



УНИВЕРЗИТЕТ  
У НОВОМ САДУ



ФАКУЛТЕТ  
ТЕХНИЧКИХ НАУКА

Трг Доситеја Обрадовића 6, 21000 Нови Сад, Република Србија  
Деканат: 021 6350-413; 021 450-810; Централa: 021 485 2000  
Рачуноводство: 021 458-220; Студентска служба: 021 6350-763  
Телефакс: 021 458-133; e-mail: [ftndeans@uns.ac.rs](mailto:ftndeans@uns.ac.rs)

ИНТЕГРИСАНИ  
СИСТЕМ  
МЕНАџМЕНТА  
СЕРТИФИКОВАН ОД:



## ZAVRŠNI RAD

# *Uređaj za crtanje zasnovan na CNC mašini*

Mentor:  
dr. Vladimir Rajs

Student  
Miroslav Blažić  
ET 26/2016



UNIVERZITET U NOVOM SADU • FAKULTET TEHNIČKIH NAUKA  
21000 Novi Sad, Trg Dositeja Obradovića 6

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj, RBR:	
Identifikacioni broj, IBR:	
Tip dokumentacije, TD:	Monografska dokumentacija
Vrsta rada, VR:	Diplomski-Bečlor
Autor, AU:	Miroslav Blažić
Mentor, MN:	dr. Vladimir Rajs, docent
Naslov rada, NR:	Uređaj za crtanje zasnovan na CNC mašini
Jezik publikacije, JP:	Srpski
Jezik ivoda, JI:	Srpski
Zemlja publikovanja, ZP:	Srbija
Uže geografsko područje, UGP:	Vojvodina
Godina, GO:	2020
Izdavač, IZ:	Autorski reprint
Mesto i adresa, MA:	21000 Novi Sad, Trg Dositeja Obradovića
Fizički opis rada, FO: (poglavlja/stranica/citata/tabela/slika/grafika/ priloga)	(10/52/24/2/35/0/10)
Naučna oblast, NO:	Mikroračunarska elektronika
Naučna disciplina, ND:	Elektronika
Predmetna odrednica/Ključne reči, PO:	Arduino CNC, Računarska numerička kontrola, Robotizovana olovka
Čuva se, CU:	Biblioteka Fakulteta Tehničkih Nauka 21000 Novi Sad, Trg Dositeja Obradovića 6
Važna napomena, VN:	Nema
Izvod, IZ:	Rad opisuje implementaciju uređaja za CNC crtanje
Datum prihvatanja teme, DP:	4.4.2020.
Datum odbrane, DO:	8.10.2020.
Članovi komisije, KO:	Predsednik: dr. Kalman Babković, docent
	Član: dr. Miodrag Brkić, docent
	Mentor: dr. Vladimir Rajs, docent



UNIVERZITET U NOVOM SADU • FAKULTET TEHNIČKIH NAUKA  
21000 Novi Sad, Trg Dositeja Obradovića 6

## KEY WORDS DOCUMENTATION

Accession number, ANO:	
Identification number, INO:	
Document type, DT:	Monographic publication
Type of record, TR:	Textual material, printed
Content code, CC:	Graduate-Bachelor thesis
Author, AU:	Miroslav Blažić
Mentor, MN:	PhD Vladimir Rajs
Title, TI:	Drawing device based on CNC machine
Language of text, LT:	Serbian
Language of abstract, LA:	Serbian
Country of publication, CP:	Serbia
Locality of publication, LP:	Vojvodina
Publication year, LY:	2020
Publisher, PB:	Author's reprint
Publication place, PP:	21000 Novi Sad, Trg Dositeja Obradovića
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendixes)	(10/52/24/2/35/0/10)
Scientific field, SF:	Microcomputer electronics
Scientific discipline, SD:	Electronics
Subject/Keywords, S/KW:	Arduino CNC, Computer Numerical Control, Pen plotter
Holding data, HD:	Library of Faculty of Technical Sciences 21000 Novi Sad, Trg Dositeja Obradovića 6
Note, N:	None
Abstract, AB:	4.4.2020.
Defended on, DE:	8.10.2020.
Defended Board, DB:	President: PhD Kalman Babković
	Member: PhD Miodrag Brkić
	Mentor: PhD Vladimir Rajs

# Sadržaj

<b>Rezime</b>	<b>5</b>
<b>Cilj</b>	<b>5</b>
<b>UVOD</b>	<b>6</b>
<b>CNC MAŠINE</b>	<b>6</b>
NC mašine	6
Pojava CNC mašina	7
Dekartov koordinatni sistem	8
Kako CNC mašine rade	10
CAD/CAM	10
<b>MEHANIČKA STRUKTURA UREĐAJA</b>	<b>11</b>
X i Y - osa	11
Z - osa	11
<b>HARDVERSKA STRUKTURA UREĐAJA</b>	<b>12</b>
Aktuatori	13
Servomotori	13
Koračni motori	14
Mikroprekidači	17
Upravljački deo	17
Pogon motora	18
H - most	19
L293D integrisano kolo	19
Serijska komunikacija	20
USB komunikacija	21
Bluetooth komunikacija	21
Napajanje	22
<b>HARDVERSKO POVEZIVANJE</b>	<b>23</b>
Povezivanje L293D	23
Povezivanje servomotora	23
Povezivanje mikroprekidača	24
Povezivanje HC-05	25

<b>SOFTVERSKI ALATI</b>	<b>25</b>
Arduino IDE	25
G-kod	26
Ručno programiranje	26
Automatsko programiranje	26
Inkscape	27
Processing	28
<b>SOFTVERSKA STRUKTURA UREĐAJA</b>	<b>28</b>
Predpocesorske direktive	28
Globalne promenljive	29
Inicijalizacija uređaja	31
Funkcije	31
Glavni program	37
<b>SASTAVLJANJE UREĐAJA</b>	<b>39</b>
Sasavljanje konstrukcije	39
Sasavljanje X i Y - ose	40
Sasavljanje Z - ose	42
Sasavljanje Elektronike	43
<b>TROŠKOVI IZRADE</b>	<b>43</b>
<b>KORISNIČKO UPUTSTVO</b>	<b>44</b>
Grafička priprema	44
Komunikacija sa uređajem	45
Puštanje uređaja u rad	46
<b>ZAKLJUČAK</b>	<b>46</b>
<b>DODACI</b>	<b>47</b>
<b>LITERATURA</b>	<b>50</b>

## Rezime

Princip CNC uređaja se zasniva na istovremenom radu dva koračna (*eng. stepper*) i jednim servo motorom, pri čemu uređaj ispisuje ulaz koji je dobijen od mikrokontrolera ATmega 328P koji je ugrađen u Arduino Nano platformu. CNC uređaj se zasniva na kontroli dve ose (X i Y) koordinatnog sistema i specijalnim mehanizmom za podizanje i spuštanje olovke. X i Y osu predstavljaju dva koračna motora izvađena iz CD čitača, dok se kontrola olovke postiže upotrebom servo motora. Za pogon motora se koristi L293D koji je kompatibilan sa Arduino. Softverski alati koji se koriste u ovom projektu za programiranje Arduino modula su: Arduino IDE, Inkscape (0.48.5) i Processing (3.0.2). Ispravnom i efikasnom upotrebom ovih alata zajedno sa električnim komponentama i električnom šemom čine ovaj projekat funkcionalnim.

## Cilj

S obzirom da je CNC danas vrlo aktuelno područje istraživanja, u nastavku ovog rada će biti realizovan model CNC-a kako bi se stekla teorijska i praktična saznanja ovog koncepta. Nakon što se upoznamo sa konceptom fokus će biti na osbosobljavanju mašine i razvoju hardvera i softvera za upravljanje mašinom koja će biti u mogućnosti da nacrti fizičku kopiju na papiru nakon što se željeni crtež nacrtu u elektronskom obliku pomoću softverskog CAM alata.

# 1. UVOD

U ovom radu se na jednostavan i razuman način upoznaje sa radom numerički upravljanim mašinama (CNC). Kod nas nema mnogo odgovarajuće literature koja se bavi ovom tematikom, te će rad biti od koristi za sve one kojima je CNC tehnologija interesantna.

U radu su obrađeni osnovni koncepti konstruisanja, elektronike i programiranja koji su potrebni da bi se steklo osnovno znanje. Rad će se najpre baviti teoretskim delom, u kojem će se govoriti kako CNC mašine funkcionišu i šta je sve potrebno za izradu, te će se svaka od stavki razjasniti ponaosob, zatim neophodnim softverskim alatima za upravljanje, i na kraju programiranjem CNC uređaja.

A kako ovaj rad ne bi ostao samo “mrtvo slovo na papiru”, realizovan je i praktični deo. U praktičnom delu rada će se fizički napraviti CNC uređaj koji će u realnosti moći da piše ili crta sve u zavisnosti šta korisnik izdizajnira na računaru preko CAM alata. Ekonomičnost izrade CNC uređaja je poprilično prilagodljiva jer se većina potrebnog materijala koristi od starih rashodovanih delova, koje ih gotovo svako može priuštiti.

Kompletan projekat je otvoren za svakoga, te se mogu preuzeti svi fajlovi sa GitHub linka, koji se nalazi u prilogima na kraju ovog rada.

## 2. CNC MAŠINE

Termin CNC je široko prihvaćen i često korišćen izraz u mašinskoj industriji. U današnje vreme mašinska industrija je nezamisliva bez savremenih CNC mašina. Oblast njihove primene je veoma široka.

### 2.1. NC mašine

Numerička kontrola, popularno poznata i kao NC (Numerical Control), često se koristi u alatnim mašinama. Numerička kontrola je definisana kao oblik programabilne automatizacije, u kojoj se proces kontroliše brojem, slovima i simbolima. Drugim rečima, NC mašina je definisana kao mašina koja se kontroliše nizom uputstava, a ta uputstva se nazivaju program. Brojevi čine osnovna programska uputstva, zbog toga je i naziv “numerička kontrola”.

Izum numeričke kontrole nastao je zahvaljujući John T. Parsons 1940. godine, kada je pokušavao automatski da generiše krivulju glodanjem rezača preko koordinatnih pokreta. Tako je Parsons došao na ideju da koristi traku sa bušenim rupama koje su sadržale sistem koordinatnog položaja za kontrolu mašinskog alata. Na ovaj način mašina je usmerena na kretanje u malim koracima.



*Slika 1.1. - Bušene trake za NC mašine.*

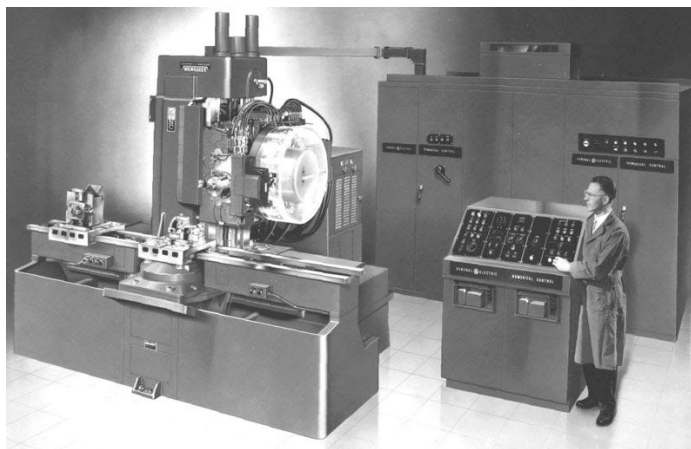
Godine 1948. Parsons je predstavio ovaj koncept Američkom ratnom vazduhoplovstvu koji je financirao dosta projekata na MIT-u. Nakon mnogo istraživanja, MIT je demonstrirao prvi NC prototip 1952. godine. Ubrzo su proizvođači alatnih mašina počeli da primenjuju velike napore kako bi uveli komercijalne NC jedinice na tržište.

MIT je i dalje nastavio da istražuje NC, i vrlo brzo razvili su APT jezik (Automatically Programmed Tools) sa kojim se moglo programirati NC mašine. Glavni cilj tog jezika je bio da obezbedi programerima lakši način da zadaju instrukcije mašinama. APT jezik je koristio naredbe na Engleskom jeziku.

APT jezik je još uvek široko korišćen u industriji, čak i nekoliko poznatih jezika se zasnivaju na konceptima APT-a.

## **2.2. Pojava CNC mašina**

U prvim godinama NC, koristile su se bušene trake za puštanje instrukcija mašini. Sa dolazkom APT jezika obeležen je dolazak CNC mašina, i vrlo brzo su NC mašine bile zamenjene sa CNC mašinama.



*Slika 2.1 - Prve CNC mašine.*



Kod CNC mašina instrukcije se više ne čuvaju na trakama, nego u memoriji kompjutera. Programer sada može lako da piše instrukcije i da ih uređuje po zahtevima.

U poređenju sa NC mašinama CNC su nudile mnogo više fleksibilnosti, efikasnosti i sposobnost. Sada se nove instrukcije mogu ugraditi u CNC regulator jednostavnim reprogramiranjem.

### 2.3. Dekartov koordinatni sistem

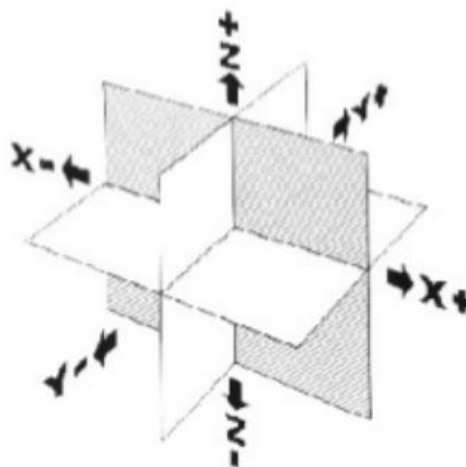
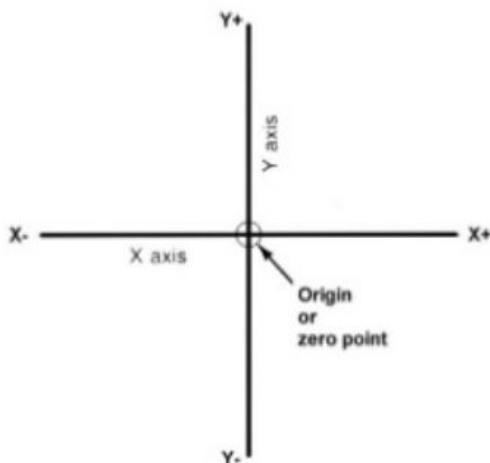
Gotovo sve što se može proizvesti konvencionalnom mašinom može se proizvesti i sa CNC mašinom, i to sa mnogo prednosti. U praksi postoje dva osnovna tipa kretanja glave alata: ravno kretanje (od tačke do tačke) i kontinuirano kretanje.

Dekartov ili pravougaoni koordinatni sistem je osmislio Francuski matematičar i filozof Rene Dekart. Sa ovim sistemom se određuje položaj tačke u prostoru gde je takav koordinatni sistem definisan središtem koordinatnog sistema 0, i tri oijentirane ose (x, y i z) s odgovarajućim jediničnim dužinama. Koordinate svake tačke u takvom sistemu zadate su uređenim skupom od 3 broja, na primer (3, -1, 5) koji označavaju odgovarajuće koordinate u trodimenzionalnom matematičkom prostoru, gde su koordinate predstavljene orjentiranim normalnim udaljenostima od neke tačke do odgovarajuće ravni. U trodimenzionalnom koordinatnom sistemu nazivi osa (apscisa i ordinata) nisu mandatorne, ali ako se upotrebljavaju tada je uobičajeno da se treća, z osa, nazvati aplikata. Na isti način je uobičajeno da se x i y osa postave u horizontalnu ravan, a preostala z osa normalno na njih. Konačno, trodimenzionalni koordinatni sistem se dili na osam područja, „oktanata”, omeđenih sa odgovarajućim delovima ravni.

Dekartov koncept se savršeno uklapa u CNC konstrukciju jer se pozicioniranje zasniva na tri ose kretanja. U našem slučaju kretanje funkcioniše na sledeći način: na ravnoj vertikali X-osa se kreće horizontalno desno-levo, radna površina Y-ose kreće horizontalno gore-dole, Z-osa se kreće vertikalno gore-dole i ona je fiksirana za X-osu.

