

ROBOTSKA ROKA

RAZISKOVALNA NALOGA
ROBOTSKA ROKA

STROJNIŠTVO

LUKA URANIČ TM/4.a

SREDNJA ŠOLA TEHNIŠKIH STROK ŠIŠKA
Mentor: dr. Rok Juhant

2020

KAZALO VSEBINE

Kazalo slik	4
POVZETEK	5
UVOD	6
HIPOTEZE	6
TEORETIČNI DEL	7
KRMILNIK ARDUINO MEGA 2560	7
KORAČNI MOTOR	8
UNIPOLARNI KORAČNI MOTOR	8
BIPOLARNI KORAČNI MOTOR	9
SISTEM KORAČNIH MOTORJEV	9
NAPAJALNIK	10
KINEMATIKA ROBOTA	10
DIREKTNA KINEMATIKA	11
INVERZNA KINEMATIKA	12
EKSPERIMENTALNI DEL	13
KONSTRUIRANJE ROBOTSKE ROKE	13
3D TISKANJE ROBOTA	14
POSTOPEK IZDELAVE OD KONSTRUKCIJE DO IZDELKA	15
POVEZAVA KORAČNIH MOTORJEV Z GONILNIKI IN KRMILNIKOM	18
IZDELAVA SIMULACIJSKEGA PROGRAMA	20
IZDELAVA PROGRAMA ZA TESTIRANJE ROBOTSKE ROKE	22
REZULTATI	25
RAZPRAVA	26
ZAKLJUČEK	27
LITERATURA	28

KAZALO SLIK

Slika 1: izdelek	5
Slika 2: Arduino Mega 2560	7
Slika 3: primer koračnega motorja	8
Slika 4: primerjava unipolarnega in bipolarnega koračnega motorja.....	9
Slika 5: sistem koračnih motorjev	10
Slika 6: napajalnik	10
Slika 7: kinematika robota.....	11
Slika 8: direktna kinematika.....	12
Slika 9: inverzna kinematika	12
Slika 10: konstrukcija robotske roke 1	13
Slika 11: konstrukcija robotske roke 2	13
Slika 12: konstrukcija robotske roke 3	14
Slika 13: konstrukcija robotske roke 4	14
Slika 14: konstrukcija izdelka v programu za konstruiranje	15
Slika 15: prenos konstruiranega izdelka v program za razslojevanje	16
Slika 16: postavitev izdelka v programu za razslojevanje	16
Slika 17: razslojevanje	17
Slika 18: prenos g-kode na sd kartico	17
Slika 19: natisnjen izdelek.....	18
Slika 20: povezava koračnih motorjev z gonilniki in krmilnikom	18
Slika 21: povezava koračnega motorja z gonilnikom in krmilnikom	19
Slika 22: simulacijski program 1	20
Slika 23: simulacijski program 2.....	21
Slika 24: simulacijski program 3.....	21
Slika 25: testni program 1	22
Slika 26: testni program 2	23
Slika 27: testni program 3	24
Slika 28: končni izdelek	25

POVZETEK

V raziskovalni nalogi je predstavljena izdelava robotske roke z šestimi osmi. Opisan je celoten postopek sestave in izdelave robotske roke, vključno z postopkom konstruiranja robotske roke, 3D-tiskanja ogrodja, povezavo motorjev, krmilnika, gonilnikov za motorje in napajalnika. Prav tako pa je opisano delovanje kinematike robotske roke, simulacijski program, ki simulira premikanje robotske roke ter testni program, ki premika vse motorje v eno smer za določeno število korakov in nato spremeni smer.



Slika 1: izdelek

UVOD

Ker me avtomatizacija in robotizacija zalo zanimata, sem se odločil izdelati svojo robotsko roko. Zadal sem si cilj, da robotsko roko skonstruiram tako, da bo sposobna premagovati svojo težo, 3D natisnem ogrodje, sestavim in povežem vse dele med seboj ter jo sprogramiram tako, da se je sposobna gibati v želenem tri-dimenzionalnem prostoru.

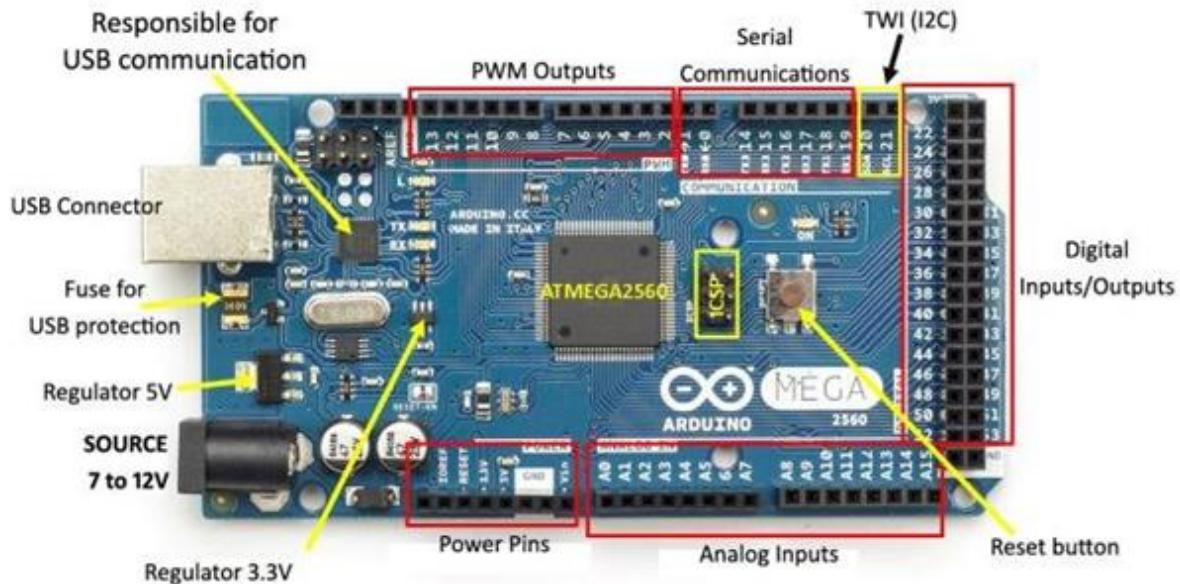
HIPOTEZE:

1. Krmilnik Arduino Mega 2560 je sposoben krmiliti robotsko roko s šestimi osmi.
2. Robotska roka iz PLA plastike je sposobna premagovati svojo težo in se gibati v svojem delovnem prostoru.

TEORETIČNI DEL

Robotska roka je sestav mehanskih sklepov, ki se gibljejo in pozicionirajo s pomočjo koračnih motorjev ter simulirajo gibanje človeške roke. Robotska roka se giblje tako, kot ji naroča program, ki ga je možno spremiščati (reprogramirati).

KRMILNIK ARDUINO MEGA 2560



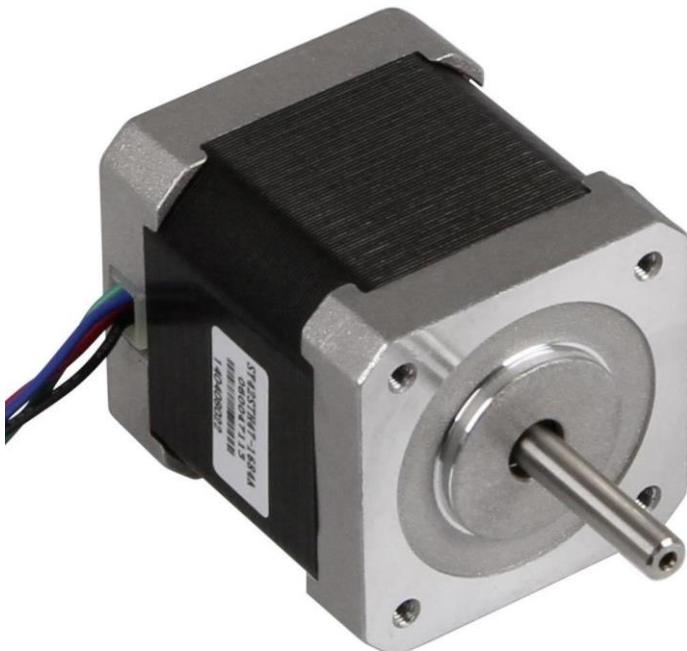
Slika 2: Arduino Mega 2560

Arduino Mega 2560 je mikro krmilnik, ki je snovan na ATmega2560. Ima 54 digitalnih vhodov / izhodov (od teh jih 14 lahko uporabimo kot PWM izhode), 16 analognih vhodov, 4 UART (strojni serijski porti), 16 MHz kristalni oscilator, USB povezavo, priključek za napajanje, gumb za ponastavljanje in ISCP pine (MISO, MOSI, SCK, VCC, GND, RESET).

