovog mjerenja možemo zaključiti da je masa baterija s elektroničkim komponentama oko 275 grama. Robot je premašio svoj početni zahtjev do 500 grama za 35%. Povećanje mase smanjuje teoretski mogući skok robota, izračunato koristeći jednadžbu (2.7), na 0,982 m.

#### **4.2. Spajanje robota na OpenHD sustav**

Spajanjem na robot putem sučelja OpenHD dobije se video transmisija iz perspektive robota na upravljačko računalo. Na slici 47 prikazan je robot i njegovo okruženje, dok je na slici 48 prikazana perspektiva robota snimljena iz sučelja za video transmisiju na upravljačkom računalu.



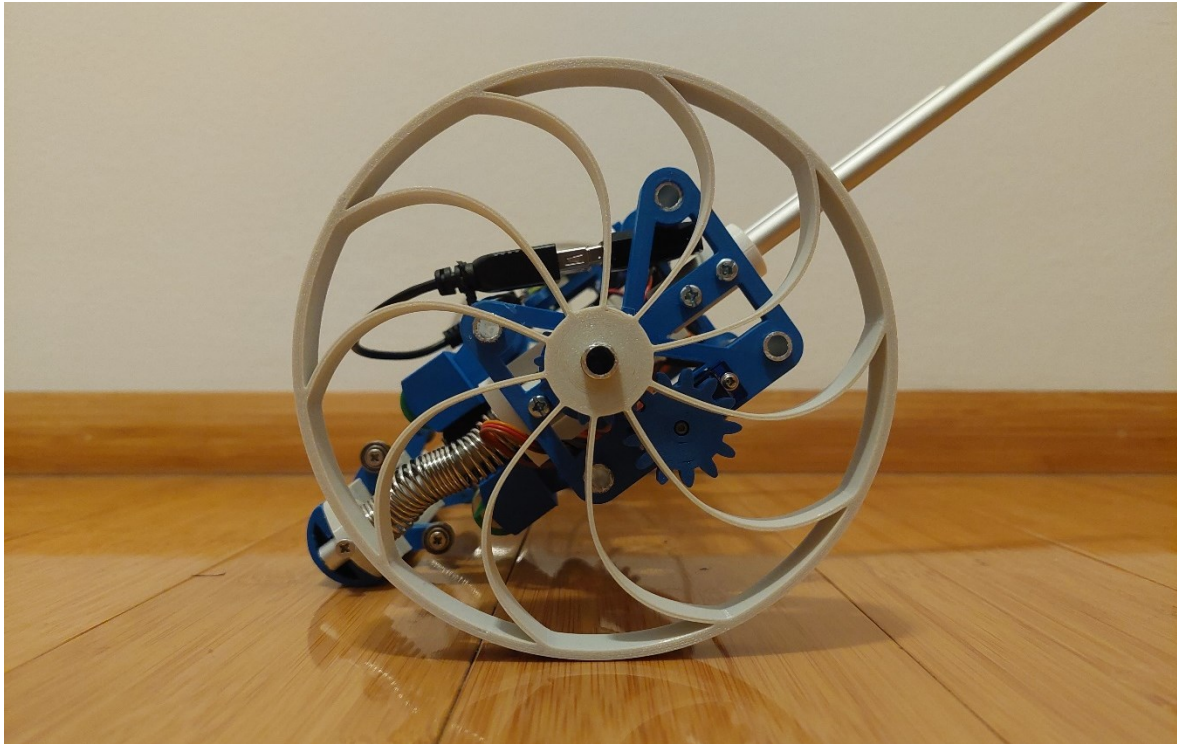
Slika 47. Slika robota i njegovog okruženja



