

TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA

GETALDIĆEVA 4. ZAGREB

ZAVRŠNI STRUČNI RAD:
PLOTER ZA PCB PLOČICE

MENTOR: Zoran Kauzlarić

UČENIK: Demion Šavrljuga

RAZRED: 4.M

Zagreb, travanj 2021.

Sadržaj

1. Uvod.....	1
2. Koračni motori	2
2.1. O koračnim motorima	2
2.2. Načini izvedbe koračnih motora	3
2.3. Načini upravljanja koračnih motora	4
3. Mehanizmi linearnog prijenosa.....	6
3.1. Vrste mehanizama linearnog prijenosa	6
3.2. Remeni prijenos na ploteru	6
4. Upravljačka elektronika	8
4.1. Arduino i CNC „štit“	8
4.2. Napajanje i aktuatori	9
4.3. Driveri	10
4.4. Interfejs (LCD i odabiranje datoteke sa SD kartice)	12
5. Programska podrška.....	14
5.1. Firmware plotera – GRBL.....	14
5.2. Universal Gcode Sender i kako ga koristiti.....	15
5.3. Stvaranje G-koda – Inkscape.....	16
5.4. Program za način rada sa SD karticom	18
6. Zaključak.....	29
7. Literatura.....	30
8. Popis slika	31

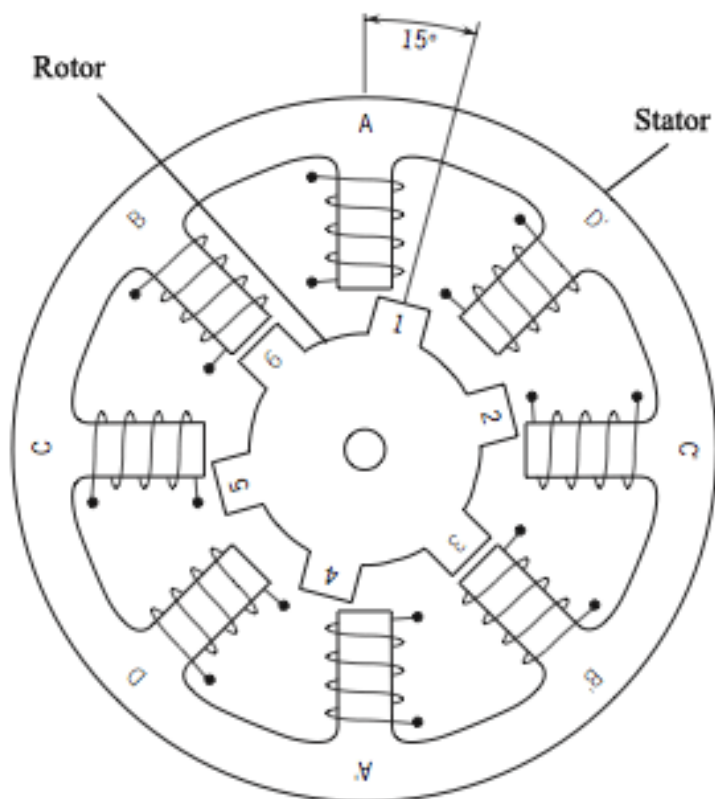
1. Uvod

Područje mehatronike je široka skupina disciplina čiji je cilj rješavanje relativno širokog spektra problema u industriji. Glavna područja koja su zastupljena u mehatronici su elektronika, mehanika i računalstvo. Zbog toga ovaj projekt je odabran kako bi pokazao svladano znanje iz tih triju disciplina. Današnje CNC mašine su jako važan dio proizvodne i metaloobradne industrije, a isto tako su mašine za koje su važna znanja iz prije navedenih triju glavnih grana mehatronike. Zato je ideja za ovaj projekt bila napraviti podvrstu CNC mašina, to jest ploter za PCB pločice. Ploter je podvrsta CNC mašina koja ima dvije analogne osi za pomicanje olovke i jednu binarnu os za dizanje i spuštanje olovke. Ploter je jako sličan i laserskim rezačima po konstrukciji te je ploter bio jeftin način za izradu CNC mašine. Za izradu i dizajn plotera potrebna su znanja o mehanici kako bi se mogla napraviti sama konstrukcija i napraviti sistem linearnih prijenosa koji omogućuju glatko kliženje te imaju veliku preciznost. Znanja iz elektronike su potrebna za izradu kontrolne elektronike te znanje iz računalstva su potrebna za rađanje programa za tu elektroniku. Ovaj ploter je namijenjen ispisivanju vodova po bakrenim pločicama flomasterom kako bi u procesu jetkanja dobili PCB pločicu, ali uz promjenu alata na glavi plotera on se može koristiti kao laserski rezač, CNC ruter ili za graviranje. U ovom dokumentu se prikazuje proces realiziranja ideje projekta te se iznose informacije koje su bile potrebne za savladavanje ovih inženjerskih zadataka.

2. Koračni motori

2.1. O koračnim motorima

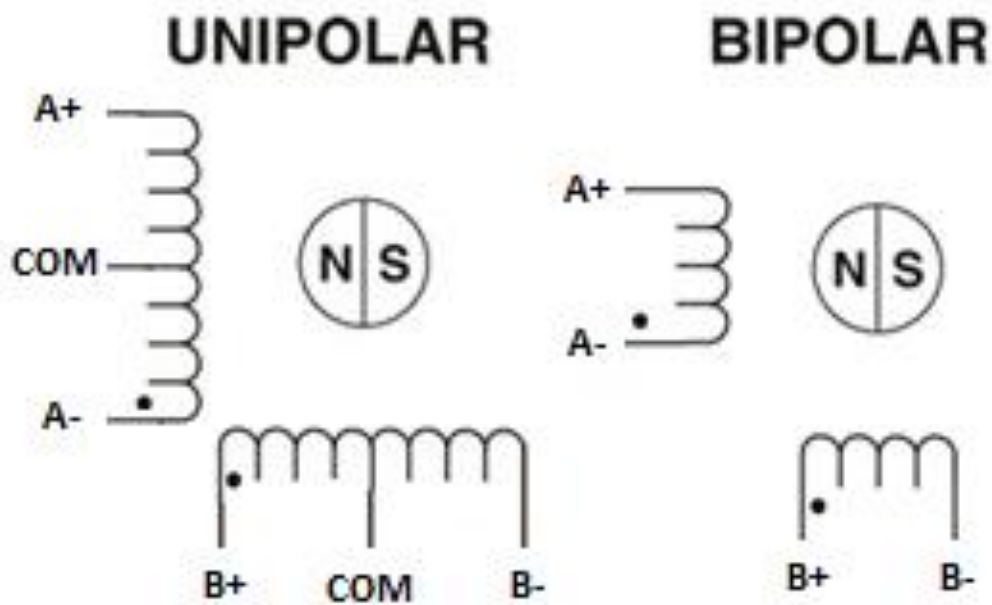
U današnje vrijeme preciznost je jedna od najključnijih stvari u većini tehničkih industrija. Robotske ruke koje se bave točnim pozicioniranjem stvari, CNC mašine koje trebaju u stotinke milimetra obraditi premete i puno drugih stvari, sve zatražuju visoki nivo preciznosti i snage svojih aktuatora za odrađivanje svojih zadataka. Zbog tih potreba visoke preciznosti i snage koje ne možemo dobiti iz tradicionalnih DC motora osmišljeni su koračni motori. Koračni motor je sinkroni DC motor bez četki te se za razliku od običnih DC motora ne vrte kontinuirano dok su samo spojeni na struju već se vrtnja njihove osovine ostvaruje izmjenom faza zavojnica kroz koje prolazi struja. Takva vrsta konstrukcije motora nam omogućava brzo zaustavljanje, mijenjanje smjera i održavanje pozicije pod velikim opterećenjem. Metalni stator koračnog motora u sebi drži elektromagnete, to su najčešće dio same konstrukcije statora (izbočine unutar statora tj. „zubi“ ovijeni zavojnicama). Zavojnice elektromagneta su grupirane po fazama, zavojnice istih faza su zajedno spojene i kroz njih teče ista struja. Rotor koračnog motora je napravljen od željeza ili trajnih magneta. Motor radi tako da se puštanjem struje kroz određene faze (elektromagnete) rotor motora poravnava s novonastalim magnetskim poljem. Brzom izmjenom tih faza s pravim redoslijedom faza možemo dobiti brzo okretanje osovine motora. Kontroliranje koračnog motora je relativno kompleksno zato jer mu je potrebno jako brzo promjenjivati tok struje na fazama. Zato se koračni motori inače kontroliraju pomoću više H-mostova ili s posebnim električnim čipovima, to jest „driver-ima“.