Arduino Mega 2560 lahko programiramo z programskega jezikom Arduino, ki temelji na programskem jeziku C, za pomoč pri programiranju lahko uporabljam knjižnice, ki ti omogočijo lažje upravljanje senzorjev in aktuatorjev.

KORAČNI MOTOR

Koračni motor, je brezkrtačni enosmerni elektromotor, ki celotno rotacijo razdeli na več enakih delov (korakov). Položaj motorja lahko nadzorujemo brez dodatnega senzorja.



Slika 3: primer koračnega motorja

Ameriško nacionalno združenje proizvajalce električne energije (NEMA) standardizira različne vidike koračnih motorjev. Običajno jih imenujemo z NEMA DD, kjer je DD premer čelne plošče v palcih pomnožen z 10 (npr. NEMA 17 ima premer 1,7 palca). Za opis koračnih motorjev so na voljo še dodatki, ki jih lahko najdemo v standardu ICS 16-2001 (poglavlje 4.3.1.1.).

Obstajata dve osnovni ureditvi navijanja za elektromagnetne tuljave v dvofaznem koračnem motorju: bipolarni in unipolarni.

UNIPOLARNI KORAČNI MOTOR

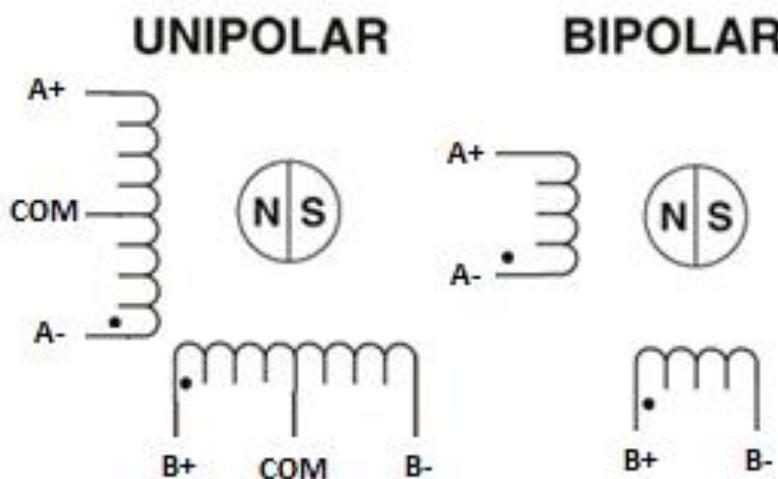
Unipolarni koračni motor ima eno navitje s sredinsko pipo na fazo. Vsak odsek navitij je vklopljen za vsako smer magnetnega polja. Ker je v tej postavitvi mogoče obrniti magnetni pol brez preklopa smeri toka, je komutacijski tokokrog lahko zelo preprost za vsako navijanje. Običajno je glede na fazo sredinska pipa vsakega navitja pogosta: podajanje treh vodov na fazo in šest odvodov za značilni dvofazni motor. Pogosto sta ti dve fazni komori notranje povezani, tako da ima motor le pet odvodov.

BIPOLARNI KORAČNI MOTOR

Bipolarni motorji imajo eno fazo navitja. Tok v navitju je treba obrniti, da lahko obrnemo magnetni pol, zato mora biti pogonski krog bolj zapleten.

Tipičen vzorec vožnje za dvokolesni bipolarni koračni motor je: A+ B+ A- B-.

- V tuljavo A spustimo pozitivni tok, nato odstranimo tok iz tuljave A.
- Napajamo tuljavo B s pozitivnim tokom, nato odstranimo tok iz tuljave B.
- Tuljavo A napajamo z negativnim tokom, nato odstranimo napajanje na tuljavi A.
- Napajamo tuljavo B z negativnim tokom, nato odstranimo tok iz tuljave B.

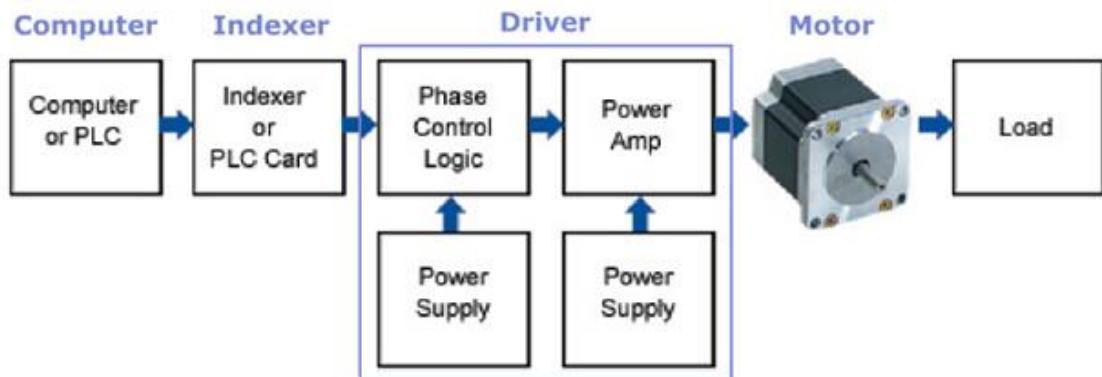


Slika 4: primerjava unipolarnega in bipolarnega koračnega motorja

SISTEM KORAČNIH MOTORJEV

Sistem koračnega motorja je sestavljen iz treh osnovnih elementov, ki so pogosto kombinirani z nekaterim uporabniškim vmesnikom (gostiteljski računalnik, PLC).

- Krmilnik – je mikroprocesor, ki lahko generira koračne impulze in smerne signale za gonilnik.
- Gonilnik – pretvori ukazne signale krmilnika v moč, ki je potrebna za napajanje navitij motorja. Obstajajo številne vrste gonilnikov z različnimi napetostmi in tokovnimi nazivi ter gradbeno tehnologijo. Niso vsi gonilniki primerni za pogonjanje vseh motorjev, zato pri načrtovanju sistema za nadzor gibanja odločilni postopek izbere gonilnikov.
- Koračni motor – je elektromagnetna naprava, ki pretvori digitalne impulze v mehansko vrtenje gredi. Prednosti koračnih motorjev so nizki stroški, visoka zanesljivost, velik navor pri nizkih hitrostih in preprosta robustna konstrukcija, ki deluje v skoraj katerem koli okolju. Glavna pomanjkljivost uporabe koračnega motorja je resonančni učinek, ki se pogosto kaže pri nizkih hitrostih in zmanjšuje navor z naraščajočo hitrostjo.



Slika 5: sistem koračnih motorjev

NAPAJALNIK

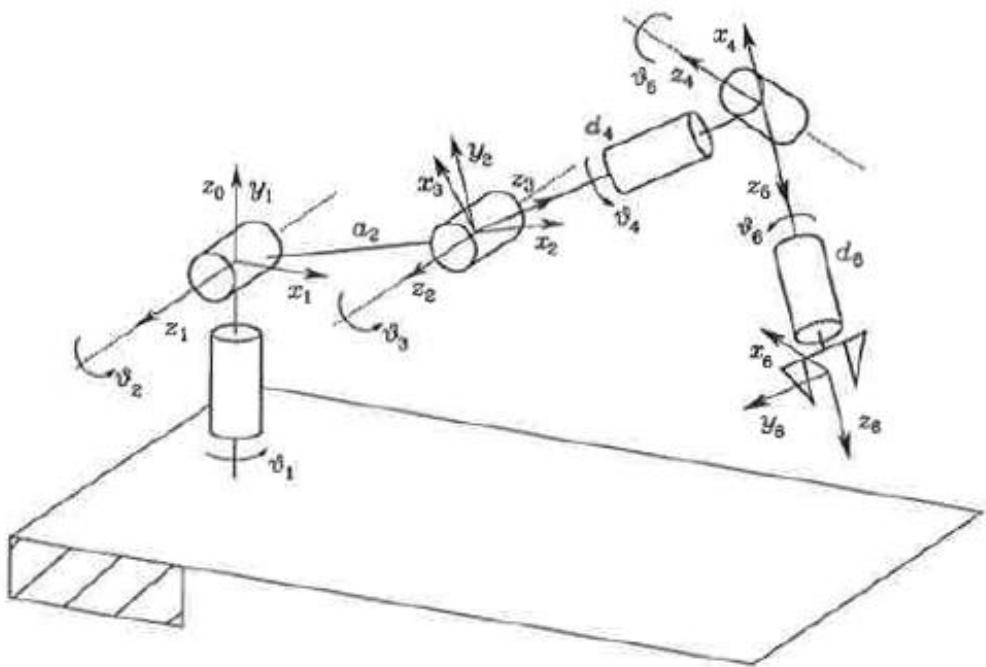
Za napajanje koračnih motorjev je potreben velik tok, zato bom robotsko roko napajal z 250 W 24V 10A napajalnikom. Napajalnik dobi napajanje iz navadne vtičnice 230 V in napetost pretvori na 24 V.