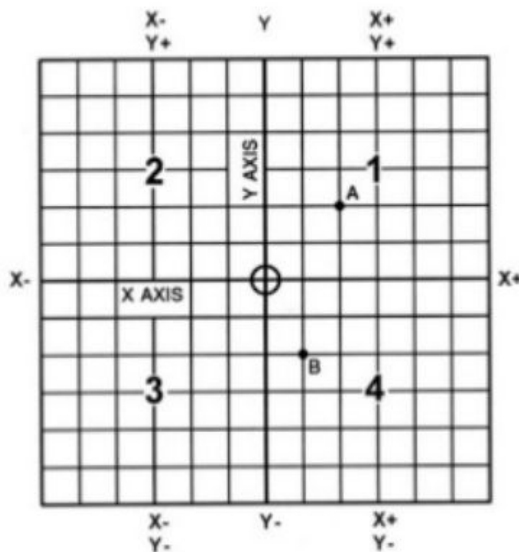
CNC sistemi se u velikoj meri oslanjaju na upotrebu pravougaonih koordinata zato što programer može da precizno da locira svaku tačku.

X i Y ose su horizontalne i one predstavljaju tabelarno kretanje, što je prikazano na slici 1.2.1., presečene linije formiraju prave uglove i formiraju nultu tačku. Z osa predstavlja vertikalno kretanje alatne glave, nadogradnjom ove ose dobija se trodimenzionalno kretanje, što je prikazano na slici 1.2.2, gotovo uvek se u CNC tehnologiji koristi ovakvo kretanje.



**Slika 2.2.** - 2-dimenziono kordinatno kretanje.      **Slika 2.3.** - 3-dimenzionalno kordinatno kretanje.

Na slici su prikazana četiri kvadranta koja se dobiju međusobnim ukrštanjem XY ose koordinatnog sistema. Kvadranti su numerisani suprotno od smera kazaljke na satu sa brojevima 1, 2, 3 i 4. Plus (+) i minus (-) znakovi ukazuju na pravac od nulte tačke duž ose kretanja. Sve pozicije koje su smeštene kvadrantu 1 će biti pozitivne ( $X+$  i  $Y+$ ), u kvadrantu 2 će biti negativna  $X-$  pozicija i pozitivna  $Y+$  pozicija, u kvadrantu 3 će sve pozicije biti negativne ( $X-$  i  $Y-$ ), u kvadrantu 4 će  $X+$  pozicija biti pozitivna a  $Y-$  negativna. Kao što je prikazano na slici tačka A bi imala vrednost 2 za X osu i vrednost 2 za Y osu, pod pretpostavkom da svaki kvadrat na slici iznosi 1,000. Tačna lokacija tačke A će biti  $X+ 2,000$  i  $Y+ 2,000$ . Za tačku B tačna lokacija bi bila  $X+ 1,000$  i  $Y- 2,000$ . U CNC programiranju nije neophodno navesti znak plus (+), jer se pretpostavlja da je pozitivna vrednost. Međutim, od velike je važnosti navesti znak minus (-), jer će se u suprotnom vrednost pretpostavljati pozitivno.



**Slika 2.4.** - Ukrštanje XY ose.

Tačna zabeleška za obe tačke (A i B) koju CNC mašina može da razume bi se uradilo na sledeći način:

A	X2.000 Y2.000
B	X1.000 Y-2.000

## 2.4. Kako CNC mašine rade

CNC mašina se sastoji od kompjutera na čijoj memoriji se nalazi program za rezanje metala prema zahtevima. Svi procesi sečenja koji treba da se izvrše i sve krajnje dimenzije se unose preko programa u računar. Računar tako zna šta treba da uradi, te na taj način sprovodi proces sečenja. CNC mašina funkcioniše slično kao robot koji se mora hrantiti programom i tako sledi vaša uputstva.

Kao rezultat pomeranja glave alata kroz više tačaka u materijalu dobijamo sečenje ili oblikovanje materijala po željenom dizajnu koji je predstavljen određenim položajem tačaka na kompjuteru. Kompjuter kao zadatak CNC mašini konstantno šalje položaj tačaka sve dok posao u potpunosti ne bude obavljen.

Neke od uobičajenih alatnih mašina koje mogu da se pokreću na CNC-u su:

- Strug,
- Glodalica,
- Bušilica,
- Lasersko sečenje

Glavna svrha ovih mašina je uklanjanje nekog materijala kako bi se dobio željeni oblik, poput okruglog, pravougaonog, i slično...

Sa tradicionalnim metodama ovim mašinama upravljaju operatori koji su stručnjaci za rad ovih mašina. Većini poslova je bila potrebna mašinska preciznost, te je bio neophodan stručan operator kako bi se mašina obavila svoj posao veoma precizno.

Danas je uloga CNC operatora svedena na minimum, kao i njegove odgovornosti. Današnja uloga operatora se uglavnom samo svodi na pokretanje programa za mašinu, ubacivanje alata u mašinu, dok ostatak posla automatski obavlja mašina.

Većina proizvođača CNC mašina kupuju kontrolne jedinice koje stavljaju u svoje mašine. Jedni od najpoznatijih i najrasprostranjenijih kontrolnih jedinica u svetu su: Fanuc iz Japana, Siemens i Heidenhein iz Nemačke...

## 2.5. CAD/CAM

CNC je doneo veliku promenu u proizvodnoj industriji, pa je sledeći razvoj bila kombinacija CAM (Computer Aided Manufacturing) i CAD (Computer Aided Designing), koja je nazvana CAD/CAM.

CAD - se odnosi na softverski alat pomoću kojeg se dizajnira proizvod i sa kojim se dokumentuje proces dizajniranja. Može se koristiti za dizajniranje 2-dimenzionalnih ili 3-dimenzionalnih nacrti, koji se prilikom rotiranja mogu posmatrati iz bilo kog ugla.

CAM - se odnosi na softverski alat pomoću kojeg se kreiraju detaljne instrukcije (na primer, G-kod) prethodno nacrtanog CAD crteža. CAM je namenjen za prevođenje crteža i podataka u detaljne instrukcije koje će pokrenuti autoamtizovan alat. Na primer, 2D crtež nacrtan preko CAD softverskog alata se može koristiti za vođenje olovke kako bi se fizički oslikao crtež.

CAD/CAM softver se koristi za dizajn i proizvodnju prototipova, gotovih proizvoda i za pokretanje proizvodnje. Integrirani CAD/CAM sistem nudi jedno celokupno rešenje za proizvodnju kroz dizajn.

CAD/CAM aplikacije se koriste za dizajniranje procesa proizvodnje i programa za CNC obradu. CAM se služi modelima i sklopovima koje su kreirane u CAD softveru za generisanje putanja alata da bi dizajn pretvorili u fizičke delove.

### **3. MEHANIČKA STRUKTURA UREĐAJA**

Da bih se glava alata CNC mašine mogla slobodno kretati, neophodno je da obezbedimo 3-dimenzionalni prostor, odnosno X, Y i Z osu, koje u praksi označavaju visinu, širinu i dubinu.

#### **3.1. X i Y - osa**

X i Y osa služe da bismo obezbedili kretanje tačke u dve dimenzije.

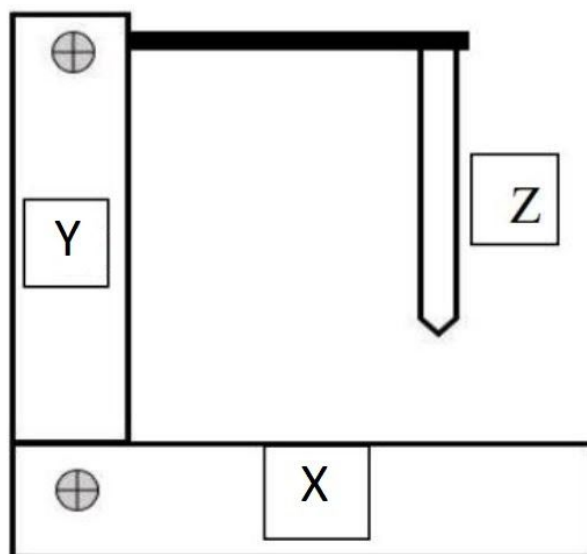
X osa se koristi kao baza koja se postavlja horizontalno i na nju se postavlja radna površina CNC mašine. Sa Y osom se dobija mogućnost horizontalnog kretanja radne površine gore-dole.

Y osa se postavlja vertikalno i to pod uglom od 90 stepeni u odnosu na X osu. Sa Y osom se dobija mogućnost kretanja horizontalnog kretanja desno-levo.

Na obe ose (X i Y) se nalazi po jedan motor na kojem je fiksirana pužasta osovinica. Ta osovinica daje mogućnost da se preko pužastog prenosa odvija kretanje X i Y ose. Kao rezultat toga dobija se dvodimenzionalno kretanje tačke.

#### **3.2. Z - osa**

Z osa se kod CNC mašina gotovo uvek koristi kao nosilac glave alata. Z osa je pričvršćena za Y osu koja se kreće horizontalno gore-dole, a Z osa se kreće vertikalno gore-dole. Sa Z osom dobijamo potpuno kretanje u 3-dimenzionalnom prostoru.

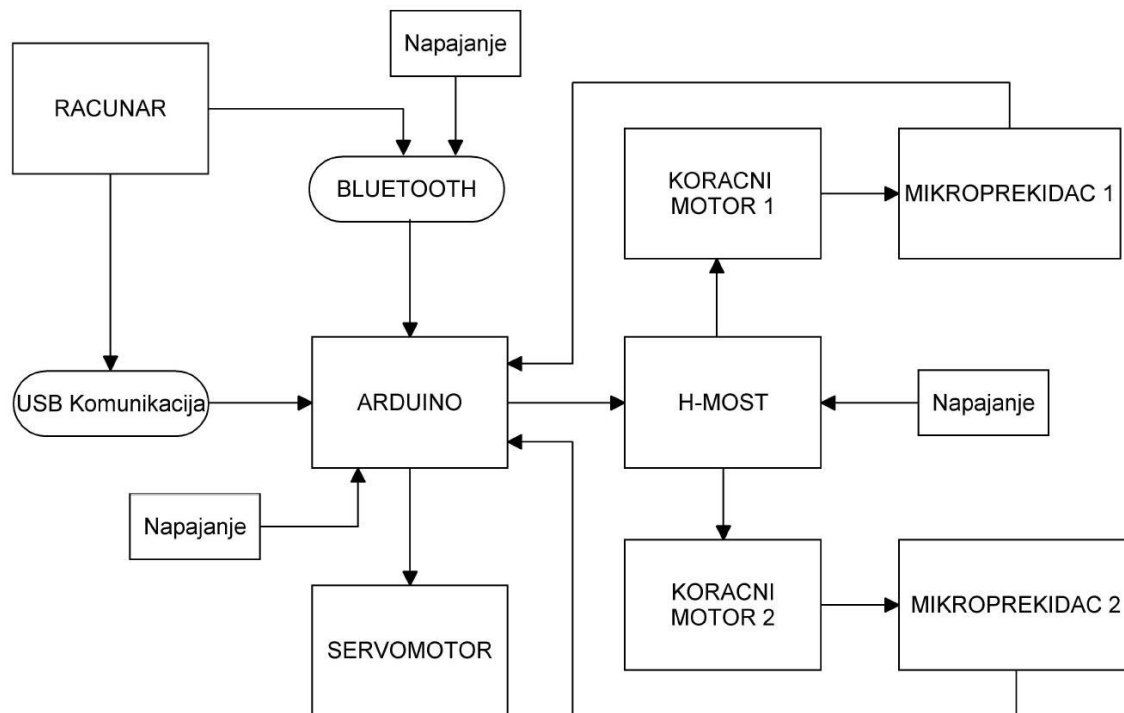


*Slika 3.1. - Mehanička skica CNC uređaja.*

Glava alata CNC uređaja je olovka, kojoj je potrebno da ima samo dva stanja, odnosno piše ili ne piše. Stoga je neophodno da se ugradi dodatni motor na Z osu koji će biti u stanju da povuče olovku gore u slučaju kada ona ne treba da piše. Ukoliko ne bi bila ugrađena Z osa (ili motor na Z osi) olovka nikada ne bi bila u stanju da se povuče na gore, a ishod toga bi bio da olovka ne bi mogla nikad napraviti prekid ukoliko joj je potreban, odnosno mogla bi ispravno da crta samo crteže koji ne zahtevaju prekid olovke.

## 4. HARDVERSKA STRUKTURA UREĐAJA

Uređaj je zamišljen da se ostvari komunikacija sa računarom na dva načina: ili preko USB serijskog kabela za Arduino Nano, ili preko bluetooth veze. Oba načina komunikacije su serijskog tipa koji omogućavaju Arduino da primi informacije o crtežu (G-kod) koje će se kao rezultat odraziti na fizičke pomeraje aktuatora. Posrednik između koračnih motora i Arduina je H-most drajver sa kojim Arduino upravlja preko kontrolnih signala. Napajanje za koračne motore se primenjuje na H-most. Preko mikroprekidača se ostvarju povratna sprega između koračnih motora i Arduina. Upotrebom mikroprekidača se određuje nulti položaj CNC uređaja. Servo motor ima tačnu informaciju o svom trenutnom položaju koji se zadaje preko Arduina, tako da između servomotora i Arduina nema potrebe praviti povratnu spregu. Sve komponente (Bluetooth, Arduino, Servomotor, koračnimotori) se napajaju sa istim naponom od 5V, što će rezultirati da na PCB ploči bude samo jedan priključak za napajanje kompletnog uređaja.



*Slika 3.2. - Blok šema uređaja.*

## 4.1. Aktuatori

Aktuatori su komponente neke mašine koji su odgovorni za kretanje i upravljanje mehanizmom ili celokupnim sistemom. Svaki aktuator zahteva kontrolni signal i izvor napajanja. Kada kontrolni signal prima informaciju, aktuator tad pretvara električnu energiju u mehaničke pokrete, odnosno, aktuatori su deo mehanizma sa kojim upravljački sistem deluje na okruženje.

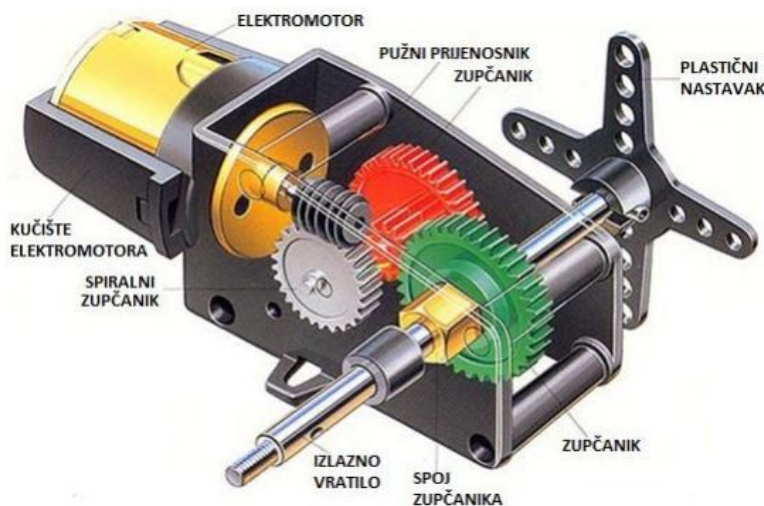
## 4.2. Servomotori

Servomotor je aktuator koji se sastoji od jednosmernog DC motora čija se brzina okretanja redukuje u svrhu povećanja obrtnog momenta. Redukcija brzine okretanja se obavlja preko prenosioca, uglavnom zupčanika, te se na taj način povećava obrtni moment motora. S obzirom na to da se kod ovog tipa motora povećava obrtni momenat uz pomoć mehaničkih delova koji direktno utiču na brzinu okretanja tj. povećavanjem obrtnog momenta se smanjuje brzina okretanja osovine DC motora. Na primer, ukoliko je prenosni odnos zupčanika 1:4, to znači da će se na izlazu servomotora motora osovina 4 puta sporije okretati u odnosu na osovinu DC motora.

Servomotori uglavnom imaju neku vrstu elektroničkog enkodera za merenje pozicije tj. da bismo dobili informaciju za koliko se stepeni zarotirala osovina. Može da radi u dva režima: ili da stoji, ili da se okreće pod punom brzinom.

Servomotori su jedinstveni po tome što osovina motora može da se pomera po tačnim uglom, većina njih se okreće za samo 180 ili 270 stepeni, međutim postoje i servomotori koji mogu da se okreću za punih 360 stepeni. Servomotor je “svestan” svog položaja iako se na njega primeni spoljna sila koja će da pomeri osovину motora.