Slika 2.1.1. rotor i stator koračnog motora

2.2. Načini izvedbe koračnih motora

Dvije najosnovnije vrste koračnih motora su unipolarni koračni motor i bipolarni koračni motor. Razlika u njima je njihov unutarnji međusobni spoj zavojnica te zbog toga i način kontroliranja koračnog motora. Unipolarni motor ima svaku zavojnicu jednim krajem povezanu na zajednički vodič koji se inače spaja na masu (GND), ali može se i na pozitivni dio napajanja. Prednost unipolarne izvedbe koračnog motora je to što ima puno lakši način upravljanja od bipolarnog. Unipolarnom motoru dovoljno je samo staviti neku vrstu prekidača (najčešće MOSFET) na svaku zavojnicu te koristeći njih propuštati struju po redu kroz zavojnice. Na drugu ruku kod bipolarnih motora dvije zavojnice nisu spojene zajedno na jednim krajevima zavojnice već je svaka faza (zavojnica) zasebna. Loša strana toga je to što ova izvedba koračnog motora je teža za upravljanje, potrebno je koristiti H-mostove na svakoj zavojnici ili koristiti „drajver“ čip za koračne motore. Dobre strane bipolarnih koračnih motora su to što su puno jači za istu veličinu te su puno efikasniji, to je razlog zašto se danas u industriji najčešće koriste bipolarni koračni motori.



Slika 2.2.1. dijagram unipolarnog i
bipolarnog koračnog motora

2.3. Načini upravljanja koračnih motora

Pošto su bipolarni koračni motori najčešće korišteni u industriji i pošto ovaj ploter koristi bipolarne koračne motore u ovom paragrafu fokusirati ćemo se na upravljanje bipolarnih motora. Bipolarni motori imaju više načina upravljanja koji svaki ima svoje prednosti i mane: punim koracima, polu koracima i mikrokoracima. Mi ćemo ove načine rada promatrati na primjeru dvofaznog koračnog motora. Polu koračno upravljanje postiže se puštanjem struje kroz obje faze u cijelo vrijeme, no svaka promjena smjera struje u fazama stvara pomak (tj. Korak), slijed smjerova struje na fazama prikazan je u tablici (2.3.1.). U polu koračnom načinu upravljanja koračni motor treba napraviti dvostruko veći broj koraka nego u puno koračnom načinu rada. Do toga dolazi jer u polu koračnom upravljanju za razliku od puno koračnog upravljanja ne propuštamo 100% struje kada prelazimo iz 0% u novi korak već imamo među korak kod kojeg propuštamo 50%-70% te onda tek u idućem koraku prelazimo na 100% izvorne struje. Polu koračno upravljanje nam stvara dvostruko veći broj koraka po rotaciji što znači da se koračni motor okreće za manji broj stupnjeva po koraku. S time dobivamo veću preciznost pozicioniranja, ali gubimo

snagu pri okretanju. Mikrokorlačno upravljanje je slično polu korlačnom upravljanju, poanta je postavljanje među koraka između minimuma i maksimumima struje kroz faze. Mikrokorlačno upravljanje je ime za sve načine upravljanja koji stvaraju veći broj koraka od polu korlačnog (četvrtina korlačno, osmina korlačno, šesnaestina korlačno... (rezolucija se povećava uvijek dvostruko)). Svaki put kada smanjimo rezoluciju koraka stvaramo još jedan međukorak između 0% i 100% struje na fazi. Tako se sve većim povećanjem koraka sve više približavamo izgledu sinusoide, sinusoida među fazama je pomaknuta za 90 stupnjeva jedna od druge. Za okretanje motora u drugom smjeru, za sve ove metode, potrebno je samo ponavljati ove korake u suprotnom smjeru.

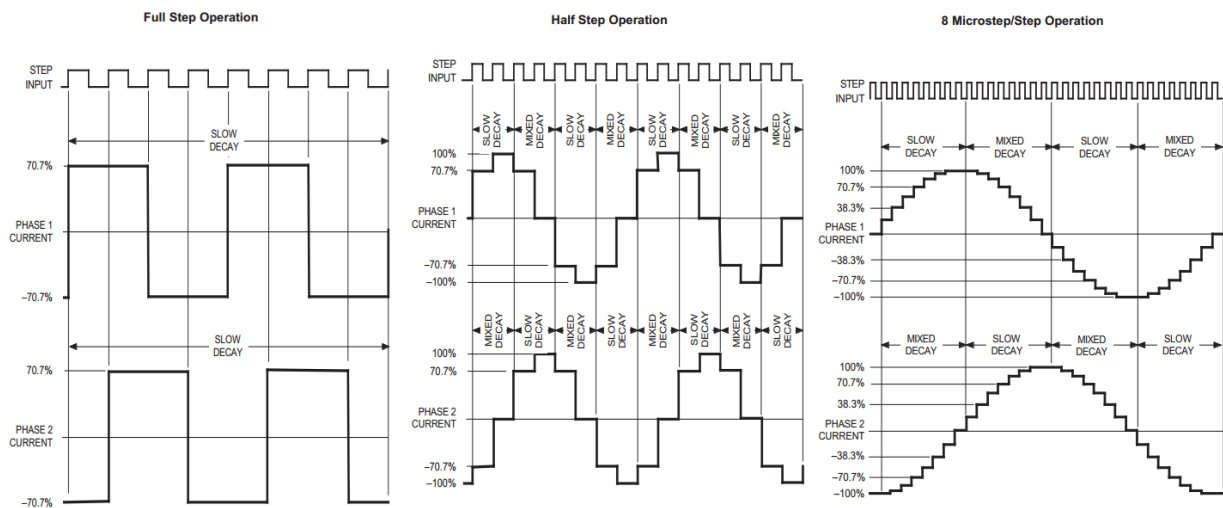
Index	1a	1b	2a	2b
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1
5	1	0	0	1
6	1	1	0	0
7	0	1	1	0
8	0	0	1	1

Alternate Full Step Sequence
(Provides more torque)

Index	1a	1b	2a	2b
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0
11	0	1	0	0
12	0	1	1	0
13	0	0	1	0
14	0	0	1	1
15	0	0	0	1
16	1	0	0	1

Half Step Sequence

Slika 2.3.1. tablica stanja faza pri rotaciji



Slika 2.3.2. grafovi upravljanja korlačnim motorom

3. Mehanizmi linearnog prijenosa

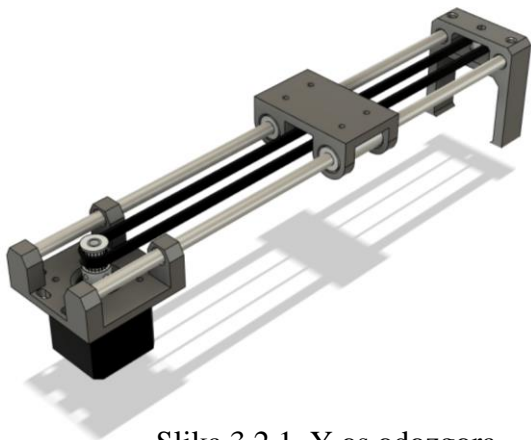
3.1. Vrste mehanizama linearnog prijenosa

CNC mašine su strojevi koji obavljaju pomake u kartazijskom (kordinatnom) sustavu, za takav sustav prikladni su nam linearni aktuatori, ali mi na CNC mašinama najčešće koristimo rotacijske aktuatore, to jest motore. Da bi iz takvih rotacijskih aktuatora postigli linearne pomake koristimo mehanizme koji pretvaraju rotacijske pomake u linearne pomake te takve mehanizme nazivamo mehanizmima linearnog prijenosa. Linearni prijenos zasniva se na pretvaranju rotacijskog pomaka u linearni pomak, to se može ostvariti na dosta načina, ali u industriji postoji par mehanizama koji se najčešće koriste. Zupčasti prijenos se inače koristi za prijenos rotacijskog pomaka, ali ako umjesto drugog zupčanika koristimo zupčastu letvu, tada dobivamo linearni pomak umjesto rotacijskog. Vijčani prijenos koristi osovinu s navojem koja se okreće te na nju postavljamo maticu (ili bilo koji drugi predmet s pripadajućim provrtom i navojem) koja je učvršćena tako da se ne može rotirati. Na taj način okretanjem osovine s navojem matica će se umjesto rotiranja, gibati linearno po osovini. Lančani prijenos koristi lanac koji spaja dva lančanika te inače služi za prijenos rotacijskog gibanja sa jednog lančanika na drugi, ali ako mi spojimo nešto na napeti lanac i tu stvar učvrstimo za nekakav sistem linearnih vodilica, tada dobijemo sustav linearnog prijenosa. Na kraju ćemo još spomenuti remeni prijenos koji je dosta sličan lančanom i on se koristi na X i Y osima plotera pa ćemo ga zato detaljnije objasniti.