Slika 6: napajalnik

KINEMATIKA ROBOTA

Kinematika robotov uporablja geometrijo za preučevanje gibanja večosnih verig, ki tvorijo strukturo robotskih sistemov. Poudarek na geometriji pomeni, da so povezave robota modelirane kot toga telesa in za sklepe se predvideva, da zagotavljajo čisto vrtenje. Kinematika robotov proučuje razmerje med dimenzijami in povezljivostjo kinematičnih verig ter položajem, hitrostjo in pospeševanjem vsake izmed povezav v robotskem sistemu, da bi načrtovali in nadzirali gibanje ter izračunali sile in navore aktuatorje.



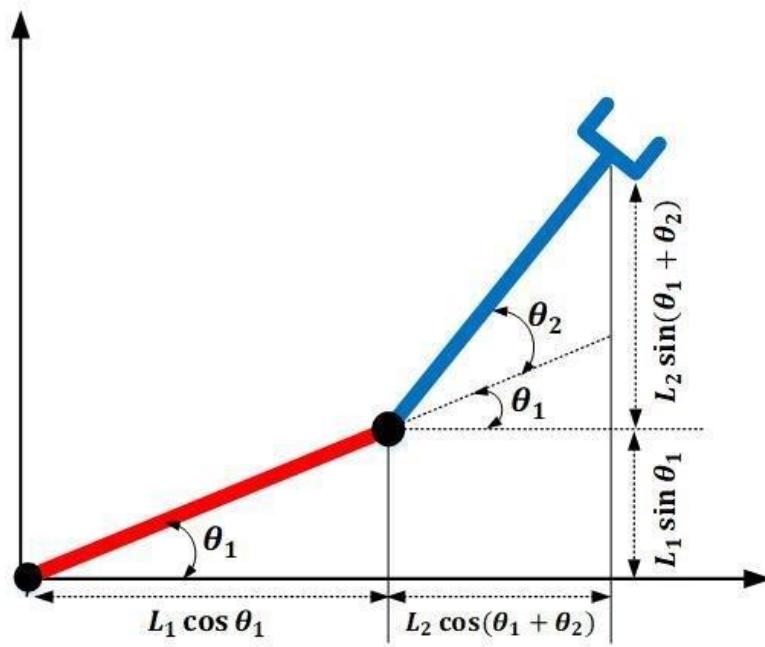
Anthropomorphic arm with spherical wrist

Slika 7: kinematika robota

DIREKTNA KINEMATIKA

Direktna kinematika se nanaša na uporabo kinematičnih enačb robota za izračun položaja končne točke robotske roke. Enačbe kinematike robota se uporabljajo v robotiki, računalniški igrah in animacijah. Povratni postopek, ki izračuna parametre sklepa, ko končna točka robota doseže določeno točko se imenuje inverzna kinematika.

Kinematične enačbe za serijsko verigo robota dobimo s togo transformacijo, s katero določimo relativno gibanje, ki je dovoljeno pri vsakem sklepu, in ločeno togo transformacijo za določitev dimenzijske povezave. Rezultat je zaporedje togih preobrazb, ki izmenično povezujejo in pretvarjajo iz osnovne verige v njeno končno vez, ki je enačena z določenim položajem za končno vez.

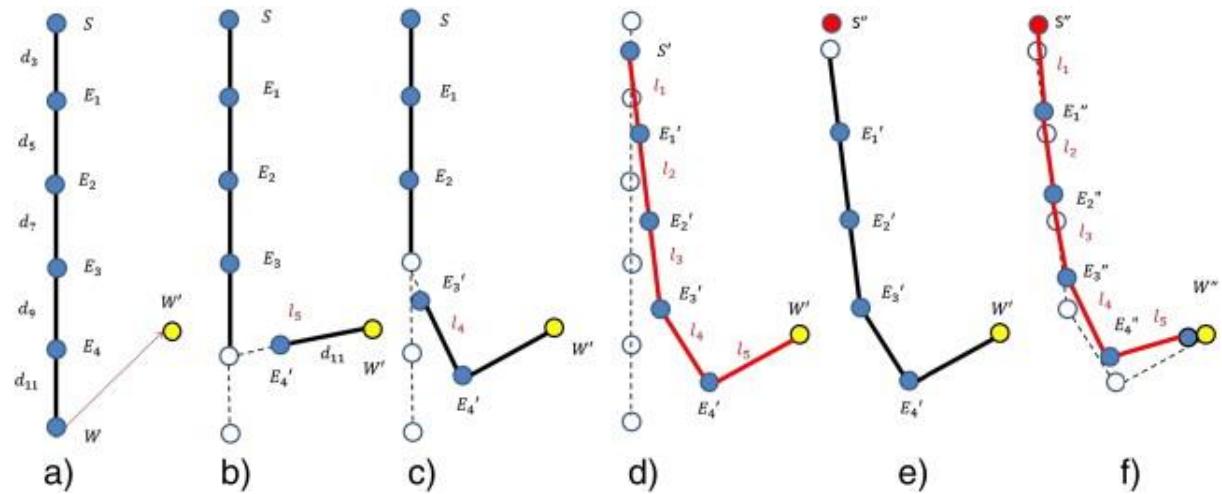


Slika 8: direktna kinematika

INVERZNA KINEMATIKA

Inverzna kinematika je matematični postopek obnavljanja premikov predmeta v svetu iz nekaterih drugih podatkov.

Inverzna kinematika v robotiki uporablja enačbe kinematike za določitev skupnih parametrov, ki zagotavljajo želeni položaj za vsakega od parametrov končne točke robota. Specifikacija gibanja robota, tako da njegovi učinki dosežejo želene naloge, je znana kot načrtovanje gibanja.



Slika 9: inverzna kinematika

EKSPERIMENTALNI DEL

V eksperimentalnem delu vam bom predstavil, kako sem roko izdelal, vse on konstruiranja, 3d tiskanja, sestavljanja komponent, povezovanja motorjev z gonilniki in krmilnikom, izdelava simulacijskega programa na osebnem računalniku in izdelava programa za testiranje robotske roke.

KONSTRUIRANJE ROBOTSKE ROKE

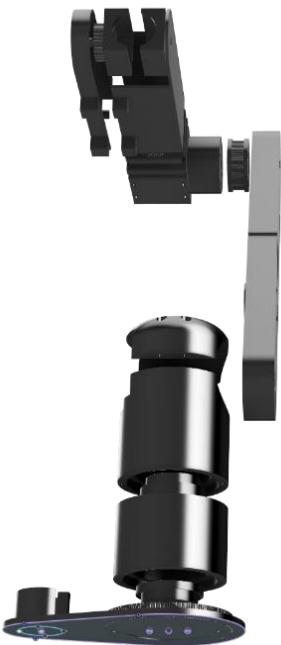
Robotsko roko sem konstruiral v programu Autodesk Fusion 360. Roko sem poizkusil skonstruirati čim bolj efektivno glede na motorje, ki sem jih kupil. Roka ima 6 rotacijskih osi ozziroma sklepov. Trije sklepi so vertikalni – to so sklepi 1, 4, 6 in trije horizontalni – to so sklepi 2, 3, 5. Konstruiranje je omejevalo velikost 3D tiskalnika in sam postopek 3D tiskanja.