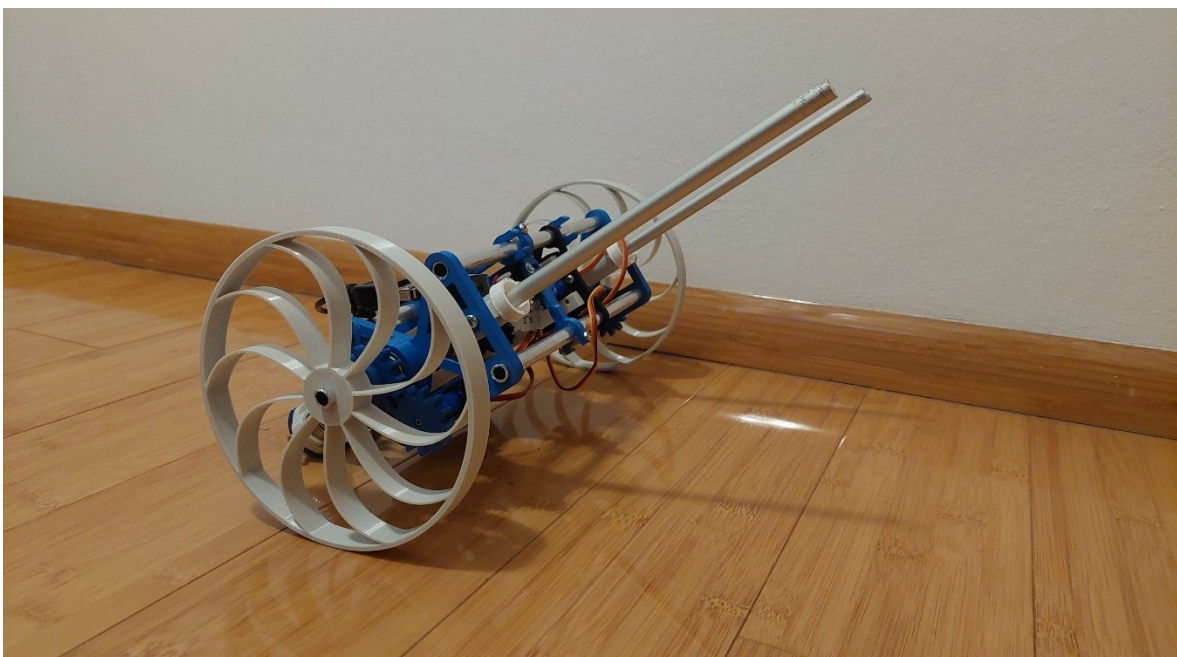
Slika 48. Snimka iz perspektive robota

### 4.3. Mehanizam skakanja

Mehanizam skakanja sa sabijenom oprugom spreman za skakanje prikazan je na slikama 49 i 50.



**Slika 49. Mehanizam skakanja sa sabijenom oprugom slikano iz profila**



**Slika 50. Mehanizam skakanja sa sabijenom oprugom**



## 5. ZAKLJUČAK

Razvoj robota skakača predstavljen u ovom radu rezultat je sveobuhvatnog procesa projektiranja i optimizacije ključnih sustava s ciljem postizanja funkcionalnog i prilagodljivog rješenja za istraživanje nepristupačnih terena. Posebna pažnja posvećena je konstrukciji mehanizma skakanja temeljenog na oprugama, čime je omogućeno učinkovito savladavanje prepreka, dok je integracija pogonskog sustava i elektronike omogućila daljinsko upravljanje. Robot skakač naglašava izradu dijelova male mase, modularnog dizajna i jednostavne konstrukcije, čime se povećava njegova prilagodljivost različitim uvjetima primjene. Elektronički sustav baziran na mikrokontroleru Arduino Nano i računalu Raspberry Pi Zero 2 W omogućava preciznu kontrolu i daljinsko upravljanje uz pomoć video prijenosa putem OpenHD sustava.

Dodatna optimizacija mase robota predstavlja važan smjer za daljnje unapređenje performansi, posebno kada je riječ o sustavima koji se oslanjaju na vertikalno skakanje. Smanjenje mase omogućilo bi postizanje većih visina skoka uz smanjenje energetske potrebe, čime bi se povećala ukupna učinkovitost robota. U tom kontekstu, daljnja istraživanja mogla bi se usmjeriti na zamjenu postojećih materijala lakšim, poput karbonskih vlakana, te na primjenu rešetkastih struktura unutar 3D ispisanih dijelova kako bi se zadržala potrebna čvrstoća uz smanjenu težinu. Također, optimizacijom baterijskog sustava s većom gustoćom energije mogao bi se smanjiti udio mase energetske komponente, što bi dodatno doprinijelo boljem omjeru mase i performansi. Takva unaprjeđenja značajno bi proširila mogućnosti primjene robota u istraživanju nepristupačnih terena i prelasku prepreka.