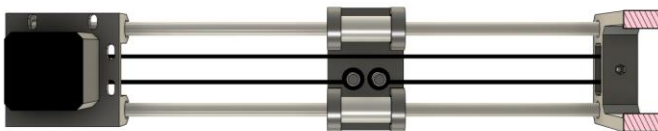
3.2. Remeni prijenos na ploteru

Ova vrsta prijenosa je odabrana zato što je dosta jeftina za izvesti i daje prilično precizno gibanje. Remen koji ovdje koristimo je zupčasti remen (GT2 timing belt). Uz sami remen da bi uspjeli napraviti sistem linearnog prijenosa potrebni su nam zupčasta spojka za osovinu motora (GT2 spojka za 5mm osovine – 10 zubi), rotacijski ležaj odgovarajućeg promjera koji je isti kao i vanjski promjer spojke, te metalne šipke i nešto s čime ćemo ih učvrstiti (mi ih učvršćujemo u samu konstrukciju plotera). Zupčasta spojka se spaja na osovinu motora te preko nje pokrećemo cijeli ovaj sistem, remen se pokreće jer ga zupci spojke zahvaćaju i guraju. Nasuprot spojci na drugom krajnjem mjestu sistema nalazi se

ležaj učvršćen u konstrukciju plotera, njegova uloga je omogućiti remenu da se okreće oko njega bez puno trenja te da time što je druga kontaktna točka remena stvara napetost. Remen koji mi imamo nije zatvoreni krug već je traka, remen se postavlja tako da se prvo jednim krajem veže za jedan dio pomičnog komponenta kojega će micati, zatim ga provlačimo oko zupčaste spojke, vodimo ga do druge strane gdje prolazi oko ležaja te se na kraju veže za drugi kraj pomičnog komponenta. Remen se ne veže doslovno, već imamo male kvačice koje drže da se remen „ne odveže“ od pomičnog komponenta. Da bi se pomična komponenta micala samo linearno potrebno joj je dati neko uporište, za to koristimo metalne šipke koje su učvršćene u konstrukciju plotera. Pomična komponenta ima u sebi linearne ležajeve klize po metalnim šipkama i tako osiguravamo stabilnost pomične komponente bez stvaranja velikog trenja. Na slikama ispod (3.2.1., 3.2.2.) prikazano je kako cijeli ovaj sistem na kraju izgleda odozgora i od ispod. X i Y os obje imaju praktički isti sistem linearnog prijenosa, ali na X osi to je sve zakrenuto da bude vertikalno kako bi smanjili količinu površine koju cijela konstrukcija zauzima na radnoj površini plotera (zakrenuli smo sve za 90 stupnjeva jer tako ćemo moći imati veći maksimalni put na Y osi).



Slika 3.2.1. Y os odozgora

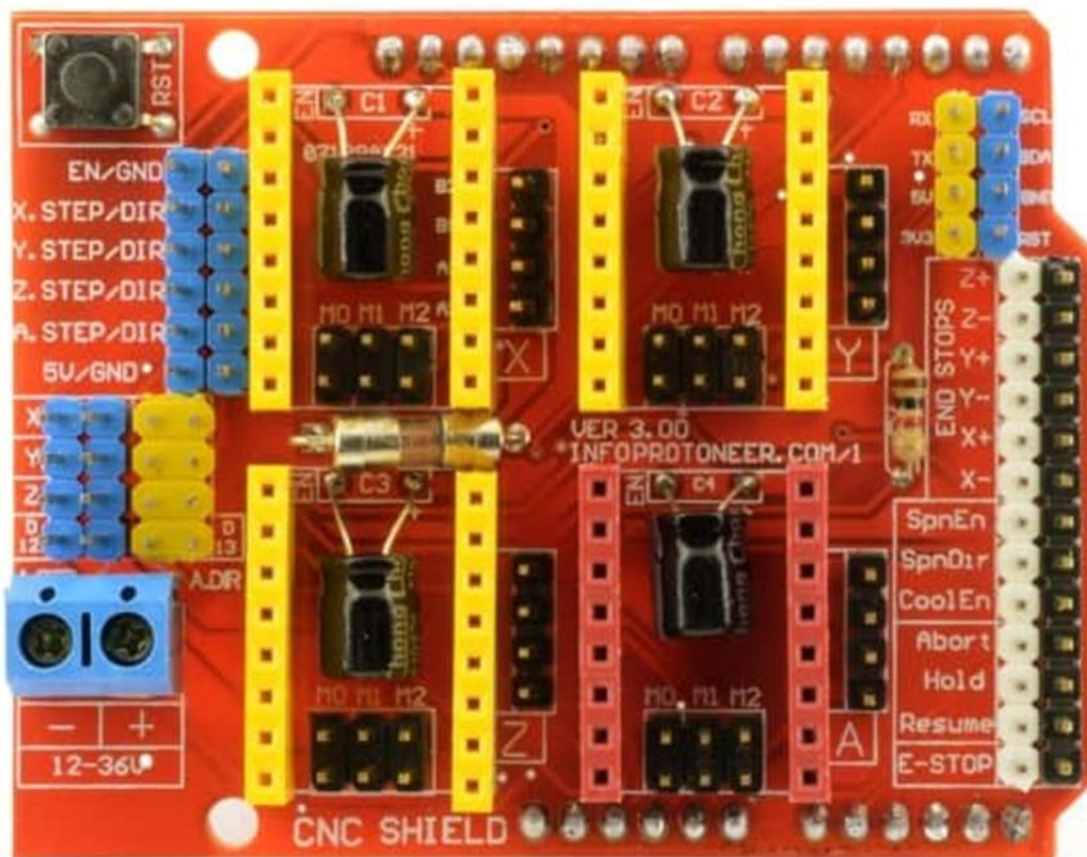


Slika 3.2.2. Y os odozdola

4. Upravljačka elektronika

4.1. Arduino i CNC „štit“

Ploter da bi mogao upravljati motorima treba način procesiranja G-koda. Tuda dolazi atmega328p mikroupravljač. U ovom projektu koristi se kroz ARDUINO UNO razvojnu ploču zato što ona ima već ugrađen serijski port za komuniciranje s računalom kao USB tip B ženski konektor. Prilikom korištenja plotera na način slanja G-koda preko kompjutera arduino UNO se kablom spaja od svojeg USB porta do USB porta kompjutera. Kada koristimo način rada preko SD kartice tada se ne smijemo kablom spajati na kompjuter te tada arduino UNO dobiva G-kod s SD kartice preko drugog arduina (arduino NANO-a) koji je na arduino UNO spojen preko RX i TX pinova. Oni isto serijski komuniciraju, ali sada arduino nano čita G-kod preko kartice i šalje ga UNO-u za razliku od kompjutera. Na arduino UNO (arduino koji upravlja aktuatorima) je isto tako spojen dodatni modul za lakše korištenje drivera za koračne motore, to jest „CNC shield“ (u nastavku teksta zvat ćemo ga samo štit). Štit se spaja na gornju stranu arduina te on na sebi ima naznačene te postavljene priključke za pinove driver-a koračnih motora, graničnih prekidača i puno druge elektronike koja se može koristiti na CNC mašini, ali mi većinu tih pinova ne koristimo za ploter. Poanta štita je dati nam način da lako spojimo sve potrebne komponente (drivere i graničnike) te aktuatora (koračne motore i servo motor) na arduino bez da moramo raditi vlastitu PCB pločicu. Štit ima priključke za četiri drivera koračnih motora (zajedno sa zaštitnim kondenzatorima već spojenim na štitu), ali mi za ploter koristimo samo dvije osi s koračnim motorima te zato koristimo samo dva priključna mjesta. Granični prekidači se na štitu spajaju na mjesta označena sa X- i Y- (naravno po svojim pripadajućim osima). Na priključku obilježenim sa Z+ se spaja narančasta žica (žica za upravljanje) servo motora te još na štit spajamo servo motor s njegove druge dvije žice na +5V i GND. Još na kraju je potrebno spojiti napajanje za koračne motore na označene konektore na štitu (plavi šarafni terminali za žice). (Ispod je slika izgleda CNC štita.)