Slika 10: konstrukcija robotske roke 1



Slika 11: konstrukcija robotske roke 2



Slika 12: konstrukcija robotske roke 3



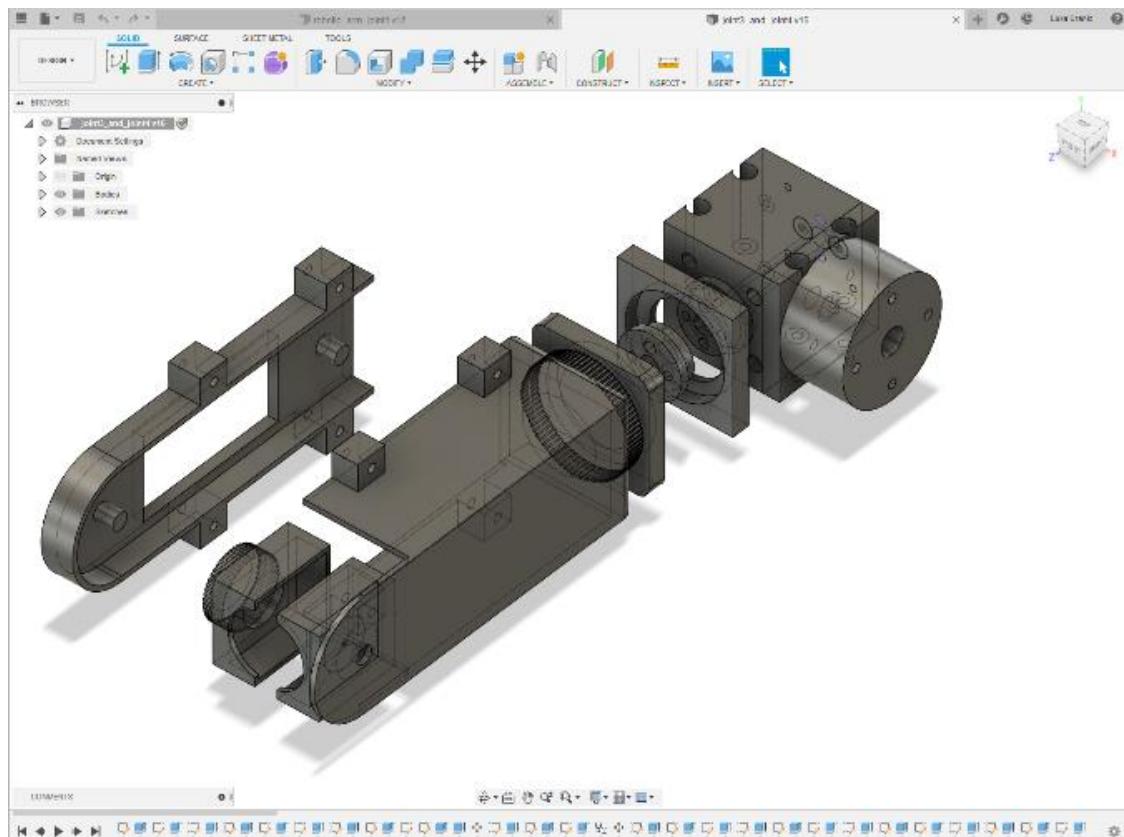
Slika 13: konstrukcija robotske roke 4

3D TISKANJE ROBOTA

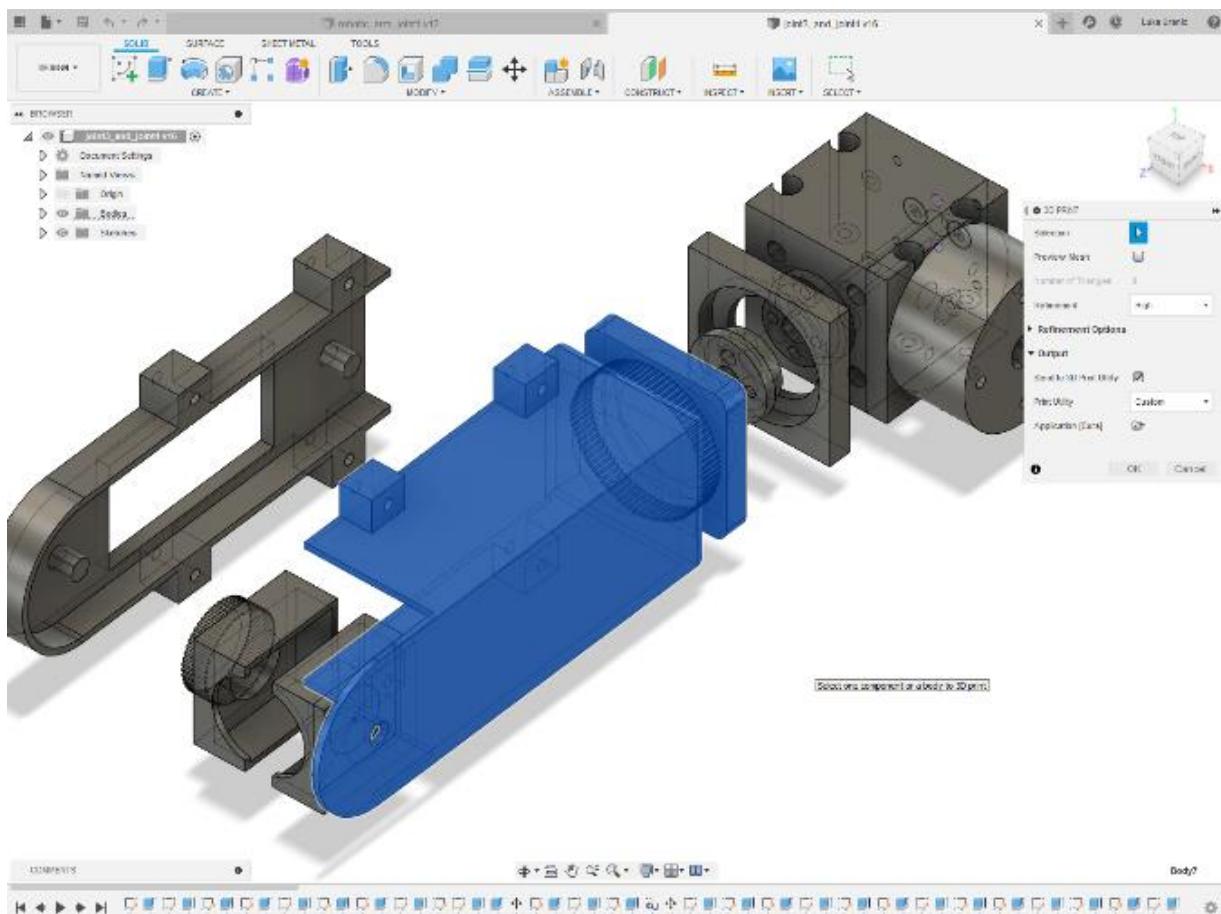
Roko sem 3D tiskal na 3D tiskalniku (Creality Ender 3) s plastiko PLA. Velikost tiskalnega prostora na tiskalniku je $235 \times 235 \times 250 \text{ mm}^3$, zato so vsi kosi morali biti prilagojeni tej velikosti. 3D tiskanje je zelo počasen postopek, hitrost tiskanja je približno 50 mm/s.

POSTOPEK IZDELAVE OD KONSTRUKCIJE DO IZDELKA

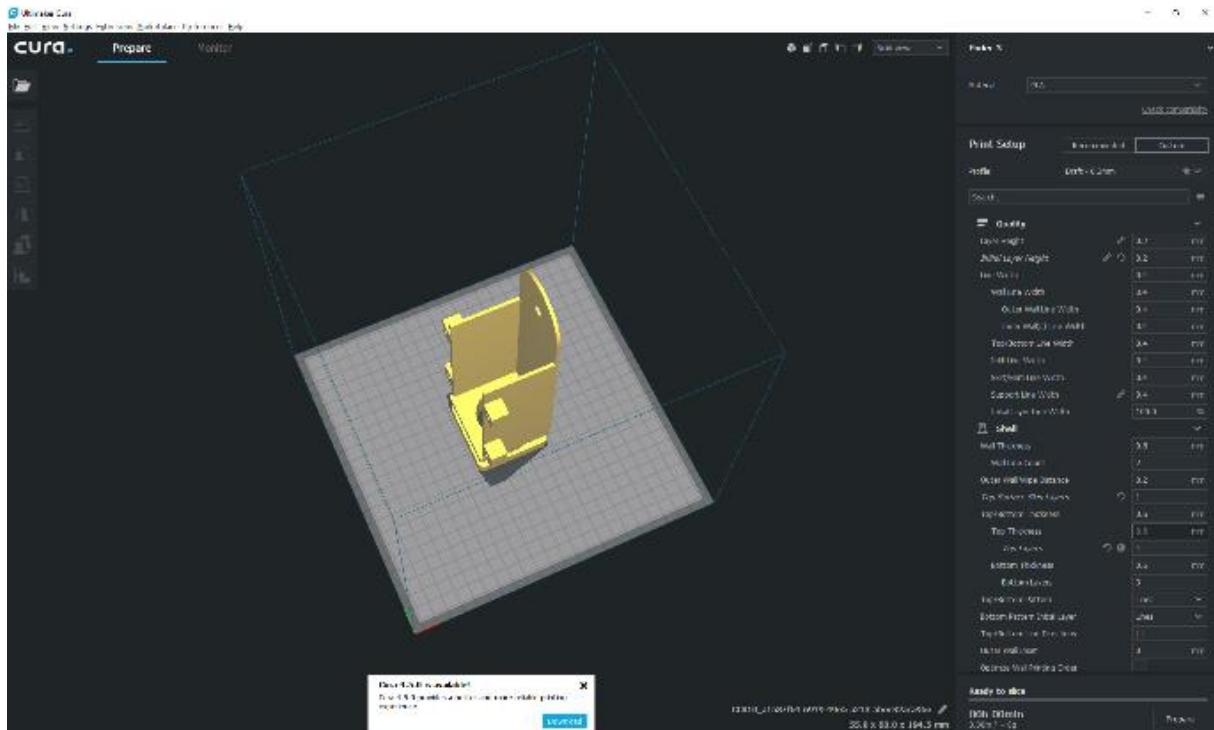
1. Konstrukcija izdelka v programu za konstruiranje (Autodesk Fusion 360).
2. Prenos konstruiranega izdelka v program za razslojevanje (Ultimaker Cura).
3. Postavitev in poravnava izdelka v programu.
4. Postopek razslojevanja: pri tem program izdelek razdeli na sloje velikost 0.1-0.3mm in za vsak sloj izračuna g-kodo, ki jo lahko spremenjamo ročno ali s spremenjanjem določenih parametrov (hitrost, odmik, temperatura, postavitev podpor, nastavitev lastnosti podpor, gostota polnila itd.).
5. Prenos g-kode na mini SD kartico.
6. Vstavitev kartice v tiskalnik in izbira programa tiskanja.
7. Priprava tiskalne površine.
8. Tiskanje izdelka.
9. Odstranitev izdelka s tiskalnika