Servomotori imaju 3 priključka: VCC, GND i Signal. VCC i GND priključci služe za napajanje servomotora, dok se na Signal priključak dovodi pulsno širinska modulacija (PWM signal) određene frekvencije.



*Slika 3.3. - Unutrašnjost servomotora.*

### 4.3. Koračni motori

Koračni motori se koriste u velikom broju uređaja kao što su 3D štampači i razne CNC mašine. Reč je o DC motorima kojima se osovina okreće u koracima ili mikrokoracima, te su od velike koristi kada se preciznost dovodi u pitanje.

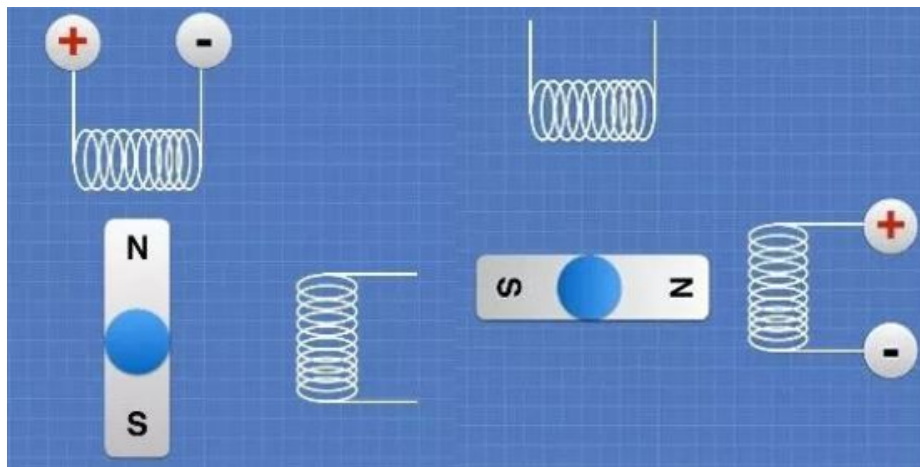
Za razliku od DC motora koračni motori se pokreću primenom impulsa na krajeve njihovih unutrašnjih kalemova. Svaki impuls pomera motor za po jedan korak ili za deo koraka koji se nazivaju mikrokoraci.

Iako postoji dosta sličnosti između servomotora i koračnog motora, ne treba ih mešati, jer oni rade totalno dve različite stvari. Za razliku od servomotora koji u svakom trenutku zna svoju poziciju, koračni motor ne zna za svoju trenutnu poziciju. Koračni motor može da se pomeri u tačan položaj u odnosu na kad je započeo kretanje (na primer, 36 stepeni u smeru kazaljke na satu), ali će za razliku od servomotora njegov položaj biti neusklađen ukoliko se na koračni motor primeni spoljna sila tako da motor “prokliže”.

Zbog toga što se kreću u koracima, ovi motori se uglavnom ne koriste za potrebe glatkog kontinuiranog kretanja. Između svakog koraka postoji izvesno vreme čekanja (u milisekundama ili mikrosekundama) pa će se stvarati vibracije, što je korak veći veća će i

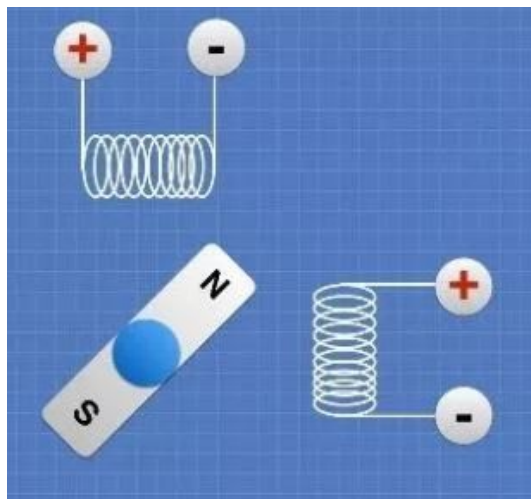
vibracija biti. Međutim, ukoliko se upotrebe mikrokoraci i mehanički prenos preko zupčanika moguće je ostvariti glatko kontinuirano kretanje, a vibracije smanjiti na minimum.

Koračni motori imaju magnetno jezgro koje je uglavnom okruženo sa dva kalema. Precizno upravljanje sa strujom u kalemu postiže se kretanje osovine u diskretnim koracima, kao što je prikazano na slici:



*Slika 4.1. - Princip rada koračnog motora.*

U prvom delu slike napon je primenjen na kalem od gore, pa je magnetna osovina privučena ovim kalemom. U drugom delu slike, napon je na kalem primenjen sa desne strane, tako je i osovina privučena sa te strane. Skok između dve pozicije je jedan korak, po prethodnoj slici je za 90 stepeni. Međutim, moguće je da se motor postavi u položaj između dva kalema, u našem slučaju za 45 stepeni, što je poznato kao mikrokorak.



*Slika 4.2. - Mikrokorak koračnog motora.*

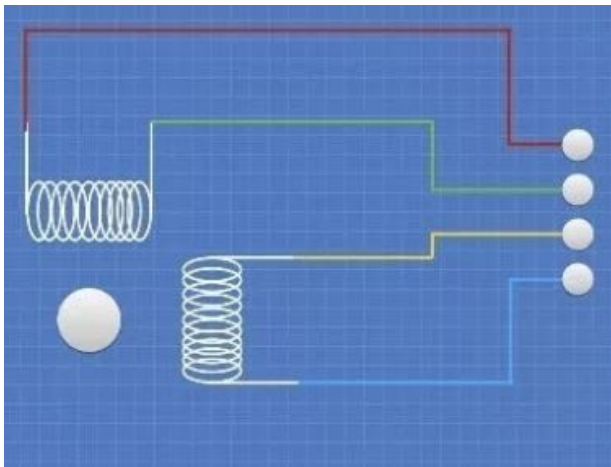
Po prethodnoj slici, struja je primenjena na oba kalema u jednakom iznosu. To dovodi do položaja motora na pola puta između oba kalema, što se naziva polu-korak. Ovaj princip se može primeniti i na četvrt-korak, osmo-korak pa čak i šesnaesto-korak. Svi ovi koraci se odnose na mikrokorake. Mikrokoraci se postižu kontrolom odnosa struje koja je primenjena



na oba kalema da bi se osovina motora privukla u položaj između kalema, ali bliže jednom kalemu nego drugom. Upotrebom mikrokoraka omogućuje se rotiranje motora za određen broj stepeni, ili na određen deo stepena, što direktno utiče na preciznost.

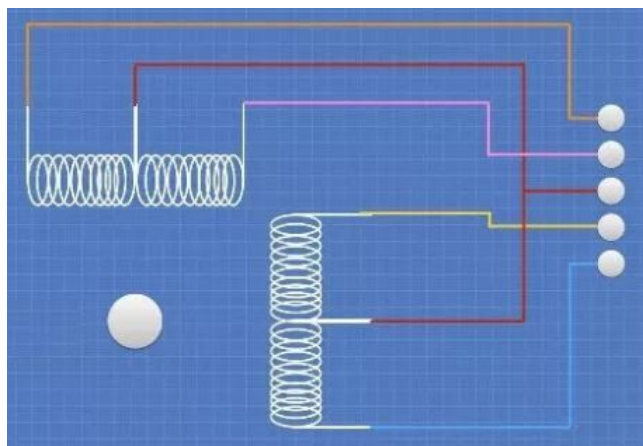
Postoje dva tipa koračnih motora: bipolarni i unipolarni. Od unipolarnog se može napraviti bipolarni.

Bipolarni koračni motor - Sastoji se od 2 kalema i 4 izlazna priključka, po 2 za svaki kalem. Prednost bipolarnih koračnih motora je u tome što koriste ceo namotaj kalema, tako da su efikasniji.



**Slika 4.3.** - Bipolarni koračni motor.

Unipolarni koračni motor - Takođe sadrži 2 kalema koji u centru imaju još jedan izlaz, tako da postoje 3 izlaza za svaki kalem. Ovo rezultira 6 izlaza na motoru, međutim mnogi unipolarni koračni motori imaju samo 5 izlaza, jer su središnji izlazi povezani unutar koračnog motora.



**Slika 4.4.** - Unipolarni koračni motor.

U svakom trenutku koristi se samo jedna polovina svakog kalema. Uglavnom se pozitivan napon se primenjuje na sredini kalema, a negativan se primenjuje na jednu od strana kalema.

Kao i kod bipolarnog motora, unipolarni se takođe pomera za po jedan korak kada se struja ukloni sa jednog kalema i primeni na drugi kalem. Takođe isti princip važi i za mikrokorake kao i kod bipolarnih.

Unipolarni motori su lakši za upravljanje u odnosu na bipolarne, jer nema potrebe da se polaritet struje promeni da bi se promenio smer okretanja. Međutim unipolarni koriste samo jednu polovinu kalema u svakom trenutku, zbog toga nemaju neku veliku efikasnost. Postoje unipolarni motori koji imaju dva namotaja po kalemu, što rezultuje 8 izlaza na motoru.

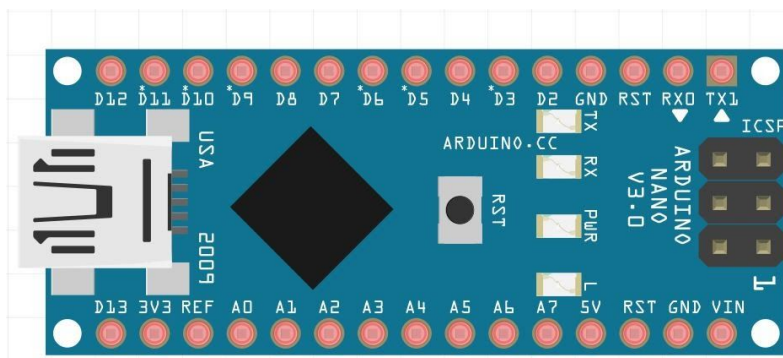
#### 4.4. Mikroprekidači

Mikroprekidači su mali senzori koji deluju na kretanje ili pritisak. Ukoliko se na njihov vrh (dugme) primeni pritisak oni će zatvoriti ili otvoriti strujno kolo, sve u zavisnosti od proizvođača samog mikroprekidača. Njihova primena je višestruka i veoma su pogodni za detekciju pokreta.

S obzirom da koračni motori nemaju početnu informaciju o svojoj poziciji, ukoliko bi se CNC uređaj iz nekog razloga u toku rada “zaglavila” ili prinudno prekinula, sledeći put kada se pokrene glava alata neće biti na svom početnom položaju. Zbog toga se koriste mikroprekidači, tako što se pre svakog pokretanja mašine X i Y osu voditi početni položaj, mikroprekidač će da detektuje taj položaj i zaustaviti motore.

#### 4.5. Upravljački deo

Upravljački modul koji se koristi u ovom radu je Arduino Nano. Arduino Nano je mala kompletna ploča koja je zasnovana na ATmega328P mikrokontroleru. Nudi istu električnu povezanost i specifikacije kao i Arduino UNO.



*Slika 4.5. - Arduino Nano.*

U opticaju su 14 digitalnih ulazno/izlaznih priključaka (od kojih se 6 može upotrebiti za PWM izlaz) i 8 analognih ulaza. Digitalni priključci se mogu upotrebiti za interfejs sa senzorima tako što će se definisati kao ulazni priključci ili se mogu koristiti kao pogonsko opterećenje (na primer pokretanje motora) tako što ćemo ih definisati kao izlaze. Radni napon svih priključaka je od 0 do 5 volti.

Arduino nano može da napaja preko Mini-B USB konektora, preko čega se takođe i odvija serijska komunikacija sa kompjuterom. Koristi kristalni oscilator od 16 Mhz, poseduje ICSP priključak za upisivanje firmvera (eng. firmware), kao i dugme za reset.

Serijski priključci 0 (Rx) i 1 (Tx) se mogu upotrebiti za prijem i prenos serijskih podataka. Spoljni prekidi (eng. External Interrupt) se mogu realizovati preko priključka 2 i 3, tako što će se konfigurisati kao okidači (eng. trigger) na niskoj naponskoj vrednosti, rastućoj ili opadajućoj ivici signala, a mogu da se konfigurišu da reaguju na promenu napona.

SPI komunikacija može da se odvija preko priključaka 10 (SS), 11 (MOSI), 12 (MISO) i 13 (SCK), dok se I2C komunikacija može odvijati preko A4 (SDA) i A5 (SCA) priključka.

## 4.6. Pogon motora

Priključiti motor direktno na priključak mikrokontrolera je poprilično loša ideja. Mikrokontroleri su projektovani da na izlaznim priključcima obezbede malo struje, odnosno daleko manje struje nego što je potrebno motoru. Ukoliko bi povezali motor direktno na mikrokontroler, motor bi povukao veću struju nego što je mikrokontroler predviđen, pa bi kao ishod toga došlo do disipacije, veća struja bi dovela mikrokontroler u zagrevanje, odnosno električna energija bi se pretvarala u toplotnu koja može trajno oštetiti mikrokontroler. Još jedna opasnost koja vreba mikrokontroleru pri direktnom povezivanju motora jeste to što je motor ujedno i generator, te će se pri pokretanju mikrokontrolera proizvesti elektromotorna sila (EMF) koja takođe može biti dovoljno velika da trajno ošteti mikrokontroler.

U svakom mikroračunarskom sistemu sa motorom neophodan je posrednik između mikrokontrolera i motora, taj posrednik je motor drajver, koji se napaja sa posebne grane i taj napon se primenjuje na motor. Mikrokontrolerom se preko kontrolnih signala upravlja drajverom, dok se motor se napaja preko drajvera, što na kraju rezultuje da možemo upravljati motorm preko mikrokontrolera a da napajanje motora i mikrokontroler budu galvanski razdvojeni.

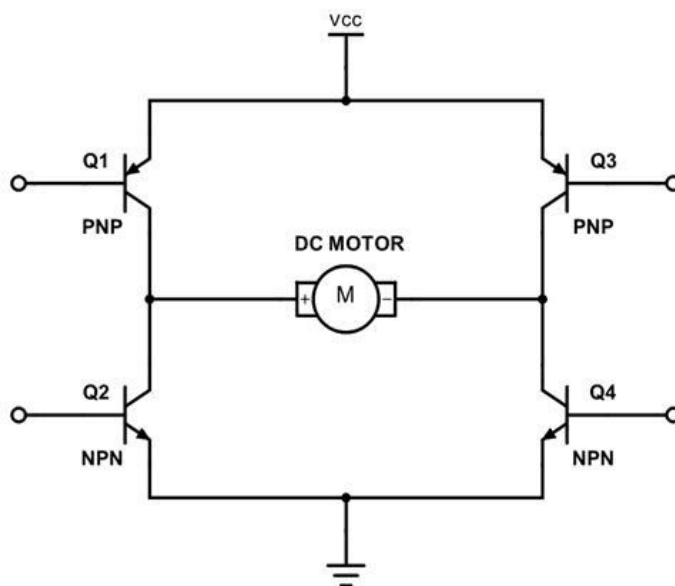


*Slika 4.6. - Indirektno upravljanje motorom.*

Na slici je prikazano indirektno upravljanje motorom bez povratne sprege. Ukoliko bi sistem zahtevao da u svakom trenutku znamo tačan položaj motora ili informacije o brzini kretanja motora tada bi zatvorena povratna sprega bila od suštinske važnosti. Zatvorena povratna sprega se može realizovati pomoću enkodera, koji bi u svakom trenutku mikrokontroleru isporučivao informacije o brzini i smeru motora.

### 4.6.1. H - most

H most je jednostavno elektronsko kolo koje se povezuje na mikrokontroler i služi za kontrolu motora u oba smera. Motor se okreće unapred ili unazad, s obzirom na polaritet motora koji je povezan na H most.



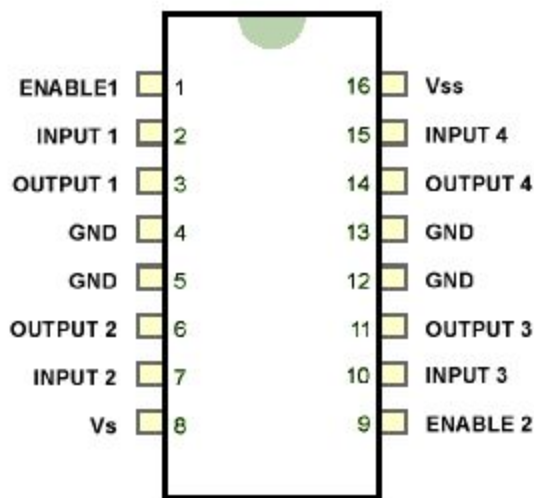
*Slika 4.7. - Električna šema H-mosta.*

Na slici je prikazan H-most drajver koji ima mogućnost da pokreće DC motor u oba smera, zbog toga ovaj drajver se naziva Full-bridge drajver. Princip rada je vrlo jednostavan, kada su tranzistori Q1 i Q4 u provodnom stanju a tranzistori Q2 i Q3 u neprovodnom tada će motor početi da se okreće na jednu stranu u zavisnosti kako je sam motor povezan na priključcima. Ukoliko bi Q1 i Q4 bili u neprovodnom stanju a Q2 i Q3 u provodnom, tada bi se motor zavrteo u suprotnom smeru.