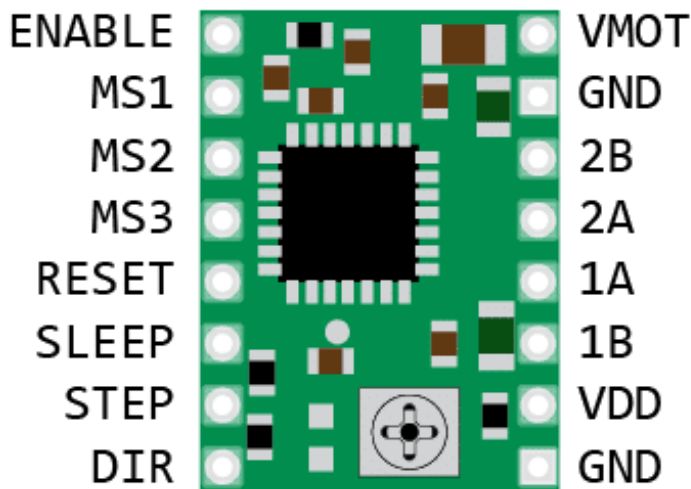


Slika 4.3.1. CNC štit

4.2. Napajanje i aktuatori

Napajanje za ploter dobiva se preko „barrel jack-a“ ugrađenog u kućište plotera. Pošto naši aktuatori, koračni motori (us 17hs4401s - model), koriste puno veći napon od ostale elektronike i aktuatora za Z os (SG90 servo motor) potreban nam je neki način da smanjimo napon u jednoj grani koja će napajati ostalu elektroniku. U tu svrhu koristimo „step down buck converter“, to je modul na bazi LM2596 čipa koji nam omogućava da smanjimo napon od 3V - 40V do 1.5V - 35V, mi ćemo koristiti taj modul kako bi stvorili izvor od 5V za ostalu elektroniku bez velikog gubitka snage. Ploter je moguće konfigurirati tako da radi s naponima od 12V do 35V podešavanjem potencijometra na „buck converter-u“, ali napajanje koje trenutno planiramo da se spaja na „barrel jack“ je 16V i 4.5A pa je i na to „buck converter“ postavljen.

pozitivan dio napajanja za motor, a na VDD dajemo +5V. GND se spaja na GND od napajanja zajedno s GND od 5 voltnog napajanja (u ovom slučaju koristimo arduino za napajanje čipa). Još je potrebno za sigurnost od visokih šiljaka napona napajanja staviti između VMOT i GND kondenzator od barem 47 μ F. ENABLE uključuje i isključuje izlaze FETOVA čipa pa ćemo ga držati u jedinici(+5V) da ENABLE bude isključen. SLEEP pin nam služi za minimiziranje potrošnje struje dok motor ne radi, mi tu funkciju nećemo koristiti te zato ćemo taj pin držati u jedinici. RESET pin nam služi za instantno prebacivanje u određeni mikrokorak koji ovisi u kojem modu mikrokoraka smo trenutno te su te pozicije specificirane u dokumentaciji čipa. Taj pin isto ne koristimo pa je isto spojen na +5V. 1A i 1B su pinovi na koje spajamo jednu fazu motora, 2A i 2B su pinovi za drugu fazu motora. MS1, MS2 i MS3 su pinovi kojima određujemo rezoluciju koraka (puno koračna, polu koračna...). Kako trebamo postaviti jedinice i nule na ulazima tih pinova za određene rezolucije možemo vidjeti u tablici ispod (4.3.2.). Još na kraju su nam ostali STEP i DIR pinovi. STEP pin nam služi za rađanje koraka, svakim rastućim impulsom(+5V) na STEP pinu koračni motor napravi jedan korak. Smjer tog koraka ovisi o DIR pinu; kada je na DIR pinu +5V motor će se okretati u smjeru kazaljke na sat, kada je na DIR pinu 0V motor će se okretati obrnuto od kazaljke na satu.



Slika 4.3.1. izgled driver modula

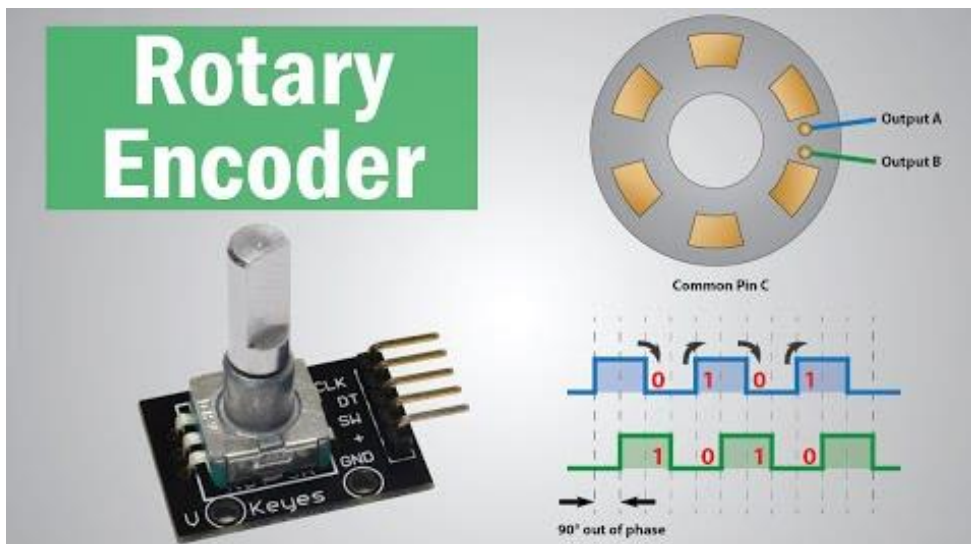
MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

Slika 4.3.2. tablica postavljanja rezolucije koraka

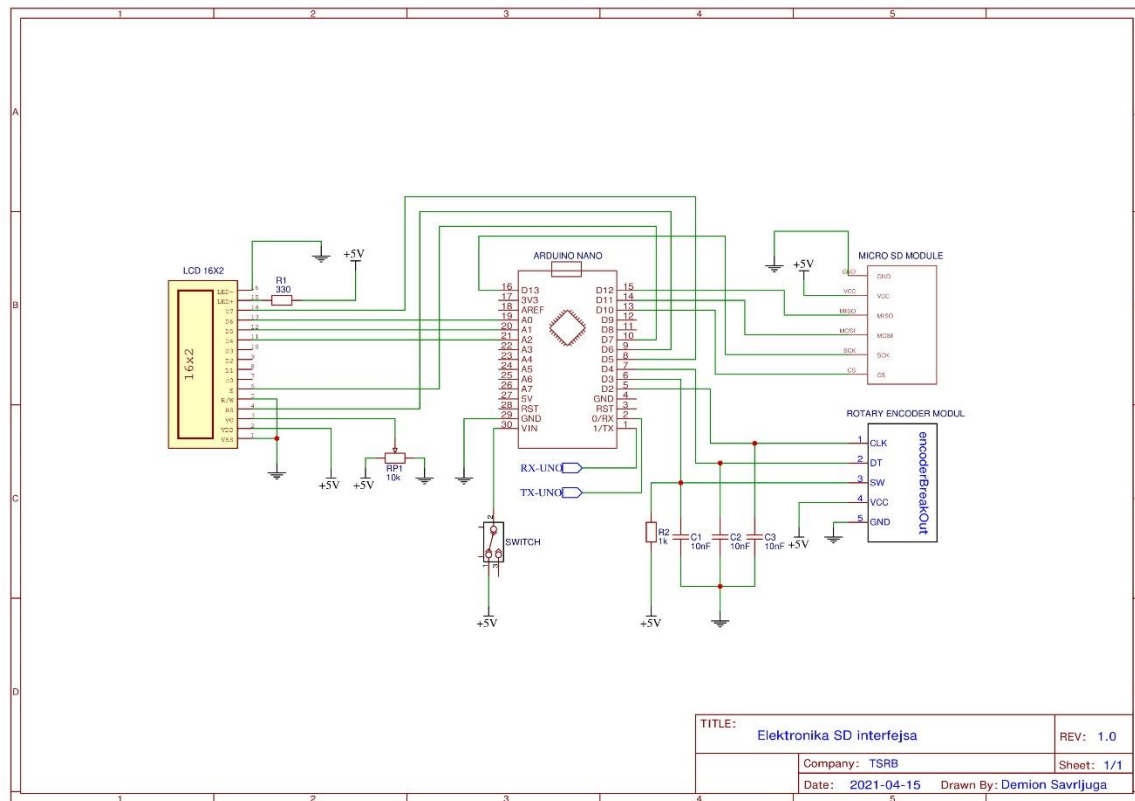
4.4. Interfejs (LCD i odabiranje datoteke sa SD kartice)