## LITERATURA

- [1] 'Grasshopper' robot sets high-jump record, dostupno na <https://www.newscientist.com/article/dn13951-grasshopper-robot-sets-high-jump-record/>, pristupljeno 7.7.2024
- [2] Boston Dynamics Sand Flea Robot Demonstrates Astonishing Jumping Skills, dostupno na <https://spectrum.ieee.org/boston-dynamics-sand-flea-demonstrates-astonishing-jumping-skills>, pristupljeno 7.7.2024
- [3] Zhihuai Miao, Jixue Mo, Gang Li, Yinghao Ning and Bing Li: Wheeled hopping robot with combustion-powered actuator, 2017.
- [4] Cornel Ciupan, Emilia Ciupan and Emanuela Pop: Algorithm for designing a hydraulic scissor lifting platform, 2019
- [5] OpenHD, WiFi Adapters, dostupno na <https://openhdl.gitbook.io/open-hd/hardware/wifi-adapters>, pristupljeno 16.11.2024
- [6] Pololu, FS90R Micro Continuous Rotation Servo, dostupno na <https://www.pololu.com/product/2820/specs>, pristupljeno 18.11.2024
- [7] Tower Pro, MG995, dostupno na <https://towerpro.com.tw/product/mg995/>, pristupljeno 18.11.2024
- [8] Soldered, Mikroprekidač KW12-3, dostupno na <https://soldered.com/hr/proizvod/mikroprekidac-kw12-3/>, pristupljeno 18.11.2024
- [9] Arduino, Arduino Nano, dostupno na <https://store.arduino.cc/en-hr/products/arduino-nano?srsId=AfmBOor-ZqtCxq8RB8jQmYBdBXwyqNcGS0LYgEDqNeyobMMbpxe9WON3>, pristupljeno 18.11.2024
- [10] Raspberry Pi, Raspberry Pi Zero 2 W, dostupno na <https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/>, pristupljeno 18.11.2024
- [11] Tp-link, Archer T3U, dostupno na <https://www.tp-link.com/us/home-networking/usb-adapter/archer-t3u/>, pristupljeno 18.11.2024
- [12] Soldered, Pretvarač logičkih razina, dostupno na <https://soldered.com/hr/proizvod/pretvarac-logickih-razina/>, pristupljeno 18.11.2024