### 4.6.2. L293D integrisano kolo

U ovom radu korišćen je L293D drajver koji u sebi ima ugradjen H most. Drajver u sebi ima dvostruki upravljački modul (eng. Full bridge), što znači da se može kontrolisati dva motora i to u oba smera ili jedan koračni motor s obzirom na to da je za njega potrebno 4 priključka.

L293D je namenjen za pogon induktivnih potrošača kao što su releji, solenoidi, DC i koračni motori. Komponenta se izrađuje u PDIP kućištu, i ima 16 priključaka kao što je prikazano na slici. Četiri ulazna drajvera (1A, 2A, 3A, 4A) koja upravljaju stanjima pripadajućim izlazima (1Y, 2Y, 3Y, 4Y). Drajveri se aktiviraju u paru, gde za aktiviranje prvog para moramo dovesti 5V na priključak 1 (1,2 EN). Komponenta takođe sadrži i zamajne diode za zaštitu logičkih sklopova od struje koje znaju prouzrokovati induktivni potrošači.



*Slika 4.8. - Izlazni priključci L293D drajvera.*

Napajanje logičkog sklopa je odvojeno od napajanja potrošača. Na VCC1 priključak se dovodi 5V i to je napajanje za logički sklop, dok se na VCC2 dovodi napon od 5-36V koji služi za napajanje potrošača.

Na priključke 1 i 9 se dovodi PWM impuls, koji će prouzrokovati da na izlazima (1Y, 2Y, 3Y, 4Y) dobijemo napon koji je proporcionalan srednjoj vrednošću faktora ispune PWM signala.

Bitan parametar kod rada sa H mosta je frekvencija PWM signala koju definiše mikrokontroler. Visoke frekvencije (>20kHz) su pogodne za suzbijanje vibracija u električnim motorima pri nižoj brzini obrtaja motora, s tim da visoka frekvencija ima i svojih nedostataka u pogledu prekomernog zagrevanja tranzistora na drajveru, odnosno veća je disipacija. Zbog toga je definisana maksimalna frekvencija u katalogu od strane proizvođača (datasheet), koja iznosi 5kHz za L293D.

Izlazni napon je 1.4V manji od VCC2 (pad napona), pri čemu se rasipna energija pretvara u toplotu, pa se elemenat zagreva, što je ujedno i mana ovog drajvera, te se preporučuje dodatno ugraditi hladnjake za L293D.

## 4.7. Serijska komunikacija

Potrebno je obezbediti komunikaciju između računara i Arduina. Najpre zbog upisivanja koda u Arduino, a potom i za korisničku upotrebu mašine. Preko računara CNC uređaju zadajemo G-kod u kojem se nalaze sve informacije i putanje po kojim se glava uređaja kreće.

U ovom radu je moguće uspostaviti komunikaciju između računara i Arduina na dva načina: preko usb kabela i preko bežične bluetooth veze. Oba načina komunikacije su serijskog tipa.

Serijski tip komunikacije se odnosi na prenos podataka jedan po jedan bit, odnosno jedan bit iza drugog, po čemu je i dobila ime.

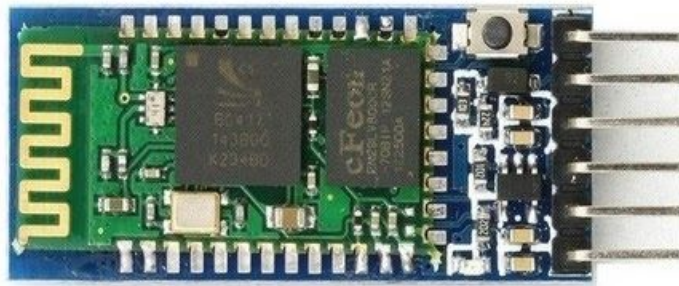
#### 4.7.1. USB komunikacija

Najčešći vid komunikacije Arduina sa računarom je preko USB kabela koji je u osnovi serijska komunikacija. Svaki put kad se upisuje programiran kod u Arduino, bitovi se preko USB kabela prenose jedan po jedan i smešta se u glavni čip (ATmega328). Na Arduino ploči postoje led indikatori Tx i Rx koji će treperiti ukoliko se koristi serijska komunikacija. Rx treperi kada Arduino prima podatke, dok Tx treperi kada Arduino šalje podatke.

#### 4.7.2. Bluetooth komunikacija

Drugi vid komunikacije između računara i Arduina jeste preko Bluetooth konekcije, koja će dati mogućnost da CNC mašina bude potpuno mobilna (bežična) ukoliko joj za napajanje obezbedmo odgovarajuću bateriju.

U ovom radu se koristi SPP (Serial Port Protocol) Bluetooth modul HC-05, koji se može podesiti da bude master ali i da bude slave uređaj.



*Slika 4.9. - Fizički prikaz HC-05.*

Specifikacije HC-05:

- Frekvencija 2.4GHz
- Modulacija: GFSK (Gaussian Frequency Shift Keying)
- Domet: do 10 metara
- Brzina: oko 1Mbps
- Radni napon: 3.3V ili 5 V
- Potrošnja struje: 50mA
- Dimenzije: 26.9mm x 13mm x 2.2 mm

Poseduje 6 izlaznih priključaka:

- Key/EN: Koristi se za dovođenje modula u režim AT komandi. U osnovi, ovaj priključak radi u režimu podataka (data mode). Key/EN priključak mora da bude visok da bi Bluetooth radio u komandnom režimu. Podrazumevani baud rate u komandnom režimu je 38400bps, dok je podrazumevani baud rate za režim podataka 9600bps.

- VCC: 5 ili 3.3 V
- GND: Uzemljenje
- TXD: Transmit Serial Data. Za ovaj priključak se vezuje izlaz RX od Arduina. Podaci sa računara stižu do HC-05 pa se dalje preko TXD isporučuju u Arduino.
- RXD: Recieve Serial Data. Za ovaj priključak se vezuje TX sa Arduina. Preko ovog priključka se podaci sa Arduina isporučuju u na HC-05 koji dalje prosleđuje podatke računaru.

## 4.8. Napajanje

Potrebno je obezbediti odgovarajuć napon za: upravljački deo, Bluetooth i za pogon motora. Moderne Arduino ploče omogućavaju da imaju više od jednog izvora napajanja. Arduino je u mogućnosti da se napaja sa USB kablom koji je povezan na kompjuter ili preko spoljnog izvora napajanja.

Ukoliko napajamo Arduino preko USB kabela struja će se isporučiti na ploču od 500mA na 5V ukoliko je računar prepoznao uređaj. Ukoliko računar nije prepoznao uređaj struja isporučena na ploču će biti 100mA na 5V.

Arduino možemo da napajamo i preko spoljnog konektora na Arduinu. Na taj konektor možemo da primenimo napon između 7V i 12V preko adaptera (pretvarač iz naizmenične u jednosmernu struju). U ovom slučaju na Arduino se isporučuje između 7V i 12V, na ploči se nalazi naponski regulator koji će ovaj napon da spusti na 5V, što je operativni napon Arduino ploče.

Moguće je Arduino napajati sa 5V ukoliko bismo iskoristili priključke Vin i GND. Vin priključak zaobilazi naponski regulator i sve bezbednosne mere koje su prisutne na Arduino, zbog toga je potrebno da napon od 5V bude stabilan i regulisan, najviši napon koji je dozvoljeno primeniti na Vin priključak je 5.5V, u suprotnom ploča može da se trajno ošteti.

Integrisano kolo L293D ima dva priključka za napajanje Vcc1 i Vcc2. Za pokretanje integrisanog logičkog kola koristi se Vcc1, njemu kao i Arduinu je potrebno 5V. Drugim priključkom, Vcc2 služi za napajanje motora, koji za L293D može da bude od 0V do 36V, u zavisnosti koliki napon je potreban za motor.

Motore koje ćemo u ovom radu koristiti napajaju se sa 5V, odnosno svaka komponenta koju ćemo ugraditi u CNC uređaj se napaja sa 5V.

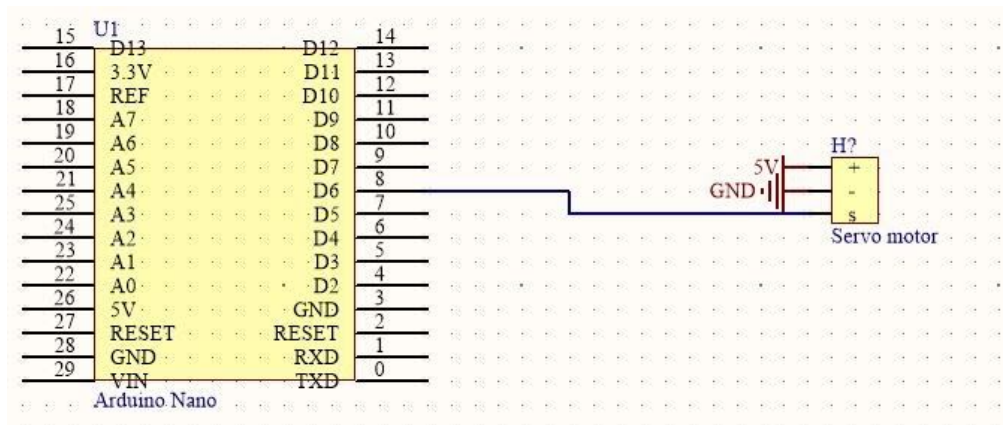
Bez obzira, koliko god deluje primamljivo da preko USB kabela napajamo Arduino i iz njega napajamo L293D na koji su priključeni motori, ipak nije nikako preporučljivo. Iz razloga zato što Arduino na izlaznom priključku za napajanje ima ograničenu struju koju može da isporuči, koja iznosi oko 200-300mA. Motori su u ovom radu najveći potrošači, i njihova potrošnja struje uglavnom nije konstantna. U zavisnosti od osovinskog opterećenja menja se i potrošnja struje. Što je motor više opterećen i potrošnja struje je veća.

Obezbedićemo stabilan i regulisan napon od 5V, te ćemo ga dovesti na priključke Vin i GND na Arduinu. Takođe, tih 5V ćemo dovesti i na L293D na priključke Vcc1 (napajanje integrisanog kola) i Vcc2 (napajanje motora). Na ovaj način ćemo rešiti napajanje.







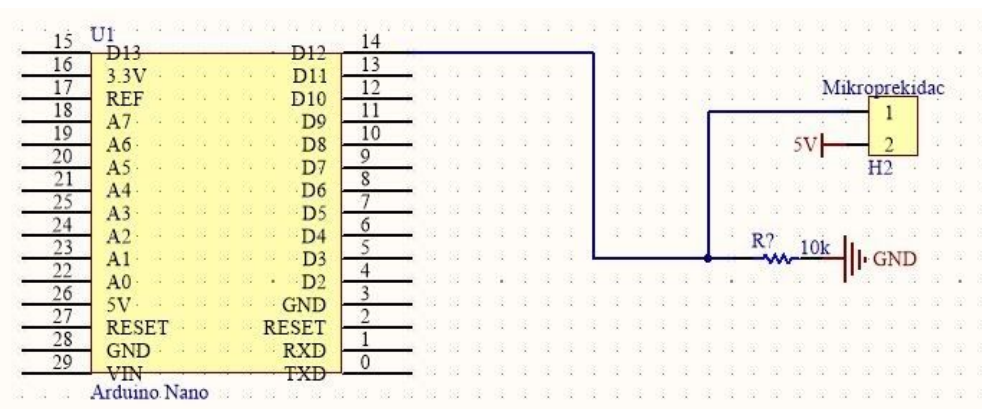


*Slika 5.2. - Povezivanje servomotora.*

Servo motor ima 3 izlazna priključka: VCC koji se povezuje na 5V, GND, i kontrolni signal koji se povezuje direktno na Arduino. Arduino servomotoru zadaje fizički pomeraj za tačno određeni ugao.

### 5.3. Povezivanje mikroprekidača

Izlazi mikroprekidača koji se koriste u ovom radu su u kratkom spoju sve dok se ne pritisne dugme na njima, tada oni prestaju da provode struju. Poželjno je da u ovom delu kola koristimo pull-down ili pull-up otpornik, kako bismo se osigurali od neželjenih signala koji se stvaraju prilikom šumova i interferencije signala. Upotrebom ovih otpornika svi neželjeni signali će otići u otpornik, te na taj način ostvarujemo preciznu detekciju LOW i HIGH logičkog stanja.



*Slika 5.3. - Povezivanje mikroprekidača..*

Kada je mikroprekidač otvoren (OFF stanje) digitalni ulaz Arduina je povezan sa GND preko pull-down otpornika, pa na taj način Arduino očitava logičko stanje LOW. Kada je mikroprekidač zatvoren (ON stanje) digitalni ulaz Arduina će očitavati logičko stanje HIGH.

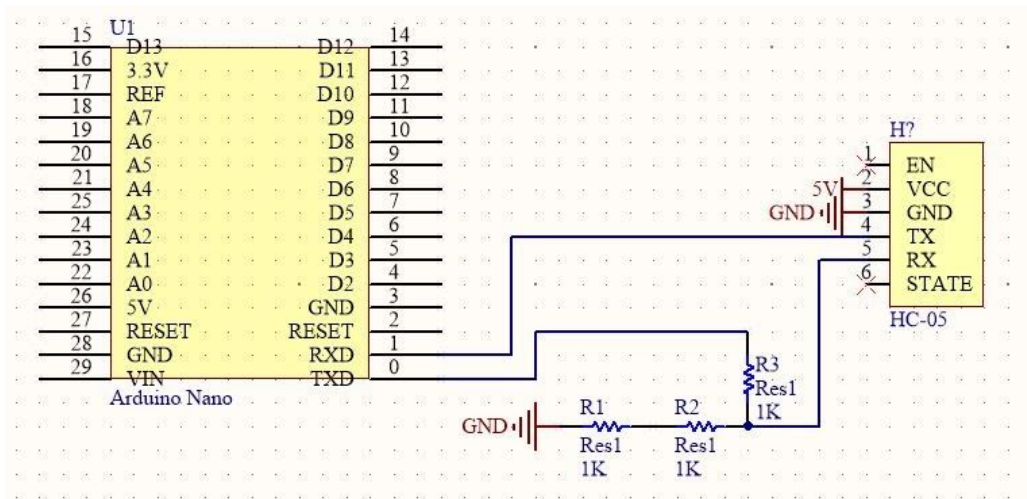
S obzirom da mikroprekidači nisu polarizovani, nije bitno je na koji će se priključak dovesti 5V a na koji pull-down otpornik.

## 5.4. Povezivanje HC-05

Bluetooth HC-05 modul ima 6 izlaznih priključaka STATUS, RX, TX, GND, VCC, EN. Radni napon modula je u opsegu od 3.6V do 6V.

Informacije koje bluetooth na računar se šalju na HC-05, dalje se prenose preko TX (HC-05) priključka putem serijske veze na RXD (Arduino) priključak. Logički napon TX (HC-05) priključka zahteva 5V za HIGH stanje i 0V za LOW stanje.

Informacije koje se šalju sa Arduinoa na HC-05 idu preko priključka TXD (Arduino) na RX (HC-05). Logički napon RX (HC-05) je 3.3V za HIGH stanje i 0V za LOW stanje. Izlazni priključak na Arduinou TXD daje 5V, a za prijem podataka na RX (HC-05) je potrebno 3.3V, zato se između postavlja naponski razdelnik, koji će se upotrebom odgovarajućih otpornika smanjiti napon sa 5V na 3.3V.



Slika 5.4. - Povezivanje mikroprekidača.