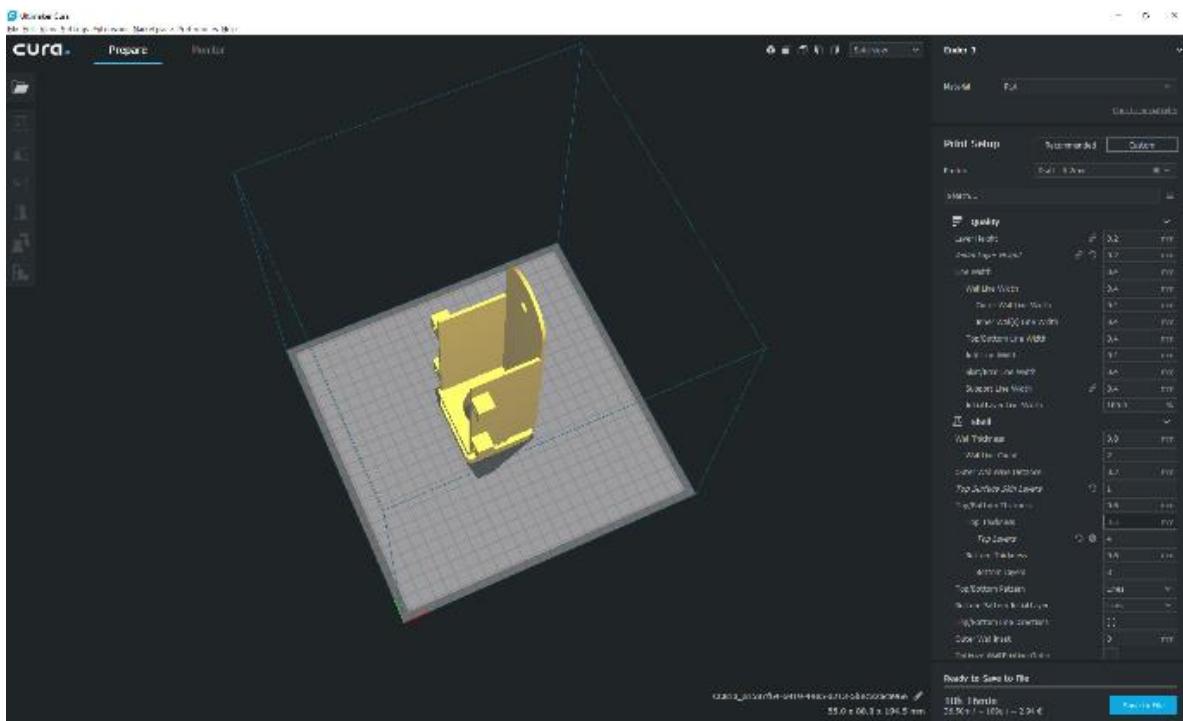
Slika 14: konstrukcija izdelka v programu za konstruiranje



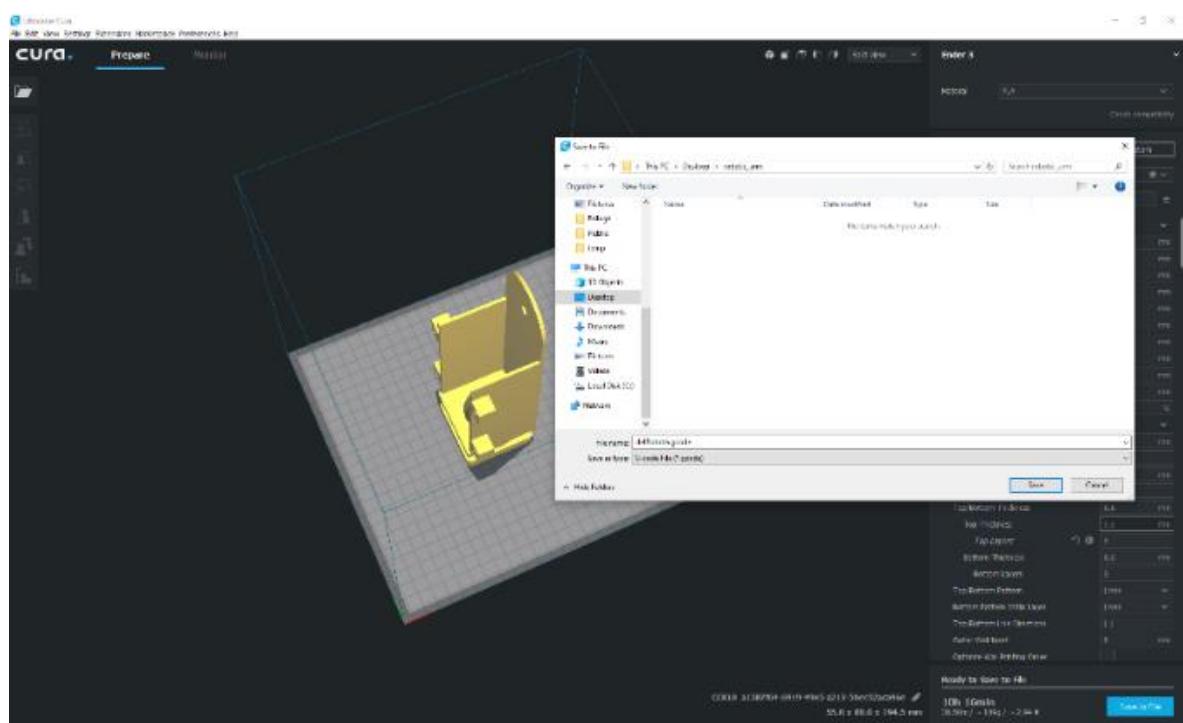
Slika 15: prenos konstruiranega izdelka v program za razslojevanje



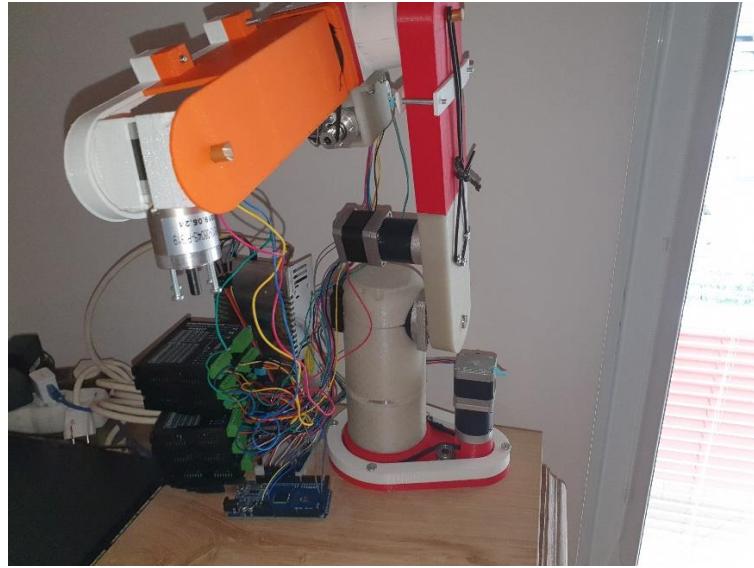
Slika 16: postavitev izdelka v programu za razslojevanje