## PRILOG – UPRAVLJAČKI KOD

### Arduino

```
#include <Servo.h>

// Format dodatnih komandi:
// W0 - zaustavi namatanje
// W1 - započni namatanje
// J1 - dozvoli skok
// JS - status skoka (odgovor: JR-spreman, JB-zauzet)

class RobotControl {
private:
    Servo leftServo;      // Servo za kretanje - lijevi
    Servo rightServo;    // Servo za kretanje - desni
    Servo winchServo;    // Servo za namatanje
    Servo jumpServo;     // Servo za skakanje

    const int SERVO_STOP = 90;
    const int SERVO_MAX_CW = 0;
    const int SERVO_MAX_CCW = 180;

    // Pinovi
    int leftPin;
    int rightPin;
    int winchPin;
    int jumpPin;
    int jumpSwitchPin;    // Pin za prekidač skoka

    // Stanja
    bool inverted;
    bool isWinding;      // Status namatanja
    bool jumpEnabled;    // Dozvola za skok
    bool jumpInProgress; // Skok u tijeku

    int mapMotorSpeed(int speed, bool reverse) {
        int mappedSpeed = map(speed, 0, 255, 0, 90);
        return reverse ? SERVO_STOP - mappedSpeed : SERVO_STOP + mappedSpeed;
    }

public:
    RobotControl(int leftServoPin, int rightServoPin, int winchServoPin,
                 int jumpServoPin, int switchPin) {
        leftPin = leftServoPin;
        rightPin = rightServoPin;
        winchPin = winchServoPin;
    }
};
```

```
    jumpPin = jumpServoPin;
    jumpSwitchPin = switchPin;

    inverted = false;
    isWinding = false;
    jumpEnabled = false;
    jumpInProgress = false;
}

void begin() {
    leftServo.attach(leftPin);
    rightServo.attach(rightPin);
    winchServo.attach(winchPin);
    jumpServo.attach(jumpPin);

    pinMode(jumpSwitchPin, INPUT_PULLUP);

    stop();
    stopWinch();
    stopJump();
}

// Osnovne funkcije za kretanje (nepromijenjene)
void setInverted(bool invert) { inverted = invert; }

void forward(int speed) {
    if (inverted) { backward(speed); return; }
    leftServo.write(mapMotorSpeed(speed, false));
    rightServo.write(mapMotorSpeed(speed, true));
}

void backward(int speed) {
    if (inverted) { forward(speed); return; }
    leftServo.write(mapMotorSpeed(speed, true));
    rightServo.write(mapMotorSpeed(speed, false));
}

void rotateLeft(int speed) {
    if (inverted) { rotateRight(speed); return; }
    leftServo.write(mapMotorSpeed(speed, true));
    rightServo.write(mapMotorSpeed(speed, true));
}

void rotateRight(int speed) {
    if (inverted) { rotateLeft(speed); return; }
    leftServo.write(mapMotorSpeed(speed, false));
    rightServo.write(mapMotorSpeed(speed, false));
}
```

```
}

void stop() {
    leftServo.write(SERVO_STOP);
    rightServo.write(SERVO_STOP);
}

// Funkcije za namatanje
void startWinch() {
    isWinding = true;
    winchServo.write(SERVO_MAX_CCW); // Pokretanje u željenom smjeru
}

void stopWinch() {
    isWinding = false;
    winchServo.write(SERVO_STOP);
}

bool isWinchActive() {
    return isWinding;
}

// Funkcije za skakanje
void enableJump() {
    jumpEnabled = true;
    if (!jumpInProgress && digitalRead(jumpSwitchPin) == HIGH) {
        startJump();
    }
}

void startJump() {
    jumpInProgress = true;
    jumpServo.write(SERVO_MAX_CW); // Pokretanje u željenom smjeru
}

void stopJump() {
    jumpInProgress = false;
    jumpEnabled = false;
    jumpServo.write(SERVO_STOP);
}

void updateJump() {
    if (jumpInProgress) {
        if (digitalRead(jumpSwitchPin) == LOW) { // Prekidač aktiviran
            jumpServo.write(SERVO_STOP);
            jumpInProgress = false;
            jumpEnabled = false;
        }
    }
}
```

```
    }
  }
}

String getJumpStatus() {
  if (jumpInProgress) return "JB"; // Busy
  return "JR"; // Ready
}
};

const int LEFT_SERVO_PIN = 9;
const int RIGHT_SERVO_PIN = 10;
const int WINCH_SERVO_PIN = 11;
const int JUMP_SERVO_PIN = 12;
const int JUMP_SWITCH_PIN = 2; // Prekidač na digitalnom pinu 2
const long BAUD_RATE = 9600;

RobotControl robot(LEFT_SERVO_PIN, RIGHT_SERVO_PIN, WINCH_SERVO_PIN,
                  JUMP_SERVO_PIN, JUMP_SWITCH_PIN);
String inputBuffer = "";

void setup() {
  Serial.begin(BAUD_RATE);
  robot.begin();
}

void processCommand(char command, String value) {
  // Postojeće komande za kretanje
  if (command == 'F' || command == 'B' || command == 'L' ||
      command == 'R' || command == 'S' || command == 'I') {
    processCommand(command, value.toInt());
    return;
  }

  // Nove komande
  switch (command) {
    case 'W': // Winch control
      if (value == "1") robot.startWinch();
      else if (value == "0") robot.stopWinch();
      break;

    case 'J': // Jump control
      if (value == "1") robot.enableJump();
      else if (value == "S") {
        Serial.println(robot.getJumpStatus());
      }
      break;
  }
}
```

```

    }
}

void loop() {
    // Ažuriraj status skoka
    robot.updateJump();

    // Čitaj komande
    while (Serial.available() > 0) {
        char inChar = Serial.read();

        if (inChar == '\n') {
            if (inputBuffer.length() >= 2) {
                char command = inputBuffer.charAt(0);
                String value = inputBuffer.substring(1);
                processCommand(command, value);
            }
            inputBuffer = "";
        } else {
            inputBuffer += inChar;
        }
    }
}
}

```

## Raspberry Pi

```

import serial
import time
import smbus
import math
from threading import Thread

class MPU6050:
    def __init__(self, bus=1, address=0x68):
        self.bus = smbus.SMBus(bus)
        self.address = address

        # Konfiguracija MPU6050
        self.bus.write_byte_data(self.address, 0x6B, 0) # Probudi MPU6050

        # Konfiguracija filtera
        self.ALPHA = 0.96 # Komplementarni filter konstanta

        # Skaliranje za pretvorbu sirovih podataka
        self.ACCEL_SCALE = 16384.0 # Za ±2g
        self.GYRO_SCALE = 131.0 # Za ±250°/s

        # Početni kutovi
        self.angle_x = 0
        self.angle_y = 0