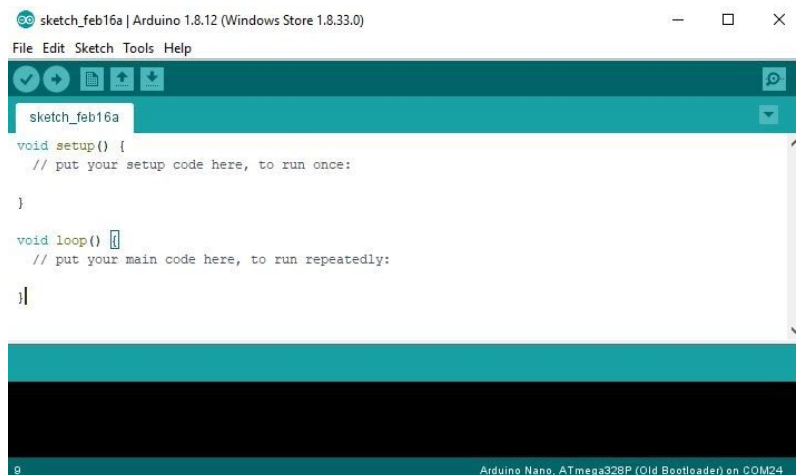
$$R_{12} = R_1 + R_2 = 2K$$

$$V_{out} = R_2 / (R_{12} + R_3) * V_{in} = (2k / 2k + 1k) * V_{in} = 3.33V$$

## 6. SOFTVERSKI ALATI

### 6.1. Arduino IDE

Arduino IDE je besplatan softver otvorenog koda (open source), vrlo je lak za upotrebu i omogućava upisivanje koda na ploču. Radi na Windows, Mac OS i Linux okruženju. Napisan je u Javi i zasnovan je na Processing-u. Arduino IDE se može koristiti za bilo koju Arduino ploču.



*Slika 6.1. - Arduino IDE.*

## 6.2. G-kod

G-kod je zajednički naziv i najčešće korišćen programski jezik za numeričke kontrole. Najčešće se koristi za kontrolu automatizovanih mašina. To je jezik preko kojeg ljudi govore kompjuterizovanim mašinskim alatima da nešto naprave. Tim jezikom se ukazuju instrukcije kao što su: gde motor treba da se pomeri, koliko brzo da se kreće i kojom putanjom da se kreće.

Pod programiranjem CNC mašine se podrazumeva opisivanje geometrijskih informacija po dekartovom koordinatnom sistemu. Te informacije su nam potrebne kako bi CNC mašina znala po kojim tačkama treba da se kreće i šta treba da uradi na određenoj tački položaja. Programiranje CNC mašine se može obaviti ručno ili automatizovano preko kompjutera.

### 6.2.1. Ručno programiranje

Podrazumeva ručno pisanje G-koda prema prethodno definisanom nacrtu. Kod ručnog programiranja je moguće koristiti bilo koji editor teksta. Svaku poziciju preko koje bi glava CNC mašina morala da predje potrebno ju je najpre izračunati i ručno upisati u editor teksta u skladu sa G-kodovom terminologijom. Programer koji programira na ovaj način praktično vodi glavu CNC mašine od tačke do tačke. Ručno programiranje je najstariji i tehnološki najniži nivo programiranja NC i CNC mašina.

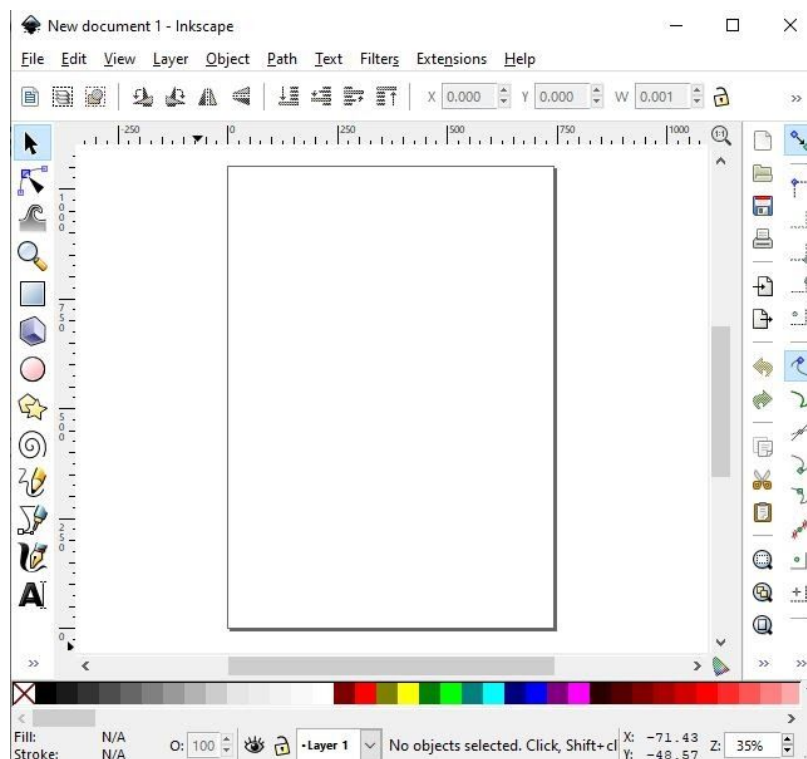
### 6.2.2. Automatsko programiranje

Automatizovano programiranje se odnosi na CAM softverske alate koji automatski generišu G-kod prema prethodno dizajniranom nacrtu. Da bi se postigao automatski generisan G-kod potrebna nam je programska podrška, odnosno softverski alat koji je u mogućnosti da recimo neki vektorski nacrt pretvori u G-kod kako bi glava CNC mašine znalala po kojim tačkama

treba da se kreće i šta na kojoj tački treba da uradi. Danas se gotovo svaki nacrt za CNC obradu programira automatski, odnosno preko CAM softverskog alata.

### 6.3. Inkscape

Inkscape profesionalni alat za dizajn vektorske grafike koji radi na Windows, Mac OS i Linux okruženju. Koriste ga profesionalni dizajneri, hobisti, istraživači... za kreiranje grafike kao što su ilustracije, ikone, logo-e, veb grafika... Potpuno je besplatan i softver otvorenog koda (open source). Grafički prikaz Inkscape prikazan je na sledećoj slici:



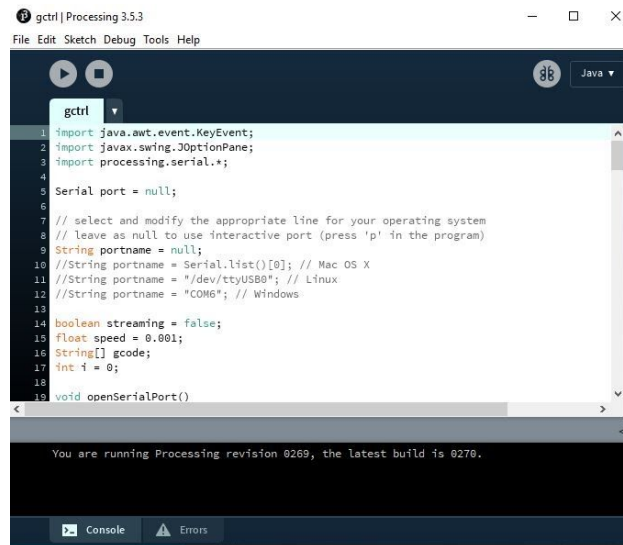
*Slika 6.2. - Grafički prikaz Inkscape.*

U ovom radu ga koristimo kako bismo preko njega nacrtali željeni dizajn koji želimo da CNC mašina iscrta. Potrebno je da sa interneta preuzmemo Inkscape verziju 0.48.5 kako bi mogli da unutar Inkscape instaliramo dodatak (Add-on) za dobijanje G-koda odgovarajućeg crteža nacrtanog u Inkscape-u. Taj dodatak se zove MakerBot Unicorn G-Code, koja se više ne razvija i zbog toga je bitno instalirati na računar Inkscape verziju 0.48.5.

Instalacija MakerBot Unicorn G-Code dodatka se odvija na sledeći način. Nakon što je na računar instalirana odgovarajuća verzija Inkscape-a, potrebno je najpre raspakovati inkscape-unicorn-master.zip, zatim otići na /src direktorijum i kopirati sadržaj u C:\Program Files\Inkscape\share\extensions.

## 6.4. Processing

Processing je fleksibilan softverski blok za crtanje (sketchbook) i programski jezik za učenje kako se programira u kontekstu vizualne umetnosti. Desetine hiljada studenata, hobista, umetnika, dizajnera... koristi Processing za učenje i za razvoj prototipova. Processing je potpuno besplatan i otvorenog koda (open source), dostupan je za Windows, Linux, Mac OS, Android i ARM okruženjima.



*Slika 6.3. - Grafički prikaz Processing-a..*

GTCRL je aplikacija napisana u Processing-u preko koje se korisnički pristupa i zadaju komande CNC mašini, to je korisnička spona između računara i CNC mašine. Preko ove aplikacije korisnik učitava G-kod koji se dalje putem serijske komunikacije (Bluetooth ili USB kabela) šalje na Arduino. Korisnik takođe može i da zaustavi mašinu u bilo kom trenutku. Važno je izabrati odgovarajuć serijski port na kojem je priključen Arduino, u suprotnom računar i Arduino neće imati komunikaciju.

## 7. SOFTVERSKA STRUKTURA UREĐAJA

### 7.1. Predprocesorske direktive

Za ispravan rad koračnog i servo motora koriste se Servo.h i Stepper.h biblioteke, koje se importuju na sledeći način.

```
#include <Servo.h>
#include <Stepper.h>
```

*Servo.h* biblioteka ima ugrađene funkcije kao što su:

- attach() - dodela promenljive priključka na koju je nakačen servomotor.
- write() - zadati servomotoru ugaoni pomeraj pod određenim uglom.

- `writeMicroseconds()` - zadaje se po standardu vrednost uS, koja će napraviti ugaoni pomeraj pod određenim uglom.
- `read()` - očitati trenutni ugao servomotora.
- `attached()` - proverava da li je servomotoru dodeljena promenljiva za priključak
- `detach()` - skida promenljivu priključka servomotora

*Stepper.h* biblioteka ima ugrađene funkcije kao što su:

- `Stepper(steps, pin1, pin2, pin3, pin4)` - kreiranje nove instance *Stepper* klase gde parametri predstavljaju priključke na koje je povezan koračni motor.
- `setSpeed(rpm)` - postavka brzine koračnog motora po jedinici obrtaji-u-minuti.
- `step(steps)` - pokreće koračni motor za određeni broj koraka po brzini koja je prethodno definisana funkcijom `setSpeed(rpm)`.

Definiše se konstanta *LINE\_BUFFER\_LENGTH* i dodeljuje joj se vrednost 512, koja se ne može promeniti u toku rada programa. Broj 512 označava broj karaktera jedne linije koja je serijskom komunikacijom poslata na Arduino.

```
#define LINE_BUFFER_LENGTH 512
```

## 7.2. Globalne promenljive

Definisanje promenljivih tipa integer. Promenljive *penZUp* i *penZDown* se odnosi na vrednost za koji ugao će servomotor da se zarotira, ukoliko se zarotira za 85 stepeni olovka će se podignuti, dok će za 60 stepeni olovka da se spusti. Promenljive *mikroprekidacX* i *mikroprekidacY* se odnosi na priključke koji su povezani sa Arduino. Na priključak 7 je povezana led dioda a servomotor je povezan na priključak 6. Koračni motori izvađeni iz CD čitača će napraviti 20 koraka za jednu punu rotaciju, zbog toga je promenljivoj *stepsPerRevolution* dodeljena vrednost 20.

```
const int penZUp = 85;
const int penZDown = 60;
const int mikroprekidacX = 12;
const int mikroprekidacY = 13;
const int ledPin = 7;
const int penServoPin = 6;
const int stepsPerRevolution = 20;
```

Kreiranje objekta *penServo* za kontrolu servomotora kao i kreiranje objekata za kontrolu koračnih motora *myStepperY* i *myStepperX*, koji su povezani sa Arduino priključcima 8, 9, 10, 11 i 2, 3, 4, 5

```
Servo penServo;
Stepper myStepperX(stepsPerRevolution, 8, 9, 10, 11);
Stepper myStepperY(stepsPerRevolution, 2, 3, 4, 5);
```

Kreiranje strukture *point* u kojoj su definisane x, y i z promenljive kojima se globalno

pristupa: *actuatorPos.x*, *actuatorPos.y*, *actuatorPos.z*. Promenljive sadrže vrednost o trenutnoj poziciji koračnog i servo motora.

```
struct point {  
    float x;  
    float y;  
    float z;  
};  
struct point actuatorPos;
```

Inicijalna postavka: *StepInc* - promenljiva koja će se koristiti za otkrivanje promene na koordinatama, *StepDelay* - vrednost koja će se koristiti za vremensko kašnjenje koraka koračnog motora, *LineDelay* - vremensko kašnjenje pre slanja informacija na serijski izlaz, *penDelay* - vremensko kašnjenje pre slanja naredbi za ugaoni pomeraj servomotora. Vrednost kašnjenja odnosi se na milisekunde.

```
float StepInc = 1;  
int StepDelay = 0;  
int LineDelay = 50;  
int penDelay = 50;
```

Koračni motor kad napravi 6 koraka x i y osa će napraviti pomeraj od 1mm. Do ovog podatka se došlo eksperimentalno, pušten je koračni motor od 100 koraka da nacrtava pravu liniju, izmerena je dužina linije koja iznosi 16,5mm,  $100\text{koraka}/16.5\text{mm} = 6\text{koraka/mm}$ .

```
float StepsPerMillimeterX = 6.0;  
float StepsPerMillimeterY = 6.0;
```

Kretanje x i y ose je ograničeno, od 0mm do 40mm. Minimum predstavlja početni položaj CNC uređaja. Z osa je predstavljena preko 0 ili 1, odnosno servomotor spušta ili podiže olovku.

```
float Xmin = 0;  
float Xmax = 40;  
float Ymin = 0;  
float Ymax = 40;  
float Zmin = 0;  
float Zmax = 1;  
float Xpos = Xmin;  
float Ypos = Ymin;  
float Zpos = Zmax;
```

U toku razvoja softvera koristila se promenljiva *verbose*, kako bi na serijski izlaz prikazale dodatne informacije pri debugovanju, koje se mogu uključiti dodelom vrednosti *true*.

```
boolean verbose = false;
```



### 7.3. Inicijalizacija uređaja

Pri pokretanju Arduino aplikacije prvo se izvršava funkcija *setup()*, koja se izvršava samo jednom nakon reseta ili pri prvom pokretanju uređaja. Funkcija *setup()* ona inicijalizuje promenljive, ulazno-izlazne priključke, biblioteke, itd...