Slika 17: razslojevanje



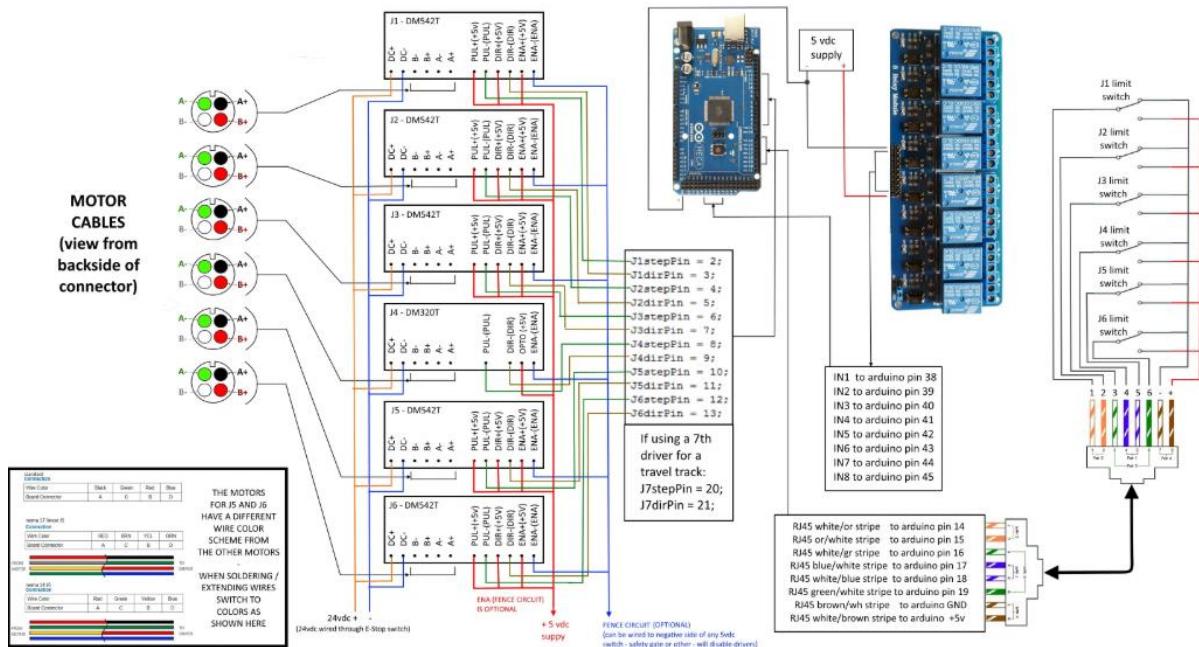
Slika 18: prenos g-kode na sd kartico



Slika 19: natisnjen izdelek

POVEZAVA KORAČNIH MOTORJEV Z GONILNIKI IN KRMILNIKOM

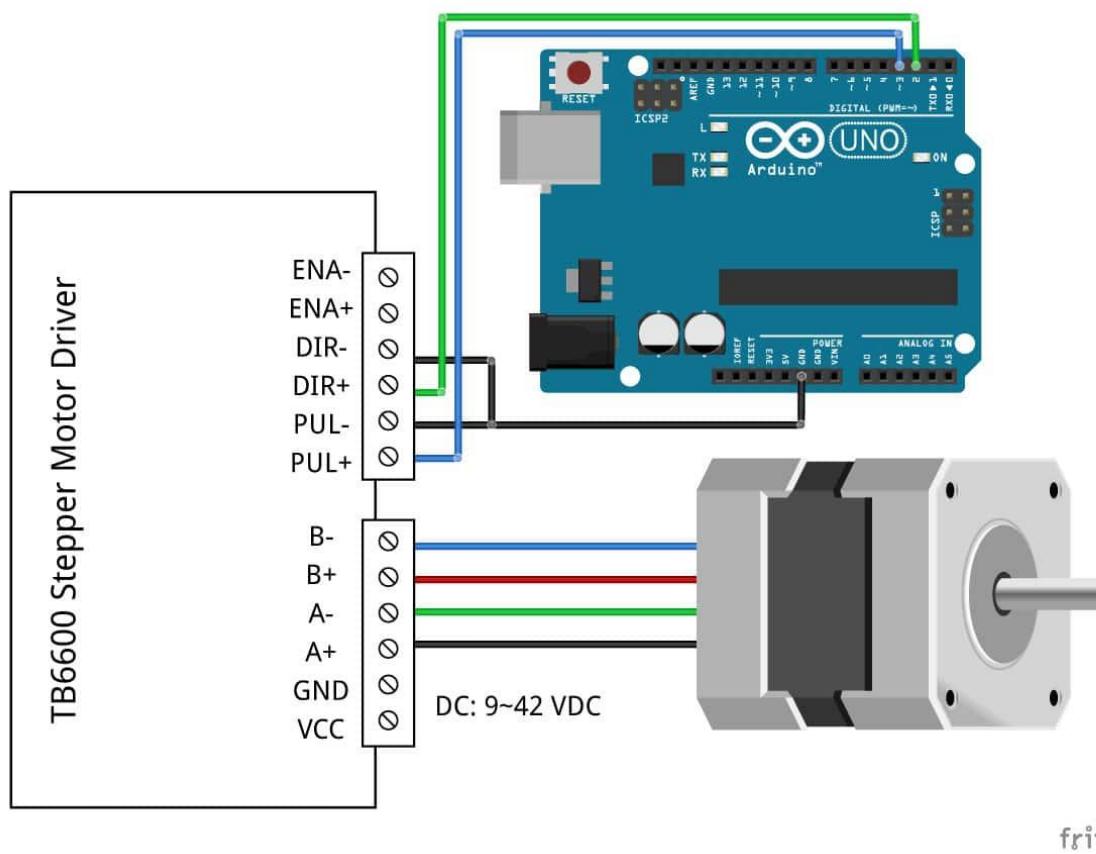
Spodnja shema prikazuje, kako povezati koračne motorje z gonilniki, napajalnikom, krmilnikom ter kako na krmilnik povezati releje s krmilnikom za dodatne aktuatorje, kot je na primer prijemovalo robotske roke.



Slika 20: povezava koračnih motorjev z gonilniki in krmilnikom

Naslednja shema prikazuje kako povezati posamezni motor z gonilnikom, napajalnikom in krmilnikom. Jaz sem uporabil napajalnik z 24V 10A, gonilnike DM542T (5 krat) in DM320T (1 krat) in naslednje koračne motorje:

- Nema 17 koračni motor L = 39mm, prestavno razmerje 10:1, planetarni menjalnik visoke natančnosti, dolžina izhodne gredi 9.5mm
- Nema 23 koračni motor L = 56mm, prestavno razmerje 50:1, planetarni menjalnik visoke natančnosti, dolžina izhodne gredi 9.5mm
- Nema 17 koračni motor L = 39mm, prestavno razmerje 50:1, planetarni menjalnik visoke natančnosti, dolžina izhodne gredi 9.5mm
- Nema 11 koračni motor L = 51mm, prestavno razmerje 14:1, planetarni menjalnik visoke natančnosti, dolžina izhodne gredi 9.5mm
- Nema 17 z navojno palico premer = 8mm, dolžina = 200mm
- Nema 14 koračni motor L = 28mm, prestavno razmerje 19:1, planetarni menjalnik visoke natančnosti, dolžina izhodne gredi 9.5mm



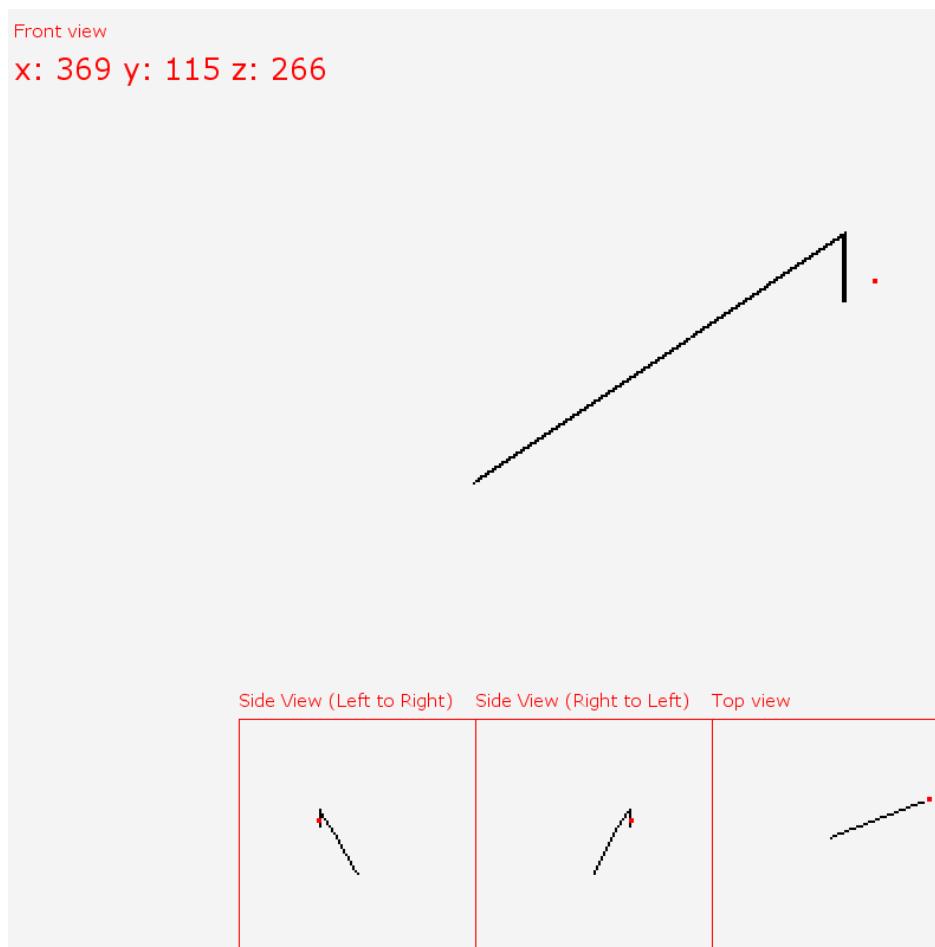
Slika 21: povezava koračnega motorja z gonilnikom in krmilnikom