```

```

self.last_time = time.time()

# Offset vrijednosti (treba kalibrirati)
self.gyro_offset_x = 0
self.gyro_offset_y = 0
self.gyro_offset_z = 0

# Kalibriraj žiroskop
self.calibrate_gyro()

def read_word(self, register):
    """Čita 16-bitnu vrijednost iz registra"""
    high = self.bus.read_byte_data(self.address, register)
    low = self.bus.read_byte_data(self.address, register + 1)
    value = (high << 8) + low

    # Konverzija u signed int
    if value >= 0x8000:
        value = -((65535 - value) + 1)
    return value

def calibrate_gyro(self, samples=1000):
    """Kalibrira žiroskop računanjem offset vrijednosti"""
    print("Kalibriranje žiroskopa...")
    sum_x = sum_y = sum_z = 0

    for _ in range(samples):
        sum_x += self.read_word(0x43)
        sum_y += self.read_word(0x45)
        sum_z += self.read_word(0x47)
        time.sleep(0.004)

    self.gyro_offset_x = sum_x / samples
    self.gyro_offset_y = sum_y / samples
    self.gyro_offset_z = sum_z / samples
    print("Kalibracija završena!")

def get_angles(self):
    """Računa kutove nagiba koristeći akcelerometar i žiroskop"""
    # Čitanje akcelerometra
    acc_x = self.read_word(0x3B)
    acc_y = self.read_word(0x3D)
    acc_z = self.read_word(0x3F)

    # Čitanje žiroskopa
    gyro_x = self.read_word(0x43) - self.gyro_offset_x
    gyro_y = self.read_word(0x45) - self.gyro_offset_y

    # Računanje kutova iz akcelerometra
    acc_angle_x = math.atan2(acc_y, math.sqrt(acc_x**2 + acc_z**2)) *
180 / math.pi
    acc_angle_y = math.atan2(-acc_x, math.sqrt(acc_y**2 + acc_z**2)) *
180 / math.pi

    # Vrijeme od zadnjeg čitanja
    current_time = time.time()
    dt = current_time - self.last_time
    self.last_time = current_time

```



```

    # Računanje kutova iz žiroskopa
    gyro_angle_x = gyro_x / self.GYRO_SCALE * dt
    gyro_angle_y = gyro_y / self.GYRO_SCALE * dt

    # Komplementarni filter
    self.angle_x = self.ALPHA * (self.angle_x + gyro_angle_x) + (1 -
self.ALPHA) * acc_angle_x
    self.angle_y = self.ALPHA * (self.angle_y + gyro_angle_y) + (1 -
self.ALPHA) * acc_angle_y

    return self.angle_x, self.angle_y

class RobotController:
    def __init__(self, serial_port='/dev/ttyUSB0', baud_rate=9600):
        self.serial = serial.Serial(serial_port, baud_rate, timeout=1)
        self.mpu = MPU6050()
        self.running = True

        # Pokretanje thread-a za praćenje kuta
        self.angle_thread = Thread(target=self._angle_monitoring)
        self.angle_thread.daemon = True
        self.angle_thread.start()

        time.sleep(2) # Čekaj da se Arduino resetira

    def _angle_monitoring(self):
        """Thread funkcija za kontinuirano praćenje kuta"""
        while self.running:
            self.current_angle_x, self.current_angle_y =
self.mpu.get_angles()
            time.sleep(0.01) # 100Hz očitavanje

    def get_current_angles(self):
        """Vraća trenutne kutove nagiba"""
        return self.current_angle_x, self.current_angle_y

    def _send_command(self, command, value):
        """Šalje komandu u formatu CMD+VALUE\n"""
        message = f"{command}{value}\n"
        self.serial.write(message.encode())

    # Osnovne funkcije za kretanje
    def forward(self, speed):
        self._send_command('F', min(255, max(0, speed)))

    def backward(self, speed):
        self._send_command('B', min(255, max(0, speed)))

    def rotate_left(self, speed):
        self._send_command('L', min(255, max(0, speed)))

    def rotate_right(self, speed):
        self._send_command('R', min(255, max(0, speed)))

    def stop(self):
        self._send_command('S', 0)

    def set_inverted(self, inverted):
        self._send_command('I', 1 if inverted else 0)

```

```
# Funkcije za namatanje
def start_winch(self):
    self._send_command('W', 1)

def stop_winch(self):
    self._send_command('W', 0)

# Funkcije za skakanje
def enable_jump(self):
    self._send_command('J', 1)

def get_jump_status(self):
    self._send_command('J', 'S')
    response = self.serial.readline().decode().strip()
    return response

def is_tilted(self, threshold=30):
    """Provjerava je li robot previše nagnut"""
    x, y = self.get_current_angles()
    return abs(x) > threshold or abs(y) > threshold

def close(self):
    """Zatvara sve veze i zaustavlja monitoring"""
    self.running = False
    self.angle_thread.join()
    self.serial.close()
```