Ploter osim načina rada gdje direktno prima G-kod od kompjutera još može i primati G-kod sa SD kartice preko drugog arduina. Način rada preko kompjutera se koristi tako da se spajamo kablom direktno na arduino UNO, u načinu rada preko SD kartice spajamo se na arduino UNO preko drugog arduina (arduino NANO-a) koji na sebe ima spojeni modul za čitanje micro SD kartica. Da bi mogli odabrati koju datoteku želimo čitati s SD kartice, imamo na kućištu plotera pričvršćene LCD i inkrementalni enkoder koji su isto spojeni na arduino NANO. Kada prekidačem uključimo način rada čitanja sa SD kartice na LCD-u će nam se pokazati inicijalizacijske poruke te poslije toga okretanjem enkodera možemo odabrati koju datoteku želimo čitati s kartice i slati našem ploteru. Okretanjem enkodera u smjeru kazaljke na sat ili obrnuto spuštamo se ili dižemo kroz listu datoteka na SD kartici. Enkoder nam daje te informacije o smjeru okretanja po principu dvaju pulsova koji su fazno pomaknuti. Enkoder unutar sebe ima disk koji ima radijalno postavljene bakrene segmente koji su spojeni na +5V dok su pinovi A i B spojeni na GND. Ticala A i B pinovi su spojeni na pomičnu osovinu enkodera (ono što mi okrećemo prstima) te kada okrećemo tu osovinu i kako A i B ticala dodiruju bakrene dijelove diska i nevodljive dijelove diska mi mehanički stvaramo na izlazima A i B pinova impulse. Brojanjem impulsa možemo znati za koliko inkremenata (ili stupnjeva) je enkoder bio zakrenut, ali pošto su A i B ticala malo odvojena jedno od drugoga, jedan impuls će uvijek malo kasniti za drugim. Impuls koji će se prije dogoditi ovisi o tome u kojem smjeru okrećemo osovinu enkodera. Zbog toga možemo pomoću mikroupravljača pratiti padajuće ili dižuće impulse na jednom od A

i B pinova te kada se impuls digne ili spusti (ovisno o tome što smo programski odabrali) provjerimo stanje drugog pina; ako je u tom trenu na drugom pinu stanje 0 osovina se okretala u jednu stranu, a kada je stanje 1 osovina se okretala u drugu stranu (koja vrijednost drugog pina znači smjer kazaljke na sat, a koja obrnuto od kazaljke na sat ovisi isto o tome koji smo pin odabrali za ovo očitavanje). Na slici ispod (4.4.1.) možemo vidjeti fizički izgled modela enkodera koji se koristi u ovom radu, dijagram unutrašnjeg diska sa A i B ticalima te kako izgleda graf impulsa na izlazima A i B pinova kod okretanja enkodera. Ovaj enkoder još ima i mogućnost da služi kao gumb koji mi koristimo da bi potvrdili odabir datoteke kada okretanjem nađemo željenu datoteku. Kada pritisnemo enkoder SW pin koji inače nije spojen na GND se spoji na GND te tako na tom pinu možemo očitavati pritisak enkodera.



4.4.1. enkoder i graf njegovih izlaza



Slika 4.4.2. shema SD interfejsa

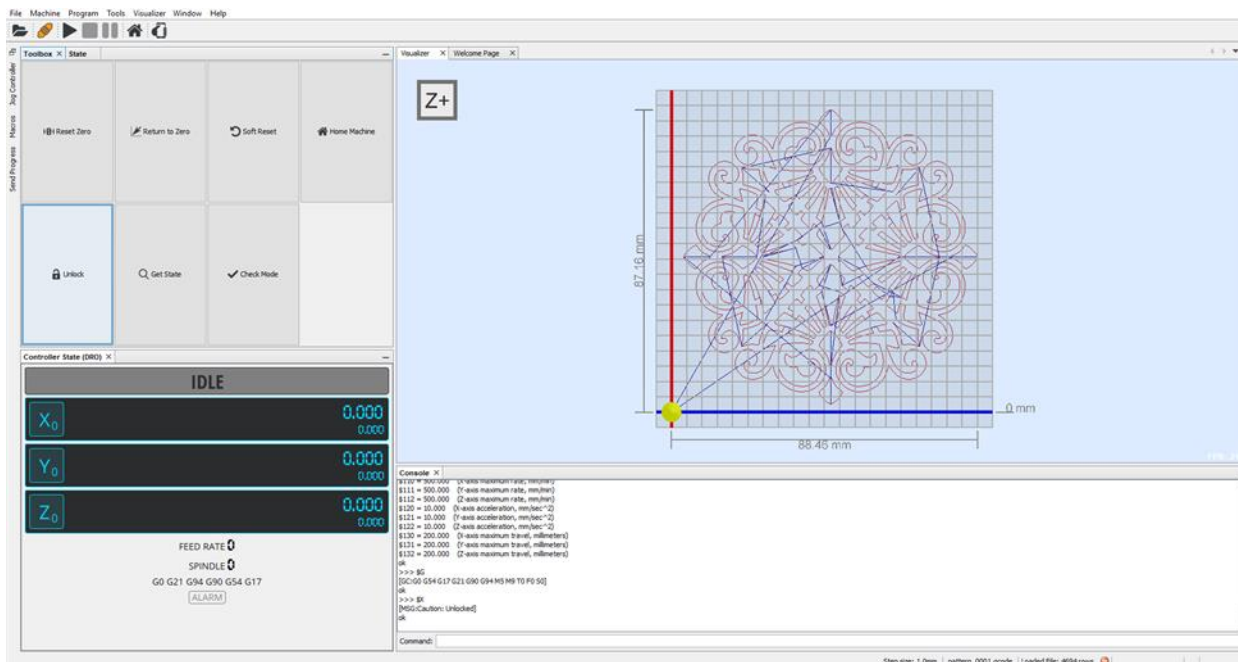
5. Programska podrška

5.1. Firmware plotera – GRBL

CNC mašine da bi mogle čitati G-kod te pretvarati te naredbe u pomake aktuatora i druge radnje, trebaju na svojoj procesnoj jedinici imati neki program koji se time bavi. Ovaj ploter koristi „open source“ „firmware“ zvani GRBL. GRBL je najčešće korišteni firmware u ovakvim CNC nekomercijalno napravljenim mašinama pa tako ga i mi koristimo u ovom ploteru. Prednosti GRBL software-a te razlozi zašto je on bio odabran za ovaj projekt su: besplatan je, lako podesiv, može raditi s tri osi (X, Y i Z) i lako se konfiguriše za karakteristike naše mašine. Za ploter koristimo modificiranu verziju GRBL-a koja se zove „GRBL servo“ (verzija: 1.1f), ta verzija nam je potrebna jer koristimo servo motor za Z-os umjesto koračnog motora, što nije moguće napraviti u običnoj verziji GRBL-a.

5.2. Universal Gcode Sender i kako ga koristiti

Kako bi GRBL mogao raditi treba od nekud primiti G-kod. Ovdje smo isto izabrali besplatnu „open source“ opciju: UGS – Universal Gcode Sender. UGS nam omogućava slanje G-koda putem USB porta. Da bi koristili način slanja G-koda preko računala trebamo otvoriti UGS, spojiti USB kablom arduino UNO sa kompjuterom, programski se u UGS-u priključiti na arduino te odabrati datoteku iz koje želimo čitati i slati G-kod. To je tok započinjanja slanja G-koda ploteru (i njegovog rada), ali UGS ima još nekoliko sposobnosti koje su nama korisne u radu s ploterom. UGS ima jedan komandni prozor na doljnoj strani sučelja preko kojega možemo direktno pisati i slati naredbe ploteru te vidjeti naredbe koje je ploter primio i njegove povratne poruke (možemo vidjeti error-e). UGS nam daje mogućnost da preko njegovog „setup WIZARD“-a konfiguriramo parametre CNC mašine. Iako se to samo jednom na početku koristi to je jako korisna funkcija za konfiguriranje mašine. S lijeve strane sučelja imamo gumbe za osnovno upravljanje CNC mašinom: homing (doalzak u koordinatu (0, 0)), opuštanje motora (inače ne možemo rukom pomicati osi), resetiranje, pauziranje slanja G-koda i još par funkcija. Na kraju vjerojatno jedna od najvećih prednosti kada koristimo UGS, a ne šaljemo G-kod sa SD kartice je mogućnost simuliranja putova. Gornji desni prozor namjenjen je da nam prikaže kako će izgledati putanja glave naše mašine te kada će se olovka dizati i spuštati. To nam omogućava da lakše uhvatimo greške ili da vidimo efekte koje dobivamo promjenama parametara. Kada UGS šalje G-kod simulacija on u pravom vremenu pokazuje kako glava plotera putuje u prostoru. Na slici ispod (5.2.1.) možemo vidjeti primjer izgleda UGS sučelja.