Najpre je potrebno podesiti brzinu za serijski prenos podataka. Brzina se ogleda u bitovima po sekundi, odnosno baudima.

```
Serial.begin(9600);
```

Inicijalizacija servomotora sa postavkom da bude podignut. Sledi kašnjenje od 200 milisekundi. Inicijalizacija brzine koračnog motora izražena u obrtajima u minuti. Postavka mikroprekidača kao digitalni ulaz i diode kao digitalni izlaz.

```
penServo.attach(penServoPin);  
penServo.write(penZUp);  
delay(200);  
  
myStepperX.setSpeed(250);  
myStepperY.setSpeed(250);  
  
pinMode(mikroprekidacX, INPUT);  
pinMode(mikroprekidacY, INPUT);  
pinMode(ledPin, OUTPUT);
```

Inicijalizacija obaveštenja koja se šalju na serijski izlaz.

```
Serial.println("Mini CNC Plotter alive and kicking!");  
Serial.print("X range is from ");  
Serial.print(Xmin);  
Serial.print(" to ");  
Serial.print(Xmax);  
Serial.println(" mm.");  
Serial.print("Y range is from ");  
Serial.print(Ymin);  
Serial.print(" to ");  
Serial.print(Ymax);  
Serial.println(" mm.");
```

### 7.4. Funkcije

Postavka z ose na minimalnu vrednost, odnosno osovina servomotora će se pozicionirati pod uglom od 60 stepeni, što je dovoljna vrednost kako bi bila u mogućnosti da piše.

```
void penDown() {
```



```

    penServo.write(penZDown);
    delay(LineDelay);
    Zpos=Zmin;
    if (verbose) {
        Serial.println("Pen down.");
    }
}

```

Postavka z ose na maksimalnu vrednost, odnosno osovina servomotora će se pozicionirati pod uglom od 85 stepeni, što je dovoljna vrednost kako bi se olovka odvojila od površine papira.

```

void penUp() {
    penServo.write(penZUp);
    delay(LineDelay);
    Zpos=Zmax;
    if (verbose) {
        Serial.println("Pen up!");
    }
}

```

Pozicioniranje oba koračna motora na početno stanje. Ukoliko X i Y podloga već nisu pritisnuli mikroprekidače za odgovarajuću osu, koračni motori će se pomerati za po jedan korak unazad sve dok X i Y podloga fizički ne dođu do mikroprekidača.

```

void PozicionirajNaPocetak() {
    while(1){
        if (digitalRead(mikroprekidacX) == LOW) {
            Serial.println("X-osa je pozicionirana na pocetak!");
            break;
        } else if (digitalRead(mikroprekidacX) == HIGH) {
            myStepperX.step(-1);
        } else {
            Serial.println(" X Nesto drugo je problem.");
        }
    }
    while(1){
        if (digitalRead(mikroprekidacY) == LOW) {
            Serial.println("X-osa je pozicionirana na pocetak!");
            break;
        } else if (digitalRead(mikroprekidacY) == HIGH){
            myStepperY.step(-1);
        } else {
            Serial.println(" Y Nesto drugo je problem.");
        }
    }
}

```

Funkcija `drawLine` povlači liniju na pairu nakon što se olovka spusti dole. Funkcija ima dva parametra  $x1$  i  $y1$  koji označavaju koordinate gde nacrt linije treba da bude započet. Ukoliko je koordinata izvan ograničenja radne površine funkcija će zameniti vrednost koordinate sa maksimalim ograničenjem radne površine, a zatim će koordinate da konvertuje u broj koraka koji je potreban da se povuče prvi deo linije na papiru. Promenljivama  $x0$  i  $y0$  se dodeljuje trenutna pozicija koordinati, a potom se otkriva promena koordinati.  $sx$  i  $sy$  sadrže vrednost o pravcu kretanja linije, dok se preko  $dx$  i  $dy$  izražava nagib linije, na primer ukoliko odnos  $dx/dy$  iznosi  $\frac{1}{3}$  to znači da će Y koračni motor napraviti 3 koraka dok X napravi 1 korak. Nakon što uređaj nacrtala liniju potrebno je da sačuva informaciju o trenutnoj poziciji motora kako bi uređaj znao odakle da započne crtanje sledeće linije.

```
void drawLine(float x1, float y1) {
    if (verbose)
    {
        Serial.print("fx1, fy1: ");
        Serial.print(x1);
        Serial.print(",");
        Serial.print(y1);
        Serial.println("");
    }

    if (x1 >= Xmax) {
        x1 = Xmax;
    }
    if (x1 <= Xmin) {
        x1 = Xmin;
    }
    if (y1 >= Ymax) {
        y1 = Ymax;
    }
    if (y1 <= Ymin) {
        y1 = Ymin;
    }

    if (verbose)
    {
        Serial.print("Xpos, Ypos: ");
        Serial.print(Xpos);
        Serial.print(",");
        Serial.print(Ypos);
        Serial.println("");
    }

    if (verbose)
    {
        Serial.print("x1, y1: ");
        Serial.print(x1);
        Serial.print(",");
    }
```

```

    Serial.print(y1);
    Serial.println("");
}

x1 = (int)(x1*StepsPerMillimeterX);
y1 = (int)(y1*StepsPerMillimeterY);
float x0 = Xpos;
float y0 = Ypos;

long dx = abs(x1-x0);
long dy = abs(y1-y0);
int sx = x0<x1 ? StepInc : -StepInc;
int sy = y0<y1 ? StepInc : -StepInc;

long i;
long over = 0;
if (dx > dy) {
    for (i=0; i<dx; ++i) {
        myStepperX.step(sx);
        over+=dy;
        if (over>=dx) {
            over-=dx;
            myStepperY.step(sy);
        }
        delay(StepDelay);
    }
}
else {
    for (i=0; i<dy; ++i) {
        myStepperY.step(sy);
        over+=dx;
        if (over>=dy) {
            over-=dy;
            myStepperX.step(sx);
        }
        delay(StepDelay);
    }
}

if (verbose)
{
    Serial.print("dx, dy:");
    Serial.print(dx);
    Serial.print(",");
    Serial.print(dy);
    Serial.println("");
}

```

```

if (verbose)
{
    Serial.print("Going to ");
    Serial.print(x0);
    Serial.print(",");
    Serial.print(y0);
    Serial.println("");
}

delay(LineDelay);
Xpos = x1;
Ypos = y1;
}

```

Funkcijom *processIncomingLine* se u Arduinou očitavaju naredbe poslate sa GCTRL Processing aplikacije kako bi se prepoznale odgovarajuće naredbe G-koda kao što su G ili M komande po jednoj poslatoj liniji. Arduino će da protumači naredbu, te će u skladu sa naredbom tako i odreagovati koračni ili servo motor. Funkciji se prosleđuju dva parametra kao što su: linija komandi koja je poslata sa GCTRL tipa *char*, i broj karaktera po poslatoj liniji tipa *int*. Pretpostavlja se da naredbe koje se šalju neće biti veći od 64 karaktera, shodno tome je deklarirana promenljiva *buffer*. Sve dok je trenutni broj karaktera sa linije manji od ukupnog ispitivaće se da li je pristigla jedna od sledećih komandi: U - podizanje olovke (potrebno samo za grbl kontroler), D - spustanje olovke (potrebno samo za grbl kontroler), G - tumačenje G instrukcija, M - tumačenje M instrukcija. Processing GCTRL ne šalje na serijski ulaz u arduino U i D komande, one će se koristiti ukoliko se za slanje na serijski ulaz koristi grbl kontroler. Najpre se kroz petlju *while (isdigit(line[currentIndex]))* ispituje da li je pristigla numerička vrednost sa linije, i ukoliko jeste indeksi od tih numeričkih vrednosti se skladište u niz *buffer*. Te numeričke vrednosti su važne kako bi se moglo tačno zaključiti koja je tačno G ili M komanda (na primer, *G90* ili *M114*). Ukoliko je pristigla *G90* komanda, to znači da CNC uređaj započinje crtanje, te će uređaj preko funkcije *PozicionirajNaPocetak()* proveriti preko mikroprekidača da li su X i Y osa pozicionirani na početak, ukoliko nisu funkcija će ih pozicionirati. Ukoliko je pristigla *G1* komanda, znači da uređaj treba da povuče liniju koja je jedan deo celokupnog crteža. Komanda M sadrži instrukcije za spuštanju (*M300 S30*) i podizanju olovke (*M300 S50*), kao i izveštaj o pozicioniranju koordinata (*M114*). Ukoliko Arduino nije u mogućnosti da protumači M komandu, poslaće poruku na serijski izlaz da tačno određena komanda G-koda nije uspela da se protumači.

```

void processIncomingLine( char* line, int charNB ) {
    int currentIndex = 0;
    char buffer[ 64 ];
    struct point newPos;

    newPos.x = 0.0;
    newPos.y = 0.0;
}

```

```

while( currentIndex < charNB ) {
    int currentIndexSave = currentIndex;
    currentIndex++;
    int nbuffer=0;
    while (isdigit(line[currentIndex])){
        buffer[nbuffer] = line[ currentIndex ];
        currentIndex++;
        nbuffer++;
    }
    buffer[nbuffer] = '\\0';
    currentIndex = currentIndexSave;
    switch ( line[ currentIndex++ ] ) {
        case 'U':
            penUp();
            break;
        case 'D':
            penDown();
            break;
        case 'G':
            switch ( atoi( buffer ) ){
                case 0:
                case 90: //G90
                    //Serial.println(buffer);
                    PozicionirajNaPocetak();
                case 1:
                    char* indexX = strchr( line+currentIndex, 'X' );
                    char* indexY = strchr( line+currentIndex, 'Y' );
                    if ( indexY <= 0 ) {
                        newPos.x = atof( indexX + 1);
                        newPos.y = actuatorPos.y;
                    }
                    else if ( indexX <= 0 ) {
                        newPos.y = atof( indexY + 1);
                        newPos.x = actuatorPos.x;
                    }
                    else {
                        newPos.y = atof( indexY + 1);
                        indexY = '\\0';
                        newPos.x = atof( indexX + 1);
                    }
                    drawLine(newPos.x, newPos.y );
                    actuatorPos.x = newPos.x;
                    actuatorPos.y = newPos.y;
                    break;
            }
            break;
        case 'M':
            switch ( atoi( buffer ) ){

```

```

    case 300:
    {
        char* indexS = strchr( line+currentIndex, 'S' );
        float Spos = atof( indexS + 1);
        // Serial.println("ok");
        if (Spos == 30) {
            Serial.println("Pen DOWN");
            penDown();
        }
        if (Spos == 50) {
            Serial.println("Pen UP");
            penUp();
        }
        break;
    }
    case 114: // M114
        Serial.print( "Absolute position : X = " );
        Serial.print( actuatorPos.x );
        Serial.print( " - Y = " );
        Serial.println( actuatorPos.y );
        break;
    default:
        Serial.print( "Command not recognized : M");
        Serial.println( buffer );
    }
}
}
}

```

## 7.5. Glavni program

U telu funkcije *void loop()* nalaze se naredbe koje se beskonačno ponavljaju nakon što se izvrši inicijalizacija uređaja. Najpre se definišu promenljive glavnog programa a potom arduino ulazi u beskonačno izvršavanje petlje *while (1)*. Konstantno se očitava da li je linija, odnosno poruka pristigla na serijski ulaz, i ukoliko jeste uređaj će da odreaguje u skladu sa pristiglom porukom. Parametri *'\n'* i *'\r'* označavaju jednu potpunu liniju, odnosno poruku koja se šalje funkciji *processIncomingLine(line, lineIndex)* na tumačenje. Pre nego što se poruka pošalje na tumačenje potrebno joj je dodati *'\0'* kako bi funkcija znala da je to kraj poruke. Logičke promenljive *lineIsComment* i *lineSemiColon* se koriste za komentare G-koda koje je potrebno ignorisati. Ukoliko pristigla poruka sadrži broj karaktera više od dozvoljenog (512) na konzoli će se prikazati greška *"ERROR - lineBuffer overflow"*. Ukoliko su karakteri pristigle poruke malim slovima, uradiće se konverzija u velika slova, iz razloga što su instrukcije G-koda naznačene striktno velikim slovima. Ukoliko u toku rada uređaja na GCTRL Processing aplikaciji pritisnemo taster X to znači da želimo prinudno da zaustavimo trenutno plotovanje, tada GCTRL šalje komandu *'Q'*, na koju će Arduino odreagovati sopstvenim resetom, kako bi sve promenljive postavio na inicijalne vrednosti.

```

void loop()
{
    delay(200);
    char line[ LINE_BUFFER_LENGTH ];
    char c;
    String str;
    int lineIndex;
    bool lineIsComment, lineSemiColon;
    lineIndex = 0;
    lineSemiColon = false;
    lineIsComment = false;

    while (1) {
        while ( Serial.available()>0 ) {
            c = Serial.read();

            if (( c == '\n') || (c == '\r') ) {
                if ( lineIndex > 0 ) {
                    line[ lineIndex ] = '\0';
                    if (verbose) {
                        Serial.print( "Received : ");
                        Serial.println( line );
                    }
                    processIncomingLine( line, lineIndex );
                    lineIndex = 0;
                }
                else {
                    // Empty or comment line. Skip block.
                }
                lineIsComment = false;
                lineSemiColon = false;
                Serial.println("ok");
            }
            else {
                if ( (lineIsComment) || (lineSemiColon) ) {
                    if ( c == ')' ) lineIsComment = false;
                }
                else {
                    if ( c <= ' ' ) {
                    }
                    else if ( c == '/' ) {
                    }
                    else if ( c == '(' ) {
                        lineIsComment = true;
                    }
                    else if ( c == ';' ) {
                        lineSemiColon = true;
                    }
                }
            }
        }
    }
}

```

```

else if ( lineIndex >= LINE_BUFFER_LENGTH-1 ) {
    Serial.println( "ERROR - lineBuffer overflow" );
    lineIsComment = false;
    lineSemiColon = false;
}
else if ( c >= 'a' && c <= 'z' ) {
    line[ lineIndex++ ] = c-'a'+'A';
}
else {
    line[ lineIndex++ ] = c;
}
}

if (c == 'Q') {
    Serial.println("RESET Q");
    delay(500);
    resetFunc();
}
}
}
}

```

## 8. SASTAVLJANJE UREĐAJA

Najpre se sastavlja mehanička konstrukcija a potom se na mehaničku konstrukciju postavlja PCB električna ploča prema odgovarajućoj električnoj šemi i sve neophodne aktuatore će se preko konektora povezati sa PCB pločom.

### 8.1. Sasavljanje konstrukcije

Za sastavljanje konstrukcije potrebno je nešto od alata i materijala kao što su:

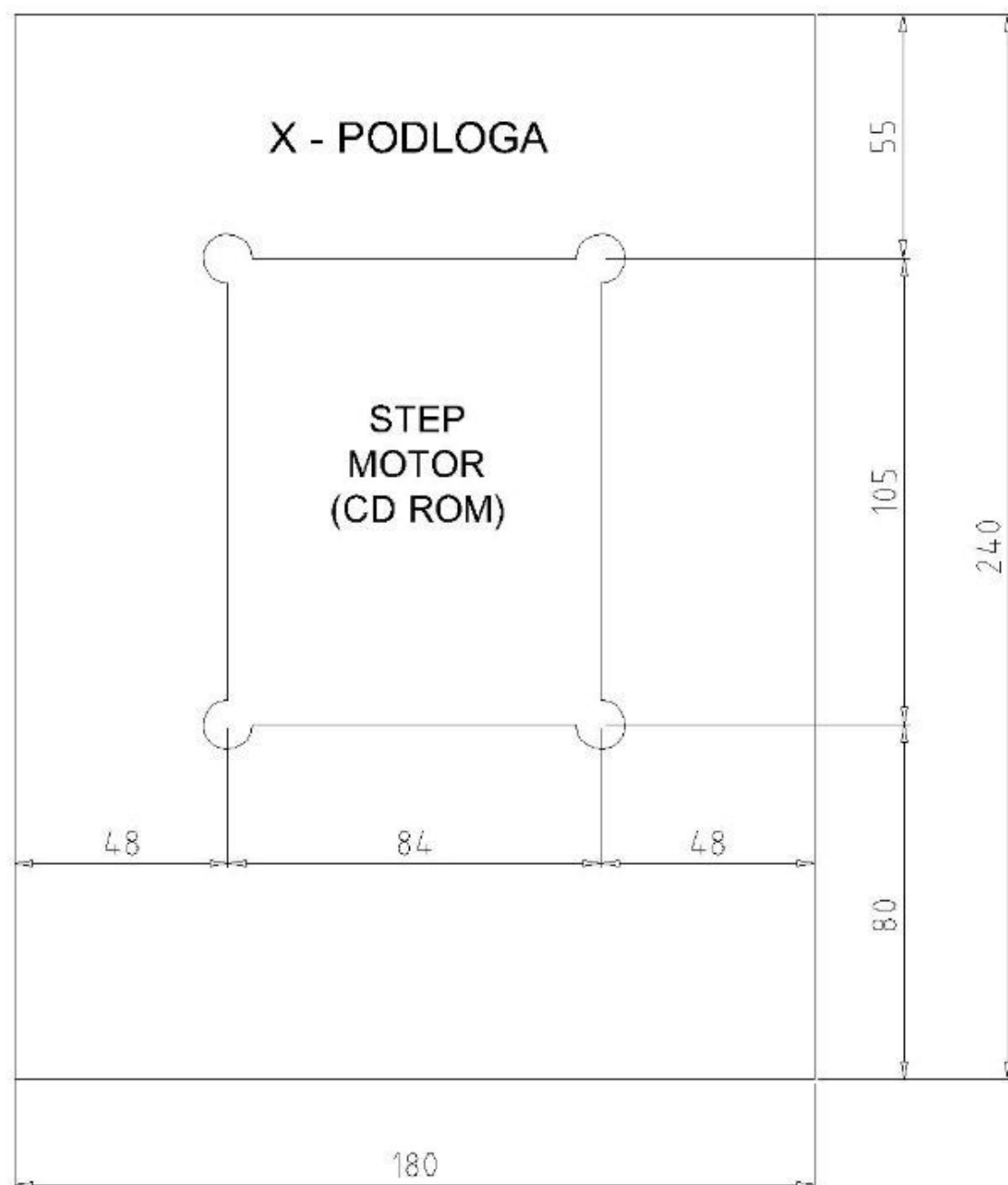
- Ubodna testera, koristi se za rezanje podloge (iverica)
- Bušilica, koristi se za bušenje neophodnih rupa
- 2x CD čitača
- Lemilica i žica za lemljenje
- Unimer, koristi se za ispitivanje koračnih motora izvađenih iz CD čitača
- Odstojnici, služe da se napravi fizičko rastojanje između podloge i koračnog motora
- L profil od 90 stepeni, služi za sastavljanje X i Y podloge u jednu celinu
- Šrafovi
- Odvijač sa kojim se pričvršćuju koračni motori
- Plastični pištolj sa kojim se pričvršćuju Z podloga i mikroprekidači

Potrebno je da CD čitače rasklopiti i izvaditi iz njih koračne motore sve zajedno sa šinama. Mehanizam koračnih motora zajedno sa šinama je važan kako bi se osa kretala pravolinijski.