IZDELAVA SIMULACIJSKEGA PROGRAMA

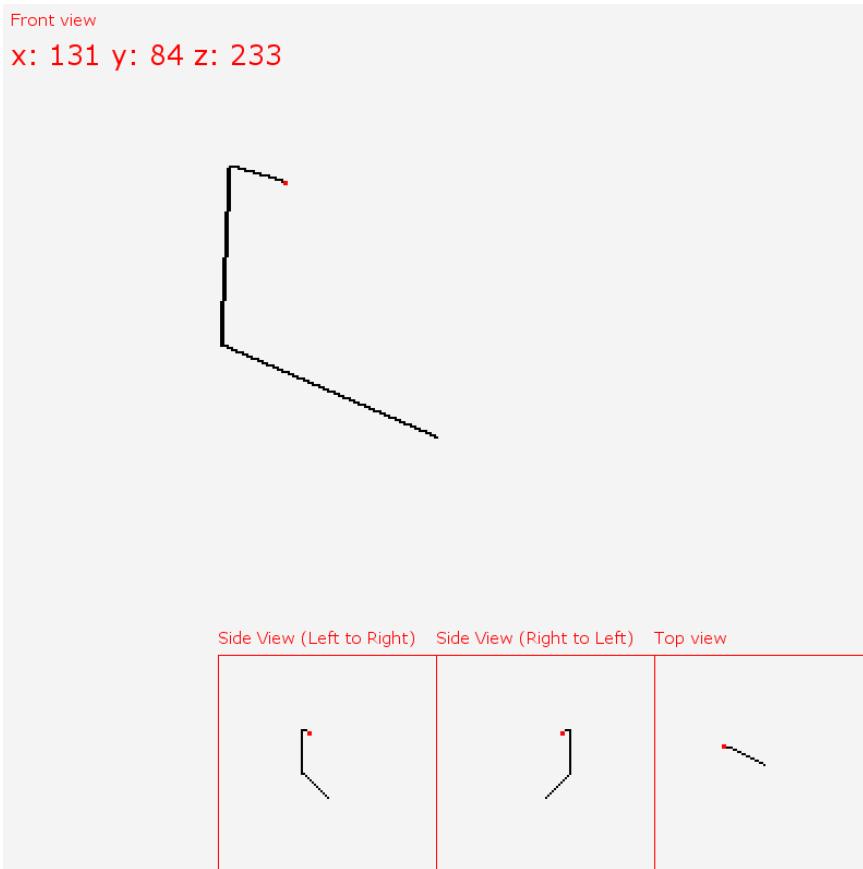
Za simulacijo gibanja robota sem izdelal simulacijski program s programskim jezikom Java. Program simulira gibanje robota v prostoru upošteva tako direktno kot inverzno kinematiko. V simulaciji roka sledi točki, ki se premika v prostoru in tako uporablja linearno transformacijo. Poleg tega lahko točki sledi tako, da ima zadnji sklep usmerjen proti točki ali pa je usmerjen navzdol oziroma tja kamor si to želimo, tako lahko simuliramo premikanje roke, kot ga robotske roke po navadi uporabljajo. Simulacijski program ni primeren za programiranje dejanske robotske roke, saj nima implementirane nobene zaščite proti premajhnimi oziroma prevelikimi koti na posameznih sklepih. Manka tudi omejitev prostora gibanje roke na prostor, ki je roki na voljo (roka se lahko zabije sama vase). Simulacijski program nam omogoča ugotavljanje efektivnosti programa saj lahko vidimo koliko časa se določen del programa izvaja, tako lahko prilagodimo program krmilniku, ki ima omejeno procesorsko hitrost. Simulacijski program ima implementiranih 5 od 6 osi (manjka 4 os).

Simulacijski program: <https://lukauranic.com/roboticArm/roboticArm.jar>

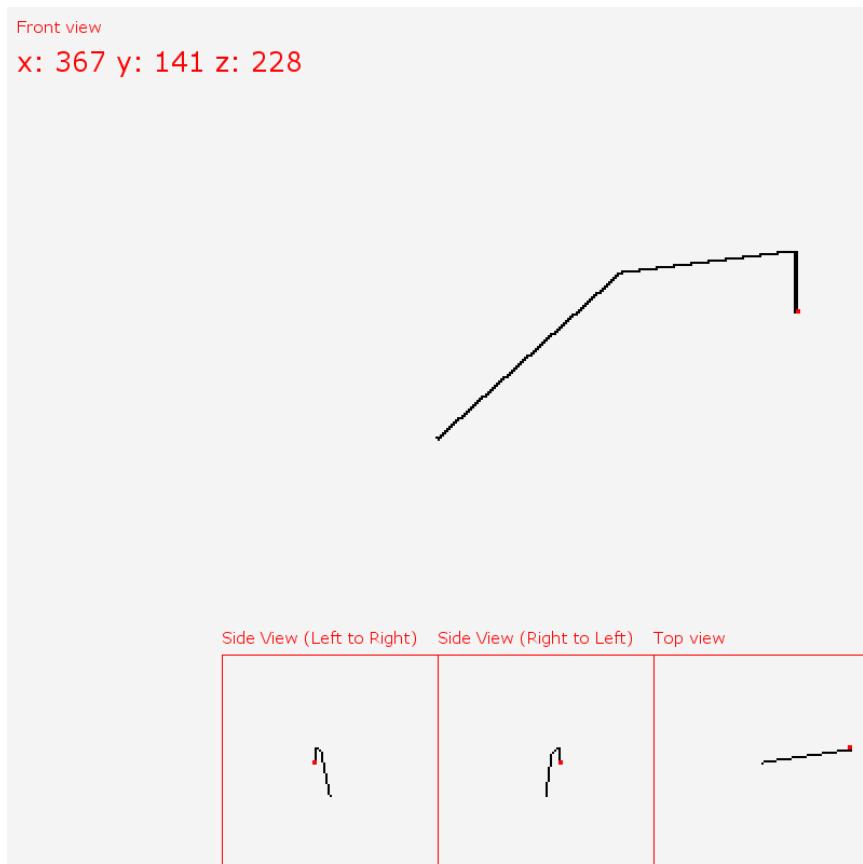
Koda simulacijskega programa: <https://lukauranic.com/roboticArm/src.zip>



Slika 22: simulacijski program I



Slika 23: simulacijski program 2



Slika 24: simulacijski program 3

IZDELAVA PROGRAMA ZA TESTIRANJE ROBOTSKE ROKE

Ko povežemo motorje, gonilnike, krmilnik in napajalnik, moramo napisati program, ki bo krmilil določen izdelek, v tem primeru robotsko roko. Spodnji program je napisan v programskem okolju Arduino in krmili 6 koračnih motorjev.

```
1 int stepPin = 2;
2 int dirPin = 3;
3 int enPin = 4;
4
5 int stepPin2 = 5;
6 int dirPin2 = 6;
7 int enPin2 = 7;
8
9 int stepPin3 = 8;
10 int dirPin3 = 9;
11 int enPin3 = 10;
12
13 int stepPin4 = 52;
14 int dirPin4 = 53;
15
16 int stepPin5 = 11;
17 int dirPin5 = 12;
18 int enPin5 = 13;
19
20 int stepPin6 = 22;
21 int dirPin6 = 23;
22 int enPin6 = 24;
23
24 void setup() {
25     // Sets the two pins as Outputs
26     pinMode(stepPin,OUTPUT);
27     pinMode(dirPin,OUTPUT);
28     pinMode(enPin,OUTPUT);
29     pinMode(stepPin2,OUTPUT);
30     pinMode(dirPin2,OUTPUT);
31     pinMode(enPin2,OUTPUT);
32     pinMode(stepPin3,OUTPUT);
33     pinMode(dirPin3,OUTPUT);
34     pinMode(enPin3,OUTPUT);
35     pinMode(stepPin4,OUTPUT);
36     pinMode(dirPin4,OUTPUT);
37     pinMode(stepPin5,OUTPUT);
38     pinMode(dirPin5,OUTPUT);
39     pinMode(enPin5,OUTPUT);
40     pinMode(stepPin6,OUTPUT);
41     pinMode(dirPin6,OUTPUT);
42     pinMode(enPin6,OUTPUT);
43
44     digitalWrite(enPin,LOW);
45     digitalWrite(enPin2,LOW);
46     digitalWrite(enPin3,LOW);
47     digitalWrite(enPin5,LOW);
48     digitalWrite(enPin6,LOW);
49 }
```