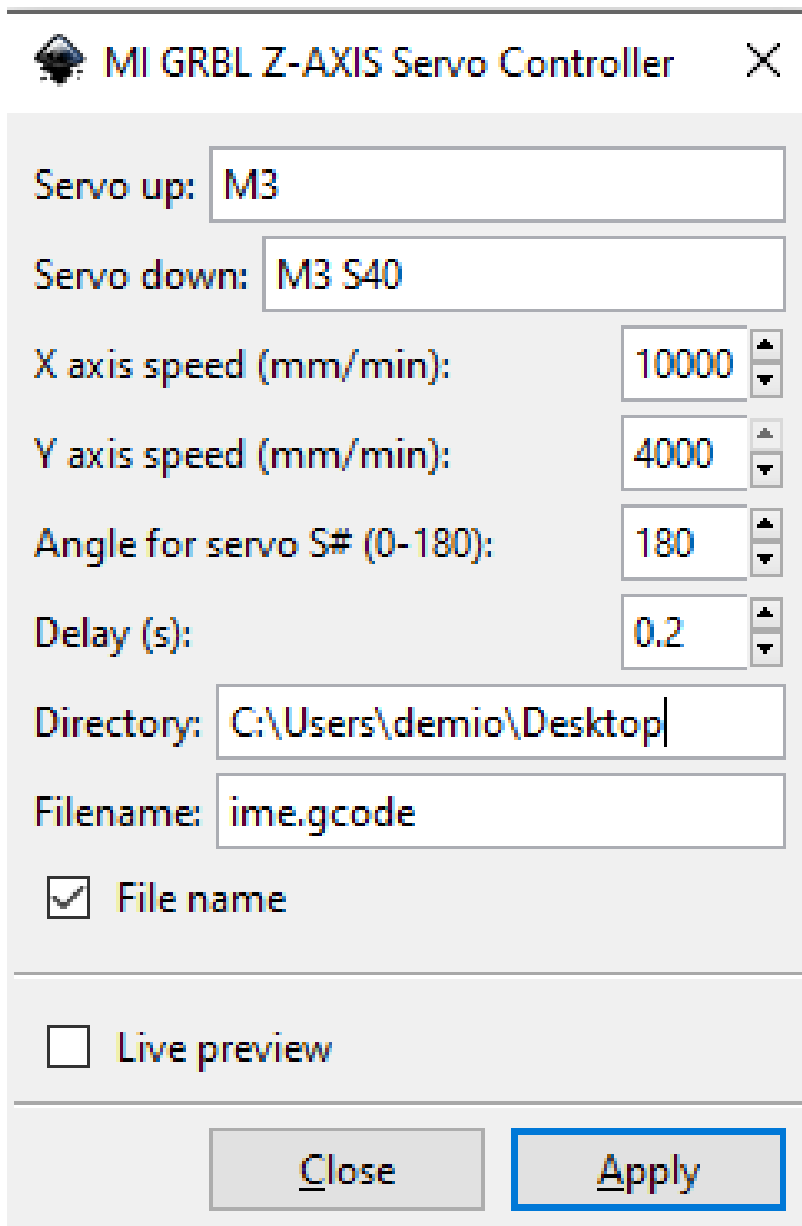


Slika 5.2.1. sučelje UGS-a

5.3. Stvaranje G-koda – Inkscape

Da bi uopće mogli slati G-kod prvo ga trebamo stvoriti. Ovdje opet koristimo besplatan „open source“ program: Inkscape. Inkscape nije program ekskluzivno dizajniran za stvaranje G-koda, njegova glavna uporaba je crtanje u grafičkom dizajnu, ali Inkscape ima dodatak („extension“) koji mu omogućava da stvara G-kod. Taj dodatak je MI-GRBL i on omogućava Inkscape-u da pretvara vektorske slike u G-kod. Mi da bi stvorili G-kod trebamo već imati vektorsku sliku (*.svg datoteka) pa je zato trebamo od negdje ili skinuti ili pretvoriti neku drugu vrstu slike (*.jpg, *.png) u *.svg vektorsku sliku. Pretvorbu slike radimo tako da ubacimo sliku u Inkscape, odaberemo u izbornicima „Path“->„Trace bitmap“ te poslije toga odaberemo „Path“->„Object to path“, tako ćemo iz bilo koje slike dobiti vektorsku datoteku (vektorske datoteke možemo dobiti ovim načinom samo iz crno bijelih slika, ako pokušamo dobiti iz normalnih slika rezultati neće biti dobri). Sada kada imamo vektorsku sliku možemo konačno dobiti G-kod. Na početku je potrebno otvoriti vektorsku sliku te namjestiti veličinu papira u programu na veličinu radne površine plotera (210x210), program će koristiti taj papir kao referencu za površinu našeg plotera te sve što je izvan papira neće se moći ispisati. Kada smo postavili sliku i površinu stišćemo ctrl+A

za označivanje svega što smo učitali na papiru te odlazimo do izbornika „Exstension“->„MI-GRBL“. Tada će nam se otvoriti prozorčić za stvaranje G-koda i u njega trebamo upisati ove parametre prikazane u slici ispod (5.3.1.) (naravno Filename i Directory mogu biti drugačiji). Kada smo završili s ovim procesom imamo G-kod spreman za slanje preko UGS-a ili preko SD kartice.



MI GRBL Z-AXIS Servo Controller

Servo up: M3

Servo down: M3 S40

X axis speed (mm/min): 10000

Y axis speed (mm/min): 4000

Angle for servo S# (0-180): 180

Delay (s): 0.2

Directory: C:\Users\demio\Desktop

Filename: ime.gcode

☒ File name

☐ Live preview

Close Apply

Slika 5.3.1. Inkscape parametri za
stvaranje G-koda

5.4. Program za način rada sa SD karticom

Kako nam ploter ne bi morao biti uvijek spojen na računalo na ploter je bila nadodana mogućnost čitanja G-koda sa SD kartice. Princip tog načina rada započinje tako da pomoću prekidača promijenimo način rada plotera na način rada sa SD karticom. Tada se pale LCD i arduino NANO te program na njemu kreće. Program prvo šalje na LCD informacije o tome je li LCD radi, zatim da li se SD kartica inicijalizirala te ako je, onda ulazimo u odabir datoteka. Ako se u procesu inicijalizacije programa pronađe neka greška arduino će na LCD poslati poruku o toj grešci te tada trebamo riješiti grešku i ponovno pokrenuti program prekidačem. Kada dođemo u fazu odabira datoteka koristimo enkoder spojen na arduino NANO za prolaženje kroz listu datoteka i pritiskujemo ga za potvrdu odabira. Kada odaberemo datoteku arduino na LCD šalje podatke o uspješnosti ili ne uspješnosti čitanja datoteke te ako je sve u redu on započinje slanje G-koda arduino UNO-u. Za komunikaciju među arduino-ima koristimo serijsku komunikaciju (to jest RX i TX pinove). Ciklus komunikacije arduino-a je slijedeći: arduino NANO čita liniju odabrane datoteke, zatim je šalje arduino UNO-u, UNO kada odradi liniju G-koda šalje povratnu „ok“ poruku da je spreman primiti iduću liniju, NANO prima tu poruku te šalje iduću liniju. Taj proces se ponavlja dok NANO ne pošalje sve linije koda iz datoteke, tada prestaje čitati datoteku i komunicirati s UNO-om. Nakon završetka datoteke NANO šalje na LCD poruku da je posao obavljen i čeka novi input korisnika. Kada dođemo u tu fazu NANO će čekati dok netko ne pritisne enkoder i tada ćemo se vratiti opet u dio programa gdje odabiremo datoteku za ispis i cijeli proces odabira datoteke i ispisa može se ponovno ponoviti (ovo će se ponavljati u nedogled). Ako za vrijeme komunikacije između arduina dođe do greške, NANO će prestati komunicirati te će na LCD poslati poruku da se dogodio ERROR. Ovo je bilo pojednostavljeno objašnjenje kako radi ovaj način rada plotera, ispod ovog teksta se može naći izvoran kod te u njegovim komentarima dubinsko objašnjenje programa.

```

// U konfiguracijama možemo mijenjati na kojem fizičkom pinu i portu
mikroupravljača se određeni element nalazi
// Varijable koje se koriste za određeni komponent nalaze se u pripadajućim
konfiguracijama
//LIBRARIES//
#include <SD.h>
#include <SPI.h>
#include <LiquidCrystal.h>
#include <avr/io.h>
#include <avr/interrupt.h>
//Beginning of Auto generated function prototypes by Atmel Studio
void go_through_menu();
void chose_another_file_prompt();
void communicate(String file_name);
String get_serial();
void wait_serial();
void empty_serial_buffer();
void LCD_initialization();
void SD_initialization();
void save_file_names();
void setup_encoder();
//-----//

//LCD - konfiguracija//
#define d4 A2
#define d5 A1
#define d6 A0
#define d7 5
#define rs 6
#define en 7

LiquidCrystal lcd(rs, en, d4, d5, d6, d7); // deklaracija pinova koje lcd
koristi i postavljanje u 4-bitni mod rada
//-----//

//SD kartica - konfiguracija i varijable//
#define chipSelect 10
String file_names[30]; // polje za spremanje imena datoteka
char *file_pointer; // pointer na ime datoteke