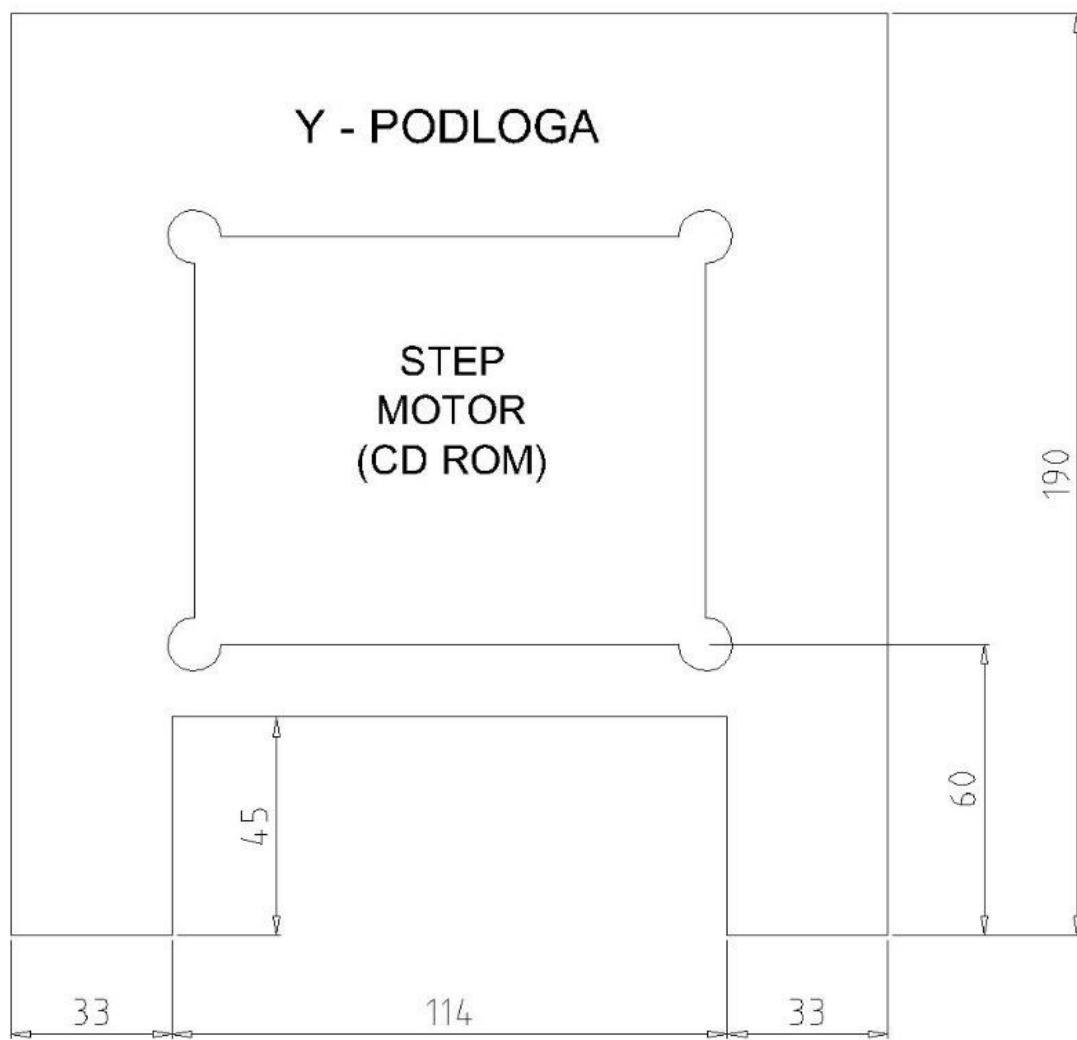


## 8.2. Sasavljanje X i Y - ose

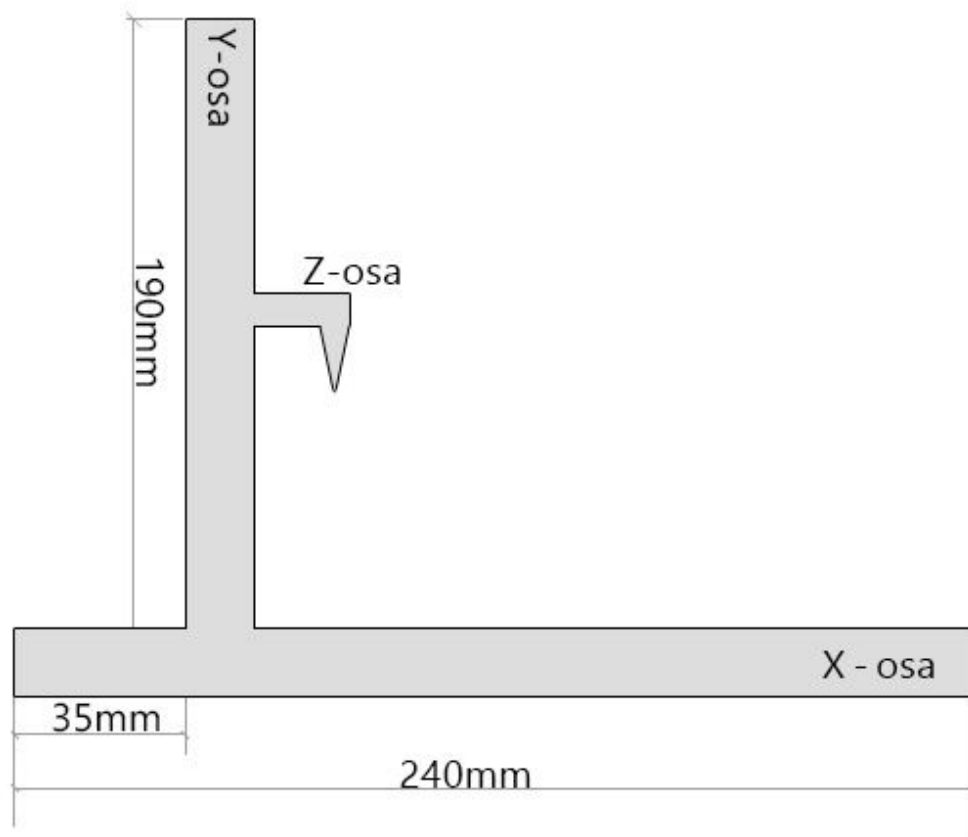
Prema sledećim nacrtima su jasno definisane dimenzije po kojima se sastavljaju X i Y podloga.



*Slika 8.1. - Nacrt za X - podlogu.*



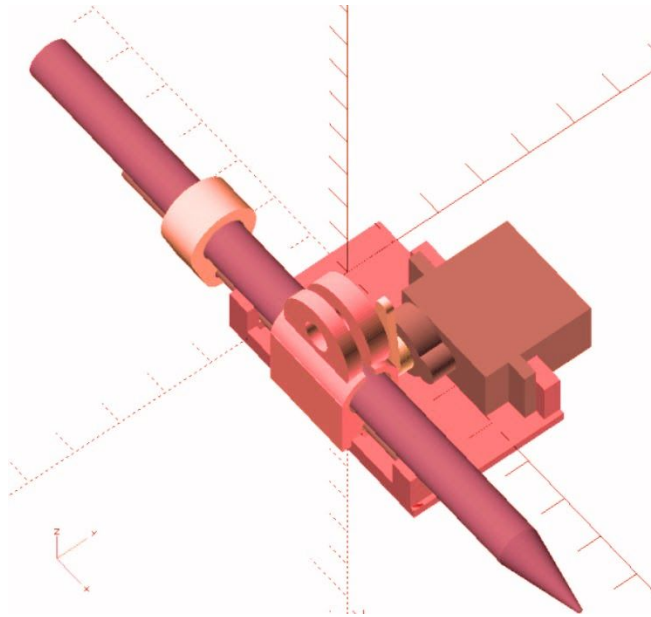
**Slika 8.2.** - Nacrt za Y - podlogu.



*Slika 8.1. - Sastavljanje X i Y podloge.*

### **8.3. Sasavljanje Z - ose**

Na github linku (u odeljku “Dodaci”) se nalaze 3D modeli mehanizma za Z osu, koji je potrebno odštampati na 3D štampaču. Ovaj mehanizam služi da preko jednog servo motora omogućimo spuštanje (kad olovka treba da piše) i podizanje olovke (kad olovka ne treba da piše.). Taj mehanizam se pričvršćuje plastičnim pištoljem za Y osu.



*Slika 8.4. - 3D model Z - ose.*

#### **8.4. Sasavljanje Elektronike**

Nakon što je izrađena PCB ploča, potrebno je zalemiti sve neophodne komponente za ploču.

Potrebno je ispitati faze koračnih motora, tako što se Unimer podesi na režim zujalice i proveriti koja su 2 od 4 izlaza koračnog motora u kratkom spoju. Ukoliko su 2 izlaza u kratkom spoju to znači da je jedna faza koračnog motora pronađena. Preostala dva izlaza su druga faza koračnog motora, ipak poželjno ih je ispitati kako bi smo bili sigurni da je motor ispravan. Ukoliko na jednom motoru se ne nađe dva kratka spoja, to znači da motor nije ispravan. Kod neispravnog motora namotaji su pregoreli, te nema kratkog spoja u fazi.

Na PCB se nalaze 4 uzastopna konektora za koračne motore, svaka faza koračnog motora se vezuje za jedan konektor. Bitno je da vezati obe faze jednog motora na prva dva uzastopna konektora ili druga dva uzastopna konektora, a obe faze drugog motora na ostatak od 4 uzastopna konektora.

Konektori za mikroprekidače na PCB ploči su igle (muški Protoboard konektori). Uloga mikroprekidača je samo da otvore ili zatvore strujno kolo, te nije od značaja redosled po kojim se vezuju.

Na konektor za napajanje se vezuje 5V.

## **9. TROŠKOVI IZRADE**

Komponenta	Količina	Cena
CD-ROM	2	100,00 rsd
Servomotor	1	150,00 rsd
3D štampa	1	1.750,00 rsd
Arduino Nano	1	250,00 rsd
HC-05	1	300,00 rsd
Otpornik	6	30,00 rsd
Konektor	7	150,00 rsd
Dip-Switch3	1	80,00 rsd
Dioda	1	10,00 rsd
L293D	2	250,00 rsd
Mikroprekidač	2	150,00 rsd
PCB izrada	5	1.000 rsd
Baterija	1	800.00rsd
Kontroler za Bateriju	1	100.00rsd

## 10. KORISNIČKO UPUTSTVO

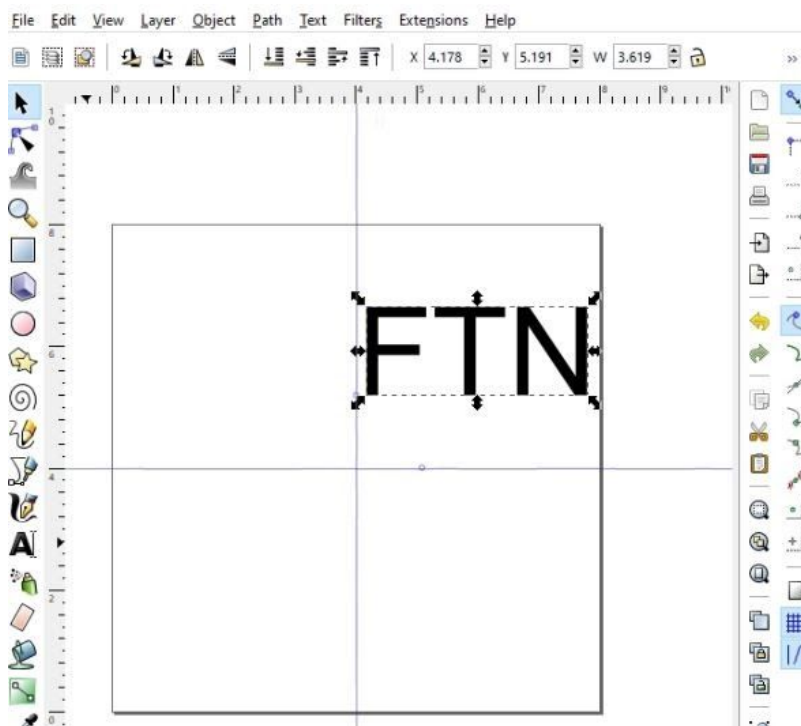
U nastavku slede instrukcije na koji način se upravlja sa CNC uređajem.

### 10.1. Grafička priprema

Pre svega, potrebno je pripremiti crtež preko Inkscape-a i pripremiti odgovarajući fajl (.gcode) koji ćemo da zadamo da se crta.

- Otvoriti *Inkscape*
- File > Document Properties
- Default Units: cm
- Units: cm
- Width: 8,00 ; Heigh: 8,00
- Podeliti dobijeni prostor na 4 jednaka kvadranta. Koristi se samo prvi kvadrant (gornji desni kvadrat), jer je to radna površina (4x4cm) po kojoj CNC može da crta.
- Proizvoljno nacrtati ili napisati bilo šta

- Path > Object to Path
- Path > Trace Bitmap
- Grays
- Scans: 2
- Update, Ok
- File Save As > Izaberi MakerBot Unicorn G-Code (\*.gcode) > Save > OK



*Slika 10.1 - Inkscape priprema.*

## 10.2. Komunikacija sa uređajem

Sa uređajem se može povezati na dva načina: USB kabelom ili preko Bluetootha. Bitno je napomenuti da nije moguće istovremeno se povezati na oba načina. Ukoliko želimo preko kabela da budemo povezani potrebno je isključiti prekidač 1 i 2 na ploči. Ukoliko želimo da se komunikacija obavlja preko Bluetootha potrebno je uključiti prekidač 1 i 2.

Ukoliko se povezujemo preko kabela potrebno je proveriti preko kog porta je povezan Arduino u Device Manager-u na Windowsu.

Ukoliko se povezujemo preko Bluetootha potrebno je najpre da se uparimo sa CNC mašinom. To ćemo uraditi na sledeći način:

- Otići na Bluetooth Device
- Kliknuti na Add a Device
- Nakon skeniranja izabrati “CNC Crtac”, zatim kliknuti na Next
- Uneti šifru: 1234, kliknuti OK
- Proveriti u Device Manager-u koji je port za bluetooth (na primer COM4)

- Izabrati odgovarajući port u *gctrl.pde* aplikaciji

### 10.3. Puštanje uređaja u rad

Processing GTCRL se koristi kao konrisnički interfejs između nas i CNC uređaja. Najpre je potrebno ga pokrenuti tako što će se pokrenuti fajl *gctrl.pde* a zatim kliknuti na Run. Nakon pokretanja će se otvoriti crni konzolni prozor preko kojeg se upravlja sa CNC-om. Najpre je potrebno izabrati odgovarajući port bilo da se radi preko USB kabela ili Bluetooth veze.



*Slika 10.2. - GTCRL Korisnički interfejs.*

Nakon provere odgovarajućeg porta u Device Manager-u, potrebno je pritisnuti taster *P* na tastaturi kako bi se označio odgovarajući port. Pritiskom na taster *G* otvara se prozor kako bi se označio odgovarajući G-kod (*.gcode*). Ukoliko se iz bilo kog razloga stvori potreba da se prinudno zaustavi uređaj to se može učiniti pritiskom na taster *X*.

## 11. ZAKLJUČAK

Oduvek su ideja i kreativnost bile početak svakog biznisa. Problemi uvek, kako svi kažu, nastaju sa realizacijom ideje. Sa CNC tehnologijom definitivno se mnogo lakše i brže dolazi od ideje do proizvoda. Upotrebom CAD/CAM alata se dizajnira sopstveni proizvod koji se pretvara u instrukcije koje CNC uređaj može da razme, što se na uređaju rezultira kao fizička kopija dizajniranog predmeta.