Slika 25: testni program I

```
50 | void loop() {
51 | // set direction to high
52 | digitalWrite(dirPin,HIGH);
53 | digitalWrite(dirPin2,HIGH);
54 | digitalWrite(dirPin3,HIGH);
55 | digitalWrite(dirPin4,HIGH);
56 | digitalWrite(dirPin5,HIGH);
57 | digitalWrite(dirPin6,HIGH);
58 | // move 6000 steps in one direction
59 | for(int i = 0; i < 6000; i++) {
60 |   digitalWrite(stepPin,HIGH);
61 |   if(i % 2 == 0) {
62 |     digitalWrite(stepPin2,HIGH);
63 |   }
64 |   digitalWrite(stepPin3,HIGH);
65 |   digitalWrite(stepPin4,HIGH);
66 |   digitalWrite(stepPin5,HIGH);
67 |   digitalWrite(stepPin6,HIGH);
68 |   delayMicroseconds(500);
69 |   digitalWrite(stepPin,LOW);
70 |   if(i % 4 == 0) {
71 |     digitalWrite(stepPin2,LOW);
72 |   }
73 |   digitalWrite(stepPin3,LOW);
74 |   digitalWrite(stepPin4,LOW);
75 |   digitalWrite(stepPin5,LOW);
76 |   digitalWrite(stepPin6,LOW);
77 |   delayMicroseconds(500);
78 | }
79 | // wait for 1000ms
80 | delay(1000);
81 | // change dir
82 | digitalWrite(dirPin,LOW);
83 | digitalWrite(dirPin2,LOW);
84 | digitalWrite(dirPin3,LOW);
85 | digitalWrite(dirPin4,LOW);
86 | digitalWrite(dirPin5,LOW);
87 | digitalWrite(dirPin6,LOW);
```

Slika 26: testni program 2

```

79 // wait for 1000ms
80 delay(1000);
81 // change dir
82 digitalWrite(dirPin,LOW);
83 digitalWrite(dirPin2,LOW);
84 digitalWrite(dirPin3,LOW);
85 digitalWrite(dirPin4,LOW);
86 digitalWrite(dirPin5,LOW);
87 digitalWrite(dirPin6,LOW);
88 // move 6000 steps in the other direction
89 for(int i = 0; i < 6000; i++) {
90   digitalWrite(stepPin,HIGH);
91   if(i % 2 == 0) {
92     digitalWrite(stepPin2,HIGH);
93   }
94   digitalWrite(stepPin3,HIGH);
95   digitalWrite(stepPin4,HIGH);
96   digitalWrite(stepPin5,HIGH);
97   digitalWrite(stepPin6,HIGH);
98   delayMicroseconds(500);
99   digitalWrite(stepPin,LOW);
100  if(i % 4 == 0) {
101    digitalWrite(stepPin2,LOW);
102  }
103  digitalWrite(stepPin3,LOW);
104  digitalWrite(stepPin4,LOW);
105  digitalWrite(stepPin5,LOW);
106  digitalWrite(stepPin6,LOW);
107  delayMicroseconds(500);
108 }
109 // wait for 1000ms
110 delay(1000);
111 |
112 //disables the driver/motor
113 //digitalWrite(enPin,HIGH);
114 }

```

Slika 27: testni program 3

REZULTATI

Robotsko roko sem izdelal s pomočjo 3D tiskanja. Pri tem sem večino delov tiskal z malo polnila (<50%). S tem sem pridobil na hitrosti tiskanja, vendar so nekateri deli premalo močni oziroma niso dovolj togi. S tem je roka izgubila na natančnosti in sposobnosti dvigovanja oziroma premagovanja težjih bremen. Kljub vsemu menim, da je cilj naloge dosežen, saj sem uspel narediti robotsko roko s svojo konstrukcijo, ki se lahko prosto giba po svojem delovnem prostoru. Prav tako je robotska roka sposobna premagovati lastno težo, kljub temu, da je narejena iz PLA plastike. Krmilnik Arduino Mega je sposoben krmiliti 6 osno robotsko roko. S tem sem dokazal, da se da izdelati robotsko roko z dokaj poceni elementi.

Hipotezi sta potrjeni, v razpravi pa so zapisane možne izboljšave sistema robotske roke.



Slika 28: končni izdelek

RAZPRAVA

Robotska roka se je sposobna gibati v tri dimenzionalnem prostoru, narejena je iz dokaj poceni elementov ni pa primerna za delo v industrijskem okolju, saj ni dovolj natančna, zaradi materialov iz katerih je narejena in postopkov prenosa momenta motorja na os oziroma sklep robotske roke. Ta ni idealen in je potreben izboljšave. Če bi hotel narediti robotsko roko, ki bi lahko bila uporabljena v industrijskem okolju, bi moral za ogrodje roke uporabiti kakšen trši material (aluminij), za prenos momenta s koračnega motorja na sklep pa bi moral uporabiti boljši prenos (veriga, zobniki). Načrtujem, da bom nalogo nadgradil tako, da bom roko povezal na internet in jo tako lahko krmilil preko spleta iz kjerkoli na svetu preko računalnika ali telefona. Prav tako pa bom večino delov naredil ponovno, saj sem opazil, da bi nekaj stvari lahko izboljšal. Tako bom izboljšal mehanske lastnosti robotske roke prav tako pa bom izboljšal tudi izgled, saj bom natisnil vse z eno barvo plastike.

ZAKLJUČEK

Z raziskovalno nalogo sem dosegel svoj cilj, ki sem si ga zadal. Uspelo mi je narediti robotsko roko. Pri tem sem se naučil veliko o kinematiki robotov, programiranju koračnih motorjev, programiranju Arduino krmilnikov, programiranju in premikanju točk v tri dimenzionalnem prostoru, konstruiranju in 3D tiskanju. S tem znanjem bom sposoben v prihodnosti narediti še veliko podobnih projektov in bom bolj zaposljiv.

LITERATURA

- Kinematika robota:
 - o https://en.wikipedia.org/wiki/Robot_kinematics
 - o https://en.wikipedia.org/wiki/Inverse_kinematics
 - o https://en.wikipedia.org/wiki/Forward_kinematics
- Koračni motorji:
 - o https://en.wikipedia.org/wiki/Stepper_motor
- Arduino:
 - o <https://en.wikipedia.org/wiki/Arduino>
- Ostalo:
 - o <https://blog.robotiq.com/how-to-calculate-a-robots-forward-kinematics-in-5-easy-steps>
 - o https://www.google.com/search?q=robot+kinematics&tbo=isch&ved=2ahUEwiXkvyh1P7nAhWOsaQKHR5NDwkQ2-cCegQIABAA&oq=robot+kinematics&gs_l=img.3..0j0i7i30l3j0i7i5i30l3j0i5i30l3.3670.4215..4495...0.0..0.124.678.0j6.....0....1..gws-wiz-img.....0i8i7i30.jGK7ijYYdCg&ei=1HxeXpfYFI7jkgWemr1I&bih=625&biw=1366#imgrc=iSjZnolr1mUOuM
 - o https://www.researchgate.net/figure/The-Inverse-Kinematics-Solutions-Based-on-Trigonometry_fig1_331685972
 - o https://www.google.com/search?q=inverse+kinematics&tbo=isch&ved=2ahUKEwiPyuy41P7nAhVIP-wKHUBwB4cQ2-cCegQIABAA&oq=inverse+kinematics&gs_l=img.3..35i39j019.48360.49827..49959...0.0..0.133.920.0j8.....0....1..gws-wiz-img.....0i7i30.4jhFBCIUmpg&ei=BH1eXo-4E8jsAfA4J24CA&bih=625&biw=1366#imgrc=qGISQPuFV5hXNM
 - o <https://www.omc-stepperonline.com/250w-24v-10a-115230v-switching-power-supply-stepper-motor-cnc-router-kits-s-250-24.html?search=power%20supply>
 - o https://www.researchgate.net/figure/Block-Diagram-1-ArduinoMEGA2560-The-Arduino-Mega-2560-is-a-type-of-microcontroller_fig5_281538436
 - o https://www.google.com/search?q=arduino+mega+pins&tbo=isch&ved=2ahUKEwjX_pOv1f7nAhUEM-wKHW7BAx4Q2-cCegQIABAA&oq=arduino+mega+pins&gs_l=img.3..0i10.23586.26570..26616...4.0..0.118.888.0j8.....0....1..gws-wiz-img.....35i39j0i67j0i5i30j0i8i30j0i30.hJZCFJFow2o&ei=H1eXteQGoTmsAfugo_wAQ&bih=625&biw=1349&hl=en#imgrc=CNBK5vBKOn4RAM