```

```

int numberOfFiles=0                                // varijabla za broj datoteka na SD
kartici
int chosen_file=0;                                // varijabla za spremanje broja
odabrane datoteke
int current_number_of_file=1;                      // varijabla za trenutni broj
datoteke koja nam je prikazana u izborniku

File root;                                         // deklariramo File(datoteka)
varijablu
//-----//

//Enkoder - konfiguracija i varijable//
#define encPORT    PORTD
#define encDDR      DDRD
#define encPIN      PIND
#define pinA        2
#define pinB        4
#define button      3
int encoder_change=0;                             // varijabla koja prati promjenu
stanja i smjer u kojemu se enkoder okrenuo
// (0=nema promjene, 1=smjer kazaljke na satu, 2=smjer obrnuto od kazaljke na
satu)
bool select=0;                                    // varijabla u koju spremamo stanje
gumba enkodera (1=pritisnuto)
// Da bi se spriječilo istitravanje na enkoderu potrebno je ovako spojiti pinove
enkodera:
// 10nF kondenzatore je potrebno staviti između pinA - GND i pinB - GND
// "button" treba imati na sebe spojen 10nF kondenzator koji ide na GND i 10kohm
// otpornik spojen na +5V
//-----//

void setup() {
    LCD_initialization();    // inicijaliziramo LCD
    //
    SD_initialization();     // inicijaliziramo SD karticu
    //
    setup_encoder();          // inicijaliziramo enkoder
    //
    save_file_names();        // spremamo sva imena datoteka u polje stringova

```

```

}

void loop() {
    go_through_menu();          // prolazak kroz izbornik datoteka
    //
    chose_another_file_prompt(); // upit za novi ciklus odabira datoteke
    //
}

//GENERALNE - FUNKCIJE//
void go_through_menu(){
    // ova funkcija prolazi kroz cijeli ciklus odabira datoteke
    root = SD.open("/");        // otvaramo SD karticu kao File varijablu
    //
    lcd.clear();                // čišćenje LCD-a
    lcd.setCursor(0, 0);        // postavljanje početka ispisa na LCD-u (stupac,
red)
    lcd.print("  CHOSE FILE");  // printamo poruku na LCD-u
    delay(1500);
    //
    current_number_of_file=1;    // nulti fajl je uvijek SYSTEM fajl kojeg ne
želimo čitati
    EIMSK|=(1<<INT0)|(1<<INT1); // uključujemo prekide za enkoder i gumb
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("FILE: ");        // nova poruka
    lcd.print(current_number_of_file); // printamo trenutni broj datoteke
    lcd.print("/");
    lcd.print(numberOfFiles);    // printamo broj čitljivih datoteka
    lcd.setCursor(0, 1);        // odlazimo na početak doljnjeg reda
    lcd.print(file_names[current_number_of_file]); // printamo ime datoteke
    encoder_change=0;           // resetiramo varijablu stanja
enkodera
    select=0;                   // resetiramo varijablu stanja gumba
    chosen_file=0;               // resetiramo broj izabrane datoteke
    while(chosen_file==0){      // dok god ne izaberemo datoteku funkcija će se
vrtiti
        if(encoder_change>0){   // dok god se enkoder ne okrene encoder_change će
biti 0

```

```

if(encoder_change==1){    // ako je enkoder okrenut u lijevo encoder_change=1
    if(current_number_of_file>1){    // ako ovo nije prva datoteka
        current_number_of_file--;    // odi jednu datoteku unazad
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("FILE: ");
        //////////////////////////////////
        lcd.print(current_number_of_file);    // blok koda gdje opet ispisujemo
    //
        lcd.print("/");    // na kojoj smo datoteci
    //
        lcd.print(numberOfFiles);
        //////////////////////////////////
        lcd.setCursor(0, 1);
        lcd.print(file_names[current_number_of_file]);
    }
}else if(encoder_change==2){
    if(current_number_of_file<numberOfFiles){ // ako ovo nije zadnja datoteka
        current_number_of_file++;    // odi jednu datoteku unaprijed
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("FILE: ");
        //////////////////////////////////
        lcd.print(current_number_of_file);    // blok koda gdje opet ispisujemo
    //
        lcd.print("/");    // na kojoj smo datoteci
    //
        lcd.print(numberOfFiles);
        //////////////////////////////////
        lcd.setCursor(0, 1);
        lcd.print(file_names[current_number_of_file]);
    }
}
}
encoder_change=0;    // resetiramo varijablu stanja enkodera
}
if(select==1){    // ako je gumb bio pritisnut
    chosen_file=current_number_of_file;    // ovo je odabrana datotekaza ispis
}
}
}

```



```

    EIMSK&=~(1<<INT0)|(1<<INT1);          // isključi prekide
    select=0;                               // resetiraj varijablu stanja gumba
    communicate(file_names[chosen_file]);    // provedi komunikaciju sa odabranom
datotekom
    root.close();                           // zatvori datoteku SD kartice
}

void chose_another_file_prompt(){
    // ova funkcija čeka dok korisnik ne pritisne gumb za ponovi odabir datoteke
    select = 0;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" READ NEW FILE?");
    //////////////////////////////////////
    lcd.setCursor(0, 1);                     // blok koda gdje ispisujemo upit
    //
    lcd.print(" [PRESS SELECT] ");           // za ponavljanje odabira fajla
    //
    delay(200);
    //////////////////////////////////////
    EIMSK|=(1<<INT0)|(1<<INT1);              // uključujemo prekide
    while(select == 0){                      // dok se enkoder ne pritisne ništa se ne
događa
        delay(1);
    }
    EIMSK&=~(1<<INT0)|(1<<INT1);              // isključujemo prekide
    select=0;                                // resetiramo varijablu stanja gumba
}

//-----//
//
//GRBL komunikacija - FUNKCIJE//

void communicate(String file_name){
    // zadaća ove funkcije je komunikacija s arduino UNO-om
    // taj proces teče ovako:
    // NANO šalje liniju koda UNO-u, čeka da dobi povratnu poruku te šalje iduću
liniju...

```

```

    char read_char;                // varijabla koja sprema pročitani char iz
datoteke
    String returning_message;      // varijabla koja sprema povratnu poruku od UNO-
a
    String line="";               // string u koji spremamo liniju pročitane iz
datoteke
    Serial.print("\r\n\r\n");     // slanje ovoga UNO-u budi GRBL
    delay(2);
    empty_serial_buffer();         // zovemo funkciju za pražnjenje serijskog
buffer-a
    File G_CODE;                  // otvaramo novu File varijablu
    G_CODE = SD.open(file_name);   // spremamo datoteku sa SD kartice
    if (!G_CODE) {                 // ako nismo uspjeli spremiti
        lcd.clear();
        lcd.setCursor(0, 0);      // ispišemo ERROR
        lcd.print("CAN'T OPEN FILE!");
        while(1){                 // zablokiraj program
            delay(1);
        }
    } else {
        lcd.clear();
        lcd.setCursor(0, 0);      // ako smo uspjeli otvoriti datoteku
        lcd.print("Reading file:"); // ispisujemo čitanje datoteke
        lcd.setCursor(0, 1);
        lcd.print(file_names[chosen_file]); // ispisujemo ime datoteke koju čitamo
    }
    //
    while (G_CODE.available()) {   // dok god nismo došli do kraja datoteke idi
kroz while
        read_char = G_CODE.read(); // zapiši pročitani char u varijablu
        if(read_char != '\n'){     // dok god nismo došli do kraja linije
            line = line + read_char; // dodaj char stringu
        }else{                     // kada dođemo do kraja linije '\n'
            line = line + '\r';     // dodaj carriage return stringu '\r'
            Serial.print(line);     // pošalji UNO-u taj string
            returning_message=get_serial(); // spremi povratnu poruku u string
            if(returning_message.indexOf("error")>0){ // ako povratna poruka je greška
                lcd.clear();
                lcd.setCursor(0, 0);
            }
        }
    }