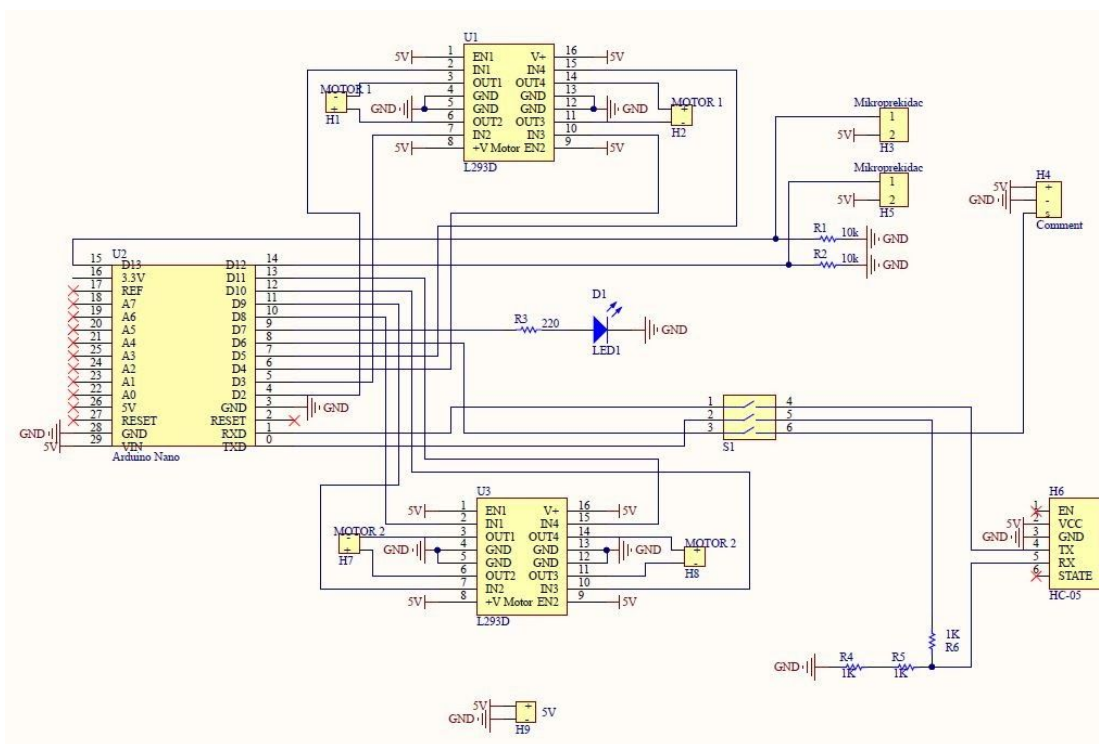
Uređaj je uspešno napravljen, te je isti tokom testiranja uspešno obavljao komunikaciju sa računarom i preko Bluetooth-a i preko USB-a. Vrlo je mali potrošač svega 700mA i napaja se preko baterije, što u bežičnom smislu uređaj čini potpunim s obzirom da se i komunikacija obavlja bežično preko Bluetooth-a, a s obzirom na njegovu veličinu lako je prenosiv. Izrada ovog uređaja se pokazala mnogo jeftinijom od mnogih komercijalno dostupnih.

Što se preciznosti tiče zadovoljavajuća je, ipak uređaj ima i svojih nedostataka:

- radna površina je na poprilično mala, 4x4 cm na kojoj se ništa ozbiljno ne može napraviti. Taj nedostatak bi mogao da se prevaziđe ukoliko bismo koristili koračne motore sa dužom pužastom osovinom ili koračne motore sa prenosnikom preko remenice i kajiša.
- koračni motori izvađenih iz CD čitača su slabi, i to direktno utiče na snagu glave alata samog uređaja, te se iz tog razloga za glavu alata ne može ništa drugo postaviti osim olovke za pisanje po papiru.

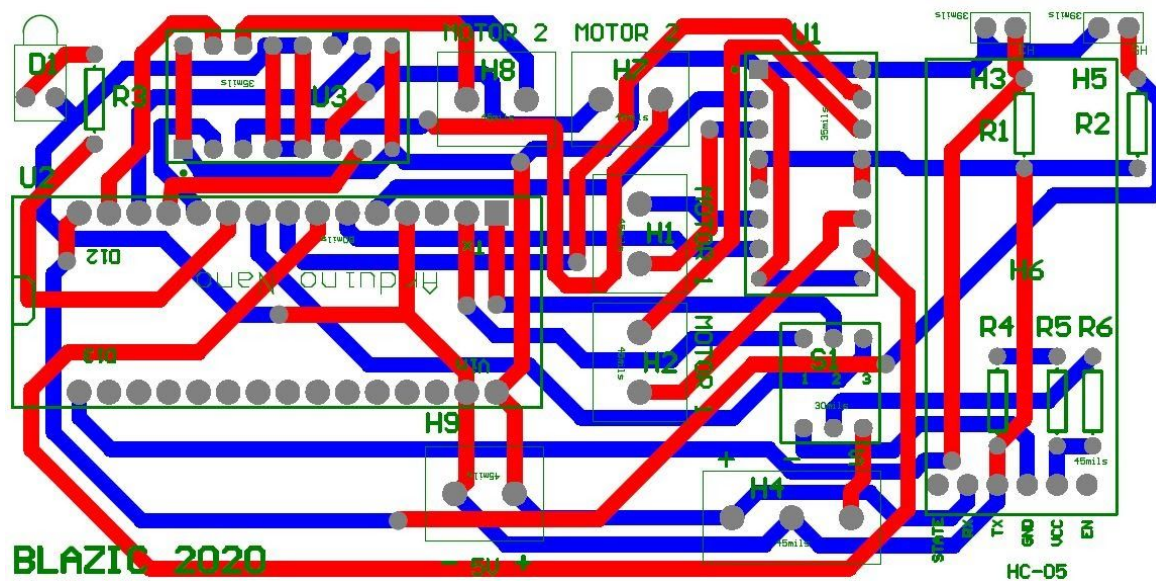
## 12. DODACI

- Kompletan razvoj projekta se nalazi na github linku i dostupan je za svakog [https://github.com/blazic016/CNC\\_ploter](https://github.com/blazic016/CNC_ploter).
- Za dizajniranje električne šeme kao i PCB ploče korišćen je *Altium Designer*.
- PCB ploča je izrađena u kini, a porudžbina napravljena preko sajta <http://jlcpcb.com>
- Dizajniranje blok šeme *Diaagram Designer*, <https://diagram-designer.en.softonic.com>
- Korisnički interfejs sa uređajem *Processing*, <https://processing.org/download>

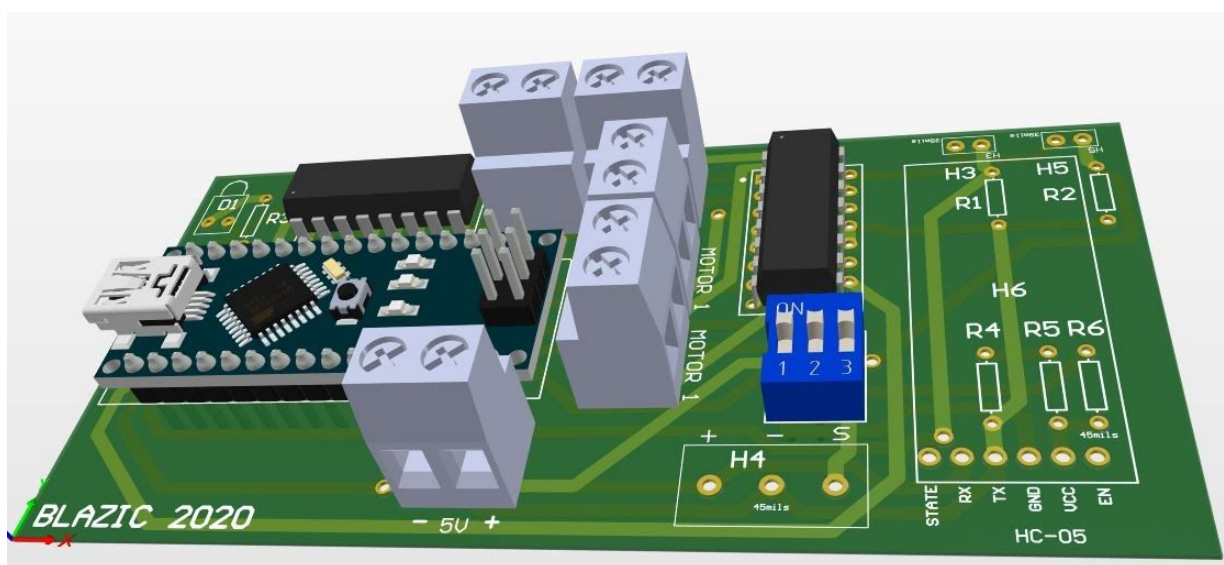


Slika 12.1. - Električna šema.

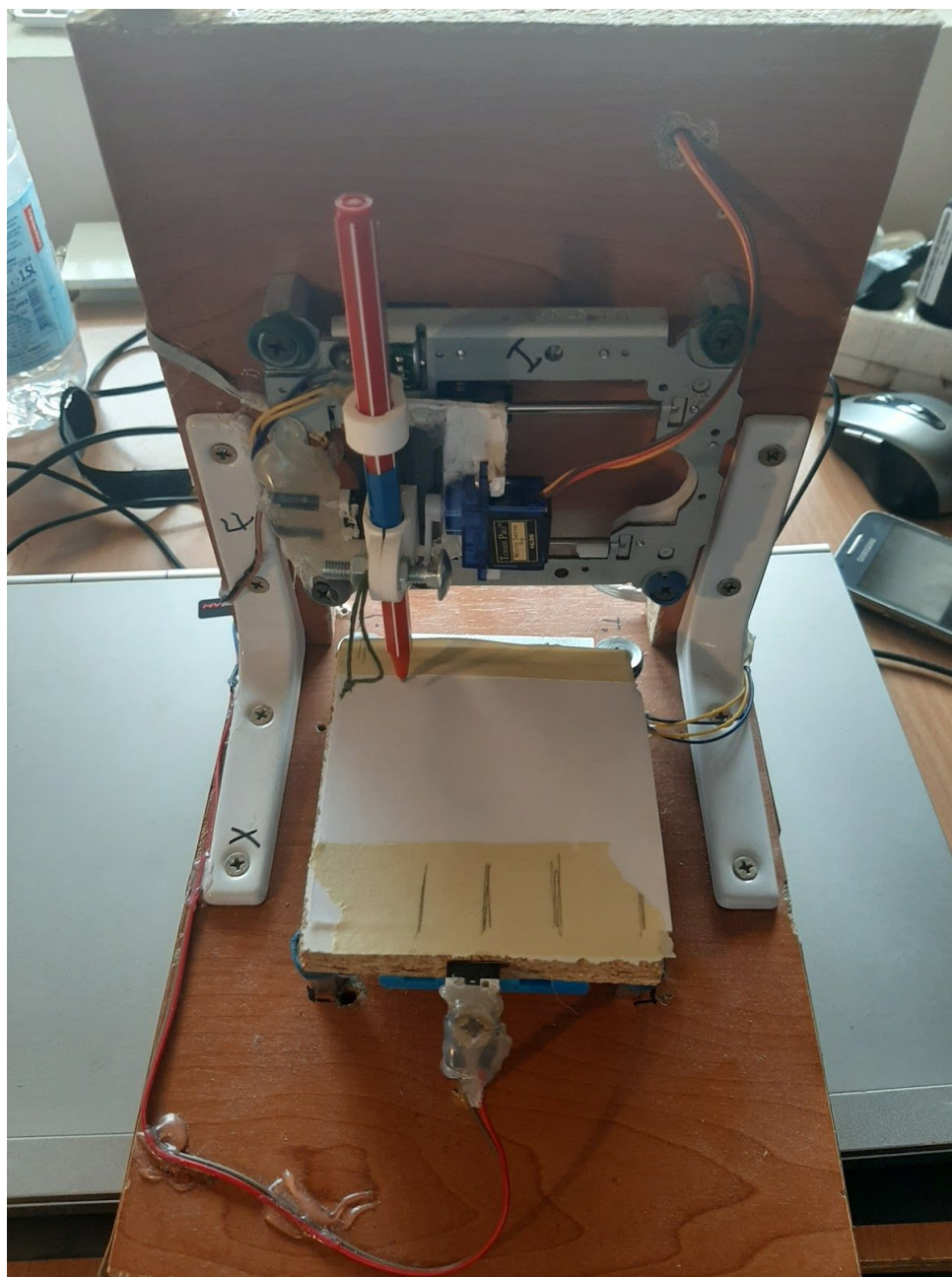




Slika 12.2. - PCB šema.



Slika 12.3. - 3D prikaz PCB-a.



*Slika 12.4. - Fizički izgled uređaja 1.*



*Slika 12.5. - Fizički izgled uređaja 2.*

## **13. LITERATURA**

- [ 1 ] Luka Mišković, *Preoblikovanje mobilne robotske platforme*, FSIB, Zagreb, 2018
- [ 2 ] T. K. Webber, *Infrared Sensors*, MCMillan, 2002.
- [ 3 ] <https://www.ardumotive.com/new-cnc-plotter.html>, 25.03.2020.

- [ 4 ] <https://www.instructables.com/id/Mini-CNC-Machine-Arduino-Based-Adafruit-Driver-Mot/>, 25.03.2020.
- [ 5 ] <https://www.instructables.com/id/How-to-Make-Mini-Home-Made-CNC-Machine-Arduino28BJ/>, 25.03.2020.
- [ 6 ] <https://www.industrija.rs/vesti/clanak/cnc-masine>, 09.04.2020.
- [ 7 ] [https://en.wikipedia.org/wiki/Cartesian\\_coordinate\\_system](https://en.wikipedia.org/wiki/Cartesian_coordinate_system), 09.04.2020.
- [ 8 ] <https://en.wikipedia.org/wiki/G-code>, 16.4.2020.
- [ 9 ] <https://www.youtube.com/watch?v=0qwrnUeSpYQ>, 01.04.2020.
- [ 10 ] <https://www.techopedia.com/definition/2063/computer-aided-design-cad>, 16.4.2020.
- [ 11 ] Tanzim Ahmed Khan, *A Micro-Controller Controlled 3 Axis CNC*, BRAC University Bangladesh, 2017
- [ 12 ] <https://www.brighthubengineering.com/manufacturing-technology/55670-what-is-numerical-control-machine/>, 17.04.2020.
- [ 13 ] Adel Messaoudi, *Implementation of Low Cost Three Axis CNC Machine with Network Control*, BRAC University BOUGARA, 2016
- [ 14 ] Barra Jabbar, *Design and Analysis Mini CNC Plotter Machine*, University of Al-Qadisiyah, 2018
- [ 15 ] Radiša Ćirković, *Programiranje CNC mašina FeatureCAM - Glodanje*, AGM Knjiga, 2016
- [ 16 ] [https://ukdiss.com/examples/computer-numerical-control.php#\\_Toc479949556](https://ukdiss.com/examples/computer-numerical-control.php#_Toc479949556), 15.04.2020.
- [ 17 ] Mohammad Kamruzzaman Khan Prince, *Implementation of a Low-cost CNC Plotter Using Spare Parts*, IJETT, 2017
- [ 18 ] <https://dronebotworkshop.com/stepper-motors-with-arduino/>, 18.04.2020.
- [ 19 ] [https://www.technobyte.org/2016/07/power-up-the-arduino-uno/#Things\\_required\\_to\\_power\\_up\\_the\\_Arduino\\_Uno](https://www.technobyte.org/2016/07/power-up-the-arduino-uno/#Things_required_to_power_up_the_Arduino_Uno), 19.04.2020.
- [ 20 ] <https://en.wikipedia.org/wiki/Actuator>, 19.04.2020.
- [ 21 ] <https://components101.com/microcontrollers/arduino-nano>, 19.04.2020.
- [ 22 ] <https://www.ladyada.net/learn/arduino/lesson4.html>, 19.04.2020.
- [ 23 ] <https://www.arduino.cc/en/Reference/Servo>, 20.04.2020.
- [ 24 ] <https://www.arduino.cc/en/Reference/Stepper>, 20.04.2020.