```

```

        lcd.print(" !FILE ERROR! ");    // ispiši da se dogodila greška
        lcd.setCursor(0, 1);
        lcd.print("RESTART MACHINE!");    // mašina se mora restartati
        while(1){                        // zablokiraj program
            delay(1);
        }
    }
    empty_serial_buffer();
    line="";                            // resetiraj string u koji spremamo liniju G-
koda
    }
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("  WORK DONE!  "); // ako smo došli do kraja datoteke
delay(2000);
G_CODE.close();                    // zatvori Fail varijablu
}

String get_serial(){
    // ova funkcija nam daje povratnu poruku koju UNO šalje
    String inLine = "";            // string u koji spremamo povratnu poruku
    wait_serial();                 // čekaj dok nešto dođe preko serijskog porta
    while(Serial.available()!=0){ // dok ima nečega na serijskom portu
        inLine=Serial.readStringUntil('\n'); // spremaj u string dok ne dođe
završetak poruke
    }
    return inLine;                 // vrati povratnu poruku
}

void wait_serial(){
    // ova funkcija čeka dok nešto ne dođe preko serijskog porta
    bool serial_available = 0;
    while(serial_available==0){
        if(Serial.available()!=0)
            serial_available = 1;
    }
}
}

```

```

void empty_serial_buffer(){
    // ova funkcija miće sve šta je u buffer-u iz buffera
    while(Serial.available())
        Serial.read();
}

//-----//
//
//LCD - FUNKCIJA//
void LCD_initialization(){
    // ova funkcija inicijalizira LCD
    lcd.begin(16, 2);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("LCD working");    // ispis poruke da LCD radi
    delay(1000);
}

//-----//
//
//SD čitač kartice - FUNKCIJE//
void SD_initialization()
{
    // ova funkcija inicijalizira SD karticu i serijsku komunikaciju
    Serial.begin(115200);    // započinjemo serijsku komunikaciju na istom
baudrate-u kao GRBL
    while (!Serial);        // dok se serijska komunikacija nije inicijalizirala
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Initializing");    // ispisujemo da se događa inicijalizacija
        lcd.setCursor(0, 1);
        lcd.print("...");
        if (!SD.begin(chipSelect)) {    // ako se SD kartica nije uspjela
inicijalizirati
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Initalization");    // ispis: Initialization FAILED
            lcd.setCursor(0, 1);
            lcd.print("FAILED");

```

```

    while (1){
        delay(1);
    }
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Initalization"); // isprintaj da je Inicijalizacija završena
lcd.setCursor(0, 1);
lcd.print("DONE");
delay(500);
}

void save_file_names() {
    // ova funkcija sprema imena datoteka u polje stringova
    File dir; // otvori Fail varijablu
    dir = SD.open("/"); // spremi datoteku u varijablu
    numberOfFiles=0;
    while (true) {
        File entry = dir.openNextFile(); // otvori datoteku
        if (! entry) { // ako nema više datoteka izađi iz loop-a
            // nema više datoteka
            break;
        }
        file_pointer=(entry.name()); // spremi pointer imena datoteke
        file_names[numberOfFiles]=file_pointer; // spremi ime datoteke u polje
stringova
        numberOfFiles++; // povećaj broj datoteka
        entry.close(); // zatvori Fail varijablu
    }
    dir.close(); // zatvori Fail varijablu
    numberOfFiles=numberOfFiles-2; // system datoteke ne možemo čitati a na kraju
su
} // liste datoteka pa smanji max broj datoteka

//-----//
//
//Enkoder - FUNKCIJE I PREKIDI//
void setup_encoder()
{

```

```

// ova funkcija postavlja prekide i port enkodera
EICRA|=(1<<ISC00)|(1<<ISC01)|(1<<ISC11);
encPORT&=~(1<<pinA)|(1<<pinB)|(1<<button);
sei();    // uključivanje globalnih prekida
}

ISR (INT0_vect)
{
    // ova funkcija kontrolira varijablu stanja enkodera
    bool Bstate=(encPIN&(1<<pinB));    // pročitaj stanje pinB
    if(Bstate==0){
        encoder_change=1;
    }else{
        encoder_change=2;
    }
}

ISR (INT1_vect)
{
    // ovaj prekid postavlja varijablu stisnutosti enkodera u 1
    select=1;
}

//-----//

```

6. Zaključak

U ovom projektu bilo je potrebno riješiti probleme konstrukcije te elektroničke i programske izvedbe. Problemi u dizajniranju konstrukcije bili su tolerancije dijelova, u industriji CNC mašine se proizvode od metalnih dijelova napravljenih na velikoj preciznosti, u ovom projektu za veliki dio stvari korišteni su 3D printani dijelovi kako bi se smanjio trošak, ali to je dovelo do problema zbog relativne nepreciznosti takvih dijelova. Ti problemi su bili riješeni učenjem iz pogrešaka te prepravljajem konstrukcije sa novijim iteracijama dijelova. Do problema u elektronici je došlo kod smetnji. Komunikacija između dva mikroupravljača događa se serijski preko dviju žica, kod komunikacije došlo je do hvatanja smetnji pa je rješenje bilo korištenje žica koje su bolje zaštićene od smetnji. Kod programa došlo je do problema zbog manjka službene dokumentacije o protokolu komunikacije sa firmware-om te je to jednostavno bilo riješeno dubokim istraživanjem sličnih projekata. Zbog ovih problema i iskustva dobivenih kroz ovaj rad, ovaj rad mi je donio znanja o rješavanju problema tolerancija kod 3D printanih dijelova, osmišljanja te realiziranja elektroničkog sistema i znanja o serijskim komunikacijama između dvaju mikroupravljača. Usprkos problemima, ovaj rad je pokazao kako je moguće napraviti CNC tip mašine. CNC tipovi mašina imaju veliki broj primjena u industriji te za inovativne ljude mogu biti jako korisni uređaji čak i na manjoj skali poput ovog plotera. CNC mašine su inače relativno skupe, ali ovaj rad pokazuje da je moguće napraviti kvalitetan CNC tip mašine poput plotera za manje od 1000kn.

7. Literatura

1.1. Što je koračni motor:

<https://uk.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/stepper-motors-guide>

1.2. Upravljanje koračnim motorima:

<https://repozitorij.unin.hr/islandora/object/unin%3A95/datastream/PDF/view>

2. Mehanizmi linearnog prijenosa:

https://www.fer.unizg.hr/_download/repository/Skripta_-_Prijenosni_mehanizmi.pdf

3.3. Driveri:

<https://youtu.be/5CmjB4WF5XA>

4.1. Firmware plotera – GRBL:

<https://github.com/gnea/grbl/wiki>

4.2. Universal Gcode Sender, te kako ga koristiti:

https://winder.github.io/ugs_website/

8. Popis slika

Slika 2.1.1. rotor i stator koračnog motora.....3

<https://circuitdigest.com/tutorial/what-is-stepper-motor-and-how-it-works>

Slika 2.2.1. dijagram unipolarnog i bipolarnog koračnog motora.....4

<https://starfishmedical.com/assets/Unipolar-vs-Bipolar-Configuration.png>

Slika 2.3.1. tablica stanja faza pri rotaciji.....5

<https://aws1.discourse-cdn.com/arduino/original/4X/c/e/0/ce0a1d8274fb3a26367f5bab3cb1dc7bb6d578e1.gif>

Slika 2.3.2. grafovi upravljanja koračnim motorom.....5

<https://blog.poscope.com/stepper-motor-driver/>

Slika 3.2.1. Y os odozgora.....7

Slika 3.2.2. Y os odozdola.....7

Slika 4.3.1. CNC štit.....9

<https://aws1.discourse-cdn.com/arduino/original/4X/5/2/a/52ad455369f7c7b6e4a1289ecb0c6b518fe05261.jpeg>

Slika 4.2.1. shema plotera.....10

Slika 4.3.1. izgled driver modula.....11

<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQC0OaV1ppKQIHlDzQM7kKl6BZ-O5gPGszA9Q&usqp=CAU>

Slika 4.3.2. tablica postavljanja rezolucije koraka.....12

<https://e-radionica.com/en/blog/hum-a4988-stepper-motor-driver/>

Slika 4.4.1. enkoder i graf njegovih izlaza.....13

<https://i.ytimg.com/vi/v4BbSzJ-hz4/hqdefault.jpg>

Slika 4.4.2. shema SD interfejsa.....	14
Slika 5.2.1. sučelje UGS-a.....	16
Slika 5.3.1. Inkscape parametri za stvaranje G-koda.